Uniwersytet Śląski
Wydział Nauk Ścisłych i Technicznych
Informatyka

Rozprawa doktorska

Identyfikacja anomalii w dziedzinowych zbiorach danych złożonych

mgr inż. Czesław Horyń

Promotor: dr hab. Agnieszka Nowak - Brzezińska, prof. UŚ

Katowice, 2024

Z głęboką wdzięcznością
dedykuję tę pracę
Rodzicom i Ani.

Wyrażam zgodę na udostępnienie mojej pracy doktorskiej dla celów naukowo-badawczych.

Data: Podpis autora pracy:

Słowa kluczowe:
Wykrywanie obserwacji odstających, Self-Organizing Map (SOM), Local Outlier Factor (LOF), Autoenkoder, Złożone anomalie, Optymalizacja rozmiaru bloku w algorytmie LOF, Outlier ensembles, Anomalie w danych kategorycznych i wielowymiarowych

---

Oświadczenie autora pracy

Świadomy odpowiedzialności prawnej oświadczam, że niniejsza praca doktorska została napisana przez mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data: Podpis autora pracy:

Spis treści

Notacja

- D - zbiór danych, macierz danych, zbiór obiektów (próbek), zestaw danych
- X_i - i-ty obiekt (próbka, obserwacja) w zbiorze danych D, każdy obiekt X_i jest reprezentowany jako wektor x zmiennych (cech, atrybutów) $[x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$, gdzie $x_{i,j} \in Q_j$
- A_j - zmienna (cecha, atrybut), to j-ta zmienna (cecha, atrybut) dla obiektu (próbki) X_i, każda zmienna (cecha, atrybut) ma dziedzinę wartości Q_j
- Q_j - określony zbiór wartości zmiennej (cechy, atrybutu) A_j
- x - wektor zmiennych (cech, atrybutów), reprezentujący zestaw wartości zmiennych (cech, atrybutów) dla danego obiektu (próbki), zapisany jako $[x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$
- $x_{i,j}$ - wartość zmiennej (cechy, atrybutu) A_j dla obiektu (próbki) X_i
- $\mathcal{H}$ - zbiór hipotez
- $h(\cdot)$ - hipoteza (uczący się model)
- $\theta^* = \arg\min_\theta f(\theta)$ - wartość zmiennej θ, która minimalizuje funkcję $f(\theta)$, θ jest wektorem parametrów modelu
- I - macierz jednostkowa
- η - współczynnik uczenia się
- ∇ - gradient, kierunek największej zmiany wartości funkcji w przestrzeni wielowymiarowej
- ϕ - funkcja aktywacji
- μ - średnia arytmetyczna
- σ - odchylenie standardowe

Komentarz: Aby uniknąć powtórzeń i zachować płynność oraz ciekawość tekstu, niektóre terminy w rozprawie stosowane są zamiennie. Terminy takie jak *zmienna*, *cecha* i *atrybut* oznaczają to samo. Podobnie, *obiekt* i *próbka* oraz *obserwacja* również są używane zamiennie.

Podziękowania

Chciałbym wyrazić moje najgłębsze podziękowania mojemu promotorowi, Profesor Agnieszce Nowak-Brzezińskiej za jej nieocenione wsparcie, cenne wskazówki i nieustającą motywację, które towarzyszyły mi przez cały okres przygotowywania tej pracy. Jej mądrość, wiedza i niezwykła osobowość były dla mnie nieocenionym źródłem inspiracji.

Szczególne podziękowania kieruję także do moich rodziców, którzy zawsze wierzyli we mnie i wspierali mnie na każdym kroku mojej edukacji. Pomimo problemów zdrowotnych, ich miłość, wsparcie i poświęcenie pozwoliły mi dążyć do realizacji moich celów.

Na koniec, chciałbym podziękować mojej żonie, Ani, za jej nieskończoną cierpliwość, zrozumienie i wsparcie. Jej miłość i wiara w moje możliwości były dla mnie ogromnym wsparciem w trudnych momentach.

Dziękuję Wam wszystkim za to, że byliście ze mną w tej podróży.

Rozdział 1

Wprowadzenie

Anomalie pojawiają się wszędzie, a ich analiza często prowadzi do głębszego zrozumienia zjawisk niż badanie samych trendów czy regularności w danych. Jednakże zrozumienie i wykrywanie anomalii wiąże się z kilkoma istotnymi wyzwaniami, z których jednym z najważniejszych jest problem nierównowagi klas. Nierównowaga klas to sytuacja, w której w zbiorze danych dominują przypadki jednej kategorii (najczęściej normalne zdarzenia), podczas gdy przypadki interesujące, takie jak anomalie, są znacznie rzadsze. Taka dysproporcja znacząco utrudnia skuteczne identyfikowanie anomalii, zwłaszcza w kontekście danych o wysokiej liczbie wymiarów. Drugim wyzwaniem jest nienadzorowane uczenie się: wykrywanie anomalii często odbywa się bez nadzoru, co wynika z braku wcześniejszej wiedzy na temat ich charakteru. Dane bez etykiet są powszechnie dostępne, a dane z etykietami są trudno osiągalne i kosztowne do zdobycia. Trzecim istotnym problemem są dane wejściowe, które w rzeczywistych zastosowaniach często składają się z atrybutów kategorycznych i/lub mieszanych. Identyfikacja przypadków, które odstają od normy, stanowi ciekawe wyzwanie w eksploracji danych. Ze względu na ogromne znaczenie wykrywania takich obiektów w praktycznych zastosowaniach, tematyka ta została wybrana jako przedmiot niniejszej rozprawy doktorskiej.

Zainteresowanie wykrywaniem anomalii wynika głównie z ich istotności w wielu realnych zastosowaniach, od prostych analiz danych relacyjnych, aż po przełomowe odkrycia w dziedzinie analizy ogromnych zbiorów danych astronomicznych. Ważne zastosowania obejmują także analizę danych biologicznych oraz wykrywanie oszustw finansowych, wśród innych istotnych dziedzin. Rzeczywiste problemy związane z wykrywaniem anomalii są złożone i wielowymiarowe, często obejmują zarówno dane numeryczne, jak i kategoryczne. W kontekście wielowymiarowego zbioru danych, wartość odstająca może być odchylona w odniesieniu do jednego z wymiarów. W takich przypadkach stosuje się różne metody analizy, w tym modele bazujące na danych historycznych, które pomagają przewidywać i wykrywać zachowania odstające od normy, jednak wybór odpowiedniego

podejścia zależy od specyfiki danych. Założeniem modeli historycznych jest przekonanie, że mechanizmy generujące dane nie uległy znaczącym zmianom w czasie, co oznacza, że statystyki opisujące system w przeszłości są adekwatne do opisu jego obecnego stanu.

Temat anomalii nawiązuje do wcześniejszych badań autora i został dodatkowo ukierunkowany przez okoliczności związane z pandemią COVID-19, co wpłynęło na tematykę i kierunek badań. Rozwój pandemii COVID-19 doskonale pokazał, jak nietypowe dane, takie jak objawy chorobowe, mogą stać się ważnym zagadnieniem dla badaczy i wymagać szczegółowej analizy. W trakcie pandemii pojawiło się wiele nietypowych przypadków zachorowań, co zainspirowało do zgłębienia zagadnienia wykrywania anomalii. Skala i dynamika pandemii podkreśliły znaczenie skutecznego identyfikowania podejrzanych przypadków, co stało się istotne dla szybkiego reagowania na potencjalne zagrożenia zdrowotne. W rezultacie w pracy skoncentrowano się na rozwoju zaawansowanych technik wykrywania anomalii, aby lepiej je zrozumieć i analizować.

Z pozoru problem wykrywania anomalii może wydawać się prostą kwestią klasyfikacji danych na normalne i anomalie. Jednak popularne algorytmy uczenia maszynowego, takie jak sieci neuronowe, maszyny wektorów nośnych czy drzewa decyzyjne, często okazują się nieskuteczne w tym kontekście z powodu drastycznej nierównowagi między anomaliami a danymi normalnymi. Taka nierównowaga prowadzi do wysokiego odsetka fałszywych negatywów, czyli pominięcia wykrycia anomalii. Różnorodność anomalii dodatkowo komplikuje problem, ponieważ anomalie mogą mieć niewiele wspólnego ze sobą. Co więcej, anomalie mogą występować w zakresach danych podobnych do normalnych, co wymaga zaawansowanej analizy relacji między danymi. Te wyzwania motywują rozwój wyspecjalizowanych algorytmów wykrywania anomalii oraz zrozumienie ich zastosowań i ograniczeń. Anomalie można zdefiniować jako sekwencję lub kombinację zachowań, działań, czy obserwacji, które znacznie odbiegają od normy. Takie anomalie, choć statystycznie rzadkie, mogą sygnalizować poważne problemy, jak awarie systemu, ataki czy oszustwa. Kluczowe cechy anomalii to:

- **rzadkość** - anomalie występują rzadziej niż typowe zachowania czy działania,
- **znaczenie** - anomalie często mają krytyczne znaczenie, np. mogą wskazywać na błędy, oszustwa, awarie systemu, ataki hakerskie lub inne poważne problemy,
- **kontekstowość** - nietypowość anomalii zależy od kontekstu. Na przykład aktywność w systemie komputerowym może być normalna w godzinach pracy, ale nietypowa w środku nocy,
- **dynamiczność** - definicja tego, co jest nietypowe, może zmieniać się w czasie, reagując na zmieniające się warunki, nowe dane lub ewoluujące normy,
- **wielowymiarowość** - anomalie mogą obejmować wiele zmiennych lub wymiarów danych, które razem tworzą obraz różniący się od normy,
- **interaktywność** - w niektórych przypadkach, nietypowość anomalii może być wykryta jedynie poprzez analizę interakcji między różnymi elementami systemu lub zestawu danych.

Rozpoznawanie i analiza takich anomalii wymaga zaawansowanych technik, takich jak analiza danych, statystyka i uczenie maszynowe, aby skutecznie je zidentyfikować i zinterpretować. Głęboka analiza problematyki oraz wyniki badań umożliwiły sformułowanie tezy pracy, zgodnie z którą:

> *Zastosowanie zaawansowanych technik, takich jak SOM, LOF z podziałem na bloki oraz autoenkodery w ramach zespołu algorytmów, umożliwia zwiększenie czułości i wydajności wykrywania anomalii w rzeczywistych, złożonych zbiorach danych, przy jednoczesnym zachowaniu wydajności procesu analizy, szczególnie dzięki optymalizacji LOF.*

Celem tej rozprawy jest opracowanie metod i narzędzi, które zwiększą czułość i efektywność wykrywania anomalii w złożonych zbiorach danych, z naciskiem na praktyczne zastosowania i optymalizację algorytmu LOF. Praca ma na celu:

- zdefiniowanie pojęcia anomalii oraz identyfikacja wyzwań związanych z ich wykrywaniem w złożonych, kategorycznych i wielowymiarowych danych,
- zastosowanie i ocena nowoczesnych technologii, takich jak uczenie maszynowe i głębokie uczenie, w kontekście identyfikacji anomalii, z naciskiem na zwiększenie czułości,
- rozwój i optymalizacja zaawansowanych technik zespołowych, w celu zwiększenia skuteczności i wydajności procesu wykrywania anomalii,
- implementacja i testowanie proponowanych metod na rzeczywistych zbiorach danych, aby ocenić ich efektywność, czułość oraz użyteczność w praktycznych zastosowaniach.

Poprzez realizację tych celów, niniejsza rozprawa dąży do stworzenia wydajnych narzędzi analitycznych, które mogą być wykorzystywane w różnych dziedzinach w celu poprawy wykrywania nieprawidłowości i potencjalnych zagrożeń w dużych i złożonych zbiorach danych. Jest to szczególnie istotne, gdyż systemy zarządzające transakcjami finansowymi, zdrowiem publicznym oraz infrastrukturą krytyczną muszą nieustannie monitorować swoje działania, aby skutecznie reagować na wszelkie nieprawidłowości lub sytuacje awaryjne. Wczesne wykrywanie anomalii w zachowaniu systemu jest bardzo ważne, zwłaszcza w kontekście przemysłowych systemów sterowania, gdzie szybkie i precyzyjne wykrycie awarii może zapobiec poważnym skutkom, takim jak uszkodzenie sprzętu, przerwy w produkcji czy nawet zagrożenie dla życia ludzkiego.

Zastosowanie odpowiednich algorytmów dostosowanych do specyficznych wymagań danej aplikacji jest niezbędne, aby skutecznie monitorować i analizować te systemy. Dlatego analiza różnych aspektów obliczeniowych problemu wykrywania wartości odstających jest nieodzowna, aby określić najskuteczniejszą metodę w danym kontekście aplikacyjnym. Wartości odstające dostarczają cennych informacji o zmieniających się właściwościach systemów fizycznych i mogą służyć jako wczesne sygnały potencjalnych zmian w ich

zachowaniu. Gdy wykrycie wartości odstających sugeruje możliwe niekorzystne zmiany, niezbędne jest posiadanie planu działania, aby te zmiany zneutralizować. Wykrywanie wartości odstających jest często niezbędne dla funkcjonowania systemu, umożliwiając mu adaptację do zmian zachodzących w czasie. Jest to forma samoregulacji, która inicjuje odpowiednie działania naprawcze we właściwym czasie, aby uniknąć negatywnych skutków. Szczególnie ważne jest to w systemach obsługujących rzeczywiste aplikacje, gdzie tolerancja na niezamierzone efekty jest niezwykle niska.

1.1 Układ pracy

Pierwsza część pracy obejmuje analizę najważniejszych aspektów problemu wykrywania anomalii w danych złożonych, istotnych dla dalszych badań naukowych. W związku z tym, początkowe rozdziały rozważają różne kwestie badawcze dotyczące wykrywania wartości odstających oraz przegląd dostępnej wiedzy na ten temat. W kolejnych rozdziałach omówione zostają istotne zagadnienia badawcze oraz przedstawione różne techniki opracowane w celu ich rozwiązania. Każdy rozdział zawiera teoretyczne tło dotyczące omawianego zagadnienia. Rozpoczyna się od omówienia definicji anomalii oraz zaproponowania własnej definicji, charakterystyk danych złożonych oraz przedstawienia wyzwań i metod ich identyfikacji. Następnie analizowane są różne teoretyczne koncepcje oraz prezentowane ich praktyczne znaczenie, w tym analizy i metody przetwarzania danych kategorycznych i wielowymiarowych. Przedstawione zostają również doświadczenia związane z konkretnymi problemami badawczymi. Druga część pracy koncentruje się na wyzwaniu wykrywania wartości odstających w zespołach danych, szczególnie omawiając rolę nowoczesnych technologii, takich jak uczenie maszynowe i głębokie uczenie, w kontekście wykrywania anomalii. Przedstawione zostają wyniki badań uzyskane z różnych podejść algorytmicznych oraz szczegóły nowoczesnych technik wykrywania anomalii, a także zaawansowane techniki zespołowe, które zwiększają skuteczność identyfikacji anomalii.

Poniższy **Układ pracy** pełni funkcję przewodnika po kolejnych rozdziałach rozprawy. Praca została zaprojektowana tak, aby stopniowo wprowadzać czytelnika w problematykę identyfikacji anomalii w danych złożonych oraz prezentować nowatorskie metody ich wykrywania. **Rozdział drugi** skupia się na omówieniu definicji anomalii, charakterystyk danych złożonych oraz przedstawieniu wyzwań i metod ich identyfikacji. **Rozdział trzeci** koncentruje się na anomaliach w danych kategorycznych i wielowymiarowych, analizując specyficzne wyzwania związane z tymi rodzajami danych oraz przedstawia metody ich przetwarzania i analizy. Omawia problemy wynikające z braku naturalnego porządku i trudności w definiowaniu miar odległości w danych kategorycznych oraz wyzwania związane z wysoką wymiarowością, co wpływa na skuteczność tradycyjnych metod detekcji anomalii. **Rozdział czwarty** omawia zastosowanie nowoczesnych technik uczenia maszynowego, w tym głębokiego uczenia, w wykrywaniu anomalii. Przedstawia autorski system Trinity SALT, który integruje algorytmy Sieci Samoorganizujących się

(SOM), Autoenkoderów (AE) i Local Outlier Factor (LOF) w celu efektywnej detekcji anomalii. **Rozdział piąty** prezentuje zaawansowane techniki zespołowe, które zwiększają efektywność identyfikacji anomalii poprzez łączenie wyników różnych algorytmów lub zastosowanie jednego algorytmu na różnych cechach i podzbiorach danych. Rozdział porównuje te metody z tradycyjnymi podejściami, podkreślając zalety współpracy między algorytmami oraz pokazuje, jak różnorodne sposoby analizy danych poprawiają wykrywanie anomalii. **Rozdział szósty** zawiera szczegółowy przegląd miar oceny skuteczności metod wykrywania anomalii, szczególnie w kontekście algorytmów nienadzorowanych, gdzie brak oznaczeń rzeczywistych anomalii utrudnia ocenę wyników. Omawia kluczowe miary oraz analizę błędów często popełnianych w tym procesie. **Rozdział siódmy** opisuje projekt i implementację systemu Trinity SALT, koncentrując się na integracji algorytmów SOM, AE i LOF oraz technicznych aspektach budowy. Omawia narzędzia użyte do konstrukcji, sposób integracji algorytmów oraz zastosowanie aplikacji webowej do interaktywnej analizy danych w czasie rzeczywistym. Rozdział przedstawia możliwości działania systemu oraz jego ograniczenia. **Rozdział ósmy** przedstawia wyniki eksperymentów z algorytmem LOF i jego podziałem na bloki z zastosowaniem optymalizacji bayesowskiej oraz z systemem Trinity SALT, oceniając jego skuteczność na różnych zestawach danych. Analiza porównuje Trinity SALT z pojedynczymi algorytmami (SOM, AE, LOF) oraz pokazuje, jak zespołowe podejście poprawia zarówno skuteczność, jak i wydajność systemu. Wyniki eksperymentów podkreślają efektywność zastosowanego nowatorskiego rozwiązania podziału na bloki w LOF oraz praktyczność zastosowań Trinity SALT z autorską techniką Maksymalnej Znormalizowanej Agregacji (MNA), która stanowi innowacyjne podejście wyróżniające się na tle tradycyjnych metod dzięki unikalnemu premiowaniu konsensusu modeli. **Rozdział dziewiąty** stanowi podsumowanie, analizuje wyniki badań oraz przedstawia ich interpretację. Omawia znaczenie uzyskanych rezultatów i proponuje dalsze kierunki badań, takie jak integracja nowych algorytmów oraz rozwój metod detekcji anomalii w czasie rzeczywistym.

Rozdział 2

Problematyka identyfikacji anomalii w danych złożonych

W każdym aspekcie naszego życia istnieje potrzeba wykrywania anomalii. Znaczenie tego zagadnienia staje się jasne, gdy przyjrzymy się kilku kluczowym przykładom: chromosomalnym anomaliom w nowotworach [1], oszustwom związanym z transakcjami kartami kredytowymi wśród miliardów legalnych operacji [2], anomaliom płodu w czasie ciąży [3], wykrywaniu spisków terrorystycznych w mediach społecznościowych [4] oraz wczesnym oznakom krachu na giełdzie [5], a także nietypowym przypadkom zachorowań, które stały się szczególnie istotne w kontekście pandemii COVID-19 [6, 7]. Pandemia ta uwypukliła znaczenie wczesnego wykrywania anomalii w danych zdrowotnych, co pozwala na szybsze reagowanie na nowe i potencjalnie groźne odmiany wirusa oraz inne zjawiska epidemiologiczne. Obserwacje odstające, zwane również anomaliami, to dane znacząco odbiegające od reszty zbioru. Ich wykrywanie jest niezbędne, ponieważ mogą wskazywać na błędy pomiarowe, oszustwa, nietypowe zjawiska lub nowe wzorce wymagające dalszej analizy. Te wyjątkowe wzorce mogą przybierać różne formy, takie jak reguły asocjacyjne, klasyfikacyjne czy opisy grupowania. W wielowymiarowych zbiorach danych, obserwacja może być uznana za odstającą w jednym wymiarze, ale nie w innym. Często obiekt jest uznawany za anomalię tylko w kontekście kombinacji dwóch lub więcej atrybutów, co nakłada wyjątkowe wymagania na metody analizy. Anomalie mogą być rezultatem błędów w danych, ale mogą również wskazywać na istnienie wcześniej nieznanych procesów.

W erze dynamicznego rozwoju technologii i rosnącej ilości generowanych danych, zbiory danych stają się coraz bardziej skomplikowane. Ilość danych, różnorodność ich typów oraz wzajemne zależności znacząco utrudniają procesy analityczne. Złożone zbiory danych charakteryzują się wielowymiarowością, różnorodnością typów, dynamicznymi zmianami oraz obecnością skomplikowanych wzorców i anomalii. Dlatego jednym z głównych wyzwań w dziedzinie identyfikacji anomalii jest precyzyjne określenie, czym właściwie jest

anomalia, którą chcemy zidentyfikować. W literaturze istnieje wiele różnych definicji tego pojęcia, co prowadzi do braku ustanowienia spójnej metodologii. Wartości odstające mogą być definiowane na różne sposoby, zależnie od przyjętej perspektywy, metody detekcji oraz struktury analizowanych danych. Ta niejednoznaczność dodatkowo komplikuje proces identyfikacji anomalii, zwłaszcza w kontekście danych wielowymiarowych, gdzie identyfikacja anomalii za pomocą standardowych narzędzi wizualizacyjnych, takich jak wykresy, diagramy czy tabele, jest praktycznie niemożliwa. Człowiek ma ograniczoną zdolność do interpretacji informacji w wielu wymiarach jednocześnie, co sprawia, że niezbędne są zaawansowane metody analizy.

Podejmując się zadania wykrywania anomalii, ważne jest zrozumienie podstawowych koncepcji matematycznych i obliczeniowych. Poznanie zbiorów próbek oraz zmiennych stanowi podstawę każdej analizy danych. Równie ważne jest zrozumienie miar odległości i podobieństwa między próbkami, takich jak odległość euklidesowa, miara Hamminga i inne. Te podstawowe koncepcje są fundamentem procesu odkrywania wiedzy KDD (*ang. knowledge discovery in databases*, KDD), który odgrywa istotną rolę w kontekście wykrywania anomalii. Dzięki ustrukturyzowanemu podejściu, takiemu jak KDD, możliwe jest zastosowanie jednolitych metod, niezależnie od specyfiki problemu czy charakterystyki danych. Standaryzacja pozwala na efektywną i spójną realizację zadań analitycznych, zapewniając jednolitość procesów oraz ułatwiając ich skalowanie i powtarzalność wyników.

Problem testowania istotności wartości odstających przyciągnął znaczną uwagę badaczy już przed 1937 rokiem [8]. Pierwsze formalne badania nad tym zagadnieniem sięgają 1777 roku [9]. W XIX wieku zajmowano się tym problemem w różnych dyscyplinach, przykładowo w pracy [10] zostały zdefiniowane niezgodne obserwacje (*ang. discordant observations*). W 1933 roku Rider [11] przedstawił szczegółowy przegląd dotychczasowych prac związanych z tym zagadnieniem. Jest to problem stary i powszechnie znany, omawiany w artykułach Grubbsa [12] z 1950 roku oraz Anscombe'a [13] z 1960 roku, jako jedno z pierwszych zagadnień, które zostało gruntownie przeanalizowane statystycznie. Czym właściwie jest problem wartości odstających? Ferguson w 1961 roku [14] pisał:

> „...można go przedstawić w następujący sposób: w próbie o umiarkowanej wielkości, pobranej z określonej populacji, wydaje się, że jedna lub dwie wartości są zaskakująco daleko od głównej grupy. Eksperymentator jest skłonny odrzucić te pozornie błędne wartości, i to nie dlatego, że jest pewien, iż wartości są fałszywe. Wręcz przeciwnie, z pewnością przyzna, że nawet jeśli populacja ma normalny rozkład, istnieje pozytywne, choć niezwykle małe, prawdopodobieństwo, że takie wartości wystąpią w eksperymencie. Jest to raczej dlatego, że uważa, iż inne wyjaśnienia są bardziej prawdopodobne, a strata w dokładności eksperymentu spowodowana odrzuceniem kilku dobrych wartości jest niewielka w porównaniu do straty spowodowanej zachowaniem nawet jednej złej wartości. Problem polega więc na wprowadzeniu pewnego stopnia obiektywności w odrzuceniu odstających obserwacji."

Intensywne badania nad wykrywaniem wartości odstających rozpoczęły się w latach 70. XX wieku [15, 16, 17, 18] i nadal są kontynuowane. Metody stosowane do identyfikacji anomalii w danych są bardzo zróżnicowane i wywodzą się z różnych obszarów statystyki oraz analizy danych. Obejmują one techniki takie jak dyskryminacja, która polega na klasyfikacji danych do różnych grup na podstawie ich charakterystycznych cech; klasyfikacja taksonomiczna, która organizuje dane w hierarchiczne struktury na podstawie podobieństw między próbkami; estymacja funkcji gęstości, która pozwala na oszacowanie rozkładu prawdopodobieństwa zmiennych losowych w celu wykrycia nietypowych wartości; wizualizacja danych, która poprzez graficzne przedstawienie ułatwia identyfikację anomalii i wzorców; oraz przetwarzanie sygnałów, które analizuje i modyfikuje sygnały w celu wykrycia i usunięcia niepożądanych zakłóceń lub identyfikacji istotnych cech sygnałów. W rozdziale przybliżono różnorodne metody wykrywania anomalii, co pozwala zyskać obraz dostępnych narzędzi analitycznych oraz ich zastosowań w rozwiązywaniu konkretnych problemów w analizie danych, a także zapewnia zrozumienie ich potencjału.

Wzrost liczby źródeł danych, ich różnorodność (np. dane strukturalne, niestrukturalne, strumieniowe) oraz rosnąca skala (duże zbiory danych, *ang. big data*) stawiają przed nami konieczność opracowywania nowych podejść do analizy. Rozwój nowych technik i algorytmów, takich jak uczenie maszynowe, głębokie uczenie, analiza grafowa, metody zespołowe i hybrydowe, wymaga ciągłego aktualizowania wiedzy. Dlatego kończąc rozdział, przedstawiono aktualne wyzwania i trendy w rozwoju analizy anomalii, co jest niezbędne w kontekście problematyki wykrywania tych nieprawidłowości w danych złożonych.

2.1 Definicja anomalii

Jak wspomniano we wstępie, definicja anomalii może być zróżnicowana i niejednoznaczna. Spośród różnych definicji proponowanych dla wartości odstających, ta przedstawiona przez Hawkinsa jest najbardziej popularna [17]. Definiuje obserwację odstającą (*ang. outlier*) jako taką, która *„odbiega na tyle od pozostałych obserwacji, że budzi podejrzenia co do swojego pochodzenia, sugerując, że mogła zostać wygenerowana przez inny mechanizm”*. W praktyce, jeżeli obserwacja odstaje na tyle od reszty danych, że przypuszcza się, iż została wygenerowana przez inny mechanizm, może to oznaczać kilka rzeczy:

- **inny rozkład statystyczny** - próbka może pochodzić z populacji o odmiennym rozkładzie prawdopodobieństwa, co może sugerować różne zjawiska lub warunki, które generują dane,
- **błąd pomiaru lub wprowadzenia danych** - wartości odstające mogą być wynikiem błędów technicznych lub ludzkich podczas zbierania lub przetwarzania danych,
- **zdarzenie rzadkie lub ekstremalne** - w niektórych przypadkach obserwacje odstające są prawidłowymi danymi, które odzwierciedlają rzadkie zjawiska lub ekstremalne przypadki w zbiorze danych.

W statystyce często przyjmuje się, że dane w zestawie pochodzą z jednolitego rozkładu. To założenie jest istotne dla wielu technik statystycznych, ponieważ umożliwia stosowanie jednolitych metod modelowania danych i przeprowadzanie standardowych testów statystycznych.

Jednak, jeśli obserwacja odstająca pochodzi z innego rozkładu, jej uwzględnienie w analizie bez odpowiednich środków ostrożności może prowadzić do błędnych wniosków. Na przykład, taka próbka może zniekształcać estymację parametrów statystycznych, takich jak średnia i odchylenie standardowe, co w konsekwencji może wpływać na wyniki testów statystycznych. Z tego powodu, w analizie statystycznej kluczowe jest rozpoznanie i odpowiednie traktowanie obserwacji odstających. Może to obejmować ich wykluczenie z analizy, przeprowadzenie dodatkowych badań w celu zrozumienia przyczyn ich występowania lub zastosowanie metod statystycznych odpornych na obecność obserwacji odstających.

Oprócz wspomnianej definicji, wartości odstające można rozpatrywać w różnych kontekstach, uwzględniając ich częstość występowania, wyjątkowość oraz wpływ:

- **na podstawie częstości występowania** - wartości odstające mogą być pojedynczymi próbkami, które znacząco odbiegają od reszty danych. Mogą również reprezentować większe grupy lub struktury, takie jak społeczności czy sieci, które nie wpisują się w dominujące wzorce danych,

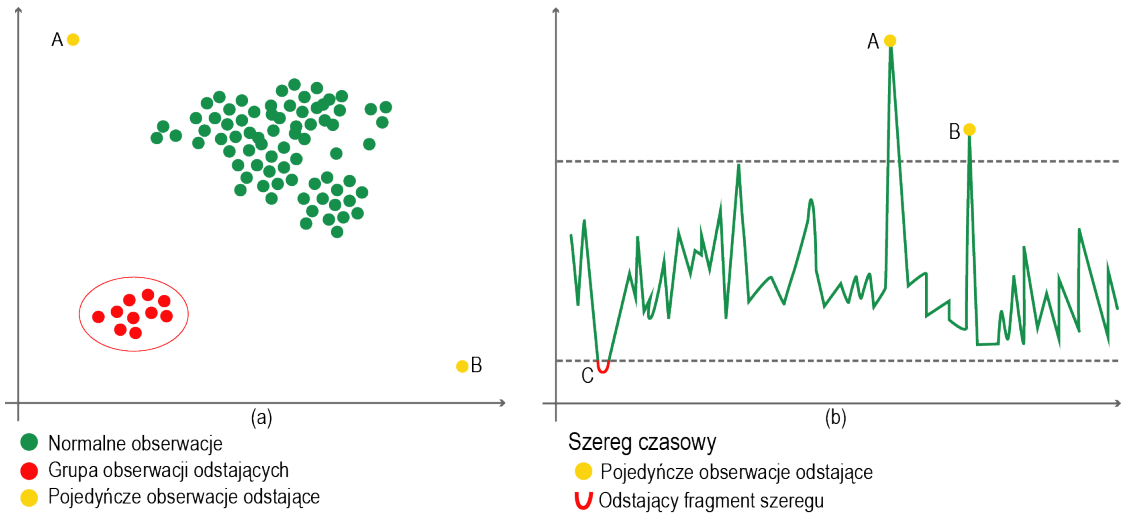
- **na podstawie wyjątkowości** - wartości te wyróżniają się w zbiorze danych z powodu swoich rzadkich cech lub wartości, co czyni je nietypowymi w kontekście całej populacji. Często są usuwane w trakcie analizy, aby uniknąć zniekształceń w modelowaniu, jednak w pewnych przypadkach mogą dostarczyć cennych informacji o nietypowych zjawiskach lub błędach w zbieraniu danych,

- **na podstawie wpływu** - choć często postrzegane negatywnie, wartości odstające mogą mieć pozytywny wpływ w niektórych kontekstach. Na przykład, w socjologii osoba o ekstremalnych poglądach może pełnić rolę lidera opinii. W kontekście naukowym, nietypowe podejścia mogą prowadzić do innowacji i przełomów, co pokazuje, że anomalie mogą przyczynić się do generowania nowych pomysłów.

Ponadto, w literaturze naukowej istnieje wiele różnych definicji obserwacji odstających, które były proponowane przez badaczy na przestrzeni lat. Na przykład, Grubbs [19] definiuje obserwację odstającą jako taką, która *„wydaje się znacząco odbiegać od pozostałych obserwacji w zbiorze danych, w którym się znajduje”*. Barnett i Lewis [18] opisują ją jako *„obserwację (lub podzbiór obserwacji), która wydaje się być niezgodna z resztą zestawu danych”*.

Natomiast według Rousseeuw i Leroy [20], możemy wyróżnić trzy główne kategorie obserwacji odstających:

- **obserwacje odstające** (*ang. outliers*) – są to przypadki, gdzie wartość zmiennej zależnej Y znacząco odbiega od reszty danych. Takie obserwacje mogą być spowodowane błędami pomiarowymi, specyficznymi warunkami eksperymentalnymi lub rzadkimi zdarzeniami. Mogą one zakłócać analizę statystyczną, wpływając na średnią, odchylenie standardowe i prowadząc do błędnych wniosków. Identyfikacja tych obserwacji jest kluczowa dla właściwego przygotowania danych do analizy,
- **obserwacje wysokiej dźwigni** (*ang. leverage*) – odnoszą się do sytuacji, w której przynajmniej jedna zmienna niezależna X znacznie odbiega od typowych wartości w zbiorze danych. Te obserwacje mają duży wpływ na linię regresji, ponieważ ich ekstremalne wartości mogą przesuwać tę linię, zmieniając tym samym kształt i kierunek modelu. Są one szczególnie ważne w analizie regresji, gdzie mogą znacząco wpłynąć na oszacowania zależności między zmiennymi,
- **obserwacje wpływowe** (*ang. influential observations*) – to obserwacje, których usunięcie z danych powoduje istotne zmiany w modelu, takie jak zmiana parametrów estymacji, kształtu krzywej regresji lub wyników testów statystycznych. Te obserwacje nie zawsze są wynikiem błędów, czasami odzwierciedlają one istotne, choć rzadkie przypadki w populacji. Wymagają one szczególnej uwagi, ponieważ ich obecność może sugerować konieczność przeglądu modelu lub zastosowania bardziej odpornych metod statystycznych, które lepiej radzą sobie z ekstremalnymi wartościami.



Rysunek 2.1: Wizualizacja anomalii w danych. (a) Punkty odstające A i B, zgrupowane anomalie oraz normalne obserwacje. (b) Szereg czasowy przedstawiający punkty odstające A i B oraz anomalny segment szeregu C. Źródło: opracowanie własne.

Rysunek 2.1 przedstawia wizualizację anomalii w danych. W części (a) zobrazowano punkty odstające A i B (*ang. point outliers*), zgrupowane anomalie (*ang. outlier cluster*) oraz normalne obserwacje. W części (b) przedstawiono szereg czasowy, w którym można zauważyć punkty odstające A i B oraz odstający fragment szeregu oznaczony jako C. Innym typem obserwacji odstających niż przedstawione na wykresie mogą być obserwacje kontekstowe i zbiorowe. Obserwacje kontekstowe są definiowane przez atrybuty kontekstowe, które określają kontekst (np. czas lub miejsce), oraz atrybuty behawioralne, które opisują właściwości samego zdarzenia (np. temperatura, poziom opadów). Z kolei, obserwacje zbiorowe obejmują sekwencje działań, które łącznie tworzą odstający wzorzec. Przykładem może być typowe zachowanie podczas ataku komputerowego, w którym pojawiają się próby uzyskania dostępu do różnych usług sieciowych, eksploatacja luk bezpieczeństwa, próby przepełnienia bufora, a następnie kopiowanie danych lub inne nieautoryzowane działania. Tego typu działania razem składają się na zbiorowy wzorzec. Znaczna różnorodność anomalii prowadzi do rozumienia ich jako koncepcji wieloznacznej. Na zakończenie tej części przedstawiono autorską definicję anomalii, która podkreśla jej unikalność i złożoność, a także wskazuje na potrzebę zaawansowanych technik do jej wykrycia i zrozumienia. Definicja ta wskazuje również na potencjalne znaczenie odkrycia anomalii w kontekście ujawniania nowych, wcześniej nieznanych mechanizmów lub zjawisk:

Anomalia to unikalny wzorzec lub zestaw wzorców w danych, który znacząco odbiega od przewidywanego zachowania. Jest wykrywalna dzięki zaawansowanym technikom uczenia maszynowego i analizy wielowymiarowej. Wzorzec odnosi się do regularności, schematu lub specyficznego układu, który można zidentyfikować w zbiorze danych. Wzorce mogą przyjmować różne formy, w zależności od rodzaju danych i dziedziny, w której są analizowane. Mogą to być sekwencje, grupy cech, zależności między zmiennymi lub inne struktury, które występują w danych w sposób powtarzalny i przewidywalny. Taka anomalia nie tylko różni się statystycznie od innych obserwacji, ale także może wskazywać na ukryte mechanizmy lub zjawiska, dotychczas nieznane lub niezrozumiane w danej dziedzinie.

2.1.1 Terminologia anomalii

Wykrywanie obserwacji odstających obejmuje szeroki zakres metod, które są do siebie podobne, lecz różnią się nazwami nadanymi przez różnych autorów. Przykładowo, autorzy opisują swoje podejścia jako wykrywanie anomalii, wartości odstających, detekcję nowości, identyfikację szumów, wykrywanie odchyleń, wykrywanie obserwacji nietypowych, punktów osobliwych czy eksplorację wyjątków. Można również spotkać terminy takie jak wykrywanie nadużyć/oszustw, nietypowych zachowań czy punktów zwrotnych, a także kilka innych określeń. Poniżej znajdują się opisy trzech najczęściej spotykanych i powszechnie stosowanych terminów w analizie danych i statystyce (anomalia, obserwacja odstająca i nowość), zdefiniowanych w literaturze naukowej przez uznanych autorów zajmujących się tą tematyką:

Anomalia

W literaturze naukowej definicja anomalii według Chandola i współautorów [21], w języku angielskim określanej jako *anomaly*, odnosi się do wzorca w danych, który znacząco odbiega od większości danych i nie odpowiada dobrze zdefiniowanemu pojęciu normalnego zachowania. Anomalia może wynikać z różnych przyczyn, takich jak złośliwe działania, awarie systemów lub inne interesujące dla analityka zdarzenia. Autorzy podkreślają, że rzeczywista istotność anomalii w życiu codziennym jest ważną cechą ich wykrywania.

W przeglądzie dotyczącym wykrywania anomalii metodami głębokimi i klasycznymi, Ruff i współautorzy [22] definiują anomalię podobnie, jako obserwację, która znacząco odbiega od pewnej koncepcji normalności. Według autorów anomalne dane mogą być bezużyteczne, na przykład gdy są spowodowane błędami pomiarowymi, lub mogą być bardzo wartościowe i zawierać klucz do nowych odkryć. Podkreślają również, że anomalia może być nazywana obserwacją odstającą lub nowością, a także określana jako niezwykła, nieregularna, nietypowa, niespójna, nieoczekiwana, rzadka, błędna, wadliwa, oszukańcza, złośliwa, nienaturalna lub po prostu dziwna – w zależności od kontekstu. Aby sformalizować swoją definicję, autorzy precyzują dwa aspekty: „koncepcję normalności" oraz co oznacza „znacząco odbiegać". Opierając się na teorii prawdopodobieństwa, definiują koncepcję normalności jako rozkład danych, który opisuje normalne zachowanie w danym zadaniu lub aplikacji. Obserwacja, która znacząco odbiega od tego normalnego zachowania – anomalia – to taka, która znajduje się w obszarze o niskim prawdopodobieństwie. Zakładając, że rozkład normalności ma odpowiadającą mu funkcję opisującą prawdopodobieństwo wystąpienia poszczególnych wartości, można zdefiniować zbiór anomalii jako te obserwacje, które mają na tyle niskie prawdopodobieństwo, że uznajemy je za wystarczająco rzadkie lub nietypowe.

Obserwacja odstająca

O wiele częściej można spotkać w literaturze definicję obserwacji odstającej, w języku angielskim określanej jako *outlier* lub *outlying observation* [19], która bywa również nazywana wartością odstającą. Obydwa terminy są poprawne i używane zamiennie, zależnie od kontekstu. Gdy mówimy o obserwacji odstającej, podkreślamy, że mamy na myśli konkretną próbkę, która różni się od reszty zbioru danych. Natomiast wartość odstająca skupia się na specyficznej wartości zmiennej w tej próbce, która jest nietypowa w porównaniu do innych wartości w zbiorze danych. W poprzednim podrozdziale krótko wspomniano trzy definicje z literatury naukowej, które teraz omówimy bardziej szczegółowo, aby lepiej zrozumieć pojęcie obserwacji odstającej.

Definicja obserwacji odstającej według Barnetta i Lewisa [18] opisuje ją jako obserwację (lub grupę obserwacji), która wydaje się być niespójna z resztą danych. Autorzy podkreślają, że ocena, czy dana obserwacja jest odstająca, jest subiektywna i zależy od obserwatora, co jest istotne przy definiowaniu pojęcia "wydaje się być niespójna". Główne

zmartwienie obserwatora to pytanie, czy te obserwacje rzeczywiście należą do głównej populacji. Jeśli nie, mogą zakłócać wyciąganie wniosków na temat populacji. Niewielka liczba fałszywych obserwacji może początkowo nie być zauważalna i nie musi znacząco zniekształcać wyników analizy. Jednak ich wpływ polega na tym, że są postrzegane jako ekstremalne i mogą powodować trudności w prawidłowym reprezentowaniu populacji oraz znacząco wpływać na oszacowania i testy parametrów w modelu populacji.

Natomiast Hawkins [17] intuicyjnie opisuje obserwację odstającą jako taką, która różni się od pozostałych na tyle, że budzi podejrzenia co do jej pochodzenia, sugerując, że mogła zostać wygenerowana przez inny mechanizm. Według autora, analiza próbki zawierającej takie obserwacje ujawnia cechy takie jak znaczne luki między obserwacjami odstającymi a resztą danych oraz wyraźne odchylenia od grupy pozostałych obserwacji, mierzone na odpowiednio znormalizowanej skali. Hawkins wskazuje na dwa podstawowe mechanizmy prowadzące do powstania obserwacji odstających. Pierwszy mechanizm to rozkłady z grubymi ogonami (*ang. heavy-tailed distributions*), gdzie dane pochodzą z rozkładu z grubymi ogonami, takiego jak rozkład t-Studenta. W takim przypadku wszystkie obserwacje są poprawne, lecz niektóre z nich mają skrajne wartości z powodu charakterystyki rozkładu. Drugi to rozkłady zanieczyszczone (*ang. contaminated distributions*), gdzie dane pochodzą z dwóch rozkładów. Jeden to „podstawowy rozkład", który generuje „dobre" obserwacje, natomiast drugi to „zanieczyszczony rozkład", który generuje „zanieczyszczenia". Jeśli zanieczyszczony rozkład ma cięższe ogony niż podstawowy rozkład, zanieczyszczenia będą się odróżniać jako obserwacje odstające, wyraźnie oddzielając się od „dobrych danych". Te dwa mechanizmy wpływają na sposób identyfikacji i analizy obserwacji odstających, podkreślając znaczenie zrozumienia źródła i natury tych obserwacji przy interpretacji wyników.

Grubbs [19] definiuje obserwację odstającą jako obserwację, która znacząco odbiega od innych elementów zestawu danych, w którym się znajduje. Według autora, taka obserwacja może być ekstremalnym przejawem losowej zmienności w danych lub wynikiem błędu lub odchylenia od przewidzianej procedury eksperymentalnej. Jego praca podkreśla konieczność identyfikacji obserwacji odstających, aby zdecydować, czy są one autentycznymi punktami danych wynikającymi z naturalnej zmienności, czy też błędami, które należy zbadać i ewentualnie wykluczyć z dalszej analizy. Obserwacje, które znacząco odbiegają od pozostałych danych, mogą być naturalnymi wynikami losowej zmienności. W takim przypadku są to autentyczne próbki danych, które powinny być uwzględniane i analizowane na równi z innymi danymi, ponieważ mogą odzwierciedlać prawdziwe cechy rozkładu. Alternatywnie, niektóre obserwacje mogą wynikać z błędów w procesie eksperymentalnym, takich jak pomyłki w pomiarach, obliczeniach lub zapisie danych. W takich sytuacjach ważne jest badanie i określenie przyczyny tych odchyleń. Jeśli zostanie ustalone, że są to błędy, można rozważyć ich wykluczenie z dalszej analizy.

Podsumowując, wszystkie definicje mają swoje źródło w statystyce i koncentrują się na obserwacjach odstających jako danych, które znacząco różnią się od reszty zbioru. Wska-

zują na konieczność szczegółowej analizy danych odstających, aby dokładnie zrozumieć, czy są one wynikiem rzeczywistych cech populacji, czy błędów. Każda definicja podkreśla różne aspekty analizy tych danych, ale wszystkie zgadzają się co do ich potencjalnego wpływu na wyniki analiz statystycznych i konieczności dokładnego badania.

Obserwacja nietypowa i niezwykła

W ramach analizy danych czasami używa się terminów obserwacja nietypowa (*ang. atypical observations*) oraz obserwacja niezwykła (*ang. unusual observations*) jako synonimów dla obserwacji odstających czy anomalii. Mimo iż te pojęcia mogą być nieraz stosowane zamiennie, różnią się one nieco w kontekście swojego znaczenia. Obserwacje nietypowe odnoszą się do danych, które nie pasują do oczekiwanego wzorca, lecz nie są ekstremalnie różne od reszty zbioru danych. Te dane mogą wynikać z naturalnej zmienności, reprezentować nowe, jeszcze nierozpoznane wzorce, lub mogą być wynikiem błędów w pomiarach lub zapisie danych. Obserwacje niezwykłe natomiast są danymi, które są rzadko spotykane lub zaskakująco różne od pozostałych danych w zbiorze. Takie obserwacje również mogą wynikać z błędów, jednak często wskazują na rzadkie lub nietypowe zdarzenia, które warto zbadać bardziej szczegółowo. Rozpoznanie i analiza zarówno obserwacji nietypowych, jak i niezwykłych, mogą dostarczyć cennych wglądów w zmieniające się trendy lub nowe zjawiska w zbiorze danych. Na przykład, w analizie ekonomicznej nietypowe obserwacje mogą wskazywać na wprowadzenie nowej polityki lub technologii, podczas gdy niezwykłe obserwacje mogą sygnalizować kryzysy ekonomiczne lub inne ekstremalne zdarzenia. W każdym przypadku kluczowe jest dokładne badanie tych obserwacji, aby ustalić ich przyczynę i potencjalny wpływ na wyniki analizy.

Odchylenie

W terminologii statystycznej często spotykamy się też z terminem odchylenie (*ang. deviations*), który opisuje różnicę między rzeczywistą obserwacją a przewidywaną wartością, taką jak średnia. Odchylenie jest bardziej ogólnym pojęciem, które odnosi się do stopnia, w jakim poszczególne dane różnią się od wartości oczekiwanej w zbiorze danych. Może być używane w różnych kontekstach statystycznych do opisywania, jak bardzo dane odbiegają od oczekiwań. Odchylenia mogą być zarówno dodatnie, jak i ujemne, co oznacza, że obserwacja może znajdować się powyżej lub poniżej przewidywanej wartości. Nie muszą one być ekstremalne ani rzadkie, mogą występować w każdej próbce danych i stanowić naturalną część zmienności statystycznej. Ważne jest zrozumienie, że odchylenia nie zawsze sygnalizują problem, stanowią one integralną część analizy danych, pozwalając na lepsze zrozumienie rozkładu i zmienności badanych zjawisk. Zatem odchylenie jest szerszym pojęciem dotyczącym różnicy między obserwacją a wartością przewidywaną, a obserwacja odstająca czy anomalia jest specyficznym przypadkiem dużego odchylenia, które może wpływać na wyniki analiz statystycznych.

Nowość

Termin nowość (*ang. novelty*) odnosi się do mechanizmu, za pomocą którego inteligentny system potrafi zidentyfikować nowe wzorce jako wcześniej nieznane. W literaturze naukowej można spotkać następujące definicje:

Wykrywanie nowości (*ang. novelty detection*) jest procesem identyfikacji wzorców w danych, które wcześniej nie były zaobserwowane, takich jak nowy temat dyskusji na forum internetowym lub pojawienie się nowego gatunku muzycznego na platformie streamingowej. W przeciwieństwie do anomalii, które są odstępstwami od normy, nowe wzorce po ich zidentyfikowaniu zazwyczaj stają się częścią standardowego modelu [21].

Markou i Singh w pracy [23] definiują nowość jako proces identyfikacji nowych lub nieznanych danych oraz sygnałów, które nie były dostępne dla systemu uczącego się podczas fazy treningowej. Podkreślają, że jest to jedno z kluczowych wymagań dla efektywnego systemu klasyfikacji lub identyfikacji, ponieważ dane testowe mogą zawierać informacje o obiektach nieznanych w momencie trenowania modelu. Dobry model musi być w stanie odróżnić znane informacje od nieznanych podczas testowania.

W drugiej części swojej pracy, autorzy koncentrują się na podejściach opartych na sieciach neuronowych do wykrywania nowości [24]. Według autorów sieci neuronowe, takie jak perceptrony wielowarstwowe, mapy samoorganizujące się i funkcje radialne, mogą być efektywnie wykorzystywane do wykrywania nowości. Wskazują również na potrzebę rozwijania adaptacyjnych sieci neuronowych, które mogą automatycznie dostosowywać swoją strukturę w odpowiedzi na nowe dane. Przykładem może być wykorzystanie algorytmów genetycznych do samodzielnego generowania nowych przykładów treningowych podczas uczenia się. Zwracają uwagę na to, że adaptacyjne metody neuronowe mają pewne zalety w porównaniu do metod statystycznych, na przykład brak a priori założeń co do właściwości danych.

W pracy [25] autorzy Saunders i Gero redefiniują pojęcie nowości, łącząc jej aspekt innowacyjny z użytecznością. Ich zdaniem nowość to coś więcej niż tylko coś nowego w dosłownym znaczeniu; musi również posiadać wartość użytkową w kontekście kreatywnego projektowania. Kreatywne projektowanie to proces prowadzący do zwiększenia liczby możliwych koncepcji. W tym ujęciu nowość polega na wprowadzeniu reprezentacji, które umożliwiają generowanie nowych pomysłów i rozwiązań. Anomalia jako nowość to nowatorski, nieprzewidywalny element w danych, który ma istotny wpływ na analizowany system. Nowość definiowana jest przez autorów jako zdolność do wprowadzania nowych, nieprzewidywalnych elementów, które znacząco zmieniają sposób, w jaki algorytm projektujący postrzega i rozwiązuje problemy. Sytuacje przewidywalne na podstawie wcześniejszych doświadczeń, nawet jeśli nigdy wcześniej nie były bezpośrednio doświadczane, nie są uważane za nowe według tej definicji. Nowość jest sytuacją, której nie można było przewidzieć na podstawie wcześniejszych doświadczeń, co oznacza, że system (algorytm) projektujący musi zaktualizować swoje wewnętrzne modele i wiedzę, aby uwzględnić te nowe elementy. Innymi słowy, nowość to coś, co znacząco zmienia sposób, w jaki

algorytm projektujący postrzega i rozwiązuje problemy, wprowadzając elementy, które wcześniej nie były brane pod uwagę. Mechanizm wykrywania nowości jest istotny w wielu dziedzinach, takich jak bezpieczeństwo systemów, diagnostyka medyczna czy analiza finansowa, gdzie identyfikacja nieznanych wcześniej wzorców może dostarczyć cennych informacji i umożliwić szybkie reagowanie na nowe zagrożenia lub szanse. Wszystkie wymienione definicje nowości mają wspólny element, który je łączy: nowość odnosi się do identyfikacji nowych, wcześniej nieznanych danych lub wzorców, które nie były dostępne lub znane podczas fazy treningowej systemu.

2.1.2 Problemy z definicją i terminologią - nowe perspektywy

W nauce i analizie danych czasami podejmowane są próby wyjaśnienia subtelnych, aczkolwiek istotnych różnic w terminach takich jak anomalia, obserwacja odstająca i nowość. Choć na pierwszy rzut oka mogą one wydawać się podobne, badacze udowadniają, że każdy tych terminów niesie ze sobą unikalne znaczenie i konsekwencje. Na przykład w świecie samochodów motocykl stanowiłby dla badacza anomalię – coś zupełnie nieoczekiwanego i odmiennego od normy. Z kolei bardzo rzadki model samochodu, taki jak DeLorean, byłby obserwacją odstającą – przypadkiem niezwykle rzadkim, ale mieszczącym się w ramach oczekiwań. Natomiast nowy, innowacyjny model samochodu elektrycznego symbolizowałby nowość – coś, co wnosi świeże, nieznane wcześniej elementy do znanej przestrzeni. Mimo prób takich szczegółowych rozróżnień, terminy te, a także wiele innych, są często używane zamiennie, co prowadzi do niejednoznaczności i trudności interpretacyjnych.

W 2021 roku Foorthuis w swojej pracy [26] zaproponował pierwszą teoretycznie uzasadnioną typologię anomalii danych, która jest niezależna od dziedziny. Autor przedstawia wszechstronną typologię anomalii, definiującą pięć kluczowych wymiarów: typ danych, kardynalność związku, poziom anomalii (rozróżnienie pomiędzy pojedynczymi, niskopoziomowymi przypadkami lub punktami danych a zagregowanymi grupami lub strukturami zbiorowymi), strukturę danych i rozkład danych. Dzięki tym wymiarom anomalie są klasyfikowane w trzy główne grupy: anomalie jednowymiarowe, wielowymiarowe i zagregowane. Każda z tych grup obejmuje trzy podstawowe typy: anomalię liczbową, kategoryczną i mieszaną, które mogą występować w każdej z tych grup, co daje łącznie dziewięć typów. Każdy z dziewięciu podstawowych typów anomalii może być dalej podzielony na bardziej szczegółowe kategorie. W sumie istnieją 63 podtypy, z których każdy opisuje specyficzne przypadki odchyleń w danych, uwzględniając różne aspekty i kombinacje atrybutów. Na przykład wielowymiarowa anomalia numeryczna może obejmować podtypy takie jak punkty znajdujące się na peryferiach zbioru danych, punkty otoczone przez normalne dane, ale nadal mające nietypowe wartości w porównaniu z otaczającymi je punktami, oraz punkty o znacząco różnej gęstości lokalnej w porównaniu z ich sąsiadami, itp. Według autora tak szczegółowa klasyfikacja umożliwia dogłębne zrozumienie oraz efektywne zarządzanie anomaliami w różnych kontekstach badawczych i praktycznych, zwłaszcza w analizach wielowymiarowych zbiorów danych, gdzie trady-

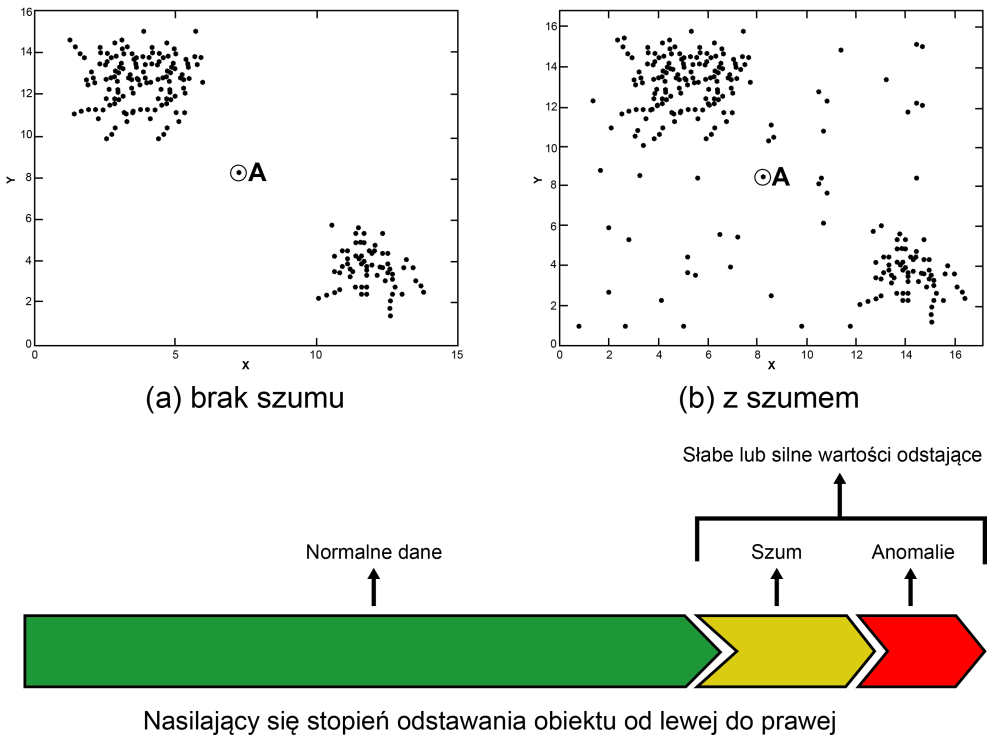
cyjne metody mogą być niewystarczające. Przedstawiona typologia może mieć istotne znaczenie dla projektowania algorytmów wykrywania anomalii oraz dla wyjaśnialności procesów analitycznych w naukach o danych. Zrozumienie różnorodnych typów anomalii może pozwolić na bardziej precyzyjne projektowanie i ocenę algorytmów, co zwiększa transparentność i możliwości interpretacji wyników.

W kontekście rosnącej krytyki metod „czarnej skrzynki" w sztucznej inteligencji, dogłębne zrozumienie natury anomalii jest decydujące dla poprawy przejrzystości i wiarygodności analiz. Autor podkreśla, że jego typologia może mieć szerokie zastosowanie w wielu dziedzinach, od cyberbezpieczeństwa po medycynę, gdzie precyzyjne wykrywanie anomalii jest niezwykle istotne. W dobie ogromnych zbiorów danych, taka typologia może umożliwić identyfikację nie tylko pojedynczych anomalii, ale również bardziej złożonych wzorców, co jest niezbędne dla nowoczesnych systemów analizy danych. Foorthuis sugeruje również, że przyszłe badania powinny skupić się na integracji tej typologii z technikami wyjaśnialnej sztucznej inteligencji (*ang. explainable artificial intelligence*, XAI), co umożliwi lepsze zrozumienie i interpretację wyników analiz anomalii. Przyszłe badania mogą także zbadać możliwość rozszerzenia typologii na inne rodzaje danych, takie jak dane czasowe i przestrzenne, co jeszcze bardziej zwiększy jej uniwersalność i użyteczność.

2.1.3 Szum

Wykrywanie anomalii jest powiązane z eliminacją szumu (*ang. noise*) i radzeniem sobie z nim. Szum to element w danych, który nie jest istotny dla analityka i przeszkadza w analizie. Usunięcie szumu polega na eliminacji tych niepożądanych elementów przed przystąpieniem do analizy. Zarządzanie szumem polega na dostosowaniu modelu tak, aby mimo obecności szumu (czyli przypadkowych zakłóceń, nieistotnych danych) nadal działał poprawnie i dostarczał wiarygodnych wyników. Szum jest najczęściej losowym zakłóceniem dodawanym do danych podczas ich generowania. Oznacza to, że rzeczywiste dane mogą być zniekształcone przez przypadkowe zmiany, które wprowadzają niepewność. Na przykład, podczas pomiaru zawsze istnieje pewien margines błędu, który może wpłynąć na wynik. Im więcej szumu, tym trudniej jest dokładnie określić prawdziwy charakter danych, ponieważ te losowe zakłócenia mogą maskować rzeczywiste informacje. Standardowo zakłada się, że szum jest równomiernie rozłożony (czyli jego średnia wartość to zero) i rozprzestrzenia się jednakowo we wszystkich kierunkach. Aby skutecznie wykrywać obserwacje odstające, trzeba umieć odróżniać je od szumu. Jednym ze sposobów jest użycie wcześniej znanych przykładów obserwacji odstających podczas uczenia maszynowego. W kontrolowanych warunkach, gdzie mamy oznaczone dane, możemy nauczyć algorytm rozpoznawać te odstające przypadki. W nienadzorowanych warunkach, gdzie dane nie są oznaczone, trudniej jest wyznaczyć granicę między normalnymi danymi a odstającymi. W takich sytuacjach szum może być mylony z obserwacjami odstającymi, ponieważ brak oznaczeń utrudnia jednoznaczną identyfikację anomalii.

Rzetelność oceny, czy dany punkt danych jest wystarczająco odchylony, aby zostać



Rysunek 2.2: Różnica między szumem a anomalią. Źródło: opracowanie własne na podstawie [27].

uznany za odstający, często zależy od subiektywnej oceny badacza. Ilustruje to rysunek 2.2 (a) i (b). Porównując oba wykresy, zauważamy, że w przypadku (b) znacznie trudniej jest jednoznacznie stwierdzić, że punkt oznaczony jako „A" stanowi rzeczywiste odchylenie od pozostałych danych. Bardzo możliwe, że punkt ten reprezentuje losowo rozmieszczony szum. Szum jest często postrzegany jako słabsza forma wartości odstających, która nie zawsze spełnia surowe kryteria wymagane do uznania punktu za istotnie odchylony. Dlatego niektórzy badacze wprowadzają rozróżnienie pomiędzy słabymi wartościami odstającymi a silnymi wartościami odstającymi, aby lepiej oddzielić szum od rzeczywistych anomalii [27].

2.2 Złożone zbiory danych a identyfikacja anomalii

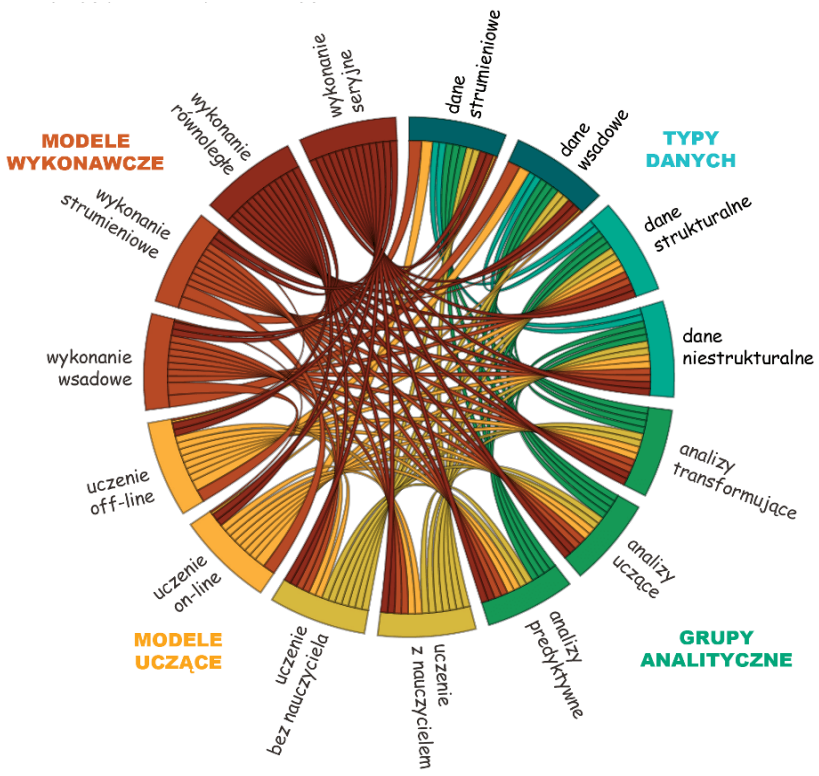
W dobie dynamicznego rozwoju technologii i rosnącej ilości generowanych danych, zbiory danych stają się coraz bardziej złożone. Duże zbiory danych (*ang. big data*) obejmują różnorodne źródła i formy informacji, takie jak posty i komentarze na mediach społecznościowych, dane transakcji internetowych, wyniki eksperymentów naukowych, oraz dane zautomatyzowane przez technologie Web mining i Web crawling [28]. Dodatkowo, dane

grafowe obrazujące relacje między stronami WWW tworzą skomplikowane sieci powiązań. Złożoność dużych zbiorów danych charakteryzuje się pięcioma kluczowymi cechami: ogromną objętością (*ang. volume*), dużą szybkością narastania (*ang. velocity*), różnorodnością struktury i treści (*ang. variety*), wiarygodnością i dokładnością (*ang. veracity*), oraz faktyczną wartością dla procesów decyzyjnych (*ang. value*). Na złożoność danych najbardziej wpływają trzy z pięciu kluczowych cech: ogromna objętość, duża szybkość narastania oraz różnorodność struktury i treści. Pozostałe dwie cechy mają znaczenie, ale ich wpływ na złożoność jest bardziej pośredni.

Wzrost liczby obserwacji może znacząco obciążyć systemy informatyczne, prowadząc do przeciążenia i wzrostu liczby błędów identyfikacyjnych. Aby umożliwić precyzyjne wykrywanie anomalii, dane muszą być analizowane z najwyższą dokładnością. Zasadniczym elementem tego procesu jest utrzymanie wysokiego poziomu szczegółowości danych, czyli ich podział na mniejsze, bardziej precyzyjne jednostki, co jest niezbędne do skutecznego odróżniania obserwacji odstających od reszty danych. W miarę jak dane stają się bardziej złożone, metody identyfikacji anomalii muszą być w stanie radzić sobie z wieloma atrybutami (wielowymiarowość) i różnorodnymi typami danych. Mimo intensywnych badań naukowych i osiągnięć, analiza i identyfikacja anomalii nie osiągnęły jeszcze pełnej dojrzałości naukowej. Dziedzina ta wciąż wymaga kompleksowego i zintegrowanego podejścia teoretycznego, które umożliwiłoby głębsze zrozumienie zjawiska anomalii oraz jego różnorodnych form, a także byłoby powszechnie akceptowane i stanowiło wyznacznik. W tym kontekście tradycyjne metody mogą okazać się niewystarczające, biorąc pod uwagę złożoność danych oraz wyzwania związane z efektywnym rozpoznawaniem i klasyfikowaniem anomalii. Dlatego konieczne jest stosowanie zaawansowanych algorytmów, które są w stanie przetwarzać złożone wzorce, by sprostać wyzwaniom, jakie stawia przed nami ta złożona dziedzina badawcza.

Nauka o danych (*ang. data science*) wydobywa wartość z danych, pomagając nam zrozumieć i wykorzystać złożone zbiory danych. Obejmuje gromadzenie, przetwarzanie, analizę i interpretację danych, aby uzyskać użyteczne informacje, między innymi identyfikować anomalie. Identyfikacja anomalii zapewnia jakość i wiarygodność danych, co jest niezbędne do przeprowadzenia dokładnych analiz i podejmowania trafnych decyzji. Każdy projekt związany z nauką o danych składa się z różnych, wzajemnie powiązanych komponentów, które obejmują typy danych do analizy, klasy stosowanych metod analitycznych, zakres używanych modeli uczenia oraz modele wykonawcze niezbędne do przeprowadzenia analizy [29]. Interakcje pomiędzy tymi składnikami tworzą złożony system wzajemnych powiązań w ramach poszczególnych etapów procesu analizy danych. Decyzje podejmowane w odniesieniu do jednego składnika analizy mają wpływ na inne aspekty procesu analitycznego. Na przykład, rodzaje danych determinują wybór klasy analitycznej i modelu uczenia, natomiast ograniczenia czasowe i strategie równoległego przetwarzania algorytmów wpływają na wybór modelu wykonawczego.

Typy danych i stosowane metody analityczne są ze sobą nierozerwalnie powiązane.



Rysunek 2.3: Relacje między komponentami nauki o danych. Źródło: opracowanie własne na podstawie [29].

Cele analityczne, wynikające z założeń biznesowych, są ściśle zależne od charakteru danych, który determinuje wybór odpowiednich metod analizy. Istnieje wiele sposobów klasyfikacji danych, a jednym z najczęściej stosowanych jest podział na dane strukturalne i niestrukturalne. Dane strukturalne cechują się jasno zdefiniowanymi polami o precyzyjnie określonym znaczeniu, co jest typowe dla danych przechowywanych w tabelach relacyjnych. Dane te są kategoryzowane według wyraźnie zdefiniowanych skal pomiarowych, takich jak skale nominalne, porządkowe czy ilorazowe. Z kolei dane niestrukturalne, takie jak tekst w języku naturalnym, obrazy, nagrania wideo i audio, charakteryzują się mniej jasno określonym znaczeniem. Przed ich analizą wymagają one wstępnego przetwarzania w celu identyfikacji i wyodrębnienia istotnych cech. Dane można także klasyfikować według szybkości ich generowania, gromadzenia lub przetwarzania. Wyróżnia się dane strumieniowe, które napływają w sposób ciągły, oraz dane wsadowe, które pojawiają się w określonych odstępach czasu.

Cele analityczne są określane na podstawie celów biznesowych, jednak typ danych również na nie wpływa. Na przykład, zrozumienie preferencji zakupowych klientów jako cel biznesowy może prowadzić do analizy koszyka zakupowego, która wymaga danych

transakcyjnych, takich jak historia zakupów w sklepie internetowym. Do analizy danych stosuje się modele uczenia maszynowego, a ich wybór zależy od charakterystyki danych i celów analitycznych. Modele te mogą obejmować różne techniki, takie jak klasyfikacja, grupowanie, analiza regresji, wykrywanie anomalii itp. Modele wykonawcze dotyczą sposobu przeprowadzania analizy i obejmują strategie równoległego przetwarzania algorytmów, co jest istotne dla skrócenia czasu odpowiedzi (*ang. latency*) i zapewnienia terminowości (*ang. timeliness*). Różne modele wykonawcze są stosowane w zależności od szybkości generowania, gromadzenia i przetwarzania danych, na przykład dane strumieniowe (*ang. streaming data*) wymagają innych modeli wykonawczych niż dane przetwarzane wsadowo (*ang. batch data*).

Interakcje między składnikami nauki o danych tworzą złożony system wzajemnych powiązań w ramach poszczególnych etapów procesu analizy. Rysunek 2.3 ilustruje relacje między różnymi komponentami nauki o danych. Te wzajemne powiązania i zależności wpływają na złożoność procesu analizy danych, którą należy uwzględnić. Złożoność ta wynika z konieczności przetwarzania i interpretacji danych w kontekście ich struktury, źródeł oraz tempa generowania. Każdy komponent analizy danych, taki jak typ danych, metoda analityczna czy model wykonawczy, dodaje kolejne warstwy złożoności, którymi trzeba zarządzać, aby uzyskać dokładne i użyteczne wyniki. W kontekście złożoności kluczowe jest zrozumienie samych danych i ich charakterystyki. Złożone zbiory danych, zarówno strukturalne, jak i niestrukturalne, można zdefiniować poprzez kilka kluczowych właściwości: wielowymiarowość, różnorodność typów danych, złożoność wzorców oraz dynamiczną zmienność. Poniżej omówione zostaną te właściwości.

Wielowymiarowość

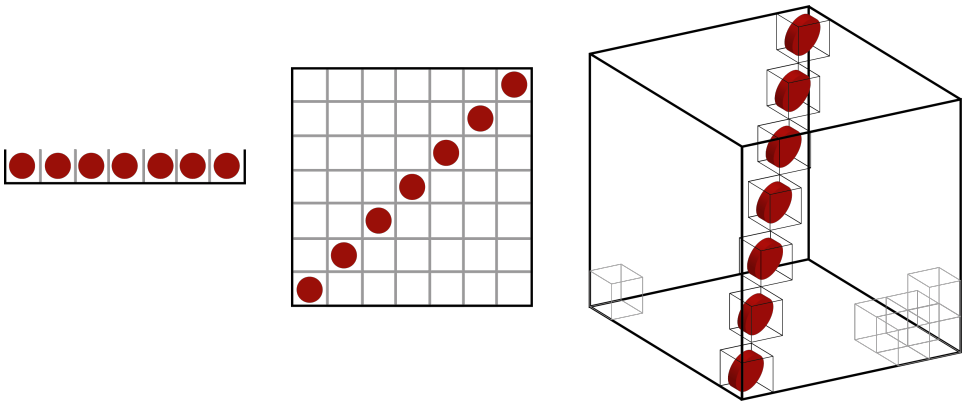
Wielowymiarowość odnosi się do danych, które zawierają wiele atrybutów, które mogą być analizowane jednocześnie. Dla danych strukturalnych jest to łatwe do zrozumienia – na przykład tabela w bazie danych może mieć wiele kolumn (wymiary), takich jak wzrost, kolor oczu, wiek itd. Każda z tych kolumn reprezentuje inny wymiar, który można analizować. W kontekście danych niestrukturalnych, wielowymiarowość jest mniej bezpośrednia, ale nadal obecna. Na przykład, teksty mogą być analizowane pod kątem różnych wymiarów, takich jak częstotliwość występowania słów, sentyment (emocje zawarte w tekście), temat, autor, data publikacji, tokenizacja czy stemming. Obrazy mogą być analizowane pod kątem wielu cech (wymiarów), takich jak kolory, kształty, obiekty, kontekst sceny, krawędzie, gradienty (zmiany intensywności kolorów lub jasności) oraz wzorce (regularne układy pikseli, takie jak tekstury). Wielowymiarowość jest kluczową cechą definiującą złożone zbiory danych, charakteryzującą się obecnością wielu atrybutów opisujących każde obserwowane zjawisko lub przypadek. Ta właściwość może znacznie utrudnić analizę danych z następujących powodów:

- **szeroka gama atrybutów** - wielowymiarowość oznacza, że dane zawierają dużą liczbę zmiennych, które opisują różne aspekty obiektów. Na przykład w medycz-

nym zbiorze danych mogą znajdować się zmienne takie jak wiek pacjenta, wyniki badań laboratoryjnych, historie chorób, objawy kliniczne oraz wyniki obrazowania medycznego. Każda z tych zmiennych wprowadza dodatkowy wymiar do analizy,

- **skomplikowane zależności między atrybutami** - duża liczba atrybutów zwiększa prawdopodobieństwo istnienia skomplikowanych zależności między nimi. Mogą to być zależności liniowe, nieliniowe, interakcje między zmiennymi oraz bardziej subtelne powiązania. Identyfikacja i modelowanie tych zależności jest nieodzowne dla pełnego zrozumienia danych, lecz jest to również trudne zadanie, wymagające zaawansowanych metod analitycznych.

W analizie wielowymiarowych danych napotykamy liczne wyzwania, z których jednym z najważniejszych jest klątwa wymiarowości, pojęcie wprowadzone przez Richarda E. Bellmana (1920-1984, matematyka) [30]. Wysoka liczba atrybutów sprawia, że przestrzeń cech staje się rozproszona, co prowadzi do problemów z efektywnością algorytmów analitycznych oraz dokładnością modeli predykcyjnych. W miarę wzrostu liczby wymiarów przestrzeń ta zwiększa swoją objętość w sposób wykładniczy, co oznacza, że ta sama liczba jednostek wypełnia coraz mniejszą jej część, co pokazano na rysunku 2.4. Klasyczne miary odległości, takie jak odległość euklidesowa, tracą swoją skuteczność, a odległości między punktami danych stają się mniej wyraziste, co utrudnia zadania takie jak klasyfikacja i grupowanie. W wysokowymiarowych przestrzeniach odległość Manhattan okazuje się być bardziej efektywna niż euklidesowa [31]. Ponadto, przestrzeń cech staje się bardziej zaszumiona, ponieważ punkty danych mogą być przypadkowo rozproszone, a odległości między nimi stają się bardziej jednorodne [32]. Wydajność wielu klasyfikatorów osiąga maksymalny poziom przy określonej liczbie cech, a następnie pogarsza się, gdy dodawane są kolejne. To zjawisko wynika z nadmiaru wymiarów, które wprowadzają szum i komplikują model. Wpływ klątwy wymiarowości na problem wykrywania anomalii został po raz pierwszy zauważony w pracy [33].



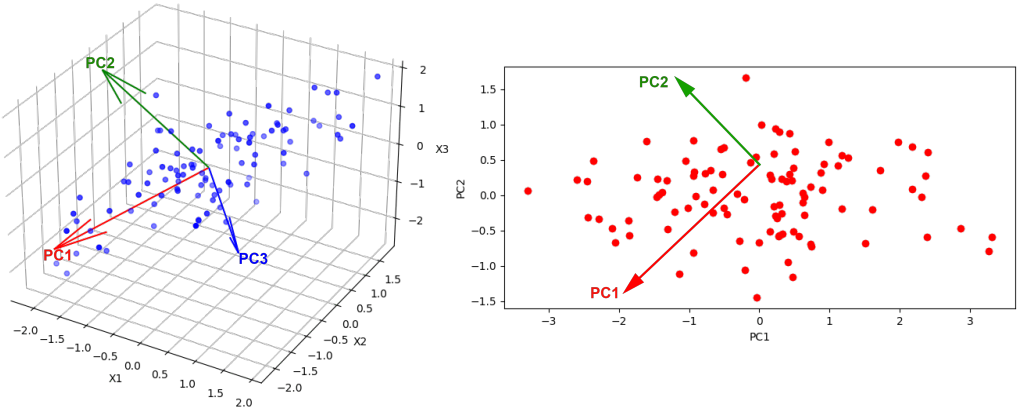
Rysunek 2.4: Klątwa wymiarowości – ta sama liczba jednostek wypełnia coraz mniejszą część przestrzeni w kolejnych wymiarach. Źródło: opracowanie własne.

Kolejnym wyzwaniem jest redundancja informacji. Często wiele atrybutów jest silnie skorelowanych, co prowadzi do nadmiarowości. Aby zaradzić temu problemowi, stosuje się techniki redukcji wymiarowości, takie jak analiza składowych głównych PCA (*ang. principal component analysis*, PCA), która przekształca oryginalne atrybuty w nowy zestaw nieskorelowanych zmiennych, zwanych składowymi głównymi, wyjaśniających większość wariancji w danych. Dzięki temu możliwe jest zmniejszenie liczby atrybutów bez utraty istotnych informacji, co ułatwia analizę i poprawia efektywność modeli predykcyjnych [34]. PCA przekształca dane na nowy układ współrzędnych, który jest liniową kombinacją oryginalnych atrybutów. Na rysunku 2.5 w wyniku zastosowania PCA dane zostały przekształcone z oryginalnych trzech wymiarów (X_1, X_2, X_3) na dwa nowe wymiary (PC_1 i PC_2). PC_1 (pierwsza składowa główna) i PC_2 (druga składowa główna) to nowe współrzędne, które zachowują jak najwięcej informacji o zmienności danych. Nowe współrzędne (zwane składowymi głównymi) są wyznaczane w taki sposób, że największa możliwa wariancja (różnorodność danych) jest reprezentowana w pierwszej składowej głównej. Pierwsza składowa główna to kierunek w nowym układzie współrzędnych, w którym dane mają największą zmienność. To znaczy, że jeżeli patrzymy na dane wzdłuż tej osi, widzimy największe rozproszenie danych. Kolejne składowe główne są ortogonalne (prostopadłe) do poprzednich. Każda kolejna składowa wyjaśnia jak najwięcej pozostałej wariancji, ale mniej niż poprzednia. Dzięki temu, że pierwsze kilka składowych głównych często wyjaśnia większość wariancji w danych, możemy zredukować liczbę wymiarów przestrzeni wielowymiarowej, używając tylko tych składowych.

PCA, dzięki swojej wszechstronności oraz zdolności do identyfikacji wzorców i trendów w złożonych zbiorach danych, jest niezwykle efektywnym narzędziem w eksploracyjnej analizie danych, w tym także w wykrywaniu wartości odstających. Oprócz swojej podstawowej roli w redukcji wymiarowości, PCA może być również wykorzystana do wykrywania anomalii. Metoda ta umożliwia skuteczne identyfikowanie wartości odstających poprzez analizę odległości punktów od składowych głównych w nowo utworzonym układzie współrzędnych [34]. Ostatnim wymienionym tutaj, choć na pewno nie jedynym, ważnym wyzwaniem w pracy z danymi wielowymiarowymi jest złożoność zarządzania zasobami obliczeniowymi. Wysoka wymiarowość danych znacznie zwiększa zapotrzebowanie na pamięć i moc obliczeniową. Aby efektywnie zarządzać dużymi i złożonymi zbiorami danych, musimy skalować algorytmy, aby mogły przetwarzać informacje w sposób optymalny i wydajny. Do tematu wielowymiarowości powrócono w kolejnym rozdziale, w podrozdziale 3.5, gdzie dogłębniej omówiono ten temat.

Różnorodność typów danych

Dane mogą występować w różnych formach i strukturach, co znacząco wpływa na sposoby ich analizy oraz metody wykrywania anomalii. Różnorodność typów danych, takich jak ilościowe, jakościowe i mieszane, umożliwia wszechstronne opisanie charakterystyki obiektów, co zwiększa szanse na odkrycie nietypowych i cennych wzorców. Każdy typ



Rysunek 2.5: Przykład analizy składowych głównych (PCA) na danych 3D, redukcja do dwóch wymiarów. Źródło: opracowanie własne.

danych wnosi unikalne informacje i wyzwania, wymagając odpowiednich metod analitycznych. Na przykład, dane ilościowe umożliwiają zastosowanie technik statystycznych, podczas gdy dane jakościowe, takie jak tekstowe, wymagają metod przetwarzania języka naturalnego. Z kolei dane przestrzenne, często zawierające zarówno elementy ilościowe, jak i jakościowe, co czyni je danymi mieszanymi, wymagają specjalistycznej analizy, która uwzględnia przestrzenny kontekst. Możemy wyróżnić następujące podstawowe rodzaje danych, które razem przyczyniają się do złożoności zbiorów danych, rysunek 2.6:

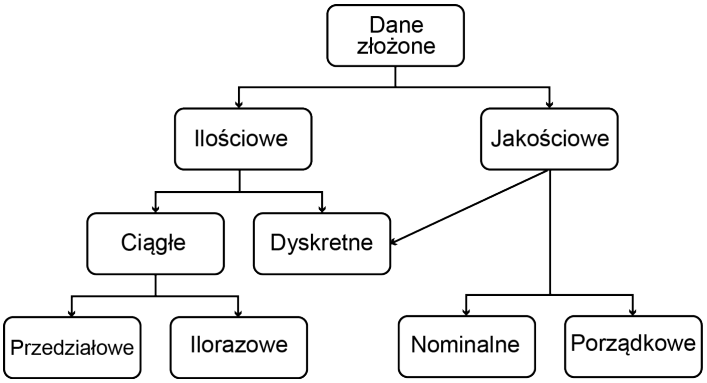
- **ilościowe** (*ang. quantitative*) pozwalające na zastosowanie technik statystycznych, takich jak analiza średnich, wariancji, korelacji i regresji. Są one również ważne dla metod uczenia maszynowego, co umożliwia identyfikację trendów oraz anomalii. Można je mierzyć i wyrażać w postaci liczbowej. Mogą one być dalej podzielone na:
  - **ciągłe** (*ang. continuous*) - dane ilościowe (numeryczne), które mogą przyjmować dowolną wartość w określonym przedziale. Oznacza to, że między dwoma dowolnymi wartościami może istnieć nieskończenie wiele wartości pośrednich. Przykładami takich danych są temperatura, wiek czy wzrost, które można mierzyć z dowolną precyzją,
  - **dyskretne** (*ang. discrete*) dane ilościowe (numeryczne), które mogą przyjmować tylko określone wartości całkowite. Te wartości są oddzielne i mogą być policzone, co oznacza, że między nimi nie istnieją wartości pośrednie. Przykładem takich danych jest liczba osób w gospodarstwie domowym.

Dane dyskretne mogą być także kategoryczne (jakościowe), jak kody pocztowe czy numery identyfikacyjne. Szczególnym przypadkiem atrybutów dyskretnych są dane bi-

narne, które mogą być zarówno ilościowe, jak i jakościowe, w zależności od kontekstu, i przyjmować tylko dwie wartości, np. prawda/fałsz, tak/nie, mężczyzna/kobieta, lub 0/1. Atrybuty binarne są często reprezentowane jako zmienne logiczne lub jako zmienne całkowite przyjmujące tylko wartości 0 lub 1. Dane ilościowe mogą być mierzone na skalach metrycznych: przedziałowych (interwałowych) lub ilorazowych (stosunkowych). Skale przedziałowe pozwalają na mierzenie różnic między wartościami, ale nie mają naturalnego punktu zerowego. Wartość zerowa na tej skali jest zazwyczaj przyjmowana arbitralnie lub na podstawie konwencji [35]. Skale ilorazowe pozwalają na mierzenie różnic między wartościami i mają naturalny punkt zerowy, umożliwiając obliczanie ilorazów wartości.

- **jakościowe** (*ang. qualitative*), które obejmują kategorie opisujące jakość lub cechy zjawisk. Dane jakościowe są używane w wielu badaniach, ponieważ pozwalają na klasyfikację i opis zjawisk, które nie mogą być wyrażone w postaci liczbowej. Dzięki nim można zrozumieć, jak różne kategorie wpływają na badane zjawiska i jakie są ich wzajemne relacje. Podstawowe własności niemetrycznych (jakościowych) skal pomiarowych przedstawiono w podrozdziale 2.3 w tabeli 2.3. Wyróżniamy kategorie:
  - **nominalne** (*ang. nominal*) to kategorie bez naturalnego porządku. Każda wartość nominalna jest wzajemnie wykluczająca się i równorzędna, co oznacza, że nie można ich uszeregować. Przykładami takich danych są płeć, kraj, gatunek zwierzęcia. Binarna zmienna kategoryczna (inaczej zmienna sztuczna, *ang. dummy variable* [36, 37]) to zmienna, która koduje obecność lub brak danej właściwości w sposób binarny, używając wartości 0 i 1. Kodowanie zmiennych jakościowych jako binarnych zmiennych kategorycznych pozwala na ich bezpośrednie wykorzystanie w algorytmach uczenia maszynowego,
  - **porządkowe** (*ang. ordinal*) to kategorie z określonym porządkiem. W przeciwieństwie do danych nominalnych, dane porządkowe można uszeregować, jednak różnice między kolejnymi wartościami nie są mierzalne. Przykładami takich danych są klasy wagowe w sportach walki, stopnie naukowe, oceny jakości usług.
- **mieszane** (*ang. mixed*) dane łączące cechy danych ilościowych i jakościowych. Przykładem mogą być dane przestrzenne, które zawierają zarówno informacje o lokalizacji (dane ilościowe), jak i opisy charakterystyk miejsca (dane jakościowe). Innym przykładem są dane demograficzne, obejmujące wiek (dane ilościowe) oraz płeć (dane jakościowe). Dane ilościowe i jakościowe różnią się skalami pomiarowymi, co utrudnia ich bezpośrednie porównanie. W rezultacie, nie wszystkie algorytmy analizy danych są przystosowane do pracy z danymi mieszanymi.

W pracy [39] autorzy zastosowali różne funkcje podobieństwa do analizy danych mieszanych, wykazując, że odpowiednie funkcje mogą ujawnić wzorce niewidoczne dla tradycyjnych algorytmów. Tradycyjne algorytmy znajdowania częstych zbiorów, takie



Rysunek 2.6: Rodzaje danych złożonych: ilościowe i jakościowe. Źródło: opracowanie własne na podstawie [38].

jak *ObjectMiner*, *STreeDC-Miner* i *STreeNDC-Miner*, zakładają, że dwa podopisy obiektów (wybrane cechy) są podobne, jeśli są identyczne. Wiele rzeczywistych problemów wymaga jednak innych metod oceny podobieństwa. W tym kontekście algorytmy *ObjectMiner* i *STreeDC-Miner* wykorzystują własność domknięcia w dół (*ang. downward closure*) do przycinania przestrzeni wyszukiwania, co może prowadzić do pominięcia ważnych wzorców. Nowe funkcje podobieństwa, nazwane elastycznym przycinaniem (*ang. relaxed prune*), zaproponowane w artykule [39], okazały się skuteczniejsze w analizie rzeczywistych problemów. *RP-Miner*, oparty na tej koncepcji, jest szybszy niż *STreeNDC-Miner* i traci mniej częstych podobnych wzorców niż *ObjectMiner* i *STreeDC-Miner*. Warto zauważyć, że choć algorytmy te nie są bezpośrednio zaprojektowane do wykrywania anomalii, wyniki uzyskane z ich zastosowania mogą czasami pomóc w identyfikacji nietypowych wzorców, które mogą sugerować obecność anomalii. Ponadto, odpowiednie funkcje podobieństwa mogą zwiększyć skuteczność wykrywania anomalii poprzez ujawnienie subtelnych wzorców niewidocznych dla tradycyjnych metod.

Dane przekrojowe, czasowe i sekwencyjne - różnorodność struktur danych

Efektywna analiza danych powinna integrować zarówno dane ilościowe, które dostarczają precyzyjnych i mierzalnych wniosków, jak i dane jakościowe, oferujące bogaty kontekst i głębsze zrozumienie. Aby uzyskać pełny obraz, niezbędne jest zastosowanie obu tych podejść. Równie ważne jest zrozumienie struktur danych, które obejmują anomalie. Strukturalne dane, charakteryzujące się jasno określonym i zorganizowanym formatem, ułatwiają przechowywanie, zarządzanie i analizowanie informacji. Stanowią one przeciwieństwo danych niestrukturalnych, które, pozbawione stałego formatu, wymagają zaawansowanych technik przetwarzania, takich jak narzędzia do analizy języka naturalnego czy algorytmy uczenia maszynowego stosowane w analizie tekstów, obrazów i dźwięków. Analiza danych czasami odbywa się na dokumentach tekstowych, które nie posiadają jednolitej struktury lub są tylko częściowo sformatowane. Niemniej jednak, większość zbiorów danych jest

zorganizowana w wyraźnie strukturalny sposób. Integracja i odpowiednie przetwarzanie zarówno danych strukturalnych, jak i niestrukturalnych, pozwala na osiągnięcie bardziej kompleksowych i trafnych wniosków.

Dane przekrojowe to forma struktury danych, która rejestruje różnorodne zmienne w określonym momencie czasowym. Są one traktowane jako niezależne i nieuporządkowane, co oznacza, że nie ma konkretnego porządku ani wzajemnego wpływu między obiektami. Jest to istotne w kontekście badań, gdyż umożliwia analizę szerokiego przekroju danych w jednym punkcie czasowym. Struktury danych używane do obserwacji zmian w czasie, dotyczących poszczególnych lub wielu obiektów, są zbierane cyklicznie, co pozwala na badanie dynamiki zjawisk lub procesów. Przykładem takich danych są szeregi czasowe, które dokumentują dane dotyczące konkretnej jednostki w regularnych odstępach czasu, umożliwiając analizę ewolucji tych zjawisk w czasie. Dane panelowe, czyli wielowymiarowe zbiory danych składające się z wielu szeregów czasowych, obejmują obserwacje dotyczące wielu obiektów przez dłuższy czas. Pozwalają one na analizę złożonych interakcji czasowych i przestrzennych. W odróżnieniu od danych przekrojowych, dane panelowe dostarczają informacji o zmianach i trendach, które zachodzą w dłuższej perspektywie czasowej. Dane sekwencyjne składają się z sekwencji indywidualnych elementów, takich jak sekwencja słów lub liter. Różnią się one od danych czasowych tym, że nie mają znaczników czasowych, zamiast tego są uporządkowane według pozycji w sekwencji. Przykładem takich danych są dane genetyczne roślin i zwierząt, które mogą być reprezentowane jako sekwencje nukleotydów, tworzących geny. Podsumowując, dane przekrojowe, dane czasowe i dane sekwencyjne reprezentują różne podejścia do struktury danych, z których każde ma swoje unikalne zastosowania i metody analizy.

Struktury danych obejmują zarówno koncepcje organizacji informacji, jak i techniczne formy ich przechowywania, takie jak macierze, tabele czy grafy. Dane przekrojowe, czasowe i sekwencyjne to różne typy zbiorów danych, które można efektywnie przechowywać w strukturach takich jak macierze i tabele. Macierze i tabele są fundamentalne dla implementacji modeli danych. Jednak w bardziej złożonych systemach, gdzie konieczne jest modelowanie relacji między wieloma obiektami, lepiej sprawdzają się modele relacyjne. Te modele umożliwiają tworzenie skomplikowanych struktur danych, w tym projektów dostosowanych do specyficznych potrzeb oraz analitycznych schematów gwiazdy (*ang. star schema*). Schemat gwiazdy jest jedną z najprostszych i najczęściej używanych metod w hurtowniach danych, gdzie centralna tabela faktów jest otoczona przez tabele wymiarów. Relacje między obiektami mogą przekazywać ważne informacje, dlatego często reprezentowane są za pomocą grafów. W grafach obiekty danych są mapowane na węzły, a relacje między nimi na krawędzie, które mogą mieć określony kierunek i wagę. Grafy, składające się z wierzchołków, krawędzi, kierunków i wag, są podstawowym elementem modelowania zaawansowanych struktur, zwłaszcza w takich zastosowaniach jak modelowanie sieci społecznościowych czy systemów sensorowych. Istnieje wiele form grafów, w tym struktury drzewiaste, które charakteryzują się brakiem cykli i hierarchiczną organizacją.

W kontekście przechowywania, grafy są zarządzane jako zestawy wierzchołków i krawędzi, reprezentowane jako listy lub macierze sąsiedztwa. Analogicznie, dane przestrzenne, które obejmują współrzędne i charakterystyki fizyczne, takie jak gęstość zaludnienia czy infrastrukturę, są często przedstawiane w formie punktów, linii czy wielokątów. Mogą być także rastrowane do postaci pikseli, co jest przydatne w analizie obrazów satelitarnych czy skanów medycznych. Dane przestrzenne zawierają atrybuty takie jak pozycje, obszary i inne cechy. Przykładem są dane pogodowe (opady, temperatura, ciśnienie) zbierane dla różnych lokalizacji geograficznych. Ważnym aspektem tych danych jest korelacja przestrzenna, gdzie obiekty fizycznie blisko siebie mają tendencję do podobieństwa w innych aspektach. Dane czasoprzestrzenne, które łączą wymiar czasowy i przestrzenny, umożliwiają szczegółową analizę dynamicznych zjawisk, co jest nieodzowne w kontekście badań historycznych, geograficznych oraz analizy treści wideo. Te techniki i metody stanowią podstawę skutecznej analizy i interpretacji różnorodnych typów danych, w tym wykrywania anomalii.

Wzorce i zmienność w danych złożonych

Złożoność wzorców odnosi się do skomplikowanych relacji i zależności, które mogą występować w danych. Wzorce te mogą obejmować różnorodne kombinacje atrybutów, sekwencje zdarzeń oraz powiązania między obiektami. Złożone dane charakteryzują się następującymi cechami:

- **wysoka kardynalność** - cechy często zawierają dużą liczbę unikalnych kategorii (etykiet),
- **różnorodne rozkłady częstości** - cechy mogą mieć bardzo zróżnicowane rozkłady częstości występowania poszczególnych kategorii, od bardzo częstych do rzadkich,
- **złożone powiązania cech** - wartość jednej cechy może wpływać na znaczenie innej, tworząc skomplikowane zależności,
- **obecność interakcji między cechami** - interakcje między wartościami dwóch lub więcej cech mogą mieć unikalne znaczenie, które nie jest widoczne przy analizie poszczególnych cech oddzielnie,
- **obecność cech szumowych** - niektóre cechy mogą nie wnosić wartości do analizy, wprowadzając mylne informacje lub zakłócenia,
- **bogactwo semantyczne kategorii** - kategorie mogą mieć bogate znaczenia semantyczne, wymagające dogłębnej wiedzy dziedzinowej do efektywnej analizy,
- **zmienna kontekstowa** - znaczenie i ważność poszczególnych cech lub ich kombinacji może się różnić w zależności od kontekstu danych,
- **wyzwania kodowania i przetwarzania** - przekształcenie danych kategorycznych

w format, który można skutecznie wykorzystać w modelach matematycznych i statystycznych, wymaga zaawansowanych technik kodowania, takich jak kodowanie one-hot, kodowanie docelowe lub stosowanie osadzeń (redukcja wymiarowości),

- **anomalie i rzadkie wzorce** - w złożonych danych kategorycznych anomalie mogą być definiowane przez nietypowe kombinacje kategorii cech.

Rysunek 2.7: Pojedyncza próbka złożonych danych z zestawu bank marketing [40] przedstawiająca 17 cech klienta banku. Każda cecha reprezentuje własny typ złożoności. Źródło: opracowanie własne.

Każda zmienna może posiadać odpowiedni typ złożoności, co pomaga lepiej zrozumieć charakterystykę i wyzwania związane z analizą wybranego zestawu danych.

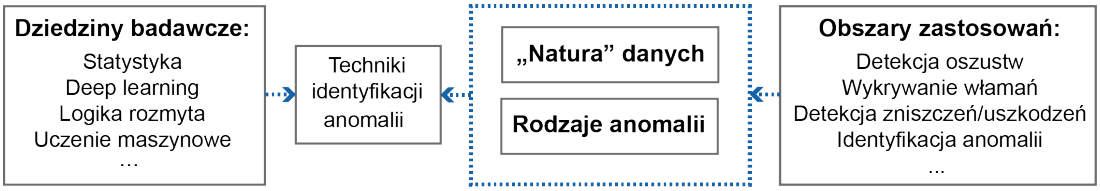
Tabela 2.1: Charakterystyka złożoności zmiennych w zbiorze danych bank marketing [40]. Źródło: opracowanie własne.

Zmienna	**Sugerowane cechy złożoności**
wiek	różnorodne rozkłady częstości
zawód	złożone powiązania cech, wysoka kardynalność, obecność interakcji między cechami, wyzwania kodowania i przetwarzania, bogactwo semantyczne kategorii
stan cywilny	wysoka kardynalność, wyzwania kodowania i przetwarzania
edukacja	bogactwo semantyczne kategorii, wysoka kardynalność, wyzwania kodowania i przetwarzania, obecność interakcji między cechami
domyślny	obecność cech szumowych, bogactwo semantyczne kategorii, zmienna kontekstowa
saldo	różnorodne rozkłady częstości, anomalie i rzadkie wzorce, złożone powiązania cech, obecność interakcji między cechami
hipoteka	zmienna kontekstowa, złożone powiązania cech, obecność interakcji między cechami
kredyt	złożone powiązania cech, zmienna kontekstowa, obecność interakcji między cechami
kontakt	wyzwania kodowania i przetwarzania, obecność cech szumowych
dzień	zmienna kontekstowa, bogactwo semantyczne kategorii
miesiąc	bogactwo semantyczne kategorii, zmienna kontekstowa
czas trwania	różnorodne rozkłady częstości
kamp. rekl.	różnorodne rozkłady częstości, anomalie i rzadkie wzorce
dni poprz. kont.	anomalie i rzadkie wzorce, różnorodne rozkłady częstości, obecność cech szumowych
licz. poprz. kont.	bogactwo semantyczne kategorii, różnorodne rozkłady częstości, złożone powiązania cech, obecność interakcji między cechami
wynik poprz. kamp.	obecność cech szumowych, bogactwo semantyczne kategorii
zapis na lokatę	zmienna celu y

Na rysunku 2.7 przedstawiono próbkę danych mieszanych ze zbioru bank marketing [40], zawierającą 17 atrybutów klienta banku wraz z przypisanymi typami złożoności. Warto zaznaczyć, że wskazane przypisania atrybutów do typów złożoności są pewnymi propozycjami i mogą się zmieniać w zależności od kontekstu analizy. W tabeli 2.1 zademonstrowano więcej sugestii przypisania złożoności dla poszczególnych zmiennych. Zarówno rysunek 2.7, jak i tabela 2.1, ilustrują wyzwania, jakie stoją przed algorytmami, które mają zidentyfikować odchylenia w przedstawionym zbiorze. Dynamiczna zmienność odnosi się do zmienności danych w czasie. Zbiory danych mogą się zmieniać w odpowiedzi na różne czynniki, takie jak sezonowe trendy, zmiany w zachowaniach użytkowników czy fluktuacje rynkowe.

„Natura” danych, rodzaje anomalii i ich wpływ na techniki identyfikacji

„Natura” danych odnosi się do właściwości i cech danych, które są analizowane. Obejmuje to różne aspekty, takie jak rodzaj danych (ilościowe, jakościowe, mieszane), rozkład danych (jak wartości są rozmieszczone w zbiorze danych, np. rozkład normalny, skośny, bimodalny), wielkość danych (liczba próbek i zmiennych), struktura danych (czy dane są przekrojowe, czasowe, sekwencyjne, hierarchiczne, przestrzenne czy relacyjne), jakość danych (czystość danych, np. brakujące wartości, szum, błędy pomiarowe), dynamika danych (czy są statyczne czy dynamiczne) oraz źródło danych (pochodzenie, które może wpływać na jakość i format danych, np. sensory, ankiety, systemy transakcyjne). Czynniki te istotnie wpływają na techniki identyfikacji anomalii, jak pokazano na rysunku 2.8.



Rysunek 2.8: „Natura” danych i rodzaje anomalii oraz ich wpływ na techniki identyfikacji. Źródło: opracowanie własne.

Zrozumienie tych aspektów danych jest ważne dla wyboru odpowiednich metod analizy oraz technik wykrywania anomalii. Rodzaj anomalii odnosi się do specyficznej kategorii lub typu anomalii wykrywanej w danych. Istnieje kilka typów anomalii, z których każdy charakteryzuje się różnymi cechami i wymaga różnych metod detekcji. Anomalie punktowe to pojedyncze obserwacje znacznie różniące się od reszty danych. Anomalie kontekstowe są nietypowe tylko w określonym kontekście. Anomalie kolektywne występują, gdy grupa obserwacji jest anomalią, mimo że poszczególne obserwacje mogą nie być odstające same w sobie. Anomalie przestrzenne dotyczą danych przestrzennych, gdzie anomalia może występować w specyficznej lokalizacji. Anomalie czasowe dotyczą danych czasowych, gdzie anomalie mogą pojawiać się w określonych momentach czasu. Rozróżnienie między tymi typami anomalii jest decydujące, ponieważ różne metody detekcji mogą być bardziej skuteczne w identyfikacji jednego typu anomalii w porównaniu do innych. Oprócz wymienionych obszarów zastosowań, analiza złożonych zbiorów danych odgrywa istotną rolę w wielu innych dziedzinach:

- **systemy wykrywania i zapobiegania włamaniom** - wywołania systemowe, ruch sieciowy oraz inne działania użytkowników są monitorowane przez systemy komputerowe. Nietypowe zachowanie może wskazywać na złośliwe działania lub naruszenia zasad,
- **oszustwa z wykorzystaniem kart kredytowych** - nieautoryzowane użycie karty kredytowej często manifestuje się przez nietypowe wzorce zachowań, takie jak

płatności z innej lokalizacji czy duże transakcje, co może być wykorzystane do wykrywania wartości odstających w transakcjach kartami kredytowymi,

- **ciekawe zdarzenia z czujników** - w wielu aplikacjach czujniki śledzą w czasie rzeczywistym różne warunki środowiskowe i parametry lokalizacji. Nagłe zmiany standardowych odczytów mogą wskazywać na interesujące zdarzenia, takie jak zużycie maszyn przemysłowych, uszkodzenia w częściach urządzeń (np. motory, turbiny) lub nieprawidłowości konstrukcyjne (pęknięcia, odkształcenia),
- **diagnoza medyczna** - wiele urządzeń medycznych, takich jak MRI, PET, EKG, analizuje dane podczas pracy. Nietypowe wzorce w tych danych zazwyczaj odzwierciedlają chorobę,
- **egzekwowanie prawa** - oszustwa w transakcjach finansowych, działalności handlowej czy wyłudzaniu ubezpieczeń zwykle wymagają identyfikacji nietypowych wzorów w danych generowanych przez podmioty dokonujące przestępstw. Często wzorce te można odkryć, analizując wiele działań danej jednostki,
- **nauka o ziemi** - znaczna ilość danych czasoprzestrzennych o wzorcach pogodowych, zmianach klimatycznych czy wzorcach pokrycia terenu jest zbierana za pomocą różnych mechanizmów, takich jak satelity lub teledetekcja. Anomalie w takich danych dostarczają istotnych spostrzeżeń o działalności człowieka,
- **wykrywanie anomalii w tekstach** - wykrywanie nowych tematów, wydarzeń lub nowych tekstów w zbiorze dokumentów, książek, artykułów.

Kategoryczne dane jakościowe, choć stanowią tylko jeden z wielu typów danych w zbiorach złożonych, odgrywają ważną rolę w wielu aplikacjach, od ankietowania po analizę sieci społecznościowych. Ich analiza wymaga specjalnych metod, takich jak kodowanie, aby mogły być efektywnie przetwarzane w modelach analitycznych. Ze względu na ich znaczenie, poświęcono im osobny rozdział 3. Dane stosowane w różnych rzeczywistych aplikacjach, jak analiza rynku, badania biologiczne czy monitoring sieci komputerowych, charakteryzują się wysoką wymiarowością. Zbiory często zawierają liczne cechy kategoryczne oprócz atrybutów numerycznych, co zwiększa ich złożoność. W związku z tym potrzebne są metody detekcji anomalii, które efektywnie skalują się wraz z rosnącą wymiarowością danych. Niska gęstość danych w przestrzeniach wysokowymiarowych dodatkowo komplikuje zadanie wykrywania anomalii, gdyż tradycyjne metody mierzenia odległości między obiektami okazują się niewystarczające. W odpowiedzi na te wyzwania, w rozdziale 3 w podrozdziale 3.5.2, szczegółowo omówiono problematykę wysokowymiarowości danych oraz metody analizy podprzestrzennej w wykrywaniu anomalii w danych kategorycznych. Skupiono się na wykrywaniu anomalii zakotwiczonych w podprzestrzeniach, co pozwala na głębsze zrozumienie i bardziej efektywne rozwiązywanie tych trudnych problemów analitycznych.

2.3 Podstawowe koncepcje matematyczne i obliczeniowe

W rozwoju nowoczesnych algorytmów wykrywania anomalii zasadnicze znaczenie mają podstawy matematyczne i obliczeniowe, które umożliwiają skuteczną analizę i przetwarzanie danych wielowymiarowych. Notacja matematyczna i matematyka stosowana stanowią fundamenty tych procesów, pozwalając nie tylko na reprezentację danych, ale również na ich efektywną obróbkę i analizę. W tym rozdziale omówione zostaną podstawowe koncepcje i definicje niezbędne do zrozumienia i stosowania zaawansowanych technik w praktyce. Notacja matematyczna, zasady algebry liniowej, teoria prawdopodobieństwa i informacji oraz obliczenia numeryczne są podstawą budowy i optymalizacji modeli zdolnych do identyfikacji obiektów odstających w dużych zbiorach danych. W dalszej części przyjrzano się także podstawowym miarom odległości i podobieństwa. Miary te, zwane również miarami proksymalnymi, są technikami matematycznymi stosowanymi do określania stopnia podobieństwa lub różnicy między obiektami w przestrzeni danych. W analizie danych, uczeniu maszynowym i innych dziedzinach nauki o danych, miary proksymalne pozwalają na ocenę i porównywanie obiektów, co jest użyteczne w grupowaniu, klasyfikacji i rekomendacjach. Są także przydatne w wykrywaniu anomalii, umożliwiając identyfikację obiektów, które znacząco różnią się od innych.

2.3.1 Zbiór obiektów i zmienne

Jednym z głównych zadań uczenia maszynowego, rozpoznawania wzorców i eksploracji danych jest konstruowanie dobrych modeli z zestawów danych. Zbiór danych to skończony zbiór obiektów $D = \{X_i\}_{i=1}^{n}$, gdzie X_i to i-ty obiekt w zbiorze. Obiekty mogą być interpretowane zarówno dosłownie, jak i metaforycznie. W kontekście badań obiekt może reprezentować konkretną rzecz, osobę, kategorię abstrakcyjną lub zdarzenie. Przykładami obiektów mogą być: użytkownik aplikacji, urządzenie elektroniczne, gatunek zwierzęcia, pacjent, miasto, firma, jezioro, eksperyment naukowy, supermarket, rynek finansowy. Zbiór danych składa się zazwyczaj z wektorów zmiennych, gdzie każdy wektor zmiennych opisuje obiekt X_i za pomocą zestawu wartości zmiennych. Zmienne nazywane są również cechami lub atrybutami, a obiekt może być nazywany próbką.

Zmienna A_j w wielowymiarowej analizie danych to odwzorowanie, które przypisuje każdemu obiektowi X_i wartość z określonego zbioru Q_j. Każdy obiekt jest opisany przez zmienne (cechy, atrybuty), a liczba tych zmiennych w zbiorze danych D nazywana jest wymiarem lub wymiarowością. W kontekście uczenia maszynowego, *model* to zazwyczaj narzędzie predykcyjne lub reprezentacja struktury danych, którą chcemy opracować lub odkryć na podstawie zbioru danych. Przykłady takich modeli to regresja liniowa, k-najbliższych sąsiadów czy algorytm k-średnich. Proces tworzenia modeli z danych nazywany jest uczeniem lub treningiem i realizowany jest za pomocą algorytmu uczącego. Otrzymany model, oznaczany jako $h(\cdot)$, często określany jest jako hipoteza lub model uczący

się. Hipotezy te są wybierane z pewnego zbioru hipotez, oznaczanego jako $\mathcal{H}$. Wyróżniamy różne rodzaje uczenia, z których najczęściej spotykane to uczenie nadzorowane i nienadzorowane. W uczeniu nadzorowanym modele uczą się na oznaczonych danych wejściowych i wyjściowych, natomiast w uczeniu nienadzorowanym modele odkrywają wzorce i struktury w nieoznakowanych danych. Notacja matematyczna używana do reprezentowania danych, wraz z podstawami algebry macierzy, stanowi fundamenty modelu obliczeniowego stosowanego w rozwijaniu algorytmów wykrywania wartości odstających w danych wielowymiarowych. W kontekście wielowymiarowej analizy danych korzysta się ze zbioru danych D i zmiennych A_j, aby dokładnie opisać i przeanalizować dane. Poniżej podsumowano najważniejsze definicje:

- **zbiór danych** (zbiór obiektów, próbek): to zestaw danych $D = \{X_i\}_{i=1}^{n}$, w którym każdy obiekt jest opisywany przez wartości cech (zmiennych),
- **zmienna** (cecha, atrybut): zmienna A_j przyjmuje wartość ze zbioru Q_j, który może zawierać liczby rzeczywiste lub kategorie, w zależności od typu zmiennej:
  - zmienne metryczne: obejmują skale ilorazowe i przedziałowe, które są często używane jako dane wejściowe do modeli regresji, sieci neuronowych i innych algorytmów uczenia maszynowego,
  - zmienne niemetryczne: obejmują skale porządkowe i nominalne, które mogą być przekształcane na zmienne numeryczne za pomocą technik takich jak kodowanie do postaci binarnego wektora (*ang. one-hot encoding*), co jest przydatne w modelach klasyfikacji, grupowania, wykrywania anomalii.

Skale pomiarowe są uszeregowane od najsłabszej do najmocniejszej: nominalna, porządkowa, przedziałowa, ilorazowa. Tabela 2.3 przedstawia kluczowe właściwości niemetrycznych skal pomiarowych. Więcej szczegółów na temat skal można znaleźć w literaturze naukowej, w tym w pracy [41]. Szczegółowe informacje na temat zmiennych metrycznych (ilościowych) i niemetrycznych (jakościowych) są dostępne we wcześniejszym rozdziale, w podrozdziale 2.2, który omawia różnorodność typów danych.

Macierz danych D pokazana w tabeli 2.2 to sposób reprezentacji informacji zbieranych w badaniach. Każdy wiersz macierzy reprezentuje obiekt, a każda kolumna odpowiada jednej zmiennej. W kontekście poprzednich definicji, gdzie obiekty danych X_i były opisane przez zmienne $A_1, A_2, \ldots, A_j$ i każdy obiekt mógł być reprezentowany jako zestaw wartości tych zmiennych, macierz danych prezentuje te informacje w uporządkowany sposób. W tej reprezentacji każde X_i odpowiada obiektowi i-temu w zbiorze danych D. A_j reprezentuje j-tą zmienną, która może być metryczna lub niemetryczna w zależności od rodzaju zmiennej. Wartości $x_{i,j}$ w macierzy odpowiadają wartościom zmiennych A_j dla danego obiektu X_i, wybranym z określonego zbioru Q_j.

Tabela 2.2: Reprezentacja macierzy danych w badaniach. Źródło: opracowanie własne.

Obiekt / Zmienna	A_1	A_2	A_3
X_1	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$
X_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_n	$x_{n,1}$	$x_{n,2}$	$x_{n,3}$

Tabela 2.3: Właściwości niemetrycznych skal pomiarowych. Źródło: opracowanie własne na podstawie prac [42] i [43].

Rodzaj skali	**Dozwolone transformacje**	**Możliwe relacje**	**Możliwe operacje**	**Dopuszczalne statystyki**
Nominalna	$z = f(x_{i,j})$, $f(x_{i,j})$ – dowolne wzajemnie jednoznaczne przekształcenie, permutacje	identyczność ($x_{i,j} = x_{k,j}$), różnica ($x_{i,j} \neq x_{k,j}$)	liczenie zdarzeń (liczba relacji identyczności, różnicy)	Liczba przypadków, moda (tylko dla zmiennych dyskretnych), korelacja kontyngencji
Porządkowa	$z = f(x_{i,j})$, $f(x_{i,j})$ – dowolna ściśle monotonicznie rosnąca funkcja	równoważność, różnica oraz większy niż ($x_{i,j} > x_{k,j}$), mniejszy niż ($x_{i,j} < x_{k,j}$)	liczenie zdarzeń (liczba relacji równoważności, różnicy, większy niż, mniejszy niż)	Mediana, percentyle, korelacja porządkowa

Normalizacja zmiennych, czyli przekształcenie ich wartości do porównywalnych skal (traktowana jako szczególny przypadek ważenia zmiennych), jest istotnym etapem przetwarzania danych, wpływającym na skuteczność modeli. Niektóre formuły normalizacyjne można znaleźć w tabeli 2.4, a obszerne opisy tych metod znajdują się w pracach [44, 45]. W tabeli 2.4:

- z_{ij} – znormalizowana wartość j-tej zmiennej A_j dla i-tego obiektu (próbki) X_i,
- $\overline{x_j}$ – średnia wartość zmiennej A_j,
- σ_j – odchylenie standardowe zmiennej A_j,
- Med_j – mediana zmiennej A_j,
- MAD_j – medianowe odchylenie bezwzględne zmiennej A_j,
- R_j – rozstęp (*ang. range*) zmiennej A_j,
- $\sum_{i=1}^{n} x_{ij}$ – suma wartości zmiennej A_j dla wszystkich obiektów,
- $\max(x_{ij})$ – maksymalna wartość zmiennej A_j dla i-tego obiektu.

Tabela 2.4: Rodzaje transformacji normalizacyjnych. Źródło: opracowanie własne na podstawie [41].

Nazwa formuły	**Formuła**
bez normalizacji	–
standaryzacja	$z_{ij} = \frac{x_{ij} - \overline{x_j}}{\sigma_j}$
standaryzacja Webera	$z_{ij} = \frac{x_{ij} - \mathrm{Med}_j}{1.4826 \cdot \mathrm{MAD}_j}$
unitaryzacja	$z_{ij} = \frac{x_{ij} - \overline{x_j}}{R_j}$
unitaryzacja zerowana	$z_{ij} = \frac{x_{ij} - \min_{i=1}^{n}(x_{ij})}{R_j}$
normalizacja w przedziale [-1;1]	$z_{ij} = \frac{x_{ij} - \overline{x_j}}{\max_{i=1}^{n}(\lvert x_{ij} - \overline{x_j} \rvert)}$
przekształcenia ilorazowe	$z_{ij} = \frac{x_{ij}}{\sigma_j}$
	$z_{ij} = \frac{x_{ij}}{R_j}$
	$z_{ij} = \frac{x_{ij}}{\max_{i=1}^{n} x_{ij}}$
	$z_{ij} = \frac{x_{ij}}{\overline{x_j}}$
	$z_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}}$
	$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}}$

2.3.2 Podstawy matematyki stosowanej w systemach uczących się

Podstawowe pojęcia matematyczne stanowią rdzeń wielu algorytmów stosowanych w systemach uczących się. Algebra liniowa, prawdopodobieństwo, teoria informacji oraz obliczenia numeryczne to ważne dziedziny matematyki stosowanej, które dostarczają narzędzi niezbędnych do analizy, modelowania i optymalizacji procesów związanych z uczeniem maszynowym i głębokim uczeniem. Zrozumienie tych pojęć pozwala na projektowanie i implementację algorytmów znajdujących zastosowanie w takich dziedzinach jak rozpoznawanie obrazów i przetwarzanie języka naturalnego. W tej sekcji przedstawiono koncepcje matematyczne niezbędne do zrozumienia funkcjonowania systemów uczących się:

- **algebra liniowa** - fundamentalna dziedzina matematyki zajmująca się operacjami na wektorach i macierzach, które są podstawą wielu algorytmów,
- **teoria prawdopodobieństwa** - matematyczny system reprezentacji niepewności, stanowiący podstawę modelowania i przewidywania zdarzeń losowych,

- **teoria informacji** - umożliwia ocenę niepewności poprzez rozkłady prawdopodobieństwa, pozwalając zrozumieć ilość i przepływ informacji w systemach,
- **obliczenia numeryczne** - metody przybliżonego rozwiązywania równań i optymalizacji, kluczowe w implementacji algorytmów bez analitycznych rozwiązań.

Więcej informacji można znaleźć w literaturze naukowej i podręcznikach do matematyki stosowanej. Celem sekcji jest zaprezentowanie ogólnego zakresu oraz podstawowych pojęć, które stanowią bazę teoretyczną dla zaawansowanych technik i algorytmów uczenia maszynowego. Obszerne opracowania na ten temat można znaleźć w takich pozycjach jak: [46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64].

Algebraiczne właściwości macierzy

Algebra liniowa to podstawowa dziedzina matematyki, niezbędna dla zrozumienia i stosowania wielu algorytmów uczenia maszynowego, zwłaszcza w głębokim uczeniu. Jest szeroko wykorzystywana w nauce i inżynierii, dostarczając narzędzi do przetwarzania i analizy dużych zbiorów danych. Algebra liniowa obejmuje podstawowe pojęcia takie jak skalary, wektory, macierze i tensory. Zrozumienie tych elementów jest konieczne dla skutecznego wykorzystania licznych algorytmów uczenia maszynowego. Szczegółowe informacje na ten temat można znaleźć w podręcznikach [46, 47, 50].

Skalary, wektory i macierze stanowią podstawowe elementy algebry liniowej. Skalar można traktować jako macierz o jednym elemencie, a wektory jako macierze z jedną kolumną. Macierze są szczególnie ważne w analizie danych, oferując szerokie możliwości obliczeniowe i analityczne. Wśród różnych typów macierzy, macierze diagonalne wyróżniają się strukturą, w której elementy poza główną przekątną są zerami. Te macierze są wydajne obliczeniowo, ponieważ mnożenie przez macierz diagonalną sprowadza się do skalowania elementów wektora przez wartości na przekątnej. Macierz jednostkowa, będąca szczególnym przypadkiem macierzy diagonalnej, charakteryzuje się jedynkami na głównej przekątnej, co czyni ją neutralnym elementem w operacjach mnożenia macierzy. Macierze mogą być prostokątne, ale ich odwracalność jest ograniczona do przypadków, gdy wszystkie elementy na przekątnej są różne od zera. Macierz diagonalna jest odwracalna, jeśli $\mathrm{diag}(v)^{-1} = \mathrm{diag}([1/v_1, \ldots, 1/v_n])$, gdzie $\mathrm{diag}(v)$ oznacza macierz diagonalną utworzoną z wektora $\mathrm{x} = [x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$, co upraszcza obliczenia. Macierz symetryczna, równa swojej transpozycji ($D = D^T$), oraz macierz ortogonalna, gdzie wiersze i kolumny tworzą zbiory wektorów ortonormalnych ($D^T D = DD^T = I$), to przykłady macierzy o interesujących właściwościach algebraicznych. Macierze symetryczne często pojawiają się tam, gdzie relacje między parametrami są wzajemne, na przykład w pomiarach odległości. Podstawowe operacje i właściwości macierzy, takie jak wyznacznik, ślad i rząd, odgrywają decydujące role w analizie danych. Wyznacznik macierzy kwadratowej wskazuje na jej odwracalność, ślad jest sumą wartości własnych, a rząd określa liczbę liniowo niezależnych kolumn. Te właściwości są fundamentem zaawansowanych metod, takich jak analiza składowych głównych PCA czy rozkład na wartości osobliwe SVD

(*ang. singular value decomposition*, SVD), które są powszechnie stosowane w redukcji wymiarów, analizie obrazów i statystyce. W zaawansowanych zastosowaniach, często potrzebujemy operować na tablicach o większej liczbie wymiarów niż tradycyjne macierze. Te wielowymiarowe tablice nazywamy tensorami. Tensor to uogólnienie pojęć skalara (pojedyncza liczba), wektora (jednowymiarowa tablica) i macierzy (dwuwymiarowa tablica) na dowolną liczbę wymiarów. Taki tensor może mieć trzy, cztery, a nawet więcej osi. Dzięki tensorom możemy reprezentować złożone struktury danych, co jest niezwykle użyteczne w wielu dziedzinach, takich jak przetwarzanie obrazów, gdzie piksele mogą być zorganizowane w trójwymiarowe tablice (wysokość, szerokość, głębokość kolorów), czy w przetwarzaniu sygnałów, gdzie czas, częstotliwość i inne parametry mogą tworzyć wielowymiarowe układy. Transpozycja macierzy jest również istotnym działaniem, polegającym na zamianie miejscami wierszy i kolumn, co jest niezbędne w wielu algorytmach i operacjach matematycznych. Wektory można traktować jako macierze zawierające tylko jedną kolumnę, a ich transpozycja przekształca je w macierz o jednym wierszu.

Prawdopodobieństwo i teoria informacji

Teoria prawdopodobieństwa umożliwia formułowanie i analizowanie twierdzeń w warunkach niepewności, natomiast teoria informacji pozwala na ilościowe ocenianie tej niepewności za pomocą rozkładów prawdopodobieństwa. Głębokie zrozumienie tych zagadnień można znaleźć w książkach [48, 49, 50, 51, 52, 53].

Teoria prawdopodobieństwa to matematyczny system reprezentacji i pomiaru niepewności, który odgrywa kardynalną rolę w uczeniu maszynowym. Umożliwia analizę i modelowanie danych obarczonych niepewnością, co pozwala algorytmom podejmować decyzje na podstawie niekompletnych i niedokładnych informacji. Systemy uczące się często muszą radzić sobie z danymi, które są niepełne lub zawierają różne stopnie niepewności. Rozkład prawdopodobieństwa opisuje, z jakim prawdopodobieństwem zmienna losowa lub zestaw zmiennych losowych przyjmie każdą z możliwych wartości. Dzięki teorii prawdopodobieństwa możliwe jest formalne modelowanie niepewności oraz określenie stopnia pewności co do wystąpienia różnych zdarzeń.

Pojęcia takie jak wartość oczekiwana, wariancja i kowariancja są ściśle związane z teorią prawdopodobieństwa, a ich zrozumienie jest nieodzowne zarówno w uczeniu maszynowym, jak i wykrywaniu anomalii. Wartość oczekiwana zmiennej losowej to średnia wartość, jaką zmienna ta przyjmuje w danym rozkładzie prawdopodobieństwa. Stanowi ona miarę centralnej tendencji i jest wykorzystywana w procesach standaryzacji i normalizacji danych. Wariancja mierzy rozproszenie wartości zmiennej losowej wokół jej wartości oczekiwanej, dostarczając informacje o stabilności i pewności modelu. Kowariancja określa stopień liniowego związku między dwiema zmiennymi losowymi, pokazując, jak zmiana jednej zmiennej wpływa na zmianę drugiej.

Rozkłady prawdopodobieństwa, takie jak rozkład normalny (Gaussa) i rozkład Bernoulliego, są szeroko stosowane w modelowaniu danych. Rozkład normalny modeluje dane ciągłe, a rozkład Bernoulliego zmienne binarne. Inne ważne rozkłady to rozkład

wykładniczy, stosowany, gdy chcemy uzyskać rozkład prawdopodobieństwa z ostrym punktem dla $x = 0$, oraz rozkład Laplace'a, w którym większość prawdopodobieństwa jest skoncentrowana wokół jednej wartości, tworząc „szczyt" na wykresie gęstości prawdopodobieństwa. W przypadku rozkładu Laplace'a ten szczyt jest wyraźniejszy i bardziej stromy niż w przypadku rozkładu normalnego. Prawdopodobieństwo warunkowe jest ważnym pojęciem, pozwalającym na określenie prawdopodobieństwa jednego zdarzenia pod warunkiem wystąpienia innego. Jest to istotne w analizie zależności między danymi oraz w konstruowaniu złożonych modeli probabilistycznych. Metody bayesowskie, oparte na regułach Bayesa, są szeroko stosowane w celu uaktualniania wiedzy na podstawie nowych obserwacji, umożliwiając adaptację modeli do zmieniających się warunków i poprawę dokładności predykcji.

W modelach głębokiego uczenia podczas pracy z rozkładami prawdopodobieństwa stosuje się różne funkcje aktywacji i przekształceń. Na przykład funkcja logistyczno-sigmoidalna mapuje wartości na przedział (0, 1), co jest przydatne przy pracy z rozkładami Bernoulliego. Funkcja ReLU (*ang. rectified linear unit*, ReLU) [65, 66, 67] ustawia wszystkie ujemne wartości na zero, co pomaga w trenowaniu głębokich sieci neuronowych. Funkcja softplus [68] jest gładką alternatywą dla funkcji ReLU (max(0, x)) i jest używana do generowania dodatnich parametrów w rozkładach normalnych. Funkcja tanh (tangens hiperboliczny) przekształca wartości na przedział (-1, 1), co może przyspieszać zbieżność w niektórych modelach. Zaproponowano również twardą funkcję tanh (*ang. hardtanh*), która działa w ten sposób, że przyjmuje wartość -1 dla argumentów mniejszych niż -1, wartość 1 dla argumentów większych niż 1, a wartości pomiędzy -1 i 1 pozostają bez zmian [69]. Funkcja softmax [70] przekształca wektory na rozkłady prawdopodobieństwa, co ma znaczenie w klasyfikacji wieloklasowej. Artykuł [68] przedstawia różnorodne klasy funkcji stosowanych w procesach modelowania.

Prawdopodobieństwo można rozumieć jako rozszerzenie logiki na sytuacje niepewne. Podczas gdy logika dostarcza formalnych reguł do rozstrzygania prawdziwości twierdzeń, teoria prawdopodobieństwa oferuje formalne zasady oceny wiarygodności tych twierdzeń, uwzględniając prawdopodobieństwo alternatywnych hipotez. Teoria informacji, ściśle powiązana z teorią prawdopodobieństwa, oferuje narzędzia do precyzyjnego oszacowania i zarządzania niepewnością za pomocą rozkładów prawdopodobieństwa. Jest to dziedzina, która umożliwia nie tylko mierzenie i analizowanie ilości informacji zawartej w danych, ale także efektywne zarządzanie niepewnością. Początkowo rozwijana do analizy komunikacji przez zakłócone kanały, teoria informacji pozwala na projektowanie optymalnych kodów oraz ocenę długości komunikatów. Dzięki tej teorii możliwe jest również określanie efektywności przesyłania danych w dyskretnych alfabetach oraz optymalizacja kodowania przy uwzględnieniu różnych schematów probabilistycznych, co ma zasadnicze znaczenie w zastosowaniach takich jak transmisje radiowe. W kontekście systemów uczących się, teoria informacji znajduje zastosowanie dla zmiennych ciągłych. Zgodnie z zasadami teorii informacji, zdarzenia o wysokim prawdopodobieństwie niosą ze sobą mało infor-

macji, natomiast zdarzenia mniej prawdopodobne przekazują więcej informacji. Istotne pojęcia, takie jak entropia Shannona, służą do charakteryzowania rozkładów prawdopodobieństwa i oceny ich podobieństw. Entropia ta mierzy średnią ilość informacji zawartą w losowym zdarzeniu, a dywergencja Kullbacka-Leiblera (KL) ocenia różnicę między dwoma rozkładami. Dywergencja KL ma liczne zastosowania, w tym optymalizację modeli probabilistycznych w uczeniu maszynowym.

Obliczenia numeryczne

Algorytmy stosowane w systemach uczących się opierają się na intensywnych obliczeniach numerycznych, obejmujących optymalizację i rozwiązywanie układów równań liniowych. Poniżej przedstawiono fundamentalną metodę optymalizacji gradientowej, odgrywającą istotną rolę w uczeniu maszynowym. Ponadto, obliczenia numeryczne wiążą się z wieloma innymi problemami oraz rozwiązaniami optymalizacyjnymi, które są szczegółowo omówione w literaturze naukowej [55, 56, 57, 50, 58, 59, 60, 61, 62, 63, 64].

Ważnym i często wykorzystywanym elementem w głębokim uczeniu jest optymalizacja gradientowa, polegająca na minimalizacji lub maksymalizacji funkcji poprzez odpowiednie dostosowanie zmiennych. Zwykle definiujemy problemy optymalizacyjne jako minimalizację funkcji celu, zwanej również funkcją kosztów lub funkcją strat. W niektórych kontekstach terminologia może się różnić, ale podstawowe zasady pozostają te same. Proces minimalizacji opisujemy za pomocą notacji $\theta^* = \arg\min_\theta f(\theta)$, gdzie θ jest wektorem parametrów modelu. W kontekście zbioru danych D, gdzie każdy obiekt X_i w D jest reprezentowany jako wektor zmiennych (cech, atrybutów) $\mathrm{x} = [x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$, optymalizacja gradientowa wykorzystuje pochodną funkcji $f(\theta)$, oznaczaną jako $f'(\theta)$ lub $\frac{df}{d\theta}$. Pochodna wskazuje nachylenie funkcji w punkcie θ, czyli jak zmiana θ wpływa na wartość $f(\theta)$. W przybliżeniu można zapisać:

$$f(\theta + \epsilon) \approx f(\theta) + \epsilon f'(\theta) \tag{2.1}$$

Pochodna jest niezbędna do minimalizacji funkcji, gdyż wskazuje, jak zmieniać θ, aby osiągnąć poprawę wartości $f(\theta)$. Na przykład, jeśli ϵ jest małe, to $f(\theta - \epsilon \cdot \mathrm{sign}(f'(\theta))) < f(\theta)$, co oznacza, że przesuwając θ w kierunku przeciwnym do pochodnej, możemy zmniejszyć wartość funkcji. Ta metoda, znana jako spadek gradientu (*ang. gradient descent*) [54], jest podstawową techniką w uczeniu maszynowym do dostosowywania parametrów modeli w celu minimalizacji funkcji kosztów i poprawy dokładności predykcji. W szczególności, hipoteza $h(\cdot)$, czyli uczący się model, jest dostosowywana poprzez aktualizację jej parametrów na podstawie gradientu funkcji kosztów obliczonego dla zbioru danych D. Zmienna $x_{i,j}$, która jest wartością cechy A_j dla obiektu X_i, oraz zbiór wartości cechy Q_j są podstawowymi elementami w procesie optymalizacji, który dąży do znalezienia najlepszej hipotezy $h(\cdot)$ spośród zbioru hipotez $\mathcal{H}$.

To Cauchy [54] już w 1847 roku wprowadził spadek gradientu, pokazując, jak bezpośrednio rozwiązywać układy równań liniowych, omijając konieczność redukcji do poje-

dynczego równania przez eliminację zmiennych, co często jest skomplikowane lub wręcz niewykonalne. W swoim artykule Cauchy opisuje, jak wykorzystać ciągłą funkcję wielu zmiennych i technikę iteracyjnego zmniejszania wartości tej funkcji, aż do osiągnięcia zera lub wartości minimalnej. Metoda ta znajduje szerokie zastosowanie w różnych dziedzinach, takich jak obliczanie orbit gwiazd, gdzie liczba równań przekracza liczbę niewiadomych, co sprawia, że precyzja wyników jest niezwykle istotna. Cauchy sugeruje również, że po uzyskaniu wstępnych przybliżeń wartości niewiadomych warto stosować metodę liniową lub metodę Newtona, co umożliwia szybkie uzyskanie dalszych przybliżeń i znacząco poprawia dokładność obliczeń. Algorytmy optymalizacyjne, które operują wyłącznie na gradientach, takie jak algorytmy spadku gradientu, są znane jako algorytmy pierwszego rzędu. Natomiast algorytmy, które korzystają z macierzy Hessego, takie jak metoda Newtona, są klasyfikowane jako algorytmy drugiego rzędu, ponieważ używają dodatkowych informacji o krzywiźnie funkcji celu, co może przyspieszyć zbieżność do rozwiązania optymalnego [58]. W niniejszej rozprawie implementacja autoenkodera bazuje na teorii Cauchy'ego, wykorzystując algorytm gradientowy do optymalizacji modelu.

W obliczeniach numerycznych na komputerach napotykamy na szereg poważnych problemów. Błędy zaokrąglenia, wynikające z ograniczonej precyzji reprezentacji liczb rzeczywistych, mogą kumulować się w długich obliczeniach, prowadząc do znacznych odchyleń wyników. Nadmiar występuje, gdy wyniki przekraczają maksymalne wartości reprezentowane przez system numeryczny, co skutkuje nieskończonościami lub wartościami skrajnymi i prowadzi do błędnych wyników w dalszych obliczeniach. Niedomiar ma miejsce, gdy wyniki są mniejsze od minimalnych wartości reprezentowanych przez system, co skutkuje zaokrąglaniem liczb bliskich zeru do zera, a tym samym błędami w dalszych obliczeniach. Błędy reprezentacji wynikają z niemożności dokładnego przedstawienia wszystkich liczb rzeczywistych w systemie binarnym, co prowadzi do dodatkowych błędów zaokrąglenia. Błędy numeryczne z kolei są skutkiem stosowanych metod obliczeniowych, szczególnie gdy metody te nie są stabilne numerycznie. Małe zmiany danych wejściowych mogą prowadzić do dużych zmian wyników, co stanowi istotne wyzwanie dla stabilności obliczeń numerycznych. Błędy algorytmiczne pojawiają się, gdy algorytm nie jest właściwie dobrany do problemu, co może skutkować wolniejszą konwergencją lub błędnymi wynikami.

Przykładem funkcji szczególnie podatnej na te problemy jest funkcja softmax, która jest wrażliwa na nadmiar i niedomiar. Podobnie logarytmy są problematyczne, ponieważ bardzo małe liczby mogą prowadzić do niedomiaru, a logarytmowanie zer prowadzi do nieskończoności lub NaN. Aby minimalizować te problemy, inżynierowie i naukowcy opracowują stabilne metody numeryczne i algorytmy odporne na błędy zaokrąglenia oraz stosują techniki takie jak normalizacja danych i transformacje skalujące, jak na przykład Theano [55, 56]. Theano jest przykładem narzędzia, które rozwiązuje wiele z tych problemów. Pracuje z symboliczną reprezentacją wyrażeń matematycznych, dostarczoną przez użytkownika w składni podobnej do NumPy. Dostęp do pełnego grafu obliczeniowego

wyrażenia umożliwia zaawansowane funkcje, takie jak symboliczne różnicowanie złożonych wyrażeń. Theano pozwala na lokalne transformacje grafu, które mogą poprawić wydajność i stabilność numeryczną wyrażeń. Po optymalizacji ten sam graf może być używany do generowania implementacji zarówno dla CPU, jak i GPU, bez potrzeby zmiany kodu użytkownika.

2.3.3 Miary odległości lub podobieństwa

Miary proksymalne, znane również jako miary odległości lub podobieństwa, stanowią ważne narzędzie matematyczne w analizie danych. Umożliwiają one precyzyjne określenie stopnia podobieństwa lub różnicy między obiektami w przestrzeni danych. W obszarze analizy danych i uczenia maszynowego, miary te są fundamentalne dla zadań takich jak grupowanie, klasyfikacja, wykrywanie anomalii. Wykorzystując miary odległości lub podobieństwa, można dokładnie ocenić, jak bardzo dany obiekt różni się od innych.

Poniżej przedstawiono podstawowe miary odległości i podobieństwa dla danych ilościowych i jakościowych. Ponadto, w podrozdziale 3.4 dotyczącym metod opartych na bliskości w danych kategorycznych, zaprezentowano w tabeli 3.2 przegląd dodatkowych, niewymienionych tutaj miar odległości i podobieństwa dla atrybutów kategorycznych. Miara odległości kwantyfikuje *odległość* lub różnicę między dwoma obiektami w przestrzeni danych, przy czym wartości te są zazwyczaj nieujemne (≥ 0). Natomiast miara podobieństwa kwantyfikuje *podobieństwo* lub bliskość między dwoma obiektami w przestrzeni danych, przy czym wartości te zazwyczaj mieszczą się w przedziale od 0 do 1, gdzie wartość 1 oznacza, że obiekty są identyczne, a wartość 0 oznacza, że obiekty są zupełnie różne. Miary te są szeroko omawiane również w literaturze naukowej, na przykład w [71, 72, 73, 43].

Miary odległości służą do obliczania odległości między obiektami X_i. W tym przypadku obiekty opisane są zbiorem atrybutów ciągłych, najbardziej popularne miary odległości między obiektami to:

- **odległość euklidesowa** - (dane ilościowe) pierwiastek sumy kwadratów różnic między odpowiednimi wartościami zmiennych (cech) A_j obiektów X_i. Odległość euklidesowa między dwoma obiektami X_i i X_k w przestrzeni n-wymiarowej jest definiowana jako pierwiastek kwadratowy sumy kwadratów różnic poszczególnych zmiennych A_j tych obiektów. Jeśli mamy dwa obiekty X_i i X_k z wartościami zmiennych $\mathrm{x_i} = [x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$ oraz $\mathrm{x_k} = [x_{k,1}, x_{k,2}, \ldots, x_{k,j}]$, wzór na odległość euklidesową d między X_i i X_k wyraża się jako:

$$d(X_i, X_k) = \sqrt{\sum_{j=1}^{n} (x_{i,j} - x_{k,j})^2} \tag{2.2}$$

- **odległość Manhattan (miejska)** - (dane ilościowe) suma bezwzględnych różnic między wartościami zmiennych A_j obiektów X_i. Definiowana jako suma modułów różnic poszczególnych zmiennych dwóch obiektów X_i i X_k, co można zapisać jako:

$$d_{\text{Manhattan}}(X_i, X_k) = \sum_{j=1}^{n} |x_{i,j} - x_{k,j}| \tag{2.3}$$

- **odległość Czebyszewa** - (dane ilościowe) maksymalna różnica wartości wśród wszystkich zmiennych A_j obiektów X_i. Jest to największa z różnic wartości zmiennych dwóch obiektów X_i i X_k, co matematycznie można przedstawić jako:

$$d_{\text{Czebyszewa}}(X_i, X_k) = \max_{j=1}^{n} |x_{i,j} - x_{k,j}| \tag{2.4}$$

- **odległość Mahalanobisa** [74] - (dane ilościowe) jest uogólnieniem odległości euklidesowej, które uwzględnia korelacje między zmiennymi. W przeciwieństwie do odległości Minkowskiego, odległość Mahalanobisa może być stosowana, gdy cechy są skorelowane. Konstrukcja tej miary bazuje na średniej arytmetycznej i odchyleniu standardowym, które nie są odpornymi estymatorami, dlatego często zastępuje się je odpornym estymatorem MCD (*ang. minimum covariance determinant*, MCD) [75]. Odległość Mahalanobisa definiuje się jako:

$$d_{\text{Mahalanobisa}}(\mathrm{x}, \mathrm{y}) = \sqrt{(\mathrm{x} - \mathrm{y})^T \mathrm{S}^{-1} (\mathrm{x} - \mathrm{y})}, \tag{2.5}$$

  gdzie x i y to wektory zmiennych, a S to macierz kowariancji.

  Estymator MCD jest bardzo pomocny w wykrywaniu anomalii w danych wielowymiarowych, ponieważ jest odporny na odchylenia. Dzięki temu metody takie jak odległość Mahalanobisa mogą skutecznie wykrywać anomalie, nawet jeśli inne metody mogłyby je przeoczyć z powodu efektu maskowania.

Miary odległości specyficzne dla danych kategorycznych to miary szczególnie użyteczne, gdy dane nie mają naturalnego porządku liczbowego:

- **odległość Hamminga** [76, 77] - (dane jakościowe/wektory binarne) liczba pozycji, w których odpowiednie wartości zmiennych A_j dwóch obiektów X_i i X_k są różne. Jest to liczba miejsc, w których odpowiadające sobie zmienne dwóch obiektów są różne. Dla wektorów $\mathrm{x_i}$ i $\mathrm{x_k}$ odległość Hamminga można obliczyć jako:

$$d_{\text{Hamminga}}(\mathrm{x_i}, \mathrm{x_k}) = \sum_{j=1}^{n} [x_{i,j} \neq x_{k,j}], \tag{2.6}$$

gdzie $x_{i,j}$ i $x_{k,j}$ to wartości zmiennych A_j dla obiektów X_i i X_k na j-tej pozycji, a wyrażenie w nawiasie kwadratowym jest operacją logiczną, zwracającą 1, gdy elementy się różnią, i 0 w przeciwnym wypadku,

- **metryka różnicy wartości atrybutów** (*ang. value difference metric*, VDM) [78] - (dane jakościowe) mierzy odległość między wartościami atrybutów, uwzględniając rozkład klas w danych. Definiowana jest jako suma kwadratów różnic prawdopodobieństw warunkowych dla klas, co można zapisać jako:

$$VDM(x_{i,j}, x_{k,j}) = \sum_{c \in C} (P(c|x_{i,j}) - P(c|x_{k,j}))^2, \tag{2.7}$$

gdzie C to zbiór wszystkich klas, a $P(c|x_{i,j})$ to prawdopodobieństwo klasy c pod warunkiem wartości atrybutu $x_{i,j}$ dla obiektu X_i,

- **metryka różnicy wartości VDM (z etykietami klas)** - (dane jakościowe) stosowana w problemach uczenia nadzorowanego dla danych kategorycznych z etykietami klas. Odległość między dwoma obiektami X_i i X_j jest definiowana jako suma kwadratów różnic prawdopodobieństw warunkowych dla klas przy każdej wartości atrybutu, co można zapisać jako:

$$VDM(X_i, X_j) = \sum_{r=1}^{m} \sum_{k} \left(P(c_k|x_{i,r}) - P(c_k|x_{j,r})\right)^2, \tag{2.8}$$

gdzie $P(c_k|x_{i,r})$ to prawdopodobieństwo klasy c_k pod warunkiem wartości atrybutu $x_{i,r}$, obliczane jako stosunek liczby obiektów w klasie c_k z wartością $x_{i,r}$ do całkowitej liczby obiektów z wartością $x_{i,r}$ w zbiorze danych:

$$P(c_k|x_{i,r}) = \frac{\text{częstość}_{c_k}(x_{i,r})}{\text{częstość}(x_{i,r})}, \tag{2.9}$$

gdzie $\text{częstość}_{c_k}(x_{i,r})$ to liczba obiektów w klasie c_k mających wartość $x_{i,r}$, a $\text{częstość}(x_{i,r})$ to liczba wszystkich obiektów mających tę wartość atrybutu $x_{i,r}$ w zbiorze danych.

Miary podobieństwa używane do analizy danych:

- **korelacja Pearsona** [79] - (dane ilościowe) miara korelacji liniowej między dwiema zmiennymi A_j i A_k na podstawie n obserwacji. Wzór na współczynnik korelacji Pearsona r między dwiema zmiennymi A_j i A_k z n obserwacjami jest określony jako:

$$r = \frac{\sum_{i=1}^{n}(x_{i,j} - \overline{x_j})(x_{i,k} - \overline{x_k})}{\sqrt{\sum_{i=1}^{n}(x_{i,j} - \overline{x_j})^2}\sqrt{\sum_{i=1}^{n}(x_{i,k} - \overline{x_k})^2}}, \tag{2.10}$$

gdzie $x_{i,j}$ to wartość i-tej obserwacji zmiennej A_j, a $\overline{x_j}$ to średnia wartość zmiennej A_j. Miara korelacji Pearsona nie jest bezpośrednią miarą podobieństwa między obiektami, lecz jest miarą statystyczną, która ocenia siłę i kierunek liniowej zależności między dwiema zmiennymi. Wymieniono ją, gdyż korelacja Pearsona może być używana do oceny podobieństwa wzorców między zmiennymi,

- **podobieństwo kosinusowe** - (dane ilościowe) cosinus kąta między dwoma wektorami cech $\mathrm{x_i}$ i $\mathrm{x_k}$, które reprezentują obiekty X_i i X_k. Na tej podstawie ocenia się podobieństwo między obiektami. Podobieństwo kosinusowe oblicza się według wzoru:

$$S_{\cos}(X_i, X_k) = \frac{\sum_{j=1}^{n} x_{i,j} \cdot x_{k,j}}{\sqrt{\sum_{j=1}^{n} x_{i,j}^2} \cdot \sqrt{\sum_{j=1}^{n} x_{k,j}^2}}, \tag{2.11}$$

gdzie $x_{i,j}$ i $x_{k,j}$ to wartości cech A_j dla obiektów X_i i X_k w j-tym wymiarze,

- **współczynnik Jaccarda** [80, 81, 82] - (dane jakościowe) stosunek liczby wspólnych cech do liczby wszystkich unikalnych cech dwóch obiektów. Można go obliczyć jako:

$$\text{Współczynnik Jaccarda } (J) = \frac{|\mathrm{x_i} \cap \mathrm{x_k}|}{|\mathrm{x_i} \cup \mathrm{x_k}|}, \tag{2.12}$$

gdzie $\mathrm{x_i}$ i $\mathrm{x_k}$ to wektory cech dla obiektów X_i i X_k,

- **współczynnik Gowera** [83] - (dane ilościowe, jakościowe i binarne) to miara podobieństwa używana do danych mieszanych. Obliczenie współczynnika Gowera polega na wykonaniu następujących kroków: dla każdej cechy określamy odpowiednią miarę odległości i przekształcamy ją do zakresu od 0 do 1. Następnie, każdą z tych skalowanych wartości ważymy zgodnie z preferencjami użytkownika. Jeśli brak jest preferencji użytkownika co do wag, wszystkie zmienne są ważone jednakowo. Ostateczna wartość współczynnika Gowera to średnia ważona tych wartości. W ten sposób tworzymy cząstkowy współczynnik podobieństwa dla każdej zmiennej, co pozwala na wszechstronną analizę danych mieszanych. Wartość cząstkowego współczynnika podobieństwa wyznaczamy zgodnie z wzorami:

  – jeśli A_r-ta zmienna jest atrybutem ilościowym to:

$$S_r(X_i, X_j) = 1 - \frac{|x_{i,r} - x_{j,r}|}{\max x_r - \min x_r} \tag{2.13}$$

  – jeśli A_r-ta zmienna jest atrybutem jakościowym to:

$$S_r(X_i, X_j) = \begin{cases} 1, & \text{jeśli } x_{i,r} = x_{j,r} \\ 0, & \text{w przeciwnym razie} \end{cases} \tag{2.14}$$

– jeśli A_r-ta zmienna jest atrybutem binarnym to:

$$S_r(X_i, X_j) = \begin{cases} 1, & \text{jeśli } x_{i,r} = 1 \text{ i } x_{j,r} = 1 \\ 0, & \text{jeśli } (x_{i,r} = 1 \text{ i } x_{j,r} = 0) \text{ lub } (x_{i,r} = 0 \text{ i } x_{j,r} = 1) \end{cases} \tag{2.15}$$

Współczynnik Gowera uwzględnia również przypadek, w którym wartości danej zmiennej są nieporównywalne (na przykład w sytuacji, gdy wartości tej brakuje dla jednego z obiektów lub gdy atrybut binarny nie występuje dla obu obiektów, czyli wartości tego atrybutu dla obiektów są równe zero). Wówczas $S_r(X_i, X_j) = 0$ oraz $w_r = 0$. Jeśli istnieje możliwość porównania r-tego atrybutu, to $w_r = 1$. Miarę podobieństwa Gowera dla obiektów X_i oraz X_j opisywanych przez n zmiennych w przypadku, gdy $\sum w_r > 0$, definiujemy następująco:

$$S(X_i, X_j) = \frac{\sum_{r=1}^{n} w_r S_r(X_i, X_j)}{\sum_{r=1}^{n} w_r} \tag{2.16}$$

Współczynnik w_r oznacza wagę przypisaną r-tej zmiennej w obliczeniach miary podobieństwa Gowera. W przeprowadzonych badaniach zastosowano tę miarę [84, 85]. Aplikacja webowa zastosowana w tych pracach jest dostępna pod linkiem [1].

2.4 Anomalie: od nauki o danych do eksploracji

Wykrywanie anomalii w danych stanowi fundamentalny aspekt analizy danych, obejmujący szeroki zakres dziedzin, od finansów po medycynę. Proces ten polega na identyfikacji nietypowych wzorców, które mogą sygnalizować istotne zdarzenia, błędy lub nowo odkryte zjawiska. Aby efektywnie identyfikować anomalie, konieczne jest zrozumienie i integracja trzech kluczowych obszarów: nauki o danych (ang. *data science*), odkrywania wiedzy w bazach danych KDD (ang. *knowledge discovery in databases*, KDD) oraz eksploracji danych (ang. *data mining*). W tabeli 2.5 przedstawiono wspólne elementy i różnice między tymi trzema kategoriami, co pomaga lepiej zrozumieć ich wzajemne zależności i zastosowania.

Nauka o danych to szerokie pojęcie, obejmujące cały cykl życia danych – od ich zbierania i przygotowania, przez analizę, aż po wdrażanie modeli i prezentację wyników. To interdyscyplinarne pole łączy w sobie elementy statystyki, informatyki i specjalistycznych dziedzin, co pozwala na kompleksowe podejście do analizy danych. Kluczowe elementy nauki o danych obejmują różne etapy pracy z danymi. Na początku mamy zbieranie danych, które polega na gromadzeniu informacji z różnych źródeł, takich jak bazy danych, API czy sensory. Następnie przygotowanie danych obejmuje procesy czyszczenia, transformacji i integracji, aby dane były gotowe do analizy. Analiza eksploracyjna danych (EDA) polega

[1] https://studies.shinyapps.io/RulesAnalysis/

Rysunek 2.9: Hierarchia pojęć: nauka o danych, odkrywanie wiedzy i eksploracja danych. Źródło: opracowanie własne.

na wstępnej analizie i wizualizacji danych w celu zrozumienia ich struktury. Kolejnym krokiem jest modelowanie, które polega na tworzeniu modeli predykcyjnych, klasyfikacyjnych czy wykrywających anomalie przy użyciu technik uczenia maszynowego. W statystyce model jest reprezentacją zależności między zmiennymi w zbiorze danych. Modelowanie to proces, w którym tworzy się reprezentatywną abstrakcję na podstawie zgromadzonych danych. Na przykład, na podstawie historii zakupów, preferencji klientów oraz danych demograficznych, można opracować model rekomendacji produktów lub model wykrywający nietypowe wzorce zakupowe, które mogą wskazywać na oszustwa lub inne nieprawidłowości. Do tego zadania potrzebne są wcześniej zgromadzone dane obserwacyjne, takie jak poprzednie zakupy, preferencje dotyczące kategorii produktów oraz dane demograficzne klientów. Model uogólnia zależności między zmiennymi wejściowymi a wyjściowymi i wykorzystuje je do przewidywania, jakie produkty mogą zainteresować nowych klientów lub identyfikowania anomalii w danych, które mogą wskazywać na istotne zjawiska.

Nauka o danych obejmuje zaawansowane procesy modelowania, które nie tylko przewidują wartości wyjściowe na podstawie nowych zmiennych wejściowych, ale również pomagają zrozumieć złożone zależności między zmiennymi. Przykładowo, analiza może odpowiedzieć na pytania takie jak: czy wiek klienta wpływa na preferencje produktowe? Czy kategoria produktów jest bardziej istotna niż historia zakupów w prognozowaniu przyszłych wyborów konsumentów? Co więcej, jak zmiany preferencji klientów lub wprowadzenie nowych kategorii produktów wpływają na istniejące wzorce? Modele te mogą być wykorzystywane zarówno w aplikacjach predykcyjnych, jak i wyjaśniających. Identyfikacja anomalii w danych stanowi jeden z istotnych aspektów nauki o danych. Anomalie, definiowane jako nietypowe lub niespodziewane obserwacje, mogą sygnalizować ważne zdarzenia, błędy w danych lub nieznane wcześniej wzorce. Techniki wykrywania anomalii, zarówno nadzorowane, jak i nienadzorowane, umożliwiają identyfikację tych wyjątkowych przypadków, co z kolei prowadzi do głębszego zrozumienia analizowanych

danych i podejmowania bardziej świadomych decyzji. Wdrażanie i monitorowanie modeli jest nieodzownym elementem tego procesu, oznacza bowiem implementację tych modeli w rzeczywistych systemach oraz stałe monitorowanie ich wydajności i dokładności. Zasadniczym etapem jest również prezentacja wyników, która powinna być przeprowadzona w sposób klarowny i zrozumiały dla decydentów oraz interesariuszy, umożliwiając im podejmowanie świadomych decyzji opartych na analizie danych. Ostatecznym celem nauki o danych jest odkrywanie nowych, wcześniej nieznanych wzorców oraz formułowanie potencjalnie użytecznych wniosków, które mogą zostać zastosowane przez użytkowników analizy w praktyce.

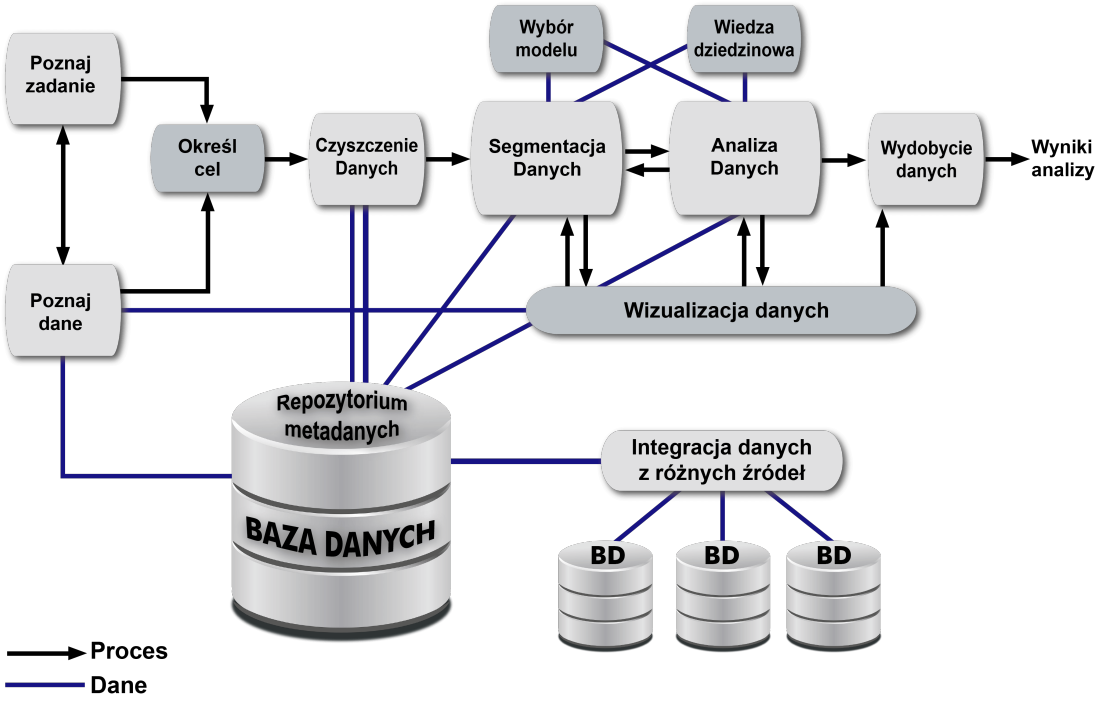
Eksploracja danych a odkrywanie wiedzy: różnice terminologiczne

Często zdarza się, że termin eksploracja danych jest używany zamiennie z odkrywaniem wiedzy w bazach danych KDD. To jednak nieporozumienie, gdyż oba te terminy odnoszą się do różnych aspektów analizy danych. Eksploracja danych stanowi jedynie fragment szerszego procesu KDD, koncentrując się głównie na algorytmach służących do identyfikacji wzorców w danych. KDD obejmuje cały kompleksowy cykl, rozpoczynający się od czyszczenia danych, poprzez ich integrację i selekcję, aż po końcową wizualizację i ocenę odkrytych wzorców. Traktowanie tych terminów jako synonimów jest błędem, gdyż nie oddaje pełni złożoności procesu KDD, w którym eksploracja danych jest tylko jednym z wielu kroków [86]. Nauka o danych to jeszcze szersze pojęcie niż KDD, obejmujące wszystkie etapy cyklu życia danych – od zbierania i przygotowania, przez analizę, aż po wdrażanie modeli i prezentację wyników. Eksploracja danych jest nieodzownym etapem w procesie KDD, który z kolei jest integralną częścią szerszego cyklu nauki o danych. Podczas gdy eksploracja danych koncentruje się na odkrywaniu wzorców, KDD obejmuje pełen zakres działań związanych z przygotowaniem i interpretacją tych wzorców. Nauka o danych natomiast rozszerza te działania o dodatkowe elementy, takie jak zarządzanie danymi, programowanie, statystyka oraz komunikacja wyników, co zademonstrowano na rysunku 2.9. Eksploracja danych jest interdyscyplinarną dziedziną zakładającą automatyzację przetwarzania przy pomocy komputerów oraz stosującą naukowe metody pozyskiwania użytecznej wiedzy z dużych zbiorów danych. Brachman i Anand [87] przedstawili praktyczny obraz procesu KDD, podkreślając jego interaktywny charakter. Proces ten jest wizualizowany na rysunku 2.10. Pierwszym etapem powinno być właściwe określenie celu problemu, co wymaga spędzania czasu z klientem i poznania mechanizmów funkcjonujących w jego organizacji. Ten wstępny proces może być bardzo czasochłonny i trudny, ale bez niego łatwo można stracić czas, odpowiadając na niewłaściwe pytania. Podstawowe składniki KDD to analiza danych (eksploracja danych) podczas której potwierdzana lub odrzucana jest hipoteza dotycząca danych przy pomocy narzędzi analitycznych i wizualizacja, która jest istotnym składnikiem każdego kroku w procesie odkrywania wiedzy. Odpowiednia prezentacja danych i ich powiązań może dać analitykowi wgląd, który jest praktycznie niemożliwy do uzyskania z tabel wyjściowych lub prostych statystyk. Kluczowa faza procesu odkrywania wiedzy musi być poprzedzona „odkryciem danych". Autorzy zwracają

Tabela 2.5: Wspólne elementy i zakresy: nauki o danych, KDD i eksploracji danych. Źródło: opracowanie własne.

Kategoria	Elementy wspólne	Zakres i cel
Nauka o danych	**czyszczenie danych**, **transformacja danych** (wspólne z KDD), **analiza i modelowanie danych** (wspólne z KDD i eksploracją danych), **analiza eksploracyjna danych (EDA)**, **prezentacja wyników** (wspólne z KDD)	Najszerszy zakres obejmujący KDD: rozwiązywanie problemów biznesowych i naukowych, zbieranie, przygotowanie, analiza, modelowanie, wdrażanie, monitorowanie, komunikacja wyników, bardziej zaawansowane wizualizacje, interaktywne dashboardy, raporty biznesowe
Odkrywanie	**czyszczenie danych**, **transformacja danych**, **eksploracja danych**, **analiza danych**, **prezentacja wyników** (wspólne z nauką o danych)	Szerszy proces niż eksploracja danych: odkrywanie wiedzy z danych i jej prezentacja jako użytecznej wiedzy, czyszczenie, integracja, wybór, transformacja danych, eksploracja danych, ewaluacja, raportowanie odkrytych wzorców, koncentruje się na odkrywaniu wiedzy z danych
Eksploracja	**odkrywanie wzorców w danych** (wspólne z nauką o danych i KDD)	Część procesu KDD: koncentracja na algorytmach, odkrywanie ukrytych wzorców i zależności. Zastosowanie metod eksploracji danych jak: klasyfikacja i predykcja, grupowanie, odkrywanie asocjacji, wykrywanie anomalii, eksploracja grafów, itp.

uwagę, że na początku dane są drogowskazem, wtedy posiadając niezbędną wiedzę eksperta z danej dziedziny możemy sformułować hipotezę. Ważnymi podprocesami związanymi z odkrywaniem wiedzy w danych są: (1) segmentacja danych, (2) wybór modelu i (3) wybór parametrów. Dane klienta praktycznie zawsze mają problemy, mogą w nich znajdować się dane niepełne, niepoprawne lub nieistotne w rezultacie proces KDD nie może się powieść bez poważnego wysiłku, aby „oczyścić" lub „wyczyścić" dane. Zrozumienie i właściwe przygotowanie danych ma duże znaczenie w kontekście odkrywania wiedzy z danych. Jest to jeden z najbardziej pracochłonnych etapów całego procesu eksploracji danych. Polega on

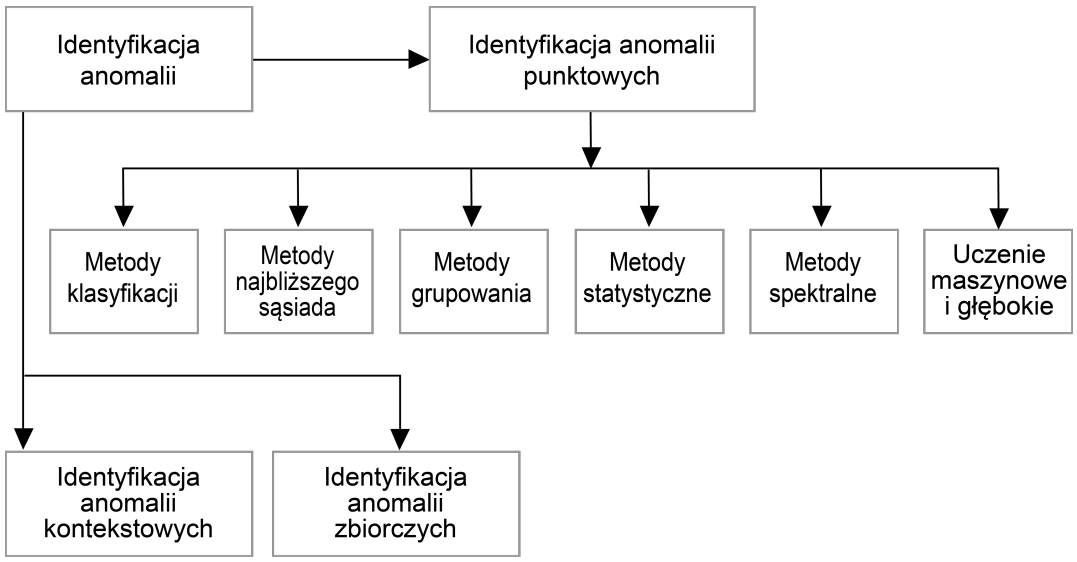


Rysunek 2.10: Schemat blokowy procesu odkrywania wiedzy w bazach danych. Źródło: opracowanie własne na podstawie [87].

na przekształceniu surowych danych w finalny zbiór gotowy do analizy. Ten etap obejmuje eliminację nieistotnych lub niepotrzebnych danych, weryfikację ich poprawności oraz oczyszczanie. Kolejnym krokiem jest wybór obiektów i zmiennych, które będą poddane analizie i które są odpowiednie do danego modelu eksploracji. Większość surowych danych przechowywanych w bazach danych jest niekompletna i zaszumiona. Często dane mogą być w formacie nieodpowiednim dla danej analizy. Inne problemy obejmują brakujące wartości oraz dane niezgodne z określonymi zasadami.

2.5 Przegląd metod wykrywania anomalii

Metody wykrywania anomalii można sklasyfikować na kilka głównych kategorii. Każda z tych metod ma swoje specyficzne zastosowania i ograniczenia, a wybór odpowiedniej metody zależy od charakterystyki danych oraz kontekstu aplikacji. Ważne jest, aby zrozumieć różnice i założenia poszczególnych kategorii, aby skutecznie zastosować techniki wykrywania anomalii w praktyce [21]. Rysunek 2.11 przedstawia klasyfikację metod identyfikacji anomalii według różnych technik i podejść, uwzględniając typ obserwacji odstającej, które mogą być stosowane przez badaczy danych. Każda kategoria została szczegółowo omówiona w kolejnych podrozdziałach.



Rysunek 2.11: Metody wykrywania anomalii. Źródło: opracowanie własne na podstawie [21].

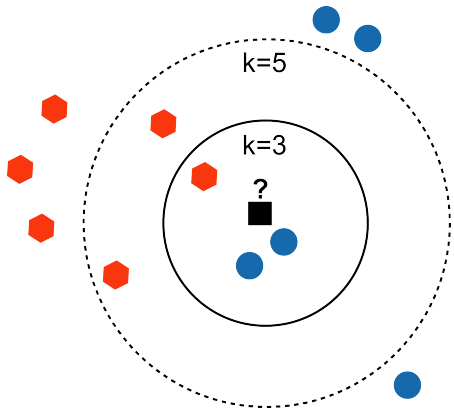
2.5.1 Metody wykrywania anomalii bazujące na mierze odległości

Metody wykorzystujące miarę odległości do wykrywania anomalii zakładają, że nietypowe obiekty są oddalone od zwartych grup typowych obiektów. To podejście jest szczególnie korzystne, ponieważ nie wymaga wcześniejszej wiedzy na temat rozkładów prawdopodobieństwa analizowanych obiektów. W ramach tego podejścia wyróżnia się dwie główne klasy metod:

- metody wyznaczające odległość w k-najbliższym sąsiedztwie,
- metody bazujące na gęstości danych.

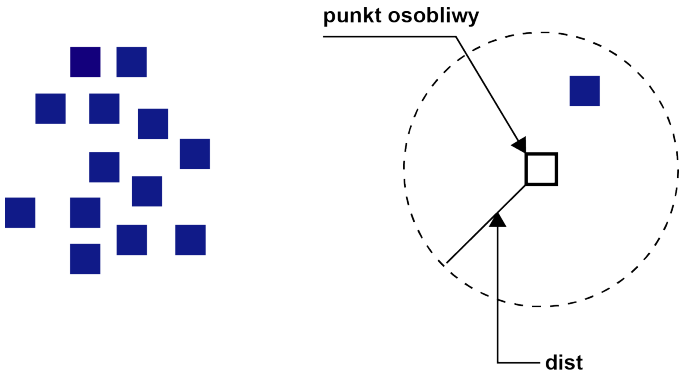
Algorytm k-najbliższych sąsiadów k-NN (*ang. k-nearest neighbors*, k-NN), zaproponowany w 1951 roku [88], znajduje szerokie zastosowanie w klasyfikacji. Metoda pokazana na rysunku 2.12 jest prosta i intuicyjna, polega na przypisywaniu obiektu do grupy, w której większość jego k najbliższych sąsiadów już się znajduje.

Algorytm k-NN często wykorzystuje indeksy wielowymiarowe, takie jak R-drzewa i KD-drzewa. Te indeksy, oparte na obserwacjach z zbioru uczącego, mają na celu zapewnienie efektywnego wyszukiwania k-najbliższych sąsiadów. W algorytmie k-NN znaczenie mają przyjęta miara odległości oraz metoda transformacji obiektu do jego reprezentacji w przestrzeni wzorców. Jak pokazano na rysunku 2.12, wybór wartości parametru k ma istotny wpływ na wynik klasyfikacji. W algorytmie k-NN anomalie są definiowane przez odległość do k-najbliższych sąsiadów. Obiekty o największych odległościach od innych są uznawane za punkty osobliwe, czyli anomalie.



Rysunek 2.12: Metoda k-NN - w przypadku $k = 3$ (mniejszy okrąg), czarny kwadrat zostanie zakwalifikowany do klasy niebieskich kropek. W przypadku $k = 5$ (większy okrąg) - do klasy czerwonych sześcianów. Źródło: opracowanie własne na podstawie [89].

Każdy obiekt X_i jest analizowany w kontekście jego *dist-sąsiedztwa*, gdzie *dist* jest progiem odległości ustalonym przez użytkownika. Obiekt X_i jest uznawany za anomalię, jeśli większość obiektów w jego sąsiedztwie nie spełnia kryterium odległości *dist*. W związku z tym, globalny punkt osobliwy w metodzie k-NN to obiekt, który ma maksymalnie k-sąsiadów w odległości *dist*, co ilustruje rysunek 2.13. Artykuł [90] omawia rozwinięcie klasycznej metody klasyfikacji k-NN. Autorzy proponują model, w którym metryka jest indukowana lokalnie dla każdego testowanego obiektu, co pozwala na lepsze dostosowanie się do lokalnych właściwości danych.



Rysunek 2.13: Globalny punkt osobliwy. Źródło: opracowanie własne na podstawie [86].

Metody wyznaczające odległość w *k*-najbliższym sąsiedztwie

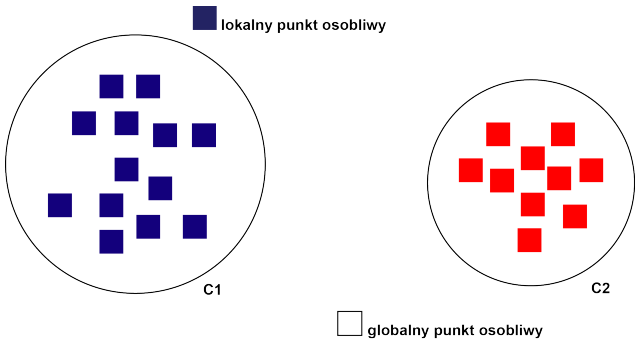
Metody wykrywania anomalii w oparciu o odległość w k-najbliższym sąsiedztwie polegają na analizie odległości między obiektami w przestrzeni wielowymiarowej. Dla każdego

obiektu obliczana jest odległość do jego k-najbliższych sąsiadów. Na podstawie tych odległości obiekt jest klasyfikowany jako normalny lub anomalia. Im większa odległość do sąsiadów, tym większe prawdopodobieństwo, że obiekt jest anomalią. Podstawowa technika wykrywania anomalii polega na obliczeniu odległości obiektu do jego k-tego najbliższego sąsiada w zbiorze danych. Ta metoda była wykorzystywana m.in. do wykrywania min lądowych na obrazach satelitarnych [91] oraz do wykrywania zwarć w uzwojeniach turbogeneratorów synchronicznych [92]. Często stosuje się próg odległości, aby określić, czy dany obiekt jest anomalią. W zależności od sposobu wyznaczania anomalii możemy wyróżnić kilka podejść [93, 94, 95, 96]:

- **metoda Ramaswamy** – dla każdego obiektu wyznaczana jest odległość d do pozostałych obiektów w jego k-sąsiedztwie. Pierwsze M obiektów o największej odległości uznaje się za anomalie,
- **metoda Angiulli i Pizzutti** – dla każdego obiektu wyznaczana jest odległość d do pozostałych obiektów w jego k-sąsiedztwie. Następnie obliczana jest suma odległości obiektu do obiektów w jego k-sąsiedztwie. Pierwsze M obiektów o największej sumie odległości uznaje się za anomalie.

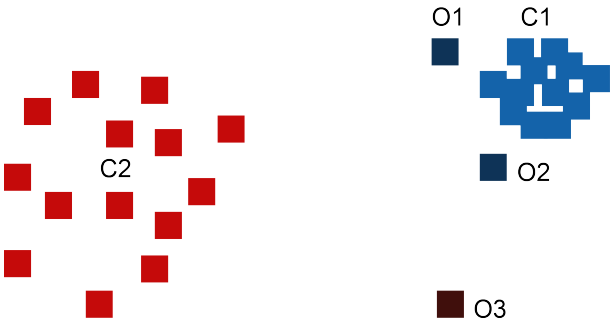
Metody bazujące na gęstości danych

Metody bazujące na analizie gęstości danych wykrywają anomalie poprzez badanie gęstości rozkładu obiektów w przestrzeni. Założeniem tych metod jest, że obiekty znajdujące się w obszarach o niskiej gęstości są prawdopodobnie anomaliami. Przykładami takich metod są algorytmy DBSCAN (*ang. density-based spatial clustering of applications with noise*, DBSCAN) [97] oraz LOF (*ang. local outlier factor*, LOF) [98]. Algorytm DBSCAN identyfikuje skupiska obiektów oraz anomalie [99] poprzez analizę gęstości obiektów. Wykrywa on obszary o wysokiej gęstości, które są interpretowane jako grupy, oraz obszary o niskiej gęstości, które są uznawane za anomalie. Wysoka gęstość odnosi się do obszarów, w których znajduje się wiele obiektów blisko siebie, co wskazuje na istnienie skupisk w danych. Algorytm bardziej szczegółowo omówiony został w podrozdziale poświęconym metodzie grupowania podobnych do siebie obserwacji 2.5.3. Algorytm LOF, zaproponowany przez Breuniga i współpracowników [98], ocenia anomalię na podstawie lokalnej gęstości sąsiedztwa obiektu w porównaniu do gęstości jego najbliższych sąsiadów. LOF oblicza się jako stosunek średniej gęstości lokalnej k-najbliższych sąsiadów obiektu do gęstości lokalnej samego obiektu. Wysoki współczynnik osobliwości LOF wskazuje, że dany obiekt jest anomalią. Algorytm LOF, wykorzystywany w badaniach przeprowadzonych w ramach niniejszej rozprawy, został szerzej opisany w rozdziale 4. Metody bazujące na gęstości danych odwołują się bezpośrednio do koncepcji grupowania gęstościowego, w której głównym parametrem jest minimalna liczba obiektów w zadanym sąsiedztwie danego obiektu, służąca do identyfikacji grup. We wcześniejszych przykładach osobliwość danego punktu dotyczyła całego zbioru obiektów, co nazywamy globalnym punktem osobliwym. W praktyce jednak często rozpatrujemy osobliwość jako stopień



Rysunek 2.14: Porównanie między globalnym a lokalnym punktem osobliwym. Źródło: opracowanie własne na podstawie [86].

izolacji danego obiektu względem jego najbliższego sąsiedztwa. Na rysunku 2.14 pokazano różnicę między globalnym a lokalnym punktem osobliwym. W tym drugim przypadku, stopień izolacji obiektu określamy na podstawie jego najbliższego sąsiedztwa. Algorytmy wykrywania lokalnych punktów osobliwych, takie jak algorytm LOF, opierają się na lokalnej gęstości sąsiednich obiektów. Współczynnik określający stopień osobliwości danego obiektu obliczamy poprzez porównanie lokalnej gęstości sąsiedztwa obiektu z gęstością jego najbliższych sąsiadów, co pokazano na rysunku 2.15.



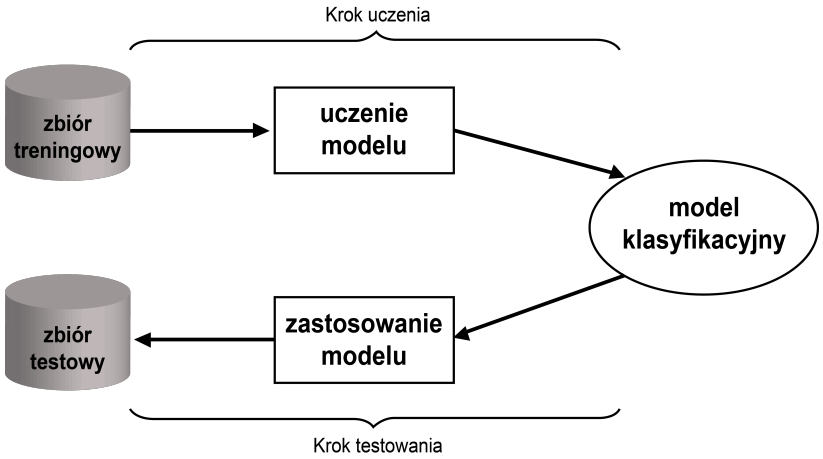
Rysunek 2.15: Ilustracja działania metody opartej na współczynniku osobliwości LOF. Obserwacje *O1* i *O2* są lokalnymi punktami osobliwymi względem grupy *C1*, natomiast obserwacja *O3* jest odstającą względem grup *C1* i *C2*. Źródło: opracowanie własne.

W literaturze naukowej istnieje wiele rozszerzeń i modyfikacji współczynnika LOF, między innymi:

- współczynnik COF (*ang. connectivity outlier factor*, COF) [100], będący również podstawą badań w ramach kilku publikacji, został uwzględniony w pięciu opracowaniach naukowych, w których autor niniejszej rozprawy miał udział [82, 85, 101, 102, 84],
- współczynnik MDEF (*ang. multi-granularity deviation factor*, MDEF) [103].

2.5.2 Metody wykorzystujące klasyfikację

Wśród metod identyfikacji anomalii szczególną uwagę zwracają techniki klasyfikacyjne. Metody te opierają się na założeniu, że możliwe jest stworzenie klasyfikatora, który skutecznie rozróżnia klasy normalne od anomalii w danej przestrzeni cech. W zależności od dostępnych etykiet w fazie uczenia, techniki klasyfikacyjne wykrywające anomalie dzielą się na dwie główne kategorie: techniki wieloklasowe i jednoklasowe. Model klasyfikacji tworzony na podstawie zbioru treningowego (np. drzewa decyzyjnego) jest następnie oceniany przy użyciu zbioru testowego. Jeśli model wykazuje odpowiednią jakość, możemy go wykorzystać do klasyfikowania nowych obiektów jako typowe lub anomalie. Etapy konstrukcji takiego modelu przedstawiono na rysunku 2.16.



Rysunek 2.16: Proces budowy modelu klasyfikacyjnego. Źródło: opracowanie własne na podstawie [86].

Sieci neuronowe

Jedną z metod klasyfikacyjnych stosowanych do identyfikacji wartości odstających są sieci neuronowe. Te modele mają zdolność do uczenia się, co polega na dostosowywaniu się do określonych reakcji na sygnały wejściowe poprzez proces trenowania. W kontekście klasyfikacji wieloklasowej, sieci neuronowe są najpierw trenowane na zbiorze danych, który reprezentują różne klasy obiektów, co pozwala im nauczyć się rozróżniać te klasy. Następnie, nowy obiekt wejściowy jest przetwarzany przez sieć. Akceptacja przez sieć oznacza, że obiekt jest typowy, natomiast brak akceptacji wskazuje na anomalię [104]. W artykule [105] przeprowadzono przegląd i klasyfikację metod wykrywania anomalii opartych na głębokim uczeniu, omawiając ich podstawowe założenia, funkcje celu oraz zalety i wady. Podkreślono w nim rosnące znaczenie technik głębokiego uczenia w detekcji anomalii. Można stosować różne rodzaje sieci neuronowych, takie jak sieci Hopfielda,

wielowarstwowe perceptrony oraz sieci autoasocjacyjne. Poniżej przedstawiono podstawowe informacje związane z wykrywaniem anomalii za pomocą autoenkoderów oraz map samoorganizujących się SOM (*ang. self-organizing map*, SOM). Techniki te są wykorzystywane w badaniach przeprowadzonych w ramach niniejszej rozprawy i zostały szerzej omówione w rozdziale 4.

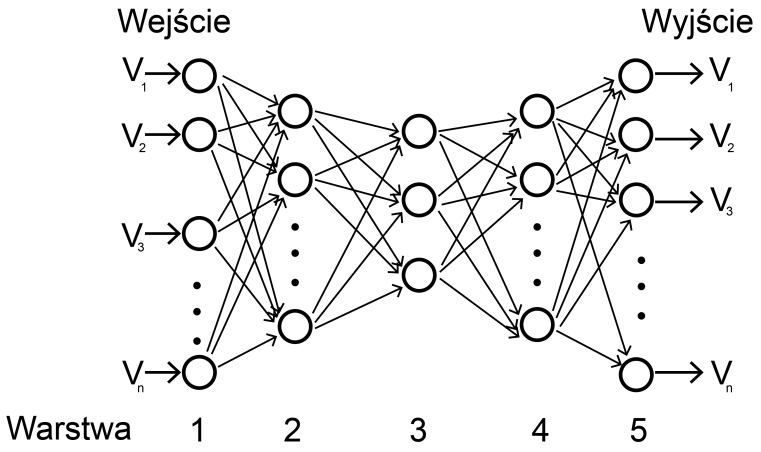
Autoenkodery są używane jako metoda jednoklasowa do wykrywania anomalii. Uczą się one kompresji i rekonstrukcji danych typowych, co pozwala na identyfikowanie danych odstających na podstawie wysokiego błędu rekonstrukcji. Autoenkodery mogą być również używane w kontekście wieloklasowym, na przykład w kombinacji z klasyfikatorami. Po kompresji danych przez autoenkoder, skompresowane reprezentacje mogą być przekazywane do klasyfikatora na przykład SVM (*ang. support vector machine*, SVM) czy sieci neuronowej w celu klasyfikacji na różne klasy.

Mapy samoorganizujące się SOM są stosowane jako metoda jednoklasowa do wykrywania anomalii poprzez analizę błędu kwantyzacji. Trenuje się je na danych typowych, a dane odstające identyfikuje się na podstawie wysokiego błędu kwantyzacji. SOM mogą być również używane do klasyfikacji wieloklasowej. Po zakończeniu treningu, SOM mogą klasyfikować nowe dane wejściowe do różnych klas na podstawie ich lokalizacji na mapie, gdzie klasy są określane przez regiony odpowiadające różnym klasom danych wejściowych. W trakcie trenowania SOM na danych typowych, sieć dostosowuje swoje wagi do wzorców danych wejściowych, tworząc topologiczną reprezentację danych. Po zakończeniu treningu, nowe dane są mapowane na istniejącą strukturę SOM. Błąd kwantyzacji, definiowany jako różnica między wejściem a reprezentacją mapy, jest wskaźnikiem do identyfikacji anomalii. Obiekty z wysokim błędem kwantyzacji są uznawane za anomalne, ponieważ nie pasują do wcześniej zdefiniowanych wzorców.

Przykład klasyfikacji jednoklasowej można znaleźć w artykule [106], gdzie w wielowarstwowej sieci jednokierunkowej z trzema ukrytymi warstwami zastosowano replikator sieci neuronowej, którego schemat przedstawiono na rysunku 2.17. Funkcja aktywacji sieci odtwarza zbiór danych wejściowych w warstwie wyjściowej, minimalizując błąd sieci na etapie uczenia. W pierwszym kroku następuje kompresja zbioru obiektów X_i w trzech ukrytych warstwach, a w drugim rekonstrukcja każdego obiektu przy wykorzystaniu wyuczonej sieci do uzyskania obiektu na wyjściu X_i'. Do wyznaczenia miary odchylenia obiektu wykorzystuje się równanie (2.17):

$$OF_i = \frac{1}{m} \sum_{j=1}^{m} (x_{ij} - x'_{ij})^2, \tag{2.17}$$

gdzie m jest liczbą cech opisujących obiekt, a x_{ij} to wartość j-tej cechy dla i-tego obiektu przed rekonstrukcją, natomiast x'_{ij} to zrekonstruowana wartość tej cechy. Różnica $(x_{ij} - x'_{ij})$ odzwierciedla odchylenie między oryginalną a zrekonstruowaną cechą. Obiekty z najwyższymi wartościami OF_i są uznawane za anomalie.



Rysunek 2.17: Replikator sieci neuronowej z n jednostkowymi wejściami i n wyjściami. Źródło: opracowanie własne na podstawie [106].

Naiwny klasyfikator Bayesa

Jednym z możliwych rozwiązań jest zastosowanie metody Bayesa, znanej jako naiwny klasyfikator Bayesa. Metoda ta wykorzystuje prosty klasyfikator probabilistyczny, który zakłada niezależność zmiennych objaśniających. Choć założenie to często nie jest spełnione w rzeczywistości, dlatego metoda jest nazywana „naiwną". Model prawdopodobieństwa oparty jest na twierdzeniu Bayesa:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.18}$$

gdzie A i B są zdarzeniami, przy czym $P(B) > 0$. $P(A|B)$ oznacza prawdopodobieństwo warunkowe zajścia zdarzenia A pod warunkiem zajścia zdarzenia B, natomiast $P(B|A)$ oznacza prawdopodobieństwo zajścia zdarzenia B pod warunkiem zajścia zdarzenia A.

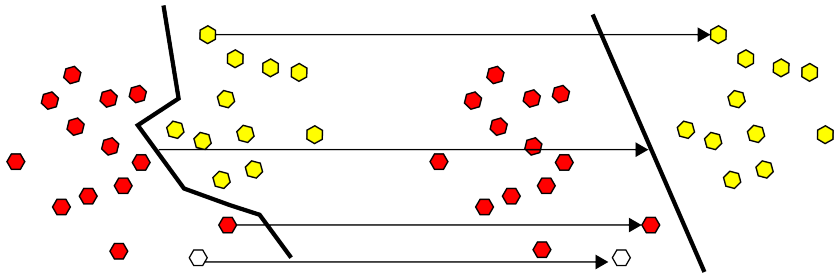
Metoda ta jest stosowana do klasyfikacji wieloklasowej. W pierwszym kroku szacowane są prawdopodobieństwa przynależności obserwacji do różnych klas, które reprezentują typowe obiekty i anomalie. W drugim kroku następuje estymacja warunkowa przynależności nowych obiektów do grup typowych obiektów i grup anomalii. Podstawowa technika opisana powyżej zakłada niezależność pomiędzy różnymi atrybutami. Zaproponowano jednak różne warianty tej techniki, które uwzględniają zależności warunkowe między atrybutami, wykorzystując bardziej złożone sieci Bayesa, takie jak warunkowe sieci Gaussa CGN (*ang. conditional gaussian network*, CGN) [107] oraz dynamiczne sieci Bayesa DBN (*ang. dynamic bayesian network*, DBN) [108], które diagnozują usterki w systemach elektronicznych. DBN modelują proces degradacji i przejścia między stanami systemu za pomocą łańcuchów Markowa, co pozwala na identyfikację nietypowych stanów systemu.

Metoda wektorów wspierających

Jeszcze innym podejściem jest metoda wektorów wspierających SVM (*ang. support vector machines*, SVM) [109]. Metoda ta jest nadzorowanym algorytmem uczenia maszynowego, który może być wykorzystywany zarówno do klasyfikacji, jak i regresji. Podstawowym celem SVM jest znalezienie hiperpłaszczyzny, która najlepiej oddziela elementy dwóch klas w zbiorze uczącym. Algorytm stara się maksymalizować margines separacji między klasami, co skutkuje zwiększeniem odporności modelu na błędne klasyfikacje.

Maszyna wektorów nośnych SVM jest w swojej podstawowej formie klasyfikatorem dwuklasowym. W rzeczywistości jednak często spotykamy się z problemami, które wymagają klasyfikacji na więcej niż dwie klasy (K > 2). Aby rozwiązać takie problemy, opracowano różne techniki łączenia wielu dwuklasowych modeli SVM w celu stworzenia klasyfikatora wieloklasowego. Jednym z często wykorzystywanych podejść [110] jest metoda polegająca na trenowaniu K oddzielnych modeli SVM, gdzie model $h_k(x)$ jest uczony przy użyciu danych z klasy C_k jako pozytywnych przykładów, a danych z pozostałych $K - 1$ klas jako negatywnych przykładów. Ta metoda jest znana jako podejście „jeden kontra reszta".

Na rysunku 2.18 pokazano przetransformowane obiekty za pomocą funkcji jądrowych na przestrzeń pokazaną po prawej stronie. Nowa przestrzeń przedstawia dwie grupy liniowo separowalne, a biały sześcian oznacza nowy obiekt. W tym przypadku metoda SVM wykorzystywana jest do klasyfikacji jednoklasowej, gdzie w pierwszym kroku poszukiwana jest hiperpłaszczyzna oddzielająca obiekty typowe od anomalii. W drugim kroku następuje weryfikacja, czy nowa obserwacja należy do wyuczonego zbioru. Jeśli nie, jest uznawana za anomalię.



Rysunek 2.18: Zastosowanie funkcji jądrowych w metodzie wektorów nośnych – „zmapowanie" oryginalnych obiektów na przestrzeń liniowo separowalną. Źródło: opracowanie własne.

W dużych zbiorach danych często wykorzystuje się klasyfikator nieliniowy z funkcjami jądrowymi (*ang. kernel functions*), które potrafią modelować skomplikowane, nieliniowe zależności pomiędzy danymi. Algorytmy SVM używają funkcji jądrowych do przekształcania oryginalnych danych na przestrzeń cech, w której dane mogą być łatwiej rozdzielane liniowo. SVM są szczególnie skuteczne w przypadku złożonych i wysokowymiarowych

danych, dzięki czemu znajdują szerokie zastosowanie w różnych dziedzinach, takich jak bioinformatyka, rozpoznawanie obrazów czy analiza tekstu. Dzięki wykorzystaniu funkcji jądrowych [111] SVM może efektywnie klasyfikować dane, które nie są liniowo separowalne w oryginalnej przestrzeni cech. Jednakże, SVM ma również pewne ograniczenia, takie jak konieczność dostosowania hiperparametrów i wydłużony czas obliczeń dla bardzo dużych zbiorów danych.

System ekspertowy

Efektywnym rozwiązaniem może być zastosowanie algorytmu opartego na regułach decyzyjnych. Systemy, które bazują na regułach decyzyjnych do reprezentowania wiedzy, nazywane są inaczej systemami regułowymi (*ang. rule-based systems*). Popularność tej metody wynika z wielu zalet tego formalizmu. Do głównych zalet reguł należą: prostota składni oraz uniwersalność zastosowań. Prostota składni sprawia, że reguły są zrozumiałe nawet dla osób, które nie są specjalistami w dziedzinie informatyki, zwłaszcza w kontekście systemów ekspertowych (SE).

Systemy ekspertowe są jednym z przykładów systemów opartych na regułach decyzyjnych, mogą być stosowane w różnych dziedzinach, takich jak medycyna, technika, ekonomia i zarządzanie, co świadczy o ich szerokim spektrum zastosowań. Wiedzę można zaimplementować do systemu ekspertowego w formie zestawu reguł decyzyjnych. Reguły w systemach ekspertowych można najogólniej zapisać w następujący sposób, opisując ich składowe tak, jak najczęściej są używane w systemach ekspertowych [112]:

$$\text{<warunki>} \rightarrow \text{<konkluzje>},$$

gdzie:

- <warunki> $= \{w_1, w_2, \ldots, w_n\}$, gdzie $w_i = \langle X_i, A_j, Q_j \rangle$ lub $\langle A_j, Q_j \rangle$,
- <konkluzje> $= \{k_1, k_2, \ldots, k_m\}$, gdzie $k_i = \langle X_i, A_j, Q_j \rangle$ lub $\langle A_j, Q_j \rangle$ lub akcje,

gdzie:

- X_i – obiekt,
- A_j – atrybut,
- Q_j – zbiór wartość atrybutu A_j, do którego należy $x_{i,j}$.

Proces indukcji reguł opiera się na zbiorze danych D, który składa się z par $\{X, y\}$, gdzie $X = [X_1, X_2, \ldots, X_n]$, czyli jest to zbiór wszystkich obiektów X_i, a każdy obiekt X_i to wektor atrybutów $[x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$, gdzie $x_{i,j} \in Q_j$, a Q_j to dziedzina wartości dla zmiennej A_j. Zmienna zależna $y = [y_1, y_2, \ldots, y_n]$ przypisana jest każdemu obiektowi X_i, przy czym wartość y_i odpowiada jednej z konkluzji k_i w zbiorze $\{k_1, k_2, \ldots, k_m\}$. Każdy warunek w_i w regule odpowiada wektorowi atrybutów obiektu X_i w zbiorze danych D, natomiast każda konkluzja k_i odpowiada wartości y_i przypisanej do obiektu X_i. W ten

sposób proces indukcji reguł przekształca zbiór danych wejściowych D oraz ich odpowiadające wartości wyjściowe y w zestaw reguł decyzyjnych, które mogą być używane do klasyfikacji nowych danych oraz wykrywania anomalii.

Systemy ekspertowe mogą wykorzystywać różne algorytmy do generowania reguł decyzyjnych, które są następnie używane do identyfikacji anomalii. Wyróżnia się dwa główne podejścia do wyodrębniania reguł:

- **algorytm sekwencyjnego pokrywania** – w pierwszym kroku generowana jest reguła pokrywająca maksymalnie dużo przypadków z danej kategorii c_i. W kolejnych krokach generowane są kolejne reguły opisujące przypadki, które nie zostały jeszcze pokryte,
- **algorytm drzew decyzyjnych** – wykorzystuje koncepcję „dziel i rządź", stopniowo dzieląc przestrzeń wejściową na coraz mniejsze podprzestrzenie.

Metoda polega na uczeniu się reguł, które opisują normalne zachowanie obiektów. Obiekty, które nie pasują do żadnej z tych reguł, są uznawane za anomalie. Najlepsze efekty można osiągnąć, gdy dostępni są eksperci dziedzinowi, którzy mogą dokładnie ocenić, co stanowi prawdziwą anomalię, a co nią nie jest.

W przypadku klasyfikacji wieloklasowej, w pierwszym kroku tworzymy reguły decyzyjne na podstawie zbioru uczącego, stosując odpowiednio dobrany algorytm. Regułom przypisujemy stopień ufności, który jest wskaźnikiem określającym stosunek liczby obiektów poprawnie zaklasyfikowanych przez regułę do liczby wszystkich obiektów w zbiorze uczącym. W drugim kroku znajdujemy regułę dla każdego testowanego obiektu. Obiekt nietypowy jest identyfikowany na podstawie odwrotności poziomu ufności powiązanego z najlepszą regułą. Klasyfikacja jednoklasowa wykorzystuje reguły asocjacyjne. Metoda odkrywania asocjacji analizuje zbiór atrybutów w bazie danych pod kątem występowania w nim powtarzających się zależności. Reguły asocjacyjne przedstawiane są w postaci implikacji i charakteryzowane dwoma parametrami:

- **wsparcie reguły** (*ang. rule support*) – parametr określający, jaki procent wszystkich transakcji w zbiorze danych zawiera daną regułę,
- **pewność reguły** (*ang. rule confidence*) – parametr określający, jaki procent transakcji zawierających poprzednik reguły zawiera również jej następnik.

Dla wygenerowanej reguły obliczane są współczynniki pewności oraz wsparcia. Jeśli ich wartości przekraczają określone minimum, reguła jest akceptowana. Wartość progowa wsparcia jest wykorzystywana do odrzucania reguł o niskim współczynniku wsparcia.

2.5.3 Metody grupowania podobnych do siebie obserwacji

Wykrywanie odchyleń w ramach metod grupowania jest integralną częścią samego procesu grupowania. Obserwacje, które nie mogą zostać włączone do żadnej grupy, mogą być traktowane jako obserwacje nietypowe. Metody te można podzielić przyjmując trzy założenia co do obserwacji odstającej:

Obserwacja odstająca nienależąca do żadnej grupy

W przypadkach, gdy obserwacja odstająca nie przynależy do żadnej z istniejących grup obiektów typowych, efektywnym rozwiązaniem może być zastosowanie algorytmu ROCK (*ang. a robust clustering algorithm for categorical attributes*, ROCK) [113]. Algorytm ten, opracowany specjalnie dla danych o atrybutach kategorycznych, wykorzystuje unikalne podejście bazujące na pojęciach sąsiadów (*ang. neighbors*) oraz połączeń (*ang. links*) [114] zamiast tradycyjnych metryk odległości, co odpowiada na ograniczenia metryk w kontekście danych kategorycznych. ROCK operuje na zasadzie określania liczby wspólnych sąsiadów między obiektami, co pozwala na grupowanie ich w sposób uwzględniający lokalne podobieństwa. Istotą algorytmu jest preferowanie połączeń między obiektami z większą liczbą wspólnych sąsiadów na wczesnym etapie procesu grupowania, co skutkuje tworzeniem bardziej spójnych grup. Sąsiadami obiektu są inne obiekty podobne do niego według wybranej miary podobieństwa, która może być metryczna (np. L1, L2) lub niemetryczna (np. zdefiniowana przez eksperta domenowego). Proces grupowania w algorytmie ROCK ma na celu nie tylko formowanie spójnych grup, ale także skuteczne oddzielanie obiektów odstających, co sprawia, że jest on szczególnie przydatny w analizie danych z dużą liczbą obiektów nietypowych. Obserwacje odstające prawdopodobnie będą miały bardzo niewielu lub żadnych sąsiadów, dlatego nie zostaną skutecznie przypisane do żadnej grupy, co ułatwia ich wykrycie jako anomalii. Algorytm ROCK jest hierarchicznym algorytmem grupowania, który przyjmuje na wejściu próbkowane obiekty danych D, z których tworzone są wstępne grupy, a następnie reszta obiektów jest przypisywana do najbardziej odpowiednich grup. Wadą jest jego złożoność czasowa, która wynosi $O(n^2 + nm_m m_a + n^2 \log(n))$, gdzie m_m oznacza maksymalną, a m_a średnią liczbę sąsiadów, a n liczbę obiektów danych. Złożoność przestrzenna tego algorytmu wynosi $O(\min\{n^2, nm_m m_a\})$. Dzięki zdolności do skutecznego grupowania i oddzielania obiektów nietypowych, ROCK jest wartościowym narzędziem w detekcji anomalii, zwłaszcza w zbiorach danych z atrybutami kategorycznymi. Zostało to przypomniane w rozdziale 3.1 omawiającym techniki grupowania danych kategorycznych.

W sytuacjach, gdy skupienie definiowane jest jako obszar przestrzeni zawierający obiekty o wysokiej gęstości, otoczony regionem o niskiej gęstości, efektywnym rozwiązaniem może być zastosowanie algorytmu DBSCAN, który grupuje skupienia jako maksymalnie gęste, połączone zbiory obserwacji. DBSCAN jest skuteczny w wykrywaniu grup o dowolnym kształcie oraz identyfikacji punktów odstających, które są izolowane od innych danych. Dzięki możliwości określania punktów wewnętrznych i brzegowych, DBSCAN jest w stanie skutecznie wykrywać nie tylko pojedyncze punkty odstające, ale również małe grupy anomalii. Może to być szczególnie istotne w analizie wykrywania oszustw, czy innych anomalii w zbiorach danych o zróżnicowanej gęstości. Czas wykonania algorytmu jest rzędu $O(n \log n)$, jeśli zapytania o sąsiednie punkty są efektywnie obsługiwane przez specjalne struktury danych, takie jak R-drzewa (*ang. R-trees*) lub drzewa KD (*ang. KD-trees*), zaprojektowane do zarządzania i wyszukiwania danych przestrzennych.

Dodatkowo powstało wiele modyfikacji algorytmu, na przykład algorytm LDBSCAN [115] rozwija koncepcję DBSCAN, wprowadzając lokalne miary gęstości do identyfikacji grup w zbiorach danych o zróżnicowanej gęstości lokalnej. LDBSCAN korzysta z lokalnego współczynnika osobliwości LOF do identyfikacji anomalii, co sprawia, że jest szczególnie przydatny w wykrywaniu małych, izolowanych skupisk danych odstających. Dzięki temu podejściu, LDBSCAN może efektywnie identyfikować zarówno duże, jak i małe grupy, jednocześnie eliminując szumy, co czyni go zaawansowanym narzędziem w analizie danych przestrzennych. Inny przykład to GDBSCAN [116]. Mechanizm konstruowania poszczególnych skupień określają następujące definicje:

- Niech $d(X_i, X_j)$ oznacza odległość między dowolnymi obiektami $X_i, X_j \in D$ i niech $\epsilon > 0$ będzie parametrem. Przez ϵ-sąsiedztwo obiektu X_i rozumiemy zbiór:

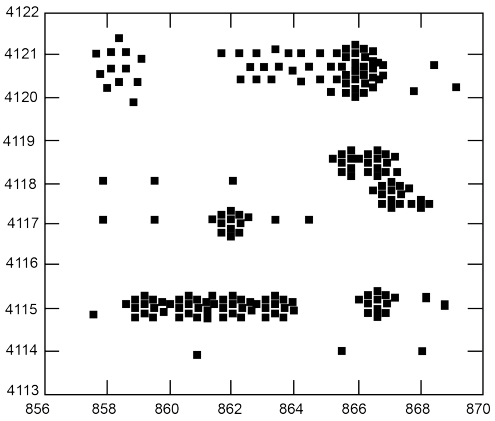
$$N_\epsilon(X_i) = \{X_j \in D : d(X_i, X_j) \leq \epsilon\}, \tag{2.19}$$

- Obiekt $X_i \in D$ nazywamy punktem wewnętrznym skupienia, jeżeli jego ϵ-sąsiedztwo zawiera przynajmniej $MinPts$ obiektów, tzn. gdy $|N_\epsilon(X_i)| \geq MinPts$, gdzie $MinPts$ jest parametrem,
- Obiekt $X_i \in D$ nazywamy punktem brzegowym, jeżeli $|N_\epsilon(X_i)| < MinPts$, ale sąsiedztwo to zawiera przynajmniej jeden punkt wewnętrzny,
- Jeżeli $X_i \in D$ nie jest ani punktem wewnętrznym, ani punktem brzegowym, to traktujemy go jako zakłócenie (anomalię).

Konstruując poszczególne skupienia, korzysta się z pojęcia *osiągalności gęstościowej*. Obiekt $X_j \in D$ jest bezpośrednio *gęstościowo osiągalny* z obiektu $X_i \in D$, jeżeli $X_j \in N_\epsilon(X_i)$, a ponadto X_i jest otoczony dostatecznie dużą liczbą innych obiektów (tzn. jest punktem wewnętrznym). Wprowadza się też pojęcie *spójności gęstościowej*: obiekty $X_i, X_j \in D$ są *gęstościowo połączone*, jeżeli istnieje taki obiekt X_k, że zarówno X_i, jak i X_j są gęstościowo osiągalne z X_k [77]. Algorytm DBSCAN jest wygodny do wykorzystania, gdy skupienia mają nieregularne kształty, a zbiór danych zawiera obserwacje odstające lub zaszumione, co przedstawiono na rysunku 2.19.

Obserwacja odstająca i centroid grupy

Przypadek, w którym obserwacja odstająca (nietypowa) znajduje się daleko od najbliższego centroida grupy, a obserwacje typowe leżą blisko centroida grupy, jest dobrze reprezentowany przez algorytm k-średnich (*ang. k-means*). Centroid to punkt, który reprezentuje środek grupy obiektów, będący średnią arytmetyczną współrzędnych wszystkich punktów w tej grupie [117, 111]. Pierwsze sformułowania dotyczące algorytmu k-średnich pojawiają się w pracach Steinhausa [118] z 1956 roku oraz Lloyda [119] z 1957 roku. W artykule [118] przedstawiono matematyczne podstawy metody k-średnich, która polega na podziale zbioru na części w sposób minimalizujący sumę momentów bezwładności tych części. To



Rysunek 2.19: Skupienia o nieregularnych kształtach i różnej gęstości zawierające elementy odstające. Źródło: opracowanie własne.

fundamentalne dzieło miało istotny wpływ na rozwój metod grupowania danych i znalazło szerokie zastosowanie w analizie danych oraz algorytmach uczenia maszynowego. Artykuł Lloyda [119] dotyczy optymalizacji kwantyzacji. Koncepcyjne podejście do minimalizacji błędów i optymalnego podziału przestrzeni ma analogie do algorytmu k-średnich stosowanego w grupowaniu danych. Algorytm ten jest jednym z najważniejszych i najbardziej popularnych algorytmów wykorzystywanych w eksploracji danych. Jego ponad 50-letnią historię opisano w pracy [117].

Algorytm polega na przenoszeniu obiektów z grupy do grupy tak długo, aż zostaną zoptymalizowane zmienności wewnątrz grup oraz między grupami. Każde skupienie jest reprezentowane za pomocą jego średniej (*ang. mean*) lub średniej ważonej (*ang. weighted mean*). Grupa jest definiowana przez swoje centrum, a zadaniem algorytmu jest znalezienie podziału zbioru n obiektów $D = \{X_1, X_2, \ldots, X_n\}$ pomiędzy k grup $C_1, C_2, \ldots, C_k$ o średnich $m_1, m_2, \ldots, m_k$, który minimalizuje funkcję kryterialną $e(k)$. W podstawowej wersji algorytmu k-średnich minimalizowaną funkcją kryterialną jest funkcja sumy błędów średniokwadratowych SSE (*ang. sum of squared errors*, SSE):

$$e(k) = \sum_{i=1}^{k} \sum_{X_j \in C_i} \text{dist}(X_j, m_i)^2, \tag{2.20}$$

gdzie X_j jest obiektem w zbiorze danych D, m_i oznacza średnią grupy C_i, a $\text{dist}(X_j, m_i)$ to odległość euklidesowa (norma L_2) między obiektem X_j a średnią (środkiem) najbliższego skupienia C_i. Wartość funkcji kryterialnej $e(k)$ jest sumą SSE poszczególnych skupień [86]. Algorytm oraz jego usprawnienia i różne warianty (np. sferyczny czy jądrowy algorytm k-średnich) zostały szeroko opisane w rozdziale drugim poświęconym algorytmom kombinatorycznej analizy skupień w książce [77].

Po wykonaniu procesu grupowania można rozpocząć identyfikację anomalii, polegającą na analizie obiektów, które znajdują się daleko od centroidów swoich skupień. Takie obiekty, ze względu na swoją odległość od środka grupy, są traktowane jako anomalie. Przykład takiego podejścia pokazano w artykule [120], gdzie zaproponowano metodę „odcinania", polegającą na wyznaczeniu takiego podzbioru obserwacji (o z góry określonej liczebności), którego usunięcie prowadzi do maksymalnej poprawy wskaźników jakości wynikowego podziału lub znacznej poprawy wartości przyjętej funkcji kryterialnej. Odcinane są obiekty, które znajdują się najdalej od centroidów grup, czyli te, które mają największy wpływ na zniekształcenie wyników grupowania. Po odcięciu tych obiektów, są one traktowane jako anomalie, a proces grupowania jest kontynuowany na pozostałych obiektach, prowadząc do bardziej wiarygodnych wyników.

Łącząc podejście probabilistyczne do grupowania obiektów z paradygmatem algorytmu iteracyjno-optymalizacyjnego k-średnich, otrzymujemy algorytm EM (*ang. expectation maximization*, EM) [121]. Omawiany wcześniej algorytm k-średnich jest specjalnym przypadkiem algorytmu EM dla mieszanin gaussowskich, gdzie zakłada się, że macierze kowariancji są sferyczne i równe oraz wagi mieszaniny są jednakowe. W takim uproszczeniu EM przekształca się w algorytm k-średnich, gdzie w kroku E każdy obiekt przypisywany jest do najbliższego centrum grupy, a w kroku M centra grup są aktualizowane jako średnie przypisanych do nich obiektów [122]. W podrozdziale poświęconym modelom generatywnym dla danych kategorycznych (3.2.2) dodatkowo omówiono algorytm EM, przedstawiając szczegółowe wyjaśnienia jego zastosowań i znaczenia w kontekście analizy danych kategorycznych. Algorytm EM naprzemiennie wykonuje dwa kroki [77]:

- **Krok E** (wyznaczanie wartości oczekiwanej parametrów) – przykładowo każdy obiekt jest przydzielany do grupy o najbliższym środku ciężkości,
- **Krok M** (maksymalizacja funkcji wiarygodności na podstawie przyjętych wartości) – na przykład stosując metodę najmniejszych kwadratów, wyznacza się nowe współrzędne środków ciężkości.

Algorytm zakłada, że dane są generowane przez pewien proces statystyczny, a celem jest znalezienie modelu, który najlepiej opisuje zbiór danych. Jest to bardziej zaawansowana odmiana algorytmu k-średnich, ponieważ potrafi radzić sobie z danymi brakującymi oraz zmiennymi ukrytymi, co jest trudne do osiągnięcia przy użyciu standardowego algorytmu. Zmienne ukryte (*ang. latent variables*) to zmienne, które nie są bezpośrednio mierzone ani obserwowane w zbiorze danych, ale które mogą wpływać na zależności i struktury w tych danych. Jednym z popularnych zastosowań algorytmu EM jest dopasowanie mieszanin gaussowskich GMM (*ang. gaussian mixture model*, GMM), gdzie wykorzystywane są rozkłady gaussowskie do modelowania danych. Model zakłada, że dane pochodzą z mieszaniny kilku rozkładów gaussowskich o różnych średnich i wariancjach. W tym przypadku algorytm EM pozwala na iteracyjne oszacowanie parametrów, takich jak średnie, wariancje oraz wagi mieszaniny, które najlepiej opisują rozkład danych. W kontekście wykrywania

anomalii, algorytm EM może być używany do identyfikacji obiektów, które nie są dobrze reprezentowane przez model generatywny, co ma znaczenie w zadaniach takich jak wykrywanie anomalii. Jeśli dane nie pasują do żadnego z rozkładów gaussowskich w mieszaninie, oznacza to, że znacznie odbiegają od wartości przewidywanych przez model, sugerując, że mogą być anomaliami. Dzięki iteracyjnemu aktualizowaniu parametrów modelu, algorytm EM może skutecznie wykrywać odchylenia w danych, które są trudne do zidentyfikowania przy użyciu innych metod.

W ramach założenia, że obserwacja odstająca znajduje się daleko od najbliższego centroida grupy, można również zastosować algorytm SOM, który jest algorytmem grupującym. Algorytm ten polega na wizualizacji wielowymiarowych danych za pomocą dwuwymiarowej mapy w taki sposób, aby obiekty bliskie sobie w przestrzeni danych były bliskie sobie na płaszczyźnie. Dane, które znajdują się daleko od swoich środków (neuronów), są traktowane jako odstające. Algorytm ten został zastosowany w badaniach i jest szerzej omówiony w rozdziale 4, w podrozdziale 4.5.1.

Obserwacja odstająca należąca do małych i rzadkich grup

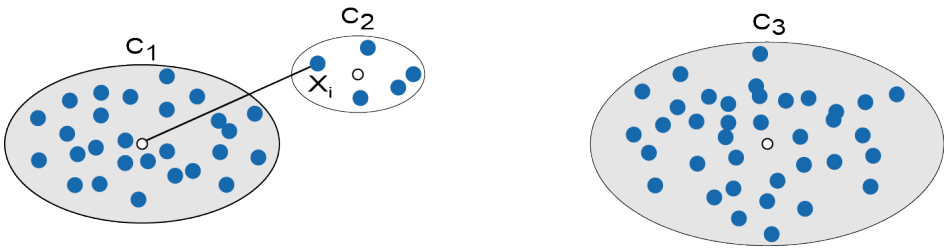
W sytuacji, gdy obserwacja odstająca należy do małych i rzadkich grup, a typowe obserwacje znajdują się w dużych i gęstych grupach, można zastosować algorytm FindCBLOF (*ang. cluster-based local outlier factor*, FindCBLOF), zaproponowany w pracy [123]. Współczynnik CBLOF to miara służąca do określenia znaczenia anomalii w kontekście lokalnym. Jest ona oparta na rozmiarze grupy, do której należy dany obiekt, oraz na odległości między obiektem a jego najbliższą dużą grupą (jeśli obiekt znajduje się w małej grupie). Algorytm FindCBLOF składa się z trzech głównych części: grupowania zbioru danych, identyfikacji dużych i małych grup oraz obliczania wartości CBLOF dla każdego obiektu:

- **grupowanie danych** - używamy dowolnego dobrego algorytmu grupowania,
- **identyfikacja dużych i małych grup** - klasyfikujemy grupy na duże i małe zgodnie z określonymi parametrami,
- **obliczenie CBLOF** - dla każdego obiektu w zbiorze obliczamy wartość CBLOF.

Wzór przedstawia sposób obliczania wartości współczynnika CBLOF dla obiektu X_i:

$$CBLOF(X_i) = |C(X_i)| \times \text{dist}(X_i, \text{centroid}(C(X_i))), \tag{2.21}$$

gdzie $|C(X_i)|$ jest rozmiarem grupy, do której należy obiekt X_i, a $\text{dist}(X_i, \text{centroid}(C(X_i)))$ jest odległością obiektu X_i od centroidu grupy $C(X_i)$. Wartości CBLOF przypisują wynik anomalii na podstawie odległości do najbliższej dużej grupy, pomnożonej przez rozmiar grupy, do której należy obiekt.



Rysunek 2.20: Dla obiektu X_i, odległość od centrum grupy C_1 jest używana do obliczenia wartości *CBLOF(X_i)*. W tym przykładzie C_1 i C_3 są identyfikowane jako duże grupy, natomiast C_2 jest uznawana za małą grupę. Białe punkty to centra grup. Źródło: opracowanie własne na podstawie [124].

Rysunek 2.20 ilustruje tę koncepcję. Obiekt X_i leży w małej grupie C_2, więc wynik byłby równy odległości do C_1, która jest najbliższą dużą grupą, pomnożonej przez rozmiar C_2.

Autorzy [124] zaproponowali inny wariant algorytmu, nazwany *unweighted-CBLOF* pokazany w równaniu (2.22):

$$unweighted-CBLOF(X_i) = \begin{cases} \min(d(X_i, C_j)) & \text{jeśli } X_i \in SC, \text{ gdzie } C_j \in LC \\ d(X_i, C_i) & \text{jeśli } X_i \in C_i \in LC \end{cases} \quad (2.22)$$

gdzie:

- X_i oznacza obiekt, dla którego obliczana jest wartość *unweighted-CBLOF*. Nazwa *unweighted-CBLOF* oznacza, że przy obliczaniu współczynnika anomalii nie uwzględnia się wagi, którą w standardowym algorytmie CBLOF jest rozmiar grupy,
- SC (*ang. small clusters*) to zbiór małych grup,
- LC (*ang. large clusters*) to zbiór dużych grup,
- $d(X_i, C_j)$ oznacza odległość obiektu X_i od centroidu grupy C_j,
- C_j to grupa LC, najbliższa obiektowi X_i,
- C_i to grupa, do której należy obiekt X_i.

Wariant ten eliminuje wagę rozmiaru grupy przy obliczaniu współczynnika anomalii, koncentrując się wyłącznie na odległości obiektu od najbliższej dużej grupy lub od centroidu własnej grupy. Dzięki temu *unweighted-CBLOF* zapewnia bardziej sprawiedliwe traktowanie obiektów w małych grupach i może prowadzić do lepszej detekcji anomalii, eliminując wpływ liczebności grupy na wynik.

2.5.4 Metody wykrywania anomalii oparte na rozkładzie danych

Metody oparte na rozkładzie danych polegają na uznawaniu danej obserwacji za anomalię, jeśli nie została wygenerowana przez założony proces stochastyczny. W tym podejściu, typowe obiekty mają wysokie prawdopodobieństwo wystąpienia w obrębie określonego modelu stochastycznego, podczas gdy anomalie mają niskie prawdopodobieństwo. W zależności od tego, czy rozkład zmiennych jest znany, można zastosować dwa rodzaje metod identyfikacji anomalii: (1) parametryczne, które zakładają określony rozkład danych, oraz (2) nieparametryczne, które nie wymagają takiego założenia.

(1) - Metody parametryczne

Metody parametryczne wymagają przyjęcia założenia dotyczącego rozkładu danych. Jeśli posiadamy podstawową wiedzę na temat rozkładu zmiennej, możemy przewidzieć, jaki będzie jej rozkład w kolejnych próbach o tej samej liczności. Zakładamy, że typowe obserwacje X_i pochodzą z parametrycznego rozkładu z wektorem parametrów Θ i funkcją gęstości prawdopodobieństwa $f(\mathrm{x}, \Theta)$, gdzie x jest wektorem wartości zmiennych opisujących obiekt X_i. Weryfikujemy zatem, czy x zostało wygenerowane z założonego rozkładu (z parametrami Θ), czy nie, co oznaczałoby, że X_i jest anomalią.

Rozkład gaussowski

W kontekście modeli opartych na rozkładzie gaussowskim, identyfikacja obserwacji odstających jest możliwa zarówno dla danych jednowymiarowych, jak i wielowymiarowych. Pierwszym krokiem w procesie detekcji wartości odstających powinna być odpowiednia wizualizacja danych, na przykład poprzez wykres kwantyl-kwantyl (*ang. Q-Q plot*) oraz histogram. Po przeprowadzeniu wizualizacji, możemy zastosować następujące metody:

- **reguła trzech sigm (odchyleń standardowych)** – zgodnie z zasadą trzech sigm, 99,7% wartości zmiennej mieści się w zakresie trzech odchyleń standardowych od średniej arytmetycznej. W związku z tym, obserwacje znajdujące się poza tym zakresem są uznawane za odstające,

- **metoda Tukeya (wykres pudełkowy)** [125] – zgodnie z metodą Tukeya, wszystkie obserwacje znajdujące się poza przedziałem $\langle Q_1 - 1,5 \cdot IQR; Q_3 + 1,5 \cdot IQR \rangle$, gdzie Q_1 oraz Q_3 są odpowiednio pierwszym i trzecim kwartylem, a IQR to rozstęp międzykwartylowy, są uznawane za obserwacje odstające,

- **zmodyfikowany wykres pudełkowy** – stosowany w przypadku rozkładów silnie asymetrycznych. Obserwacje znajdujące się w przedziale $\langle Q_1 - 1,5 \cdot \exp(-3,5 \cdot MC) \cdot IQR; Q_3 + 1,5 \cdot \exp(4 \cdot MC) \cdot IQR \rangle$, gdy $MC \geq 0$, oraz $\langle Q_1 - 1,5 \cdot \exp(-4 \cdot MC) \cdot IQR; Q_3 + 1,5 \cdot \exp(3,5 \cdot MC) \cdot IQR \rangle$, gdy $MC < 0$ (gdzie MC to medcouple [126], odporny estymator asymetrii), są uznawane za obserwacje odstające,

- **metoda Hampela** [127] – obserwacje X_i znajdujące się poza przedziałem $\langle \text{med}(D) - 3 \cdot \text{MAD}(D); \text{med}(D) + 3 \cdot \text{MAD}(D) \rangle$ są uznawane za odstające. Tutaj $\text{med}(D)$ oznacza medianę dla całego zbioru danych D, a $\text{MAD}(D)$ – odchylenie medianowe, które jest miarą odpornościową na odstające wartości w zbiorze danych [128],

- **test Grubbsa (test T)** [19] – jest to test statystyczny stosowany do identyfikacji obserwacji odstających, czyli wartości, które znacząco różnią się od reszty danych w zbiorze. Dane wykraczające poza przedział $\langle \bar{x} \pm 2 \cdot \sigma \rangle$ są uznawane za odstające. Test Grubbsa ocenia prawdopodobieństwo, że dane są obarczone błędem grubym, w oparciu o średnią $\bar{x}$ i odchylenie standardowe σ.

Dla danych wielowymiarowych możemy zastosować test Laurikkala [129]. Alternatywnie, można przeprowadzić digitalizację danych za pomocą metody zaproponowanej przez Pawlaka [130], która przekształca wielowymiarową informację w jedną wartość numeryczną, nie zaburzając relacji między obiektami, poprzez oddzielne kodowanie każdego atrybutu. W artykule Pawlaka przedstawiono matematyczny model systemu przechowywania i wyszukiwania informacji (*ang. information storage and retrieval*, w skrócie i.s.r., używanym przez autora), wykorzystujący języki pośrednie między rachunkiem zdań a rachunkiem predykatów. Zbadano logikę i kompletność tych języków, wprowadzono narzędzia algebraiczne do zarządzania dynamiką systemu oraz zaproponowano algorytmy implementacyjne. Szczególną uwagę poświęcono analizie i kodowaniu zbiorów opisowych dokumentów oraz algorytmowi implementacyjnemu opartemu na algebrze Boole'a. Dzięki temu podejściu możliwe jest dokładne przekształcanie danych wielowymiarowych w jedną wartość numeryczną, zachowując jednocześnie relacje między obiektami. Rozwiązanie to zastosowano w badaniach [96].

Analiza regresji

Obserwacje odstające możemy również wykryć, budując wielowymiarowy model regresji. W tym celu należy zbudować model, wykorzystując metodę najmniejszych kwadratów lub uogólnioną metodę najmniejszych kwadratów. W analizie regresji obserwacje nietypowe dzielimy na odstające (*ang. outliers*), wysokiej dźwigni (*ang. leverage*) oraz wpływowe (*ang. influential*). Można wyszukać obserwacje odstające, analizując reszty modelu regresji (standaryzowane, studentyzowane), a także wyszukać obserwacje wpływowe i te o wysokiej dźwigni. Obserwacje odstające mogą znacząco wpływać na wyniki modelu regresji, prowadząc do błędnych wniosków. Dlatego identyfikacja i analiza tych obserwacji jest niezwykle ważna dla poprawy jakości modelu.

Metody takie jak analiza reszt (standaryzowanych i studentyzowanych) oraz odległość Cooka (*ang. Cook's distance*) [131] i DFFITS (*ang. difference in fits*) [132] są używane do oceny wpływu poszczególnych obserwacji na model. Dzięki tym narzędziom możliwe jest nie tylko wykrywanie anomalii, ale również zrozumienie ich wpływu na analizowane dane, co prowadzi do bardziej precyzyjnych i wiarygodnych wyników.

Odległość Cooka jest to miara, która identyfikuje obiekty mające duży wpływ na dopasowane wartości modelu. Odległość Cooka łączy informacje o resztach (różnicy między obserwowanymi a przewidywanymi wartościami) oraz dźwigni (wpływ pojedynczego obiektu danych). Wskaźnik ten jest obliczany według wzoru (2.23):

$$D_i = \frac{(y_i - \hat{y}_i)^2}{p \cdot MSE} \left(\frac{h_{ii}}{(1 - h_{ii})^2} \right), \tag{2.23}$$

gdzie y_i to wartość obserwowana, $\hat{y}_i$ to wartość przewidywana, h_{ii} to dźwignia obiektu, p to liczba predyktorów, a MSE to średni błąd kwadratowy (oszacowanie wariancji reszt). Duża wartość odległości Cooka wskazuje, że obiekt danych ma silny wpływ na dopasowane wartości modelu.

DFFITS jest to miara, która wskazuje wpływ pojedynczego obiektu danych na dopasowane wartości modelu. Mierzy różnicę w dopasowanych wartościach z i bez konkretnego obiektu danych, skalowaną przez błąd standardowy dopasowania. Wzór na DFFITS jest przedstawiony w równaniu (2.24):

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{i(-i)}}{s_i \sqrt{h_{ii}}}, \tag{2.24}$$

gdzie $\hat{y}_{i(-i)}$ to przewidywana wartość bez i-tego obiektu danych, s_i to błąd standardowy, a h_{ii} to dźwignia. Ogólna zasada mówi, że wartość DFFITS większa niż $2 \cdot \sqrt{\frac{p+1}{n-p-1}}$ jest uznawana za dużą, co wskazuje na potencjalnie wpływowy obiekt. Zawsze n to liczba obserwacji, a p to liczba parametrów, w tym wyraz wolny (*ang. intercept*). Należy pamiętać, że nie jest to ścisła zasada, ale raczej wskazówka.

(2) - Metody nieparametryczne

W modelach nieparametrycznych struktura modelu jest dostosowywana do danych, a modele te nie wymagają wstępnych założeń co do struktury i liczby ich parametrów. Testy nieparametryczne są mniej restrykcyjne, jeśli chodzi o założenia co do natury i rozkładów badanych obiektów. Metody te są szczególnie użyteczne w sytuacjach, gdy nie można założyć normalności rozkładu danych lub gdy dane mają nietypowe rozkłady. Przykładami metod nieparametrycznych są:

- **test Kołmogorowa-Smirnowa (K-S)** [133, 134] – używany do porównywania rozkładu próbki z rozkładem teoretycznym lub do porównywania dwóch próbek między sobą. Test K-S jest jednym z najczęściej stosowanych testów nieparametrycznych w statystyce. Służy do oceny zgodności próbki z określonym rozkładem teoretycznym lub do porównania, czy dwie próby pochodzą z tego samego rozkładu. Gdy test wykaże istotne odchylenie, można dalej analizować konkretne obserwacje, które przyczyniają się do tego odchylenia jako potencjalne anomalie,

- **test Manna-Whitneya (M-W)** [135, 136] – to seminieparametryczny test statystyczny używany do porównywania dwóch niezależnych prób. Szczególnie przydatny, gdy nie można założyć normalności rozkładów. Test ten pozwala ocenić, czy wartości próbek pobranych z dwóch populacji są istotnie różne pod względem ich median. Test M-W może ujawnić, czy istnieje istotna różnica między medianami grup. Grupa z większą liczbą wartości odstających może powodować tę różnicę,

- **test Kruskala-Wallisa (K-W)** [137, 138] – to nieparametryczny test statystyczny używany do porównywania więcej niż dwóch grup. Często uważany jest za nieparametryczny odpowiednik analizy wariancji (ANOVA) i nie wymaga założenia normalności rozkładów danych. Stosując test Kruskala-Wallisa do wielu grup, można określić, które grupy różnią się znacząco od innych. Dalsza analiza tych grup może pomóc w zidentyfikowaniu anomalii,

- **regresja nieparametryczna** [73, 139] – takie metody jak regresja jądrowa (*ang. kernel regression*) czy splajny (*ang. splines*), gdzie struktura modelu jest elastyczna i dopasowuje się do kształtu danych. Modele regresji nieparametrycznej mogą wykazać, które obiekty danych znacznie odbiegają od przewidywań modelu. Te obiekty mogą być potencjalnymi anomaliami i wymagają dalszej analizy.

Metody nieparametryczne są cenione za swoją elastyczność i mniejsze wymagania dotyczące założeń, co czyni je odpowiednimi do analizowania danych rzeczywistych, które często nie spełniają restrykcyjnych założeń modeli parametrycznych.

Histogram

Prostym nieparametrycznym modelem gęstości prawdopodobieństwa jest histogram. W metodach bazujących na histogramie ustala się przedziały wartości zmiennej. W kroku testowym weryfikujemy, czy nowa obserwacja „wpada" do jednego z przedziałów histogramu, jeśli nie uznajemy ją za anomalię.

Jądrowa funkcja gęstości

W ramach metod nieparametrycznych można wyróżnić metody bazujące na jądrowej funkcji gęstości. Obrazowo mówiąc, metody te umożliwiają, aby modele liniowe zachowywały się w sposób nieliniowy. Jak wskazują Jacek Koronacki i Jan Ćwik [140], metody te polegają na nieliniowej transformacji obserwowanych wektorów $x_i \in \mathbb{R}^n$, przekształcając je w wektory w przestrzeni $\mathbb{R}^N$, zazwyczaj o znacznie większym wymiarze niż n (często nawet nieskończonym). Przestrzeń $\mathbb{R}^n$ jest nazywana przestrzenią obserwacji, natomiast przestrzeń $\mathbb{R}^N$ nazywa się przestrzenią cech, co pozwala matematycznie pozostać przy prostych metodach liniowych w przestrzeni $\mathbb{R}^N$. Każda obserwacja, która ma niskie prawdopodobieństwo w obszarze funkcji gęstości, jest uznawana za anomalię. Jądrowa funkcja gęstości jest często stosowana do estymacji gęstości prawdopodobieństwa zmiennej losowej i stanowi narzędzie w analizie statystycznej, szczególnie w kontekście wykrywania anomalii i modelowania rozkładów danych, które nie spełniają założeń normalności.

2.5.5 Metody spektralne

Metody spektralne odgrywają ważną rolę w nowoczesnej analizie danych, szczególnie w identyfikacji złożonych struktur grup. Grupowanie spektralne to zaawansowane narzędzie, które wykorzystuje techniki z zakresu teorii grafów i algebry liniowej, pozwalając na efektywne przetwarzanie danych o skomplikowanych i nieliniowych zależnościach. Metody spektralne umożliwiają wydobycie kluczowych cech grafów poprzez analizę ograniczonej liczby wektorów własnych macierzy opisujących grafy. Charakterystyczną cechą grupowania spektralnego jest wykorzystanie wektorów własnych do podziału na odpowiednie klasy. Laplasjan, znany również jako macierz Laplace'a, jest istotnym narzędziem w teorii grafów i analizie spektralnej. W kontekście grafów, laplasjan jest używany do analizy struktury grafu. Proces wyznaczania wektorów jest złożony obliczeniowo, jednak dostępne są efektywne metody obliczania wektora Fiedlera [141]. Pierwsze znaczące prace dotyczące zastosowania wektorów własnych laplasjanu nieskierowanego grafu pochodzą z lat 70-tych, na przykład artykuł z 1970 roku [142]. Kolejnym ważnym wkładem jest artykuł [143], który wnosi nowatorskie podejście do rozdzielania macierzy rzadkich z wykorzystaniem wektorów własnych macierzy związanych z grafem. Popularny i często cytowany jest artykuł [144], w którym zaproponowano zastosowanie algorytmu spektralnego do segmentacji obrazów.

W fazie grupowania dane są projektowane na przestrzeń o mniejszej liczbie wymiarów, co umożliwia efektywne grupowanie obiektów na podstawie ich wewnętrznych relacji. Szczegółowy przegląd różnych metod spektralnej analizy grupowania można znaleźć w pracach [145, 146, 77, 147]. Choć głównym celem tych badań jest grupowanie, metody spektralne mogą być również związane z wykrywaniem anomalii. Analiza wartości własnych i wektorów własnych macierzy podobieństw pozwala odkrywać ukryte struktury w danych, co umożliwia skuteczne grupowanie nawet w przypadku nieliniowych lub skomplikowanych relacji. Dzięki temu możliwe jest identyfikowanie anomalii jako obiektów, które nie pasują do żadnych zidentyfikowanych grup. Badania przedstawione w artykule [148] wykorzystują zaawansowane techniki spektralne do redukcji wymiarowości danych. Kluczowe w tej analizie są metody takie jak wielowymiarowe skalowanie MDS (*ang. multidimensional scaling*, MDS), lokalne osadzanie liniowe LLE (*ang. locally linear embedding*, LLE) oraz izometryczne mapowanie ISOMAP (*ang. isometric mapping*, ISOMAP). Po redukcji wymiarowości, anomalie są identyfikowane poprzez analizę rozkładów gęstości danych w zredukowanej przestrzeni wymiarowej. Wykazano, że metody te są niezwykle skuteczne w wykrywaniu anomalii w danych transportowych.

2.5.6 Identyfikacja anomalii kontekstowych i zbiorczych

Anomalie kontekstowe to obserwacje, które są uważane za nietypowe tylko w określonym kontekście. Na przykład wysoka sprzedaż lodów jest normalna latem, ale zimą byłaby anomalią. Wykrywanie takich anomalii wymaga uwzględnienia specyficznych warunków,

w jakich dane zostały zebrane. Artykuł [149] przedstawia metodę wykrywania anomalii kontekstowych o nazwie QCAD (*ang. quantile regression forests*, QCAD). QCAD oddziela cechy kontekstowe od cech behawioralnych, wykorzystując regresję kwantylową do bardziej kompleksowej analizy anomalii. Metoda ta pozwala na wyjaśnienie wykrytych anomalii, co jest istotne w zrozumieniu, dlaczego dane zdarzenie zostało uznane za anomalię. Artykuł podkreśla znaczenie kontekstu w analizie anomalii, co pozwala na bardziej precyzyjne i wyjaśnialne wyniki.

Metody głębokiego uczenia, takie jak RNN (*ang. recurrent neural networks*, RNN) i LSTM (*ang. long short-term memory*, LSTM), są szczególnie skuteczne w analizie sekwencji czasowych i identyfikacji wzorców, które wskazują na anomalie kontekstowe lub/i zbiorcze. Anomalie zbiorcze, odnoszą się do zestawów danych, które jako całość są nietypowe, mimo że poszczególne elementy mogą wydawać się normalne. Techniki takie jak autoenkodery i generatywne sieci antagonistyczne GAN (*ang. generative adversarial networks*, GAN) są często używane do wykrywania takich anomalii. Autoenkodery uczą się kompresji i rekonstrukcji danych, identyfikując anomalie na podstawie wysokich błędów rekonstrukcji. GAN generują dane, które mają być nieodróżnialne od rzeczywistych, a różnice między generowanymi a rzeczywistymi danymi mogą wskazywać na anomalie. Przykładowo, artykuł [150] przedstawia model wykrywający zbiorcze anomalie w czasie rzeczywistym, wykorzystujący LSTM do analizy serii czasowych i predykcji anomalii na podstawie błędów predykcji z wcześniejszych kroków czasowych.

2.5.7 Anomalie: uczenie maszynowe i głębokie

Rozpoznawanie wzorców to dziedzina zajmująca się automatycznym wykrywaniem regularności i anomalii w danych za pomocą algorytmów komputerowych oraz wykorzystywaniem tych wzorców do podejmowania decyzji, takich jak klasyfikacja zbioru danych D do różnych kategorii. Jest to podstawowy problem o długiej i bogatej historii sukcesów w nauce i technologii. Jednym z podejść, które znacznie poprawiło wyniki w rozpoznawaniu wzorców, jest uczenie maszynowe. Metoda ta wykorzystuje obszerny zbiór danych treningowych, na przykład zbiór N obiektów $\{X_1, \ldots, X_N\}$, do dostrojenia parametrów modelu adaptacyjnego, umożliwiając modelowi naukę klasyfikacji nowych danych na podstawie dostarczonych obiektów.

W istocie, podstawowe elementy obliczeniowe głębokiej sieci neuronowej są inspirowane klasycznymi algorytmami uczenia maszynowego, takimi jak regresja liniowa i logistyczna. Sieci neuronowe nabierają swojej mocy poprzez integrację licznych takich jednostek i wspólne uczenie ich wag, aby zminimalizować błąd przewidywań. Z tej perspektywy, sieć neuronowa może być postrzegana jako sieć elementarnych jednostek obliczeniowych, gdzie większa moc obliczeniowa jest osiągana dzięki ich specyficznemu połączeniu. Gdy sieć neuronowa jest stosowana w najprostszej formie, bez łączenia wielu jednostek, algorytmy uczenia są często redukowane do tradycyjnych modeli uczenia maszynowego [151].

W poprzednich podrozdziałach przedstawiono metody wykrywania anomalii przy użyciu różnych technik i podejść. Ten podrozdział poświęcony jest metodom uczenia maszynowego (*ang. machine learning*, ML) i głębokiego uczenia (*ang. deep learning*, DL). Chociaż bardziej szczegółowe omówienie tych metod znajduje się w rozdziale 4, a niektóre algorytmy, takie jak sieci neuronowe, były już wspomniane w ramach przeglądu metod, na przykład przy okazji metod wykorzystujących klasyfikację, metody te zasługują na szczególne uwzględnienie ze względu na dynamiczny rozwój i rosnące znaczenie w analizie danych. Dodatkowo, zostały one przedstawione na schemacie 2.11 na początku tego rozdziału jako osobna gałąź, co stanowi uzasadnienie dla powstania tego podrozdziału i krótkiego ich omówienia.

Uczenie maszynowe jest integralnym elementem sztucznej inteligencji (AI), umożliwiającym systemom komputerowym naukę z danych i stopniowe doskonalenie w realizacji określonych zadań bez potrzeby bezpośredniego programowania. AI wykorzystuje algorytmy, które automatycznie odkrywają wartościowe informacje, identyfikują ukryte wzorce, dokonują prognoz i podejmują decyzje na podstawie bieżących danych. W poprzednich podrozdziałach nie zaznaczono wystarczająco, że omawiane metody są częścią uczenia maszynowego lub głębokiego uczenia. Co więcej, nie wszystkie metody wymienione w przeglądzie technik wykrywania anomalii są bezpośrednio związane z uczeniem maszynowym lub głębokim uczeniem, choć wiele z nich jest często używanych w tych kontekstach. Większość z wymienionych metod może być stosowana w kontekście uczenia maszynowego i głębokiego uczenia, choć niektóre, szczególnie te z podrozdziału 2.5.4, są bardziej klasycznymi metodami statystycznymi. Uczenie maszynowe i głębokie uczenie często integrują te metody w bardziej złożone algorytmy i modele, aby poprawić dokładność i skuteczność wykrywania anomalii. Dzięki zdolności do automatycznego wykrywania wzorców i anomalii oferują one wyjątkowe możliwości, które są trudne do osiągnięcia za pomocą tradycyjnych technik. Dlatego też, aby uniknąć powtarzania wcześniejszych informacji, poniżej przedstawiono krótkie wyjaśnienia. Szczegółowe informacje można znaleźć w rozdziale 4. Oto krótki przegląd istotnych metod:

Metody wykorzystujące uczenie maszynowe

Uczenie maszynowe umożliwia trenowanie modeli na zbiorach danych zawierających zarówno normalne, jak i odstające obiekty. Oprócz SVM i *k*-NN, które zostały omówione w podrozdziałach 2.5.1 oraz 2.5.2, inne popularne algorytmy to:

- **drzewa decyzyjne** (*ang. decision trees*) [152] i **lasy losowe** (*ang. random forests*) [153] - algorytmy te są wykorzystywane do klasyfikacji i detekcji anomalii na podstawie struktury drzewa decyzyjnego. Dzięki ich zdolności do przetwarzania danych wielowymiarowych są efektywne w wykrywaniu nieprawidłowości w dużych zbiorach danych.

Metody głębokiego uczenia

Metody głębokiego uczenia wykorzystują sieci neuronowe o wielu warstwach, które są zdolne do automatycznej ekstrakcji cech z surowych danych. Popularne metody obejmują:

- **autoenkodery** [154, 155, 156] - te sieci neuronowe kompresują dane wejściowe do mniejszej liczby wymiarów, a następnie je rekonstruują. Anomalie są wykrywane na podstawie błędów rekonstrukcji,

- **rekurencyjne sieci neuronowe** (*ang. recurrent neural networks*, RNN) [157] oraz **długa pamięć krótkoterminowa** (*ang. long short-term memory*, LSTM) [158] - używane do analizy sekwencyjnych danych, takich jak sygnały czasowe, wykrywają anomalie na podstawie nietypowych wzorców w sekwencjach,

- **generatywne sieci antagonistyczne** (*ang. generative adversarial networks*, GAN) [159] - GAN tworzą modele generujące dane, które są nieodróżnialne od rzeczywistych danych. Anomalie są wykrywane na podstawie różnic między rzeczywistymi danymi a danymi generowanymi przez model.

2.6 Aktualne wyzwania i trendy w analizie anomalii

Dziedzina wykrywania anomalii dynamicznie się rozwija, napędzana przez rosnącą złożoność i objętość danych w różnych branżach. W miarę jak dane stają się coraz bardziej skomplikowane i różnorodne, pojawiają się nowe wyzwania związane z ich przetwarzaniem i analizą. Duże zestawy danych oraz wysoka wymiarowość mogą prowadzić do niestabilności algorytmów oraz zwiększonych kosztów obliczeniowych. Różnorodność typów atrybutów i wymiarowości danych wpływa na wybór odpowiednich algorytmów wykrywania anomalii. Dodatkowo, szum i brakujące wartości w danych mogą prowadzić do fałszywych alarmów, utrudniając dokładność analizy. Analiza danych niestrukturalnych, które są zróżnicowane i trudne do przetworzenia, stanowi dodatkową trudność. Aby sprostać tym wyzwaniom, badania i innowacje w dziedzinie wykrywania anomalii koncentrują się na następujących obszarach, poruszanych we współczesnej literaturze naukowej w kontekście rozwoju wykrywania anomalii [105, 160, 161]:

- **jakość i objętość danych**:

  Jakość danych jest priorytetowa dla skutecznego wykrywania anomalii, ponieważ od niej zależy dokładność i niezawodność procesów decyzyjnych. Dane niekompletne, niespójne lub zaszumione mogą prowadzić do błędnych wniosków, co w kontekście systemów automatycznych może skutkować kosztownymi błędami lub pominięciem istotnych sygnałów ostrzegawczych. W szczególności, w erze internetu rzeczy (*ang. internet of things*, IoT), gdzie urządzenia generują ogromne ilości danych, wyso-

ka jakość danych nabiera jeszcze większego znaczenia. Objętość danych stanowi wyzwanie pod względem ich przetwarzania i analizy w czasie rzeczywistym. Systemy wykrywające anomalie muszą być wyposażone w zaawansowane narzędzia zdolne do efektywnego zarządzania dużymi strumieniami danych, aby nie tylko szybko reagować na bieżące zagrożenia, ale również przewidywać przyszłe problemy. Nowoczesne techniki, takie jak głębokie uczenie (*ang. deep learning*) i uczenie maszynowe (*ang. machine learning*), odgrywają ważną rolę w optymalizacji procesów analizy danych. Narzędzia takie jak sieci neuronowe (*ang. neural networks*), lasy izolacyjne (*ang. isolation forests*) oraz autoenkodery (*ang. autoencoders*) zostały zaprojektowane, aby radzić sobie z rosnącą złożonością i zmiennością współczesnych zbiorów danych. Te zaawansowane podejścia umożliwiają bardziej precyzyjne wykrywanie anomalii, co jest niezbędne dla zapewnienia bezpieczeństwa i efektywności operacyjnej w różnych dziedzinach,

- **przetwarzanie w czasie rzeczywistym**:

  Wykrywanie anomalii w czasie rzeczywistym staje się coraz bardziej istotne, szczególnie w dziedzinach takich jak bezpieczeństwo sieci oraz procesy przemysłowe. Wyzwaniem w tym obszarze jest zapewnienie algorytmów o wysokiej wydajności oraz zasobów obliczeniowych, które są zdolne do obsługi nieprzerwanych strumieni danych. Algorytmy muszą być nie tylko szybkie, ale również dokładne, aby skutecznie identyfikować i reagować na anomalie w czasie rzeczywistym, co wymaga zaawansowanej optymalizacji i skalowalnych rozwiązań technologicznych. Dodatkowo, wprowadzenie rozproszonych systemów przetwarzania oraz technologii przetwarzania brzegowego (*ang. edge computing*) może znacząco poprawić zdolność do przetwarzania danych w czasie rzeczywistym, minimalizując opóźnienia i zwiększając efektywność operacyjną,

- **integracja uczenia maszynowego i AI**:

  Integracja uczenia maszynowego oraz sztucznej inteligencji (*ang. artificial intelligence, AI*) w nowoczesnych systemach wykrywania anomalii stanowi fundament współczesnych rozwiązań z zakresu cyberbezpieczeństwa. Uczenie maszynowe i AI całkowicie odmieniły sposób, w jaki interpretujemy dane, umożliwiając dynamiczne dostosowywanie się do nowych wzorców oraz automatyczne identyfikowanie anomalii z nieosiągalną wcześniej precyzją. Niemniej jednak, wdrożenie tych technologii wiąże się z pewnymi wyzwaniami. Podstawowym problemem jest potrzeba dostępu do obszernych i dobrze oznakowanych zbiorów danych szkoleniowych. Aby systemy AI mogły skutecznie rozpoznawać anomalie, muszą być trenowane na różnorodnych danych, co jest zarówno kosztowne, jak i czasochłonne. Co więcej, wysoka czułość algorytmów AI, zaprojektowanych do wykrywania nawet najdrobniejszych odchyleń od normy, może prowadzić do generowania dużej liczby fałszywych alarmów. Zarzą-

dzanie tymi fałszywymi alarmami wymaga dodatkowych zasobów i może prowadzić do tzw. „zmęczenia alarmami" (*ang. alert fatigue*), co w dłuższej perspektywie może obniżyć efektywność systemów wykrywania anomalii,

- **złożoność modeli**:

  W miarę jak systemy wykrywania anomalii ewoluują, ich złożoność rośnie. Bardziej zaawansowane modele, zwłaszcza te bazujące na głębokim uczeniu, mogą działać jako „czarne skrzynki". Oznacza to, że mimo ich wysokiej skuteczności w identyfikacji anomalii, trudniej jest zrozumieć, dlaczego podejmują określone decyzje. Brak przejrzystości w działaniu tych modeli stwarza wyzwania związane z diagnozowaniem problemów i wyjaśnianiem decyzji, co wpływa na zaufanie i akceptację tych technologii w środowisku biznesowym. Aby temu zaradzić, konieczne jest rozwijanie metod wyjaśnialnej sztucznej inteligencji XAI (*ang. explainable AI,* XAI), które umożliwiają lepsze zrozumienie i interpretację wyników generowanych przez te złożone modele,

- **adaptacja do nowych typów anomalii, skalowalność i elastyczność**:

  Systemy wykrywania anomalii muszą dynamicznie się rozwijać, aby sprostać rosnącej złożoności i różnorodności zagrożeń. Cyberprzestępcy stosują coraz bardziej zaawansowane techniki, takie jak ataki wykorzystujące luki dnia zerowego (*ang. zero-day*) oraz zaawansowane trwałe zagrożenia (*ang. advanced persistent threat*, APT). W związku z tym, niezwykle ważna staje się zdolność systemów do wykrywania nowych, wcześniej nieznanych anomalii. Tradycyjne podejścia mogą być niewystarczające w obliczu dynamicznie zmieniających się wzorców ataków i operacyjnych zakłóceń. Zaawansowane metody, takie jak głębokie uczenie i uczenie transferowe (*ang. transfer learning*), znacząco zwiększają zdolność systemów do adaptacji do nowych typów zagrożeń. Głębokie sieci neuronowe mogą wykrywać skomplikowane wzorce w danych, co umożliwia identyfikację subtelnych anomalii, które mogą wskazywać na nowe rodzaje ataków. Uczenie transferowe natomiast pozwala modelom na wykorzystanie wiedzy zdobytej na jednym zbiorze danych do efektywnego uczenia się na nowym, co jest niezwykle przydatne w dynamicznie zmieniającym się środowisku zagrożeń. Aby systemy wykrywania anomalii były skuteczne, konieczne jest ich ciągłe monitorowanie i aktualizowanie na podstawie najnowszych danych. Systemy wykrywania anomalii muszą być również skalowalne i elastyczne, aby skutecznie zarządzać rosnącą objętością i różnorodnością danych. W miarę wzrostu ilości danych, systemy te muszą być zdolne do przetwarzania ogromnych wolumenów informacji bez utraty wydajności. Elastyczność polega na zdolności systemów do adaptacji do różnych typów danych i anomalii. Zróżnicowanie danych, obejmujące zarówno dane strukturalne, jak i niestrukturalne, oraz ich dynamiczna natura, szczególnie w kontekście IoT, tworzą dodatkowe trudności

dla systemów. Aby osiągnąć wysoką skalowalność i elastyczność, konieczne jest wykorzystanie zaawansowanych technologii, takich jak przetwarzanie równoległe i rozproszone oraz nowoczesne techniki uczenia maszynowego,

- **niski współczynnik wykrywania anomalii**:

  Systemy wykrywania anomalii często zmagają się z wyzwaniem niskiego współczynnika wykrywania, wynikającym z rzadkości i różnorodności anomalii. Anomalie są z natury niezwykle rzadkie i heterogeniczne, co sprawia, że ich identyfikacja jest trudna. Skutkuje to częstymi przypadkami błędnej klasyfikacji normalnych instancji jako anomalii oraz pomijaniem skomplikowanych, rzeczywistych anomalii. Pomimo wprowadzenia licznych metod detekcji anomalii, obecne najnowocześniejsze techniki, zwłaszcza nienadzorowane, często generują wysokie wskaźniki fałszywych alarmów w rzeczywistych zestawach danych. Anomalie stanowią jedynie niewielki procent wszystkich danych, co znacznie utrudnia ich wykrycie. Nawet najbardziej zaawansowane algorytmy mogą nie wychwycić subtelnych anomalii, zwłaszcza gdy są one skomplikowane i różnorodne. Normalne przypadki są często błędnie identyfikowane jako anomalie, co prowadzi do wysokiej liczby fałszywych alarmów. To zjawisko jest szczególnie problematyczne w systemach nienadzorowanych, które nie mają dostępu do wcześniej oznakowanych danych anomalii. Chociaż zaproponowano wiele metod wykrywania anomalii, wciąż brakuje efektywnych rozwiązań, które mogą zminimalizować liczbę fałszywych alarmów przy jednoczesnym zwiększeniu dokładności wykrywania. Wykorzystanie metod zespołowych (*ang. outlier ensembles*), które łączą różne podejścia do wykrywania anomalii, może znacząco poprawić ich skuteczność. Tego typu techniki mogą wykorzystać zalety różnych algorytmów, redukując liczbę fałszywych alarmów i zwiększając skuteczność identyfikacji rzeczywistych anomalii,

- **wykrywanie anomalii w danych wielowymiarowych**:

  Wykrywanie anomalii w danych wielowymiarowych stanowi jedno z największych trudności współczesnej analizy danych. Wielowymiarowe przestrzenie danych charakteryzują się tym, że anomalie mogą być „rozmyte” i trudniejsze do zidentyfikowania, gdy liczba wymiarów rośnie. Anomalie, które są łatwe do wykrycia w danych o niskiej liczbie wymiarów, mogą stać się trudniejsze do zauważenia w danych o wysokiej liczbie wymiarów, ponieważ większa liczba wymiarów powoduje, że anomalie „znikają” w ogromnej ilości informacji. Główne wyzwanie polega na identyfikacji złożonych interakcji między cechami, które mogą wskazywać na obecność anomalii. Wielowymiarowe dane wymagają zaawansowanych metod redukcji wymiarów, które zachowują istotne informacje. Techniki takie jak analiza składowych głównych PCA, stochastyczna metoda porządkowania sąsiadów w oparciu o rozkład t (*ang. t-distributed stochastic neighbour embedding*, t-SNE), czy autoenkodery,

są powszechnie stosowane do zmniejszenia liczby wymiarów. Jednakże, nie zawsze udaje się im zachować wszystkie istotne informacje, co może prowadzić do pominięcia anomalii i trudności w identyfikacji złożonych interakcji cech. Różnorodność zależności czasowych, przestrzennych i opartych na grafach dodatkowo komplikuje proces wykrywania. Anomalie w danych czasowych mogą objawiać się jako niespodziewane wzorce lub zmiany w trendach, wymagając specjalistycznych metod, takich jak długa pamięć krótkoterminowa (*ang. long short-term memory*, LSTM) czy bramkowane jednostki rekurencyjne (*ang. gated recurrent unit*, GRU), do ich wykrywania. W przypadku danych przestrzennych, techniki takie jak splotowa sieć neuronowa (*ang. convolutional neural network*, CNN) mogą być użyteczne, ale wymagają one dostosowania do specyficznych wzorców przestrzennych. Zaawansowane podejścia, takie jak wykorzystanie głębokich sieci neuronowych do automatycznego uczenia się reprezentacji danych, mogą znacząco poprawić skuteczność wykrywania anomalii w danych wielowymiarowych. Modelowanie złożonych, często nieliniowych interakcji między cechami, może prowadzić do lepszego zrozumienia struktury danych i identyfikacji trudnych do zauważenia anomalii. Metody zespołowe mogą skutecznie radzić sobie z tym problemem, łącząc wyniki wielu różnych algorytmów w celu uzyskania bardziej precyzyjnych i stabilnych wyników,

- **efektywne uczenie się wzorców normalnych i anomalii z ograniczonych nieoznakowanych danych**:

  Efektywne wykrywanie wzorców normalnych i anomalii w sytuacjach, gdzie dostępne są jedynie ograniczone i nieoznakowane dane, stanowi istotną barierę w dziedzinie współczesnej analizy danych. Tradycyjne metody nadzorowane wymagają dużych zbiorów danych z etykietami, które są często trudne do uzyskania ze względu na wysokie koszty oraz złożoność procesu oznaczania danych. W odpowiedzi na te ograniczenia, coraz większe znaczenie zdobywają metody nienadzorowane i półnadzorowane, które uczą się na podstawie ograniczonych danych. Problemem w tym obszarze jest opracowanie metod, które skutecznie uczą się z ograniczonej liczby danych, jednocześnie zachowując zdolność do generalizacji i wykrywania nowych, nieznanych anomalii. Zaawansowane techniki, takie jak głębokie uczenie i metody zespołowe, mogą pomóc w rozwoju tej dziedziny, umożliwiając tworzenie bardziej efektywnych i adaptacyjnych systemów wykrywania anomalii,

- **wykrywanie złożonych anomalii odpornych na szum**:

  Radzenie sobie z niezgodnościami w oznakowanych danych, które często zawierają błędy, oraz adaptacja metod częściowo nadzorowanych, które zakładają, że dane treningowe są wolne od szumów, stanowi istotny problem w analizie danych. W rzeczywistości, dane te często są zanieczyszczone różnego rodzaju szumem, co utrudnia ich skuteczną analizę. Ważnym czynnikiem w wykrywaniu złożonych

anomalii jest dostosowanie metod pierwotnie zaprojektowanych do identyfikacji pojedynczych odchyleń do bardziej zaawansowanych zadań, takich jak wykrywanie anomalii warunkowych i grupowych. Wymaga to analizy różnorodnych źródeł danych, uwzględniających wysoką kardynalność, różnorodność rozkładów częstości, skomplikowane zależności i interakcje między cechami, a także radzenia sobie z szumem. Zaawansowane techniki, takie jak głębokie uczenie oraz metody zespołowe, mogą wspierać opracowywanie bardziej odpornych na szum i efektywnych systemów wykrywania anomalii, umożliwiając tym samym bardziej precyzyjną analizę danych w rzeczywistych, złożonych środowiskach,

- **wyzwania w identyfikacji istotnych podprzestrzeni**

  Identyfikacja istotnych podprzestrzeni jest wyjątkowo trudnym zadaniem, ponieważ liczba możliwych projekcji danych rośnie wykładniczo wraz ze wzrostem liczby wymiarów. Aby skutecznie wykrywać anomalie, konieczne jest przeszukiwanie danych i wymiarów w sposób zintegrowany, co pozwala na odkrycie najbardziej istotnych anomalii. Różne zestawy wymiarów mogą być istotne dla różnych anomalii, co dodatkowo komplikuje problem. Analiza podprzestrzenna jest trudniejsza niż grupowanie, ponieważ grupowanie opiera się na zbiorczym zachowaniu danych, a anomalie są z definicji rzadkie i mogą być ukryte w pełnej wymiarowości. Statystyczne agregaty na poszczególnych wymiarach często dostarczają słabych wskazówek do eksploracji podprzestrzeni w porównaniu z problemami opartymi na agregacji, jak grupowanie. W efekcie, bardziej złożone metody analizy są konieczne do identyfikacji podprzestrzeni zawierających istotne anomalie.

Przedstawione punkty napędzają bieżące badania i innowacje w dziedzinie wykrywania anomalii, koncentrując się na poprawie jakości i zarządzania dużą objętością danych, przetwarzaniu w czasie rzeczywistym, integracji uczenia maszynowego i sztucznej inteligencji, rozwoju bardziej przejrzystych i wyjaśnialnych modeli, oraz adaptacji systemów do nowych typów anomalii. Ponadto, zwiększenie współczynnika wykrywalności anomalii oraz efektywne radzenie sobie z danymi wielowymiarowymi są istotnymi celami. Ciągły postęp w tych obszarach jest niezbędny, aby skutecznie identyfikować i reagować na nowe zagrożenia w coraz bardziej złożonym i połączonym cyfrowym krajobrazie.

2.7 Podsumowanie

Rozdział ten dotyczył problematyki identyfikacji anomalii w danych złożonych. Przedstawiono definicję anomalii oraz problemy związane z jej precyzyjnym określeniem, proponując własną autorską definicję. Omówiono, jak szum może wpływać na proces identyfikacji anomalii, a także zaprezentowano charakterystykę złożonych zbiorów danych. W dalszej części rozdziału zaprezentowano podstawowe koncepcje matematyczne i obliczeniowe,

które mają znaczenie w analizie anomalii, oraz omówiono miary odległości i podobieństwa. Podkreślono znaczenie wiedzy dziedzinowej w procesie analizy anomalii, charakteryzując dziedziny, w których wykrywanie anomalii jest przydatne. Pokazano, jak wykrywanie anomalii jest powiązane z nauką o danych, procesem odkrywania wiedzy w bazach danych KDD oraz eksploracją danych. Przeprowadzono przegląd różnych metod wykrywania anomalii, w tym metod bazujących na mierze odległości, klasyfikacji, grupowaniu, rozkładzie danych, metod spektralnych oraz identyfikacji anomalii kontekstowych i zbiorczych. Wskazano także na zastosowanie uczenia maszynowego i głębokiego uczenia w analizie anomalii. Na zakończenie rozdział ten poruszył aktualne wyzwania i trendy rozwoju w analizie anomalii.

Podsumowując, analiza anomalii w złożonych zbiorach danych jest dziedziną o dużym znaczeniu praktycznym i teoretycznym, a zrozumienie różnorodnych metod oraz ich zastosowań może znacząco przyczynić się do rozwoju efektywnych technik identyfikacji anomalii w różnych kontekstach. W przyszłości warto zwrócić uwagę na dalszy rozwój algorytmów oraz integrację wiedzy dziedzinowej z nowoczesnymi metodami analizy danych.

Rozdział 3

Anomalie w zbiorach kategorycznych i wielowymiarowych

Dane z wielu rzeczywistych aplikacji mają często charakter kategoryczny i charakteryzują się wysoką wymiarowością, obejmującą setki, a nawet tysiące cech. W miarę wzrostu liczby wymiarów, skuteczność wielu tradycyjnych metod wykrywania anomalii drastycznie maleje. Ten spadek efektywności wynika z problemu znanego jako klątwa wymiarowości, omówionego w podrozdziale 2.2. Atrybuty kategoryczne, także omówione w tym rozdziale w kontekście podziału danych złożonych, stanowią istotny rodzaj danych, charakteryzujący się brakiem naturalnego porządku. Typ atrybutów używanych do opisu danych jest zasadniczym aspektem w kontekście wykrywania obserwacji odstających. W niniejszej rozprawie skupiono się szczególnie na badaniach dotyczących danych charakteryzowanych przez atrybuty jakościowe. Istotność tego podejścia podkreśla fakt, że wiele realnych zastosowań opiera się na danych opisanych za pomocą atrybutów kategorycznych, co z kolei podkreśla potrzebę opracowania skutecznych algorytmów do wykrywania anomalii w tego rodzaju danych. W przeciwieństwie do danych numerycznych, trudno jest często określić podobieństwo między różnymi wartościami tego samego atrybutu kategorycznego. Te niuanse, charakterystyczne dla danych kategorycznych i wielowymiarowych, stwarzają liczne metodologiczne i techniczne trudności dla algorytmów wykrywania anomalii. W wielu przypadkach zestawy danych zawierają zarówno atrybuty kategoryczne, jak i numeryczne. Analiza takich hybrydowych danych stwarza problemy dla aplikacji uczenia maszynowego, głównie ze względu na trudności związane z jednolitym i spójnym przetwarzaniem różnych typów atrybutów. Ten rozdział koncentruje się na analizie danych kategorycznych i wielowymiarowych, omawiając teoretyczne aspekty wykrywania anomalii oraz różnorodne metody i techniki stosowane do ich przetwarzania i interpretacji.

Na początku omówiono atrybuty kategoryczne, które są specyficznym typem danych jakościowych. Te atrybuty klasyfikują informacje na podstawie cech lub właściwości

badanych obiektów, a nie wartości liczbowych. Do takich kategorii należą płeć, status zawodowy czy poziom wykształcenia. Atrybuty kategoryczne są powszechne w danych rzeczywistych. Na przykład, dane finansowe mogą zawierać atrybuty ilościowe, takie jak saldo konta, ale większość innych atrybutów, takich jak typ konta, kategoria transakcji czy rodzaj waluty, jest kategoryczna. Dane zbierane z systemów lojalnościowych często obejmują informacje o preferencjach klientów, które również są kategoryczne. Podobnie, wiele danych w systemach rekomendacyjnych, takich jak gatunki filmów, typy produktów czy preferencje użytkowników, ma charakter kategoryczny. Dane kategoryczne mogą być mierzone za pomocą skal nominalnych, gdzie kolejność kategorii nie ma znaczenia, lub skal porządkowych, gdzie kategorie są uporządkowane w logicznej sekwencji, ale nie określają precyzyjnych odstępów między nimi. Dane kategoryczne często wymagają specjalnych metod kodowania, takich jak kodowanie zero-jedynkowe (*ang. binary encoding*) lub kodowanie one-hot (*ang. one-hot encoding*), aby mogły być efektywnie przetwarzane w statystycznych modelach predykcyjnych oraz algorytmach uczenia maszynowego. Te metody kodowania umożliwiają reprezentację kategorycznych danych w sposób, który jest zrozumiały dla modeli matematycznych i algorytmów.

Dzięki zdolności do grupowania obserwacji w łatwo interpretowalne kategorie, dane kategoryczne są nieocenione w identyfikacji wzorców i tendencji oraz w analizie zależności w różnych dziedzinach, takich jak badania społeczne, marketing, medycyna i wiele innych. Te właściwości sprawiają, że dane kategoryczne odgrywają niezastąpioną rolę w analizie danych i odkrywaniu istotnych informacji. Oto główne wyzwania, które stanowią przeszkody metodologiczne i techniczne w efektywnej analizie i przetwarzaniu danych kategorycznych:

- **nieznaczące miary statystyczne** - tradycyjne miary statystyczne, takie jak średnia i odchylenie standardowe, powszechnie stosowane w analizie danych numerycznych, są nieadekwatne i nieprzydatne w kontekście danych kategorycznych. Dane kategoryczne reprezentują jakościowe atrybuty, które nie mają naturalnego porządku ani skali liczbowej. W związku z tym próby zastosowania ilościowych miar statystycznych do danych kategorycznych prowadzą do wyników pozbawionych znaczenia i wartości interpretacyjnej. Na przykład obliczanie średniej wartości dla zbioru danych kategorycznych, takich jak kolory czy kategorie produktów, jest bezsensowne, ponieważ wartości te nie mogą być logicznie sumowane ani dzielone. Podobnie, odchylenie standardowe, które mierzy rozproszenie wokół średniej, traci sens, gdy dane nie są liczbowe. W rezultacie analizy danych kategorycznych wymagają zastosowania alternatywnych metod, bardziej dostosowanych do jakościowej natury tych danych, takich jak miary częstości występowania czy analizy rozkładów kategorycznych,
- **ograniczenia modeli liniowych** - modele liniowe, takie jak regresja liniowa, są pierwotnie opracowane do pracy z danymi numerycznymi, co sprawia, że ich zastosowanie do danych kategorycznych napotyka na istotne trudności. Nawet jeśli

dane kategoryczne zostaną przekształcone w reprezentacje binarne, co polega na zamianie każdej kategorii na oddzielny atrybut binarny, problemy te nadal się utrzymują. Reprezentacje binarne prowadzą do powstania macierzy danych o bardzo dużych rozmiarach, które są rzadkie (zawierają wiele zer), co znacznie komplikuje proces obliczeniowy. Ponadto, transformacja taka może nie zachowywać informacji o pierwotnej strukturze danych kategorycznych, przez co modele liniowe nie są w stanie w pełni wykorzystać potencjału tych danych. W konsekwencji, wyniki analiz przeprowadzonych za pomocą modeli liniowych mogą być mylące lub niekompletne, co wymaga poszukiwania bardziej zaawansowanych metod analizy danych kategorycznych, które lepiej oddają ich specyfikę i złożoność,

- **wyzwania obliczeniowe** - przekształcenie danych kategorycznych w rzadkie reprezentacje binarne to proces, który często skutkuje powstaniem niezwykle dużych macierzy danych, w których dominują wpisy zerowe. Ta ogromna skala oraz rozproszenie wartości w macierzy prowadzą do znaczących trudności obliczeniowych. Złożoność operacji na tak rozległych i rzadkich zbiorach danych dramatycznie zwiększa wymagania dotyczące pamięci i mocy obliczeniowej, co czyni standardowe techniki analizy mniej wydajnymi i bardziej kosztownymi. Ponadto, struktura tych macierzy może utrudniać optymalizację algorytmów, co przekłada się na wydłużenie czasu przetwarzania oraz zwiększenie potencjalnych błędów w analizie. W efekcie, niezbędne staje się poszukiwanie bardziej zaawansowanych i efektywnych metod, które będą w stanie poradzić sobie z takimi wyzwaniami, zachowując jednocześnie integralność i dokładność analizowanych danych kategorycznych,
- **niedostosowanie funkcji bliskości** - tradycyjne miary odległości, takie jak odległość euklidesowa, okazują się nieadekwatne w kontekście danych kategorycznych. Dane te, z natury jakościowe, nie poddają się bowiem łatwo klasycznym technikom mierzenia odległości, które zostały opracowane głównie z myślą o danych numerycznych. Aby temu sprostać, niezbędne jest opracowywanie nowatorskich funkcji bliskości, które będą w stanie precyzyjnie uchwycić specyfikę i złożoność danych kategorycznych. Takie funkcje muszą uwzględniać unikalne cechy tych danych, aby umożliwić prawidłowe modelowanie i analizę. Opracowanie takich zaawansowanych miar bliskości umożliwia skuteczne zastosowanie algorytmów opartych na bliskości w analizie danych kategorycznych, otwierając nowe możliwości w dziedzinie uczenia maszynowego i analizy danych,
- **trudności w modelach opartych na gęstości** - modele oparte na gęstości, takie jak LOCI (*ang. local correlation integral*, LOCI) [162], LOF [98] czy metoda Parzena-Rosenblatta [163, 164], napotykają na trudności przy próbach adaptacji do danych kategorycznych ze względu na swoją zależność od pojęcia odległości. Modele te, projektowane pierwotnie z myślą o danych numerycznych, wykorzystują miary odległości do estymacji gęstości prawdopodobieństwa. Jednakże, gdy dane są

kategoryczne, brak naturalnej skali liczbowej i porządku powoduje, że klasyczne miary odległości stają się nieadekwatne. W związku z tym, konieczne jest opracowanie nowych metodologii, które pozwolą na prawidłową estymację gęstości w kontekście danych kategorycznych. Tylko dzięki takim innowacyjnym podejściom możliwe będzie zachowanie integralności analizy i zapewnienie, że modele oparte na gęstości będą mogły być efektywnie stosowane również do danych jakościowych. Przezwyciężenie tych trudności otworzy nowe horyzonty w dziedzinie analizy danych, umożliwiając bardziej precyzyjne i wszechstronne modelowanie różnorodnych typów danych,

- **problemy ze skalowalnością** - przekształcanie danych kategorycznych do postaci binarnej napotyka na istotne ograniczenia skalowalności, zwłaszcza w kontekście dużej liczby możliwych wartości tych atrybutów. Każdy z atrybutów przekształcony w odrębny atrybut binarny prowadzi do powstania ogromnych macierzy danych, które stają się trudne do zarządzania i przetwarzania. Taka transformacja powoduje eksplozję wymiarowości, gdzie liczba binarnych atrybutów może przekroczyć możliwości obliczeniowe standardowych systemów. W rezultacie analiza tych rozszerzonych zbiorów danych staje się bardziej złożona i mniej efektywna, zarówno pod względem czasu, jak i zasobów obliczeniowych. Dlatego konieczne jest opracowanie zaawansowanych metod reprezentacji danych, które zachowają integralność informacji przy jednoczesnym minimalizowaniu wzrostu wymiarowości. Innowacyjne podejścia, takie jak techniki kompresji informacji, algorytmy uczenia maszynowego przystosowane do wysokiej wymiarowości czy metody hybrydowe, mogą stanowić klucz do przezwyciężenia tych trudności, otwierając nowe możliwości w efektywnej analizie danych kategorycznych,
- **współzależność atrybutów binarnych** - przekształcanie atrybutów kategorycznych na reprezentacje binarne często prowadzi do powstania atrybutów, które są silnie współzależne. Ta współzależność wynika z natury danych kategorycznych, gdzie poszczególne wartości są wzajemnie wykluczające się. W konsekwencji, taka reprezentacja danych staje się nieefektywna, ponieważ generuje redundantne informacje i zwiększa złożoność analizy. Atrybuty binarne, będące wynikiem tej transformacji, zawierają dużą ilość zer, co nie tylko zwiększa wymagania pamięciowe, ale również komplikują algorytmy analityczne, które muszą przetwarzać te dane. Ponadto, algorytmy te często nie są w stanie w pełni wykorzystać wzorców ukrytych w danych, co prowadzi do mniej dokładnych wyników analizy. Aby przezwyciężyć te ograniczenia, niezbędne jest opracowanie bardziej zaawansowanych metod transformacji i analizy danych kategorycznych, które uwzględniają współzależność między atrybutami i minimalizują redundancję informacji. Przykłady takich podejść mogą obejmować techniki redukcji wymiarowości, takie jak analiza składowych głównych dostosowana do danych binarnych, czy też zastosowanie sieci neuronowych, które są zdolne do modelowania złożonych zależności między atrybutami.

W dalszej kolejności omówiono dane wielowymiarowe, które składają się z wielu atrybutów definiujących przestrzeń cech. Przykłady obejmują multimedia, dane genetyczne, dane z badań naukowych i medyczne oraz aplikacje biznesowe. Obiekty w takich przestrzeniach są reprezentowane przez wektory cech, takie jak histogramy kolorów, wektory tekstury czy deskryptory kształtów. Wysoka liczba wymiarów prowadzi do problemu rzadkości danych, gdzie obiekty są od siebie oddalone, co utrudnia stosowanie tradycyjnych miar odległości i wykrywanie anomalii. Wysoka wymiarowość sprawia, że pojęcie bliskości traci intuicyjność i każda obserwacja może wydawać się anomalią. Tradycyjne metody wykrywania anomalii stają się mniej skuteczne, dlatego konieczne jest stosowanie nowych technik, które efektywnie analizują i interpretują dane, uwzględniając ich wielowymiarową naturę. Problemy z efektywnością dotyczą również tradycyjnych metod indeksowania. Struktury indeksowania, działające dobrze w niskich i średnich wymiarach, przestają być efektywne w wysokich wymiarach. Aby radzić sobie z wysoką wymiarowością, powszechnie stosuje się redukcję wymiarowości, np. za pomocą analizy składowych głównych PCA, która kondensuje większość informacji w kilku wymiarach.

Większość algorytmów detekcji wartości odstających konstruuje model, który odzwierciedla typowe wzorce w danych, a następnie wykorzystuje go do oceny, czy konkretny obiekt jest odstający. Ocena ta polega na porównaniu obiektu z modelem, co pozwala ustalić, jak dobrze pasuje on do ustalonych wzorców. Podstawowe modele w kontekście danych kategorycznych zostały szczegółowo opisane w kolejnych podrozdziałach. Rozdział jest podzielony na kilka części. Najpierw skupiono się na identyfikacji anomalii w danych kategorycznych. Przeanalizowano modele generatywne, które dzięki swojej zdolności do lepszego zrozumienia struktury danych, pozwalają na precyzyjniejszą identyfikację anomalii. Szczególną uwagę poświęcono procesowi generowania danych w tych modelach oraz algorytmowi EM (oczekiwanie-maksymalizacja), wykorzystywanemu do szacowania parametrów modeli generatywnych. Następnie omówiono modelowanie danych mieszanych, łączących cechy kategoryczne i numeryczne, oraz techniki identyfikacji anomalii przy użyciu modeli generatywnych. W kolejnej części przedstawiono metody oparte na modelach liniowych, które wykorzystują podejście liniowe do analizy danych kategorycznych, co umożliwia identyfikację wzorców i zależności. Omówiono również modele oparte na bliskości danych, które identyfikują podobieństwa między obiektami na podstawie ich atrybutów, co jest ważne w zadaniach takich jak grupowanie, klasyfikacja oraz identyfikacja anomalii. Kolejny podrozdział poświęcono identyfikacji anomalii w danych wielowymiarowych, które wymagają szczególnego podejścia ze względu na złożoność ich struktury. Omówiono wiele technik i algorytmów pomocnych w radzeniu sobie z wysoką liczbą wymiarów, takich jak redukcja wymiarowości za pomocą PCA oraz metody oparte na lokalnej analizie korelacji danych.

Dane z wielu rzeczywistych aplikacji są często kategoryczne i mają wysoką wymiarowość. Dlatego konieczne jest opracowanie odpowiednich technik przetwarzania danych opisanych za pomocą atrybutów kategorycznych. Skuteczność algorytmu wykrywania

wartości odstających w dużej mierze zależy od interpretacji tych wartości w kontekście atrybutów kategorycznych. Rozdział dostarcza kompleksowego przeglądu metod i narzędzi stosowanych w analizie danych kategorycznych i wielowymiarowych, umożliwiając skuteczne przetwarzanie, analizę i interpretację tych danych w różnych dziedzinach.

3.1 Identyfikacja anomalii w danych kategorycznych

Najczęściej dane kategoryczne są analizowane poprzez badanie częstotliwości występowania różnych wartości atrybutów. Oznacza to, że sprawdzamy, jak często w zbiorze danych pojawiają się poszczególne wartości danego atrybutu. Na przykład, dla atrybutu *kolor samochodu* możemy określić, ile razy pojawiają się samochody w kolorze *czerwonym*, *niebieskim*, *zielonym* itd. Celem takiej analizy jest identyfikacja wartości typowych oraz tych, które są rzadkie i odstają od normy. Możemy to robić zarówno dla pojedynczych atrybutów (np. *kolor samochodu*), jak i dla kombinacji dwóch lub więcej atrybutów (np. *kolor samochodu* i *marka*). Aby to osiągnąć, definiujemy miarę odchylenia, która ocenia, jak bardzo dana wartość różni się od innych na podstawie jej częstotliwości występowania. Na przykład, jeśli *czerwone* samochody pojawiają się bardzo rzadko w porównaniu do innych kolorów, możemy uznać je za odstające zgodnie z tą miarą odchylenia. Przykładowe wartości atrybutu kategorycznego (jakościowego) są reprezentowane przez różne kategorie, jak zilustrowano w tabeli 3.1.

Tabela 3.1: Przykłady atrybutów kategorycznych z ich wartościami. Źródło: opracowanie własne.

Nazwa atrybutu	**Kategorie (przykładowe wartości)**
Rodzaj owocu	jabłko, banan, pomarańcza, ...
Rodzaj napoju	herbata, kawa, sok, ...
Typ samochodu	sedan, SUV, kabriolet, ...
Poziom wykształcenia	podstawowe, średnie, wyższe, ...
Marka telefonu	Samsung, Apple, Huawei, ...

Charakterystykę zbioru danych kategorycznych można opisać za pomocą kilku parametrów. Po pierwsze, podstawowym elementem jest rozmiar zbioru danych, czyli liczba wszystkich obiektów, które zbiór zawiera. Po drugie, ważna jest liczba atrybutów, co odnosi się do liczby różnych kolumn w zbiorze danych. Kolejnym ważnym parametrem jest liczba wartości przypisanych do każdego atrybutu, czyli zakres możliwych wartości, jakie atrybut może przyjmować. Ostatnim ważnym szczegółem jest częstość występowania różnych wartości każdego atrybutu kategorycznego, czyli liczba wystąpień każdej wartości w zbiorze danych. Te parametry mają znaczenie dla zrozumienia struktury i rozkładu w zbiorze danych kategorycznych.

Wykrywanie anomalii w danych kategorycznych jest trudniejsze niż w danych numerycznych. Istotnym problemem jest zdefiniowanie miar podobieństwa. W przypadku danych numerycznych łatwo jest powiedzieć, że liczbie 5 jest bliżej do 6 niż do 10, ponieważ liczby mają naturalny porządek. Jednak w danych kategorycznych, takich jak kolory (np. *czerwony*, *zielony*, *niebieski*) czy typy samochodów (np. *sedan*, *SUV*, *kabriolet*), nie ma takiego naturalnego porządku. Nie możemy powiedzieć, że *czerwony* jest „bardziej" lub „mniej" od *zielonego*, ponieważ kolory nie mają uporządkowanej kolejności. To sprawia, że trudniej jest znaleźć podobieństwo między różnymi wartościami. W rezultacie wiele zadań w analizie danych staje się trudniejsze. Na przykład, znalezienie najbliższego sąsiada dla danego obiektu kategorycznego, klasyfikacja dokumentów tekstowych według kategorii, czy analiza preferencji klientów na podstawie ich wyborów produktów (np. określenie, jakie produkty są podobne do tych, które klient już kupił). To dlatego, że nie możemy używać prostych miar podobieństwa, jak w danych numerycznych, i musimy znaleźć inne sposoby porównywania wartości kategorycznych.

Jedną z takich metod jest grupowanie, które pozwala na lepsze zrozumienie struktury danych i jest pomocne w identyfikacji anomalii. W tym kontekście analiza skupień odgrywa znaczącą rolę w wykrywaniu anomalii w danych kategorycznych. Oto cztery znane algorytmy grupowania danych kategorycznych, które można wykorzystać do identyfikacji anomalii:

- **Algorytm ROCK**

  Algorytm ROCK (*ang. robust clustering using links*) [165] wykorzystuje pojęcie powiązań między obiektami danych zamiast tradycyjnych miar podobieństwa. Połączenia są definiowane jako liczba wspólnych sąsiadów między parami obiektów. Celem algorytmu jest maksymalizacja liczby połączeń wewnątrz grup oraz minimalizacja liczby połączeń między grupami. Dzięki temu obiekty o wysokim stopniu połączeń są grupowane razem, co prowadzi do bardziej znaczących grup. W algorytmie ROCK anomalie mogą być skutecznie wykrywane na dwa sposoby [165]. Po pierwsze, odpowiedni dobór progu podobieństwa τ pozwala na eliminację próbek, które mają bardzo niewielu sąsiadów lub nie mają ich wcale. Po drugie, podczas procesu grupowania, jeśli występują małe grupy próbek, które są luźno powiązane z resztą danych, mogą one zostać zidentyfikowane jako anomalie. Algorytm ROCK został szczegółowiej opisany w podrozdziale 2.5.3, gdzie omówiono metody grupowania w kontekście wykrywania anomalii,

- **Algorytm k-modes**

  Algorytm k-modes [166] to rozszerzenie paradygmatu k-średnich, dostosowane do danych kategorycznych poprzez wprowadzenie miary niepodobieństwa specyficznej dla takich danych. Algorytm ten zastępuje średnie grup modami, co jest bardziej odpowiednie dla atrybutów kategorycznych. Zamiast tradycyjnej miary odległości, k-modes wykorzystuje prostą miarę dopasowania, która określa niezgodność

między obiektami na podstawie liczby różnych wartości atrybutów. Mody grup są aktualizowane na podstawie częstości występowania poszczególnych wartości atrybutów, co pozwala na minimalizację funkcji kosztu grupowania. Dzięki tym rozszerzeniom, algorytm k-modes umożliwia efektywne grupowanie danych kategorycznych w sposób analogiczny do k-średnich. Jest on skalowalny i wydajny, co sprawia, że nadaje się do grupowania dużych zbiorów danych kategorycznych. Metody przedstawione w k-modes mogą być używane do identyfikacji anomalii poprzez analizę grup i wykrywanie obiektów, które nie pasują do żadnej z nich,

- **Algorytm Squeezer**

  Algorytm Squeezer [167] wykonuje grupowanie danych kategorycznych poprzez sekwencyjne przetwarzanie obiektów ze zbioru danych. Algorytm przetwarza dane, grupując je. Pierwszy obiekt jest przypisywany do początkowej grupy, a kolejne obiekty są przypisywane do istniejących grup lub tworzą nowe grupy w zależności od ich podobieństwa do już istniejących grup. Miara podobieństwa w algorytmie Squeezer opiera się na liczbie wspólnych wartości atrybutów między obiektem a grupą. Im więcej wspólnych wartości, tym większe podobieństwo. Algorytm nie wymaga podania liczby grup na wejściu, co czyni go elastycznym i wydajnym dla strumieni danych. Dobór odpowiedniego progu podobieństwa τ pozwala na eliminację próbek, które mają bardzo niewielu sąsiadów lub nie mają ich wcale. Próbki te są traktowane jako anomalie. Ponadto, małe grupy próbek, które są luźno powiązane z resztą danych, mogą zostać zidentyfikowane jako anomalie,

- **Algorytm k-ANMI**

  Algorytm k-ANMI (*ang. average normalized mutual information*, ANMI) [168] to wariant algorytmu k-średnich przeznaczony dla danych kategorycznych. Optymalizuje on funkcję celu opartą na wzajemnej informacji między grupami. Jakość grupowania (*ang. cluster validity*) jest mierzona za pomocą średniej znormalizowanej wzajemnej informacji ANMI, która kwantyfikuje ilość informacji wspólnej między rozkładami grup. ANMI jest miarą oceny jakości grupowania danych i jest oparta na kryterium wzajemnej informacji [169, 170], mierzącym stopień zależności między dwiema zmiennymi losowymi. Wzajemna informacja MI (*ang. mutual information*, MI) między dwoma zmiennymi losowymi X i Y mierzy, jak bardzo znajomość jednej zmiennej redukuje niepewność drugiej zmiennej. Informacja ta, $I(X;Y)$, mierzy różnicę między wspólnym rozkładem prawdopodobieństwa X i Y a iloczynem ich brzegowych rozkładów. W przypadku danych kategorycznych wzajemna informacja jest obliczana podobnie jak dla danych liczbowych, ale z wykorzystaniem częstości występowania kategorii zamiast prawdopodobieństwa. Metody oparte na MI są używane do mierzenia istotności i redundancji między cechami w celu ich selekcji. Ta miara, znana również jako *zysk informacyjny*, jest dokładnie opisana w literaturze, na przykład w książce [86]. Miara AMI (*ang. average mutual information*, AMI)

uśrednia wzajemną informację między wszystkimi parami grup i rzeczywistymi etykietami, uwzględniając rozkłady grup i etykiet. AMI jest normalizowana w celu uzyskania ANMI, która jest niezależna od liczby grup i etykiet w grupowaniu. Dzięki temu można porównywać jakość grupowania między różnymi zestawami danych i różnymi algorytmami. ANMI jest wykorzystywana do oceny, jak dobrze obiekty danych są przypisane do grup. Algorytm k-ANMI może pomóc w identyfikacji anomalii poprzez analizę obiektów, które nie pasują do żadnej grupy lub są przypisane do grup z niską wartością ANMI. Niska wartość ANMI wskazuje na słabe dopasowanie obiektów do grup, co może sugerować ich odchylenie od reszty danych.

Przykłady dostępnych algorytmów do przetwarzania danych kategorycznych

Anomalie są często definiowane na podstawie częstości występowania wartości atrybutów w danych kategorycznych. Aby uznać dane za anomalie, muszą one wykazywać charakterystyczne cechy poprzez nietypowe lub interesujące wartości jednego lub więcej atrybutów. Te nietypowe wartości zazwyczaj występują rzadziej niż wartości standardowe, więc niska częstość ich występowania wskazuje na obecność anomalii. W danych kategorycznych można zidentyfikować dwa typy anomalii:

- **Anomalia typu 1** - w tym przypadku wartości atrybutów $x_{i,j}$, opisujących obiekt X_i, występują stosunkowo rzadko,
- **Anomalia typu 2** - drugi typ anomalii występuje, gdy kombinacja wartości zmiennych kategorycznych $x_{i,j}$ opisujących obiekt X_i jest stosunkowo rzadka, mimo że każda z tych wartości $x_{i,j}$ jest często spotykana oddzielnie.

Metody oparte na częstościach wartości atrybutów są skuteczne w wykrywaniu anomalii typu 1, ale zawodzą w przypadku anomalii typu 2, ponieważ te ostatnie mogą być częścią dużego skupienia ze względu na ich bliskość do takich grup. Natomiast metody oparte na grupowaniu są zdolne do wykrywania anomalii typu 2, ale mogą mieć trudności z identyfikacją anomalii typu 1, ponieważ te anomalie, mimo swoich nieregularnych wartości, mogą zostać włączone do dużych grup.

Algorytmy ROAD (*ang. ranking-based outlier analysis and detection*, ROAD) [171], AVF (*ang. attribute value frequency*, AVF) [172] oraz algorytm zachłanny (*ang. greedy algorithm*) [173] mają pewne wspólne cechy, ale różnią się w swoich podstawowych zasadach i zastosowaniach. Wszystkie można wykorzystać do identyfikacji anomalii w danych kategorycznych. Wykorzystują one pewne formy heurystyki do podejmowania decyzji w procesie obliczeniowym i są zaprojektowane z myślą o efektywności obliczeniowej, choć na różne sposoby. Podczas gdy ROAD i AVF koncentrują się na wykrywaniu wzorców i anomalii w danych, algorytmy zachłanne znajdują szerokie zastosowanie w różnych problemach optymalizacyjnych. Poniżej przedstawiono krótkie omówienie każdego z nich:

Algorytm ROAD jest przeznaczony do wykrywania anomalii w danych kategorycznych. Działa w dwóch głównych fazach: fazie obliczeniowej i fazie rankingowej. W skrócie, algorytm ten oblicza gęstość obiektów oraz wykonuje grupowanie danych, aby określić anomalie w stosunku do dużych grup. Podczas analizy algorytm wykorzystuje dwie miary: częstość występowania wartości atrybutów oraz odległość od najbliższej dużej grupy, które są używane do identyfikacji anomalii. Gęstość obiektu określa, jak często jego atrybuty występują w zestawie danych. Na przykład, w zestawie danych dotyczących samochodów, jeśli jednym z atrybutów jest kolor, gęstość zielonego samochodu będzie zależała od tego, jak często zielony kolor pojawia się w całym zestawie danych. Jeśli zielony samochód jest rzadkością, jego gęstość będzie niska, co może sugerować, że jest anomalią. Obiekty o niskiej gęstości są sortowane rosnąco, przy czym te o najniższej gęstości są najbardziej podejrzane jako anomalie. Odległość mierzy, jak daleko dany obiekt znajduje się od najbliższej dużej grupy podobnych obiektów. Jeśli większość samochodów w zbiorze danych jest czerwonych lub niebieskich, a mamy zielony samochód, to mierzymy odległość tego zielonego samochodu od grupy czerwonych i niebieskich samochodów. Obiekty oddalone od najbliższej dużej grupy prawdopodobnie są anomaliami, ponieważ nie pasują do reszty danych. Im większa odległość, tym większe prawdopodobieństwo, że obiekt jest anomalią. Odległość mierzymy za pomocą funkcji odległości, która określa, jak bardzo obiekt różni się od innych obiektów. W danych kategorycznych, gdzie atrybuty nie są liczbowe, odległość często mierzymy na podstawie liczby różnic w wartościach atrybutów między dwoma obiektami. W ten sposób obiekty są klasyfikowane według gęstości od najniższej do najwyższej oraz według odległości do grupy od największej do najmniejszej. Łącząc oba rankingi, uzyskujemy zestaw obiektów najbardziej podejrzanych jako anomalie, niezależnie od tego, czy są typu 1, czy typu 2.

Algorytmy zachłanne są intensywnie badane i rozwijane przez licznych naukowców oraz matematyków, którzy dostrzegli ich efektywność w rozwiązywaniu określonych problemów optymalizacyjnych. Wykrywanie wartości odstających w danych kategorycznych można sformułować jako problem optymalizacyjny. Celem jest znalezienie podzbioru s obiektów, których usunięcie maksymalnie zredukuje entropię zestawu danych. Algorytm zachłanny usuwa iteracyjnie obiekty mające największy wkład w entropię, aż zostanie zidentyfikowanych k wartości odstających. W ten sposób minimalizuje entropię pozostałego zestawu danych, co umożliwia skuteczne wykrywanie wartości odstających w danych kategorycznych. Entropia mierzy niepewność związaną z atrybutem. Entropia atrybutu A_j jest opisana wzorem (3.1):

$$E(A_j) = -\sum_k p(A_{j,k}) \log(p(A_{j,k})), \tag{3.1}$$

gdzie $p(A_{j,k})$ oznacza prawdopodobieństwo, że atrybut A_j przyjmie wartość $A_{j,k}$, gdzie $A_{j,k} \in Q_j$, czyli $A_{j,k}$ należy do zbioru wartości (dziedziny) Q_j atrybutu A_j.

Zestaw danych składa się z n obiektów $\{X_1, X_2, \ldots, X_n\}$, z których każdy jest opisany przez m atrybutów kategorycznych $\{A_1, A_2, \ldots, A_m\}$. Entropia całego zestawu danych $E(D)$ jest sumą entropii poszczególnych atrybutów:

$$E(D) = E(A_1) + E(A_2) + \ldots + E(A_m) \tag{3.2}$$

Sposób postępowania:

- w każdej iteracji algorytmu obliczamy wkład każdego obiektu X_i w entropię zestawu danych D,
- następnie wybieramy obiekt, który ma największy wkład w entropię, oznaczamy go jako wartość odstającą i usuwamy go z zestawu danych,
- proces ten powtarzamy, usuwając jeden obiekt na iterację, aż do momentu, gdy zidentyfikujemy k wartości odstających.

Złożoność czasowa tego algorytmu to $O(nkm)$, gdzie n to liczba obiektów, k to liczba wartości odstających, a m to liczba atrybutów.

Algorytm AVF, czyli częstotliwości wartości atrybutów, to efektywna i skalowalna metoda służąca do identyfikacji wartości odstających w zbiorach danych kategorycznych. Jego działanie opiera się na założeniu, że obiekty odstające występują niezbyt często w zbiorze danych. Idealnym przykładem obiektu odstającego jest taki, w którym każda z wartości atrybutów pojawia się wyjątkowo rzadko. Aby określić rzadkość wartości atrybutu A_j w zbiorze danych D, metoda AVF zlicza częstotliwości wystąpień poszczególnych wartości atrybutów. Mając zestaw danych D składający się z n obiektów X_i i m atrybutów kategorycznych A_j, metoda oblicza wynik oparty na częstotliwości dla obiektu X_i (3.3):

$$\mathrm{AVF}(X_i) = \frac{1}{m} \sum_{j=1}^{m} f(x_{i,j}), \tag{3.3}$$

gdzie $f(x_{i,j})$ to liczba wystąpień wartości $x_{i,j}$ dla atrybutu A_j w całym zbiorze danych D.

Zgodnie z tą metodą, obiekty X_i o niskich wynikach AVF są uważane za wartości odstające. Po obliczeniu wyników dla wszystkich obiektów X_i w D, można wybrać k obiektów z najniższymi wynikami AVF jako wartości odstające. Złożoność obliczeniowa tego algorytmu wynosi $O(nm)$, co oznacza, że jego czas wykonania rośnie liniowo wraz z liczbą obiektów X_i i atrybutów A_j. Czyni go to wyjątkowo efektywnym, ponieważ działa przy pojedynczym skanowaniu zbioru danych D, bez potrzeby tworzenia i przeszukiwania różnych kombinacji wartości atrybutów.

Algorytm FindFPOF (*ang. frequent pattern outlier factor*, FindFPOF) [174] różni się od algorytmów ROAD, AVF i algorytmów zachłannych pod względem metodologii wykrywania anomalii. FindFPOF to metoda wykrywania wartości odstających w danych kategorycznych, oparta na analizie częstych wzorców. Algorytm rozpoczyna się od wydobycia wszystkich częstych wzorców (*ang. frequent itemsets*) ze zbioru danych, korzystając

z określonego przez użytkownika minimalnego wsparcia (*ang. minisupport*). Wzorce te reprezentują „typowe" schematy w danych i są identyfikowane za pomocą algorytmu reguł asocjacyjnych. Następnie dla każdej transakcji w bazie danych algorytm oblicza wartość FPOF, która mierzy, jak typowa jest transakcja na tle całego zbioru danych. Wartość FPOF jest sumą wsparć wszystkich częstych wzorców zawartych w danej transakcji. Transakcje o niższych wartościach FPOF są uznawane za bardziej prawdopodobne wartości odstające, ponieważ zawierają mniej typowych wzorców charakterystycznych dla reszty danych. Transakcje są sortowane według wartości FPOF w kolejności rosnącej, a te z najniższymi wartościami są klasyfikowane jako wartości odstające.

3.2 Modele generatywne dla danych kategorycznych

Model generatywny zakłada, że obserwacje pochodzą z mieszaniny różnych komponentów, każdy z własnym typem rozkładu prawdopodobieństwa. W kontekście danych kategorycznych, takie rozkłady mogą być np. rozkładem wielomianowym lub Bernoulliego.

3.2.1 Proces generowania danych w modelach generatywnych

Wybór komponentu

Każdy obiekt danych X_i jest generowany przez jeden z komponentów modelu mieszaniny, z określonym prawdopodobieństwem α_m dla każdego komponentu m. Prawdopodobieństwa te mogą być równomiernie rozłożone lub dostosowane na podstawie danych D. W modelach generatywnych komponenty reprezentują statystycznie odrębne podgrupy danych, każda z nich charakteryzująca się specyficznymi wzorcami lub cechami. W przypadku danych liczbowych, jak w modelach mieszaniny gaussowskiej, komponenty opisane są przez średnią μ i macierz kowariancji, które lokalizują i określają rozproszenie grupy. Dla danych kategorycznych, komponenty definiowane są przez prawdopodobieństwa wystąpienia poszczególnych kategorii Q_j, ilustrujące dominujące wzorce zachowań w danej grupie. Każdy komponent ma przypisaną wagę α_m, która wskazuje, jak duża część zbioru danych D jest przez niego generowana. Na przykład, waga 0,3 sugeruje, że około 30% danych pochodzi z tego komponentu.

Generacja obiektu danych

Dla wybranego komponentu m, obiekt danych X_i jest generowany zgodnie z określonym rozkładem. Jeśli rozkład Bernoulliego jest używany dla zmiennej A_j, przyjmującej wartość Q_j, stosuje się prawdopodobieństwo p_{ijm}. Rozkład Bernoulliego opisuje prawdopodobieństwo sukcesu lub niepowodzenia w pojedynczym eksperymencie, gdzie może wystąpić tylko jedno z dwóch możliwych zdarzeń. Sukces jest oznaczany jako 1, a niepowodzenie jako 0. Jeśli oznaczymy prawdopodobieństwo sukcesu jako p, to prawdopodobieństwo niepowodzenia wynosi $1 - p$.

Wzór prawdopodobieństwa dla rozkładu Bernoulliego:

$$P(X = x) = p^x \cdot (1 - p)^{1-x}, \tag{3.4}$$

gdzie:

- $P(X = x)$ to prawdopodobieństwo, że zmienna losowa X przyjmie wartość x,
- x może przyjąć tylko dwie wartości: 0 lub 1,
- p to prawdopodobieństwo sukcesu.

Jeśli $x = 1$ (sukces), to wzór redukuje się do $P(X = 1) = p$, a jeśli $x = 0$ (niepowodzenie), to wzór redukuje się do $P(X = 0) = 1 - p$.

Funkcja generatywna

Funkcja generatywna dla obiektu danych X_i, który jest opisany przez d atrybutów A_j, z których każdy przyjmuje wartość kategoryczną, opisana jest wzorem:

$$g_{m,\lambda}(X_i) = \sum_{j=1}^{d} p_{ijm}, \tag{3.5}$$

gdzie $x_{i,j}$ to wartość atrybutu A_j dla obiektu X_i, p_{ijm} oznacza prawdopodobieństwo przypisania tej wartości $x_{i,j}$ przez komponent m, a λ jest parametrem modelu, który wpływa na kształt lub wagę funkcji generatywnej.

Prawdopodobieństwo a posteriori

Po obliczeniu prawdopodobieństwa generacji danych przez wszystkie komponenty, obliczane jest prawdopodobieństwo a posteriori, że obiekt danych X_i jest generowany przez komponent m:

$$P(G_m|X_i, \lambda) = \frac{\alpha_m \cdot g_{m,\lambda}(X_i)}{\sum_{r=1}^{k} \alpha_r \cdot g_{r,\lambda}(X_i)} \tag{3.6}$$

3.2.2 Algorytm EM (oczekiwanie-maksymalizacja)

W kontekście procesu generowania danych w modelach generatywnych, opisanego w sekcji 3.2.1, algorytm EM jest szeroko stosowany do szacowania parametrów modeli mieszaninowych, szczególnie w przypadku danych kategorycznych. W porównaniu z algorytmem k-średnich, wspomnianym w podrozdziale 2.5.3, który jest szczególnym przypadkiem algorytmu EM dla danych liczbowych, w modelach generatywnych dla danych kategorycznych podejście to wymaga innego oszacowania rozkładów prawdopodobieństwa dla poszczególnych kategorii:

Krok E (Expectation)

Krok E (Oczekiwanie) polega na obliczeniu oczekiwanej wartości funkcji logarytmicznej wiarygodności parametrów modelu, bazując na aktualnych oszacowaniach tych parametrów i dostępnych danych. W kontekście modeli z danymi kategorycznymi, gdzie zaangażowane są zmienne ukryte (np. przynależność do komponentów mieszaniny), krok E wykorzystuje obecne parametry modelu do oszacowania, jak prawdopodobne jest, że dany obiekt danych pochodzi z każdego z komponentów. Oczekiwane prawdopodobieństwa te nazywane są *prawdopodobieństwami a posteriori* każdego obiektu danych względem poszczególnych komponentów. W tym kroku, dla każdego obiektu danych obliczane są prawdopodobieństwa jego przynależności do różnych komponentów modelu, co odgrywa ważną rolę w późniejszej maksymalizacji wiarygodności modelu.

Krok M (Maximization)

Krok M (Maksymalizacja) to proces aktualizacji parametrów modelu w celu maksymalizacji oczekiwanej log-wiarygodności (*ang. log-likelihood*), uzyskanej w kroku E. Na podstawie obliczonych prawdopodobieństw a posteriori, parametry każdego z komponentów (np. prawdopodobieństwa wystąpienia różnych kategorii w rozkładach Bernoulliego) są aktualizowane tak, aby jak najlepiej pasowały do obserwowanych danych. W modelach generatywnych dla danych kategorycznych, może to oznaczać aktualizację prawdopodobieństw przynależności poszczególnych obserwacji do konkretnych komponentów, co z kolei wpływa na charakterystyki tych komponentów takie jak dominujące wzorce zachowań czy rozkłady prawdopodobieństw poszczególnych kategorii.

3.2.3 Modelowanie danych mieszanych

Dane mieszane, zawierające zarówno atrybuty kategoryczne jak i liczbowe, prezentują specyficzne wyzwania analityczne, które wymagają zastosowania zaawansowanych metod modelowania w celu skutecznego przetwarzania i analizy. Modele generatywne, zwłaszcza te wykorzystujące algorytm EM, są wyjątkowo przydatne do jednolitego przetwarzania tego rodzaju danych dzięki ich zdolności do modelowania złożonych struktur danych.

Problem normalizacji w danych mieszanych

W tradycyjnych podejściach modelowania danych mieszanych, trudnością jest normalizacja różnych typów atrybutów, aby zapewnić równomierne traktowanie wszystkich cech przez algorytm. W danych mieszanych, atrybuty liczbowe mogą być skalowane różnie niż kategoryczne, co może prowadzić do nierównowagi w ich wpływie na wyniki modelowania. Modele probabilistyczne, takie jak te wykorzystujące algorytm EM, naturalnie radzą sobie z tym problemem, stosując odpowiednie funkcje prawdopodobieństwa dla różnych typów danych. Dzięki temu eliminowana jest potrzeba zewnętrznej normalizacji, co z kolei redukuje ryzyko błędów wynikających z nieprawidłowego skalowania danych.

Zintegrowane modelowanie atrybutów mieszanych

W modelach generatywnych stosuje się specyficzne funkcje prawdopodobieństwa dla różnych typów atrybutów:

- **Atrybuty ciągłe** są modelowane przy użyciu funkcji gęstości prawdopodobieństwa $f_{m,\theta}(X_i)$, np. normalnych rozkładów opisanych parametrami takimi jak średnia i wariancja dla każdego komponentu mieszaniny,
- **Atrybuty kategoryczne** są reprezentowane przez dyskretne funkcje prawdopodobieństwa $g_{m,\lambda}(X_i)$, które definiują prawdopodobieństwo wystąpienia każdej z możliwych kategorii dla danego atrybutu w kontekście każdego komponentu modelu.

Dla scenariuszy zawierających oba typy atrybutów, definiuje się wspólną funkcję gęstości:

$$h_m(X_i) = f_{m,\theta}(X_i) \cdot g_{m,\lambda}(X_i) \tag{3.7}$$

Takie podejście pozwala na holistyczną ocenę generatywną każdego obiektu, uwzględniając jednocześnie atrybuty liczbowe i kategoryczne. Dzięki temu możliwe jest dokładniejsze modelowanie zależności występujących w danych mieszanych.

W przypadku danych mieszanych, algorytm EM działa w podobny sposób jak w innych scenariuszach, które zostały opisane w sekcji 3.2.2. Algorytm oblicza oczekiwane przynależności danych do komponentów modelu w kroku E, uwzględniając różne typy atrybutów, takie jak liczbowe i kategoryczne. W kroku M algorytm aktualizuje parametry modelu w taki sposób, aby maksymalizować log-wiarygodność danych, dostosowując się do specyfiki atrybutów mieszanych.

3.2.4 Identyfikacja anomalii modelami generatywnymi

Model generatywny może być użyty do wykrywania wartości odstających przez analizę prawdopodobieństwa generacji poszczególnych obiektów przez model. W praktyce, dla każdego obiektu obliczane jest prawdopodobieństwo, że został on wygenerowany przez jeden z komponentów modelu mieszaniny. Obiekty, które mają niskie prawdopodobieństwo generacji przez jakikolwiek z komponentów, mogą być uznane za odstające. Działanie to opiera się na założeniu, że większość danych dobrze pasuje do jednego z komponentów modelu, co jest reprezentowane przez wysokie prawdopodobieństwo generacji. Obiekty odstające, które nie pasują dobrze do żadnego z komponentów, charakteryzują się niskimi wartościami prawdopodobieństwa, sugerując, że nie są one typowymi obserwacjami w ramach rozkładu modelu. Takie podejście pozwala nie tylko na identyfikację odstających wartości, ale również na zrozumienie, które cechy danych sprawiają, że nie pasują one do modelu. W rezultacie, model generatywny dostarcza nie tylko narzędzia do detekcji anomalii, ale również głębszych odkryć mogących prowadzić do dalszej eksploracji i analizy danych.

3.3 Metody oparte na modelach liniowych

Aby przeprowadzić analizę danych kategorycznych przy użyciu modeli liniowych, dane te można przekształcić w formę binarną. Każdy atrybut kategoryczny jest zamieniany na zbiór zmiennych binarnych, tzw. kodowanie *one-hot*, gdzie każda wartość atrybutu ma przypisaną swoją własną zmienną. W tej reprezentacji, dla każdej wartości atrybutu przypisuje się odrębny wymiar binarny, który przyjmuje wartość 1, gdy ta wartość jest obecna, natomiast pozostałe wymiary mają wartość 0. W świecie analizy danych i uczenia maszynowego, kodowanie *one-hot* jest jedną z najpopularniejszych technik przekształcania danych kategorycznych na format, który może być z łatwością używany w algorytmach analitycznych. Wprowadzenie sztucznej cechy (*ang. dummy feature*) dla każdej unikatowej wartości w kolumnie cechy nominalnej umożliwia precyzyjne reprezentowanie tych danych w sposób binarny.

Jeśli dany atrybut A_j posiada n_j możliwych wartości w zbiorze Q_j, końcowy zbiór danych będzie miał wymiarowość $\sum_{j=1}^{m} n_j$. Taka konwersja może znacznie zwiększyć wymiarowość danych, zwłaszcza gdy wiele atrybutów ma dużą liczbę unikalnych wartości. Jednym z wyzwań związanych z tego rodzaju transformacją jest to, że atrybuty mogą nieintencjonalnie uzyskać różny poziom istotności. Dzieje się tak, ponieważ liczba możliwych wartości n_j dla każdego atrybutu A_j może być zróżnicowana, co prowadzi do przypisania im różnej liczby wymiarów. Dodatkowo, częstotliwości występowania poszczególnych wartości atrybutów wpływają na ich względne znaczenie. Na przykład, kolumna, w której 40% wartości binarnych to 1, różni się od kolumny, gdzie ten odsetek wynosi 5%. W takich przypadkach kolumna z wyższym udziałem wartości 1 staje się ważniejsza, ponieważ wiąże się z większą zmiennością w swoich wartościach binarnych.

Normalizacja danych binarnych

Normalizacja danych binarnych ma na celu przekształcenie wartości binarnych tak, aby miały one porównywalną skalę i wpływ na analizę. Normalizacja jest szczególnie ważna, gdy różne atrybuty mają różną liczbę możliwych wartości lub różne częstotliwości występowania. Dzięki normalizacji można uzyskać bardziej rzetelne i spójne wyniki analizy oraz następujące korzyści:

- **zrównoważenie wpływu różnych atrybutów:** normalizacja zapobiega sytuacji, w której atrybuty z większą liczbą unikalnych wartości lub o wyższej częstotliwości występowania zyskują nieproporcjonalnie dużą wagę. Dzięki temu wszystkie atrybuty mają równą szansę na wpływanie na wyniki analizy,

- **poprawa wydajności modeli:** modele uczące się na znormalizowanych danych mogą działać bardziej efektywnie, ponieważ wartości atrybutów są na tej samej skali. To może prowadzić do szybszej konwergencji algorytmów i lepszej wydajności modeli,

a także poprawić stabilność wyników, zwłaszcza w przypadku skomplikowanych zestawów danych o dużej liczbie cech,

- **lepsza interpretacja wyników:** wyniki analizy są bardziej interpretowalne, gdy dane są znormalizowane. Łatwiej jest zrozumieć i porównać wpływ poszczególnych atrybutów na model,

- **redukcja wpływu wartości odstających:** normowanie pomaga w redukcji wpływu wartości odstających, które mogą znacząco wpłynąć na wyniki analizy, jeśli nie są odpowiednio znormalizowane. Jednak należy pamiętać, że w procesie normalizacji można również stracić cenne informacje dotyczące wartości odstających, co może być istotne w kontekście ich wykrywania.

Normalizacja danych binarnych nie tylko ułatwia proces analizy, ale także podnosi jej jakość, co jest istotne dla uzyskania dokładnych i użytecznych wniosków. Dzięki temu, że każda kolumna ma porównywalną wagę, badania stają się bardziej „sprawiedliwe", a ich wyniki - bardziej wiarygodne. Proces normalizacji danych binarnych przebiega następująco:

- **obliczenie odchylenia standardowego:**
dla każdej kolumny oblicza się odchylenie standardowe. Dla j-tej wartości przekształconych danych binarnych, dla której ułamek u_{ij} wpisów przyjmuje wartość 1, odchylenie standardowe określa się jako $\sqrt{u_{ij} \cdot (1 - u_{ij})}$,

- **podzielenie kolumny przez odchylenie standardowe:**
każdą kolumnę danych binarnych dzieli się przez $\sqrt{u_{ij} \cdot (1 - u_{ij})}$. Dzięki temu wariancja każdej pochodnej binarnej wynosi 1, co odpowiada normalizacji stosowanej w danych numerycznych,

- **uwzględnienie liczby możliwych wartości n_j:**
aby uwzględnić wpływ liczby możliwych wartości atrybutu n_j, wszystkie kolumny dla i-tego atrybutu dzieli się przez $\sqrt{n_j \cdot u_{ij} \cdot (1 - u_{ij})}$. Zapewnia to, że suma wariancji dla wszystkich kolumn odpowiadających danemu atrybutowi wynosi 1.

Ułamek u_{ij} w kontekście normalizacji danych binarnych odnosi się do proporcji wartości 1 w danej kolumnie j dla przekształconych danych binarnych. Innymi słowy, u_{ij} reprezentuje częstotliwość, z jaką wartość 1 pojawia się w kolumnie j wiersza i w zbiorze danych binarnych. Jeżeli, na przykład, w kolumnie j znajduje się 100 wpisów i 40 z nich to wartość 1, wówczas u_{ij} dla tej kolumny wynosi 0,4.

Wykorzystanie PCA do identyfikacji anomalii

W celu identyfikacji anomalii po normalizacji danych można zastosować metodę PCA [175, 176]. Jak omówiono w podrozdziale 2.2, PCA działa poprzez przekształcenie danych do nowej przestrzeni współrzędnych, w której pierwsze kilka składowych głównych (największych osi wariancji) zawiera większość informacji o oryginalnym zbiorze danych. Proces normalizacji danych binarnych zapewnia, że każdy atrybut ma równą wagę, co jest niezbędne przed zastosowaniem PCA. Proces normalizacji może być również rozszerzony na zbiory danych mieszanych przez dodanie znormalizowanych atrybutów numerycznych do tej reprezentacji. Normalizowanie kolumn numerycznych polega na takim przekształceniu, aby wariancja każdej kolumny wynosiła jeden. Dzięki temu metody takie jak PCA mogą być skutecznie stosowane w zróżnicowanych zbiorach danych, nie faworyzując żadnego z atrybutów. PCA jest szczególnie efektywne dla rzadkich zbiorów danych binarnych, ponieważ potrafi zredukować ich wymiarowość do niewielkiej liczby składowych. W celu wykrycia anomalii, po oszacowaniu składowych głównych, obliczane są odchylenia wzdłuż tych składowych. Suma kwadratów tych odchyleń jest modelowana jako rozkład χ^2 z d stopniami swobody. Ekstremalne wartości odchyleń są identyfikowane jako obserwacje odstające, co umożliwia ich raportowanie i dalszą analizę.

Nadzorowane modele regresji w identyfikacji anomalii

Stosowanie nadzorowanych modeli regresji do wykrywania anomalii w zbiorach danych polega na wyznaczeniu jednej cechy jako zmiennej zależnej, którą chcemy przewidzieć na podstawie pozostałych cech. Na początku procesu wybiera się atrybut, który będzie prognozowany, a następnie buduje się model regresji, aby przewidzieć jego wartość na podstawie innych atrybutów. Oblicza się błąd średniokwadratowy RMSE (*ang. root mean square error*, RMSE) dla każdej predykcji, który mierzy różnicę między rzeczywistymi a przewidywanymi wartościami. Wysokie wartości RMSE wskazują na potencjalne obserwacje odstające. Proces ten powtarza się dla różnych atrybutów, które są traktowane jako zmienne zależne, a uzyskane wyniki RMSE są uśredniane dla każdego obiektu. Dzięki temu możliwa jest dokładniejsza identyfikacja anomalii. Metoda ta jest również skuteczna dla danych mieszanych, gdzie numeryczne atrybuty są normalizowane w taki sposób, aby ich wariancja wynosiła jeden. Taki zabieg zapobiega nadmiernemu wpływowi jednego atrybutu na model.

3.4 Metody oparte na bliskości danych

Opracowanie skutecznych miar podobieństwa dla danych kategorycznych odgrywa decydującą rolę w realizacji algorytmów opartych na bliskości, takich jak grupowanie. Choć temat ten obejmuje szerokie spektrum badań, w tym podrozdziale przedstawiono najczęściej stosowane miary, które są wykorzystywane do identyfikacji obserwacji odstających. Dla

danych kategorycznych bardziej intuicyjne jest analizowanie podobieństw niż odległości, ponieważ wiele miar bazuje na porównywaniu konkretnych, dyskretnych wartości. Weźmy pod uwagę dwie próbki danych $X_i = (x_{i,1}, \ldots, x_{i,m})$ oraz $X_k = (x_{k,1}, \ldots, x_{k,m})$, gdzie $x_{i,j}$ i $x_{k,j}$ są wartościami j-tego atrybutu A_j dla obiektów X_i i X_k. Podobieństwo między tymi próbkami jest sumą podobieństw pomiędzy ich poszczególnymi atrybutami A_j. Oznacza to, że jeśli $S(x_{i,j}, x_{k,j})$ reprezentuje podobieństwo pomiędzy wartościami atrybutu A_j, to całkowite podobieństwo jest określone jako (3.8):

$$\mathrm{S}(X_i, X_k) = \sum_{j=1}^{m} S(x_{i,j}, x_{k,j}) \tag{3.8}$$

Miara podobieństwa bazuje na funkcji $S(x_{i,j}, x_{k,j})$ zastosowanej w tej definicji. Najprostszą metodą wyznaczenia wartości $S(x_{i,j}, x_{k,j})$ jest przypisanie jej 1, gdy $x_{i,j} = x_{k,j}$, oraz 0 w przeciwnym przypadku. Taki sposób nazywany jest miarą nakładania (*ang. overlap measure*). Chociaż jest to metoda dość uproszczona, zyskała dużą popularność z powodu swojej łatwości stosowania. Jej główną wadą jest jednak pominięcie względnej częstości występowania poszczególnych wartości atrybutów. Przykładowo, zgodność rzadkich wartości atrybutu A_j (np. dwóch osób z unikalnym zainteresowaniem jak kolekcjonowanie meteorytów) ma większe znaczenie statystyczne niż zgodność często występujących wartości (np. dwóch osób lubiących oglądać filmy). Tego typu różnice są szczególnie ważne przy analizie obserwacji odstających, gdzie nierównomierny rozkład wartości atrybutów może świadczyć o nietypowych cechach. Są dwa główne typy metod pomiaru podobieństwa w danych kategorycznych:

- **statystyczne cechy podobieństwa**: metody te bazują na statystycznych częstotliwościach występowania poszczególnych wartości atrybutów w celu zwiększenia precyzji obliczeń podobieństwa. Na przykład, w analizie genomowej, dopasowanie rzadkich mutacji genetycznych jest uznawane za bardziej znaczące, podczas gdy niedopasowania rzadkich wartości są uznawane za mniej istotne. Ta technika uwzględnia różnice w częstości występowania, co pozwala na lepsze wychwycenie subtelnych, ale kluczowych podobieństw w danych [177],
- **kontekstowe sąsiedztwa**: w tym podejściu do obliczania podobieństwa używa się sąsiedztw obserwacji, co pośrednio modeluje korelacje między atrybutami w określonym sąsiedztwie. Przykładowo, jeśli słowa *wykres* i *diagram* częściej współwystępują w sąsiedztwie danej obserwacji niż słowa *wykres* i *tablica*, to znaczy, że *wykres* i *diagram* są semantycznie bardziej zbliżone niż *wykres* i *tablica*. W ten sposób podobieństwo definiuje sąsiedztwo, a jednocześnie sąsiedztwo definiuje podobieństwo, co może prowadzić do wzajemnego wpływania tych dwóch pojęć na siebie.

Statystyczne cechy podobieństwa

W kontekście danych kategorycznych, obliczanie podobieństwa warto oprzeć na zagregowanych właściwościach statystycznych całego zbioru danych. Przykładowo, jeśli atrybut przyjmuje wartość *Czerwony* w 95% przypadków, a *Zielony* w 5% przypadków, to podobieństwo między często występującymi wartościami atrybutów powinno być mniej istotne niż podobieństwo między rzadkimi wartościami. Ta zasada stanowi fundament wielu technik normalizacyjnych, takich jak odwrotna częstość dokumentów IDF (*ang. inverse document frequency*, IDF) w dziedzinie wyszukiwania informacji [178]. Zróżnicowanie między wartościami atrybutów powinno być ważone w zależności od ich częstotliwości występowania. IDF jest miarą używaną w przetwarzaniu języka naturalnego oraz w wyszukiwaniu informacji i analizie tekstu, do oceny ważności jednostek leksykalnych w kontekście całej kolekcji dokumentów. IDF jest integralną częścią algorytmu TF-IDF (*ang. term frequency-inverse document frequency*, TF-IDF), który łączy częstość występowania jednostki leksykalnej w dokumencie (TF) z odwrotną częstością dokumentów zawierających tę jednostkę leksykalną (IDF). $\mathrm{IDF_i}$ oblicza się według wzoru (3.9):

$$\mathrm{IDF_i} = \log\left(\frac{|D|}{|\{d : t_i \in d\}|}\right), \tag{3.9}$$

gdzie:

- $|D|$ – liczba dokumentów w kolekcji,
- $|\{d : t_i \in d\}|$ – liczba dokumentów zawierających przynajmniej jedno wystąpienie danej jednostki leksykalnej t_i.

Im rzadziej jednostka leksykalna pojawia się w dokumentach, tym wyższa jest jej wartość IDF, co wskazuje na jej większe znaczenie w kontekście identyfikacji unikalnych cech dokumentu. Z kolei jednostki leksykalne, które występują w wielu dokumentach, charakteryzują się niską wartością IDF, co sugeruje ich mniejsze znaczenie w procesie różnicowania dokumentów. Stosowanie miary IDF jest fundamentalne w algorytmach wyszukiwarek internetowych, gdyż umożliwia zrównoważenie lokalnego znaczenia jednostek leksykalnych z ich ogólnym znaczeniem w całej kolekcji dokumentów. Dzięki temu, wyszukiwarki są w stanie dostarczać bardziej precyzyjne i relewantne wyniki, lepiej odpowiadając na zapytania użytkowników.

Różnice między wartościami $x_{i,j}$ i $x_{k,j}$ są bardziej prawdopodobne, gdy atrybut A_j posiada dużą liczbę różnych możliwych wartości n_j lub gdy $x_{i,j}$ i $x_{k,j}$ występują rzadko. W odpowiedzi na te wyzwania, miara Eskina [179] wprowadza modyfikację do tradycyjnej miary nakładania, przypisując wartość $S(x_{i,j}, x_{k,j}) = 1$ w sytuacji, gdy $x_{i,j} = x_{k,j}$, a w przypadku różnicy między tymi wartościami stosuje niezerowy wskaźnik podobieństwa wyliczany jako $n_j^2/(n_j^2 + 2)$. Zamiast bazować jedynie na liczbie unikalnych wartości atrybutu n_j, można bezpośrednio uwzględniać rzeczywiste częstości występowania $x_{i,j}$

i $x_{k,j}$, co pozwala na mniej surowe karanie niezgodności między rzadkimi wartościami. Takie podejście zwiększa dokładność i subtelność oceny podobieństwa, szczególnie w kontekście atrybutów o zróżnicowanych rozkładach częstości. Alternatywą jest miara IOF (*ang. inverse occurrence frequency*, IOF), która przypisuje wartość 1 dla zgodnych wartości oraz $1/(1+\log[f_j(x_{i,j})]\cdot\log[f_j(x_{k,j})])$ w przeciwnym przypadku. W tej formule, $f_j(x_{i,j})$ i $f_j(x_{k,j})$ reprezentują rzeczywiste liczby wystąpień wartości $x_{i,j}$ i $x_{k,j}$ dla danego atrybutu A_j. Miara IOF została pierwotnie zaprojektowana dla zadań eksploracji tekstu [180], a później dostosowana do zmiennych kategorycznych [177]. Miara ta przypisuje większą wagę niezgodnościom w przypadku wartości rzadziej występujących i odwrotnie, co pozwala na bardziej precyzyjne różnicowanie wartości o różnej częstości występowania. IOF może być również stosowana w przypadku zgodnych wartości, co sprawia, że obliczenia są zbliżone do tych używanych w analizie danych tekstowych.

Miara Goodalla, która przypisuje większe wagi rzadkim wartościom, stanowi wyjątkowo wartościowe narzędzie w analizie danych. Niestety, oryginalna metoda zaproponowana przez Goodalla [181] charakteryzuje się wysokimi wymaganiami obliczeniowymi, co ogranicza jej praktyczne zastosowania. W odpowiedzi na te problemy, Boriah i współautorzy [177] wprowadzili szereg heurystycznych przybliżeń, które zachowują istotę miary Goodalla, jednocześnie znacząco zwiększając jej efektywność obliczeniową. W literaturze zaproponowano szeroką gamę miar podobieństwa dla danych kategorycznych [182, 183, 184, 185, 186]. Szczególnie istotna jest praca Boriah i współautorów [177], która koncentruje się na analizie obserwacji odstających. Praca ta porównuje różne miary podobieństwa dla danych kategorycznych, zwłaszcza w kontekście wykrywania anomalii. Autorzy badali 14 różnych miar, oceniając ich skuteczność na zróżnicowanych zestawach danych. Główne wnioski z tej pracy są niezwykle pouczające. Wskazują one, że nie istnieje jedna uniwersalnie najlepsza miara. Każda z miar posiada swoje unikalne zalety i wady, które zależą od specyficznych cech analizowanych danych. Na przykład, miary takie jak Lin, OF i Goodall3 wykazują konsekwentnie wysoką skuteczność w różnych kontekstach, podczas gdy inne miary charakteryzują się bardziej zmienną wydajnością. Interesującym spostrzeżeniem jest komplementarność niektórych miar. Przykłady obejmują pary miar takie jak OF i IOF, czy Lin i Lin1, co sugeruje, że kombinacja różnych miar może prowadzić do lepszych wyników. Jest to niezwykle istotne, ponieważ pokazuje, że synergiczne wykorzystanie różnych podejść, które można porównać do pracy zespołowej, gdzie każda metoda wnosi swoje unikalne zalety, może maksymalizować precyzję analizy, wychwytując subtelności, które mogłyby zostać pominięte przez pojedyncze miary.

Wydajność miar podobieństwa jest ściśle związana z charakterystyką danych, które są analizowane. Na przykład miara Eskin, choć użyteczna w wielu kontekstach, wykazuje słabe wyniki w przypadkach, gdzie atrybuty mają wiele różnych wartości. Wynika to z jej ograniczonej zdolności do różnicowania pomiędzy rzadkimi a często występującymi wartościami. Z kolei miary, które potrafią wykorzystywać dodatkowe informacje zawarte w danych, takie jak rozkład częstości wartości atrybutów, na ogół osiągają lepsze wyni-

ki. Przykładem mogą być miary wykorzystujące podejście probabilistyczne lub oparte na entropii, które bardziej precyzyjnie uwzględniają strukturalne właściwości danych. W skrócie, wybór odpowiedniej miary podobieństwa jest uzależniony od specyficznych cech analizowanych danych. Nie ma jednej uniwersalnej miary, która byłaby najlepsza we wszystkich sytuacjach. Zamiast tego, skuteczność miar różni się w zależności od kontekstu aplikacji. Miary takie jak te opracowane przez Lin czy Goodall często są bardziej efektywne w analizach, gdzie istotne jest uwzględnienie rzadkich wartości lub specyficznego rozkładu danych.

Tabela 3.2, zamieszczona w pracy [177], przedstawia szczegółowe informacje na temat różnych miar podobieństwa stosowanych dla atrybutów kategorycznych. Kluczowe elementy tej tabeli to kolumna opisująca miarę $S_j(x_{i,j}, x_{k,j})$, która zawiera formułę określającą sposób obliczania podobieństwa pomiędzy dwiema wartościami $x_{i,j}$ i $x_{k,j}$ dla j-tego atrybutu. Formuły te uwzględniają specyficzne cechy danych oraz częstości występowania wartości w zbiorze danych. Dodatkowo, kolumna $w_j, j = 1 \ldots m$ wskazuje na wagę przypisaną każdemu j-temu atrybutowi. Wartości w tej kolumnie określają, jak wyniki podobieństwa dla poszczególnych atrybutów są łączone, aby uzyskać końcowy wynik podobieństwa dla całego zbioru danych. Liczba m oznacza tutaj liczbę atrybutów w analizowanym zbiorze danych. Przykładowo, dla miary Overlap, $S_j(x_{i,j}, x_{k,j})$ wynosi 1, gdy $x_{i,j}$ jest równe $x_{k,j}$, i 0 w przeciwnym razie. Wartość w_j dla tej miary jest stała i wynosi $\frac{1}{m}$, co oznacza, że każdy atrybut ma jednakową wagę w końcowym obliczeniu podobieństwa. Analizując tabelę 3.2, należy zauważyć, że całkowite podobieństwo $S(X_i, X_k)$ jest wyrażone jako suma ważona poszczególnych miar podobieństwa:

$$S(X_i, X_k) = \sum_{j=1}^{m} w_j S_j(x_{i,j}, x_{k,j})$$

Dla miary Lin1, formuła przyjmuje postać $\{Q_j \subseteq A_j : \forall q \in Q_j, \hat{p}_j(x_{i,j}) \leq \hat{p}_j(q) \leq \hat{p}_j(x_{k,j})\}$, przy założeniu $\hat{p}_j(x_{i,j}) \leq \hat{p}_j(x_{k,j})$. Dla miary Goodall1, formuła jest określona jako $\{Q_j \subseteq A_j : \forall q \in Q_j, p_j(q) \leq p_j(x_{i,j})\}$. Dla miary Goodall2, przyjmuje postać $\{Q_j \subseteq A_j : \forall q \in Q_j, p_j(q) \geq p_j(x_{i,j})\}$.

Podejście wykorzystujące kontekstowe sąsiedztwa

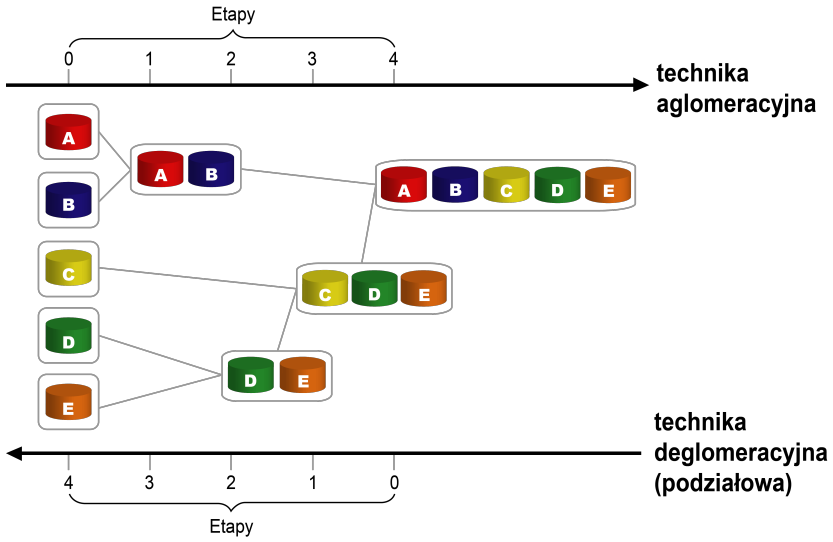
Kontekstowe podobieństwo wprowadza nowatorskie podejście do analizy danych kategorycznych, w którym relacje między próbkami danych są wykorzystywane do definiowania powiązań między wartościami atrybutów. Dzięki temu możliwe jest wychwytywanie subtelności, które mogłyby zostać pominięte przy użyciu tradycyjnych metod. To podejście jest szczególnie istotne w różnych dziedzinach, takich jak przetwarzanie języka naturalnego (*ang. natural language processing*, NLP), systemach rekomendacji oraz analizie sieci społecznościowych. W przetwarzaniu języka naturalnego, słowa występujące w podob-

Tabela 3.2: Przegląd miar odległości i podobieństwa dla atrybutów kategorycznych. Tabela przedstawia równania wykorzystywane do obliczania podobieństwa między wartościami atrybutów kategorycznych. Każda miara opisuje sposób wyznaczania podobieństwa dla poszczególnych atrybutów $S_j(x_{i,j}, x_{k,j})$, co zostało szczegółowo pokazane w kolumnie 2. Dodatkowo, w kolumnie 3 zaprezentowano sposób obliczania wagi dla każdego atrybutu w_j, która ma wpływ na ogólny wynik podobieństwa w zbiorze danych. Źródło: opracowanie własne na podstawie pracy [177].

Miara	$S_j(x_{i,j}, x_{k,j})$	$w_j, j = 1 \ldots m$
Overlap	$\begin{cases} 1 & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Eskin	$\begin{cases} 1 & \text{jeśli } x_{i,j} = x_{k,j} \\ \frac{n_j^2}{n_j^2+2} & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
IOF	$\begin{cases} 1 & \text{jeśli } x_{i,j} = x_{k,j} \\ \frac{1}{1+\log f_j(x_{i,j}) \cdot \log f_j(x_{k,j})} & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
OF	$\begin{cases} 1 & \text{jeśli } x_{i,j} = x_{k,j} \\ \frac{1}{1+\log\left(\frac{N}{f_j(x_{i,j})}\right) \cdot \log\left(\frac{N}{f_j(x_{k,j})}\right)} & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Lin	$\begin{cases} 2\log \hat{p}_j(x_{i,j}) & \text{jeśli } x_{i,j} = x_{k,j} \\ 2\log(\hat{p}_j(x_{i,j}) + \hat{p}_j(x_{k,j})) & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{\sum_{j=1}^{m}(\log \hat{p}_j(x_{i,j}) + \log \hat{p}_j(x_{k,j}))}$
Lin1	$\begin{cases} \sum_{q \in Q_j} \log \hat{p}_j(q) & \text{jeśli } x_{i,j} = x_{k,j} \\ 2\log \sum_{q \in Q_j} \hat{p}_j(q) & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{\sum_{j=1}^{m} \sum_{q \in Q_j} \log \hat{p}_j(q)}$
Goodall1	$\begin{cases} 1 - \sum_{q \in Q_j} p_j^2(q) & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Goodall2	$\begin{cases} 1 - \sum_{q \in Q_j} p_j^2(q) & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Goodall3	$\begin{cases} 1 - p_j^2(x_{i,j}) & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Goodall4	$\begin{cases} p_j^2(x_{i,j}) & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Smirnov	$\begin{cases} 2 + \frac{N - f_j(x_{i,j})}{f_j(x_{i,j})} + \sum_{q \in \{A_j \setminus x_{i,j}\}} \frac{f_j(q)}{N - f_j(q)} & \text{jeśli } x_{i,j} = x_{k,j} \\ \sum_{q \in \{A_j \setminus \{x_{i,j}, x_{k,j}\}\}} \frac{f_j(q)}{N - f_j(q)} & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{\sum_{j=1}^{m} n_j}$
Gambaryan	$\begin{cases} -\left(\hat{p}_j(x_{i,j}) \log_2 \hat{p}_j(x_{i,j}) + (1 - \hat{p}_j(x_{i,j})) \log_2 (1 - \hat{p}_j(x_{i,j}))\right) & \text{jeśli } x_{i,j} = x_{k,j} \\ 0 & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{\sum_{j=1}^{m} n_j}$
Burnaby	$\begin{cases} 1 & \text{jeśli } x_{i,j} = x_{k,j} \\ \frac{\sum_{q \in A_j} 2\log(1 - \hat{p}_j(q))}{\log\left(\frac{\hat{p}_j(x_{i,j}) \hat{p}_j(x_{k,j})}{(1 - \hat{p}_j(x_{i,j}))(1 - \hat{p}_j(x_{k,j}))}\right) + \sum_{q \in A_j} 2\log(1 - \hat{p}_j(q))} & \text{w przeciwnym razie} \end{cases}$	$\frac{1}{m}$
Anderberg	$S(x_{i,j}, x_{k,j}) = \frac{\sum_{j \in \{1 \le j \le m : x_{i,j} = x_{k,j}\}} \left(\frac{1}{\hat{p}_j(x_{i,j})}\right)^2 \frac{2}{n_j(n_j+1)}}{\sum_{j \in \{1 \le j \le m : x_{i,j} = x_{k,j}\}} \left(\frac{1}{\hat{p}_j(x_{i,j})}\right)^2 \frac{2}{n_j(n_j+1)} + \sum_{j \in \{1 \le j \le m : x_{i,j} \ne x_{k,j}\}} \left(\frac{1}{2\hat{p}_j(x_{i,j}) \hat{p}_j(x_{k,j})}\right) \frac{2}{n_j(n_j+1)}}$	

nych kontekstach mogą być uznane za semantycznie podobne. Na przykład, słowa *kot* i *pies* często pojawiają się w kontekście *zwierzęta domowe*, co sugeruje ich semantyczne podobieństwo. Analiza takich kontekstowych powiązań umożliwia lepsze zrozumienie i interpretację tekstu, co ma znaczenie w aplikacjach takich jak tłumaczenie maszynowe czy wyszukiwanie informacji. W systemach rekomendacji, produkty kupowane razem przez użytkowników mogą być uznane za podobne. Na przykład, jeśli użytkownicy często kupują razem *telefon* i *słuchawki*, te produkty są uznawane za podobne w kontekście preferencji zakupowych. Taka analiza kontekstowych sąsiedztw pozwala na tworzenie bardziej trafnych i spersonalizowanych rekomendacji, co zwiększa zadowolenie użytkowników i skuteczność systemów rekomendacyjnych. W analizie sieci społecznościowych, osoby mające wspólnych znajomych i uczestniczące w tych samych wydarzeniach są uznawane za bardziej podobne. Na przykład, jeśli dwie osoby często uczestniczą w tych samych wydarzeniach i mają wielu wspólnych znajomych, są one uznawane za podobne pod względem swoich zainteresowań i zachowań społecznych. Analiza takich kontekstowych powiązań jest ważna dla zrozumienia struktury i dynamiki sieci społecznościowych, co ma zastosowanie w marketingu, socjologii i wielu innych dziedzinach.

Znaczenie kontekstu zostało trafnie podkreślone przez filozofa Nelsona Goodmana, który argumentował, że podobieństwo między obiektem X a Y jest pojęciem bez znaczenia, chyba że można określić „w jakim aspekcie" X jest podobne do Y. Jego pogląd sugeruje, że samo stwierdzenie, iż dwa przedmioty są podobne, jest niepełne i niejasne bez dodatkowego kontekstu. W swoim wpływowym eseju zatytułowanym *„Seven Strictures on Similarity"* [187], Goodman podkreślał, że podobieństwo jest pojęciem względnym i zmiennym, które wymaga określenia kontekstu, w którym dwa obiekty są podobne. Argumentował, że bez tego dodatkowego kontekstu, stwierdzenie o podobieństwie jest pozbawione znaczenia, ponieważ każdy obiekt jest w pewien sposób podobny do każdego innego obiektu w jakimś aspekcie. Goodman zwraca uwagę na to, że podobieństwo jest często używane w sposób niewłaściwy i prowadzi do nieporozumień. Aby mierzyć kontekstowe podobieństwo, można przekształcić atrybuty kategoryczne do formy binarnej, co może ułatwić analizę poprzez wykorzystanie miar podobieństwa dla danych binarnych. Reprezentacja binarna umożliwia porównywanie wartości atrybutów za pomocą miar takich jak miara Jaccarda [188], która jest skuteczna przy danych binarnych. Jednakże, jak zauważają autorzy [189], wyniki tej miary są silnie zależne od wielkości zbioru danych, co może prowadzić do błędnych wniosków w analizach genomowych. Przekształcenie atrybutów kategorycznych do formy binarnej nie zawsze jest konieczne, ponieważ można bezpośrednio pracować na danych kategorycznych, wykorzystując relacje między próbkami danych do definiowania powiązań między wartościami atrybutów. Na przykład miara Goodalla, która nadaje większą wagę rzadkim wartościom, może być stosowana bez konieczności binaryzacji danych. Jednym z najprostszych podejść do mierzenia podobieństwa kontekstowego jest zastosowanie metody hierarchicznego grupowania. Metody te, choć pierwotnie przeznaczone do danych numerycznych, można łatwo dostosować do danych kategorycznych poprzez odpowiedni



Rysunek 3.1: Techniki hierarchicznego grupowania dla danych kategorycznych. Schemat przedstawia podejścia aglomeracyjne (łączenie obiektów w większe grupy) oraz deglomeracyjne (podział dużych skupień na mniejsze grupy). Źródło: opracowanie własne.

dobór miar podobieństwa i strategii łączenia grup. Jak przedstawiono na rysunku 3.1, techniki hierarchicznego grupowania obejmują podejścia aglomeracyjne, gdzie obiekty są łączone w większe grupy na podstawie podobieństwa, oraz deglomeracyjne, w których duże skupienia są dzielone na mniejsze, bardziej jednorodne grupy. W kontekście podobieństwa kontekstowego, techniki te są szczególnie użyteczne, ponieważ pozwalają na uwzględnienie złożonych zależności między wartościami atrybutów w danych kategorycznych. Przy okazji proces grupowania pozwala na identyfikację obserwacji, które znacząco różnią się od reszty, gdyż w miarę jak dendrogram rozwija się, nietypowe obiekty często pozostają w mniejszych, odizolowanych grupach lub są wcześnie oddzielane od głównych skupień.

Mierzenie podobieństwa kontekstowego można zrealizować na dwa różne sposoby, bazując na relacjach między wartościami atrybutów lub między obiektami w zbiorze danych. W artykule [190] autorzy opisują metody mierzenia podobieństwa między różnymi komponentami relacji binarnych (0/1) w bazach danych kategorii. Proponują oni miary podobieństwa między atrybutami, wierszami oraz subrelacjami, które znajdują zastosowanie w grupowaniu, klasyfikacji i innych procesach eksploracji danych. Miary te opierają się na kontekście poszczególnych komponentów, co pozwala odkrywać bardziej subtelne relacje między nimi. Przykładowo, dwa produkty (atrybuty) są uznawane za podobne, jeśli ich zestawy klientów (subrelacje) są podobne. Problem znalezienia miar odległości został sformułowany jako system nieliniowych równań. Autorzy przedstawiają iteracyjny algorytm ICD (*ang. iterated contextual distances*, ICD), który w praktyce szybko osiąga stabilne wartości odległości, zazwyczaj w mniej niż pięciu iteracjach. Algorytm ten wymaga tylko jednego skanowania zbioru danych. Wyniki uzyskane na sztucznych i rze-

czywistych danych pokazują, że metoda jest wydajna i dostarcza dokładnych oraz spójnych wyników, które dobrze odzwierciedlają rzeczywiste relacje między danymi. Każde z tych podejść ma swoje unikalne zalety i zastosowania. Pierwsze podejście koncentruje się na relacjach między wartościami atrybutów, aby zbudować reprezentację obiektów w formie wartości rzeczywistych. W tej metodzie każda wartość atrybutu jest przekształcana na wartość liczbową, która odzwierciedla relacje między atrybutami. To podejście zawiera wszystkie istotne informacje o korelacjach między atrybutami, dzięki czemu odległości między obiektami w tej przestrzeni rzeczywistej automatycznie kodują podobieństwo między wartościami atrybutów. Drugie podejście działa odwrotnie – koncentruje się na odległościach między obiektami, aby określić relacje między wartościami atrybutów. Najpierw tworzy się subrelacje dla każdej wartości atrybutu, a następnie oblicza się centroid dla każdej subrelacji. Centroid to punkt w przestrzeni rzeczywistej, który reprezentuje środek grupy obiektów zawierających daną wartość atrybutu. Odległość L1 (*ang. Manhattan*) między centroidami tych subrelacji jest następnie używana do mierzenia odległości między wartościami atrybutów. Ta metoda odzwierciedla, jak bardzo różnią się subrelacje odpowiadające różnym wartościom atrybutów, co pomaga w zrozumieniu bardziej subtelnych relacji między nimi.

Dane mieszane

Trudności związane z danymi mieszanymi w podejściu opartym na bliskości można rozwiązać, przypisując odpowiednie wagi komponentom kategorycznym i numerycznym oraz normalizując wartości podobieństwa w obu domenach. Parametr λ reguluje względne znaczenie tych komponentów, umożliwiając dostosowanie modelu do specyficznych cech analizowanych danych. Wartości odległości są przekształcane na miary podobieństwa, przy czym odchylenia standardowe zapewniają właściwą skalę porównań. Dzięki tej normalizacji parametr λ staje się bardziej precyzyjnym wskaźnikiem względnej wagi między komponentami numerycznymi i kategorycznymi. Ogólne podobieństwo między dwoma obiektami X_i i X_k można zdefiniować według wzoru (3.10):

$$S(X_i, X_k) = \lambda \cdot \frac{S_{\text{num}}(\mathrm{x}_{i,n}, \mathrm{x}_{k,n})}{\sigma_n} + (1 - \lambda) \cdot \frac{S_{\text{kat}}(\mathrm{x}_{i,k}, \mathrm{x}_{k,k})}{\sigma_k}, \tag{3.10}$$

gdzie:

- $\mathrm{x}_{i,n}$ i $\mathrm{x}_{k,n}$ - podzbiory atrybutów numerycznych dla obiektów X_i i X_k,
- $\mathrm{x}_{i,k}$ i $\mathrm{x}_{k,k}$ - podzbiory atrybutów kategorycznych dla obiektów X_i i X_k,
- $S_{\text{num}}(\mathrm{x}_{i,n}, \mathrm{x}_{k,n})$ i $S_{\text{kat}}(\mathrm{x}_{i,k}, \mathrm{x}_{k,k})$ oznaczają podobieństwo numerycznych i kategorycznych atrybutów odpowiednio dla obiektów X_i i X_k,
- σ_n i σ_k - odchylenia standardowe dla podobieństw numerycznych i kategorycznych.

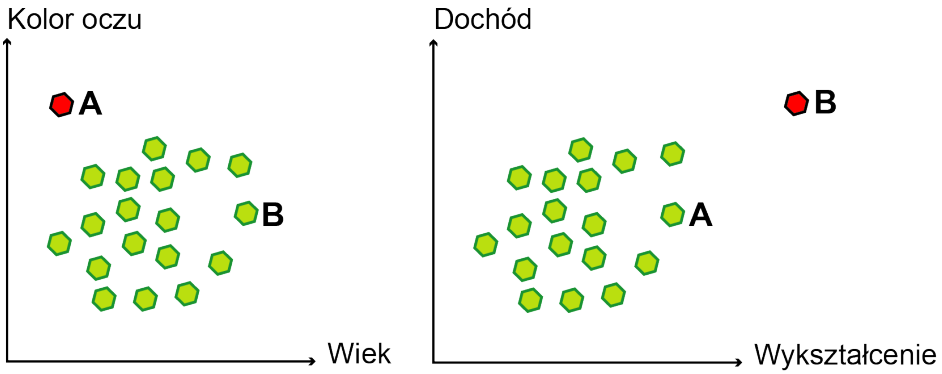
3.5 Identyfikacja anomalii w danych wielowymiarowych

Niniejszy podrozdział zajmuje się analizą problemu wymiarowości danych, ze szczególnym uwzględnieniem detekcji anomalii w wysokowymiarowych podprzestrzeniach danych kategorycznych. Rzeczywiste dane, takie jak te pochodzące z transakcji bankowych, medycznych rejestrów czy mediów społecznościowych, często charakteryzują się wysoką wymiarowością. Zawierają one zarówno liczne cechy kategoryczne, jak i atrybuty liczbowe. Taka struktura danych wymaga zastosowania zaawansowanych metod detekcji anomalii, które są zdolne do efektywnego skalowania się w miarę wzrostu liczby wymiarów. Trudność w wykrywaniu anomalii w tych danych polega na tym, że w przestrzeniach o dużej liczbie wymiarów rzadkość danych powoduje, że tradycyjne miary odległości między obiektami tracą swoją istotność.

3.5.1 Techniki selekcji cech w detekcji anomalii

Aby skutecznie rozwiązać problem identyfikacji anomalii w danych o wysokiej wymiarowości, konieczne jest stworzenie algorytmów, które będą w stanie wykrywać obiekty odstające w różnych podprzestrzeniach cech. Rysunek 3.2 doskonale ilustruje tę kwestię. Obiekty A i B, reprezentujące klientów banku, odróżniają się od reszty danych jedynie w niektórych projekcjach. Ta sytuacja wynika z faktu, że klienci są osadzeni w specyficznych podprzestrzeniach cech, co sprawia, że są widoczni tylko w wybranych widokach danych. Na pierwszym wykresie, obiekt A jest wyraźnie oddzielony od reszty danych w przestrzeni zdefiniowanej przez *kolor oczu* i *wiek*. Natomiast na wykresie obok, przestrzeń zdefiniowana przez *dochód* i *wykształcenie* jednoznacznie ukazuje obiekt B jako odrębny od reszty danych. To pokazuje, jak wybór odpowiednich cech do analizy może znacząco wpłynąć na zdolność wykrywania anomalii. W obiektach można mierzyć wiele zmiennych, takich jak temperatura, ciśnienie, wilgotność czy prędkość. Znaczące odchylenia w zachowaniu obiektu mogą być widoczne tylko w niewielkiej części tych zmiennych. Innymi słowy, tylko niektóre z mierzonych wartości mogą wykazywać anomalie, podczas gdy reszta pozostaje w normalnym zakresie. Na przykład, podczas monitorowania zdrowia pacjenta, anomalie mogą być zauważalne tylko w parametrach takich jak ciśnienie krwi czy tętno, mimo że wiele innych cech pozostaje w normie.

Wysoka wymiarowość danych znacznie komplikuje proces wykrywania obserwacji odstających z powodu obecności szumu i nieistotnych cech, które mogą obniżać skuteczność detekcji. W takich przypadkach zaleca się wybór odpowiednich podzbiorów cech za pomocą miar oceniających ich przydatność, aby zwiększyć efektywność wykrywania. Każdy obiekt danych jest opisany przez zestaw cech. Głównym celem jest identyfikacja tych cech, które umożliwiają skuteczniejsze wykrywanie anomalii. Istotne jest, aby zrozumieć, że zbiór najlepszych m cech niekoniecznie jest równy m najlepszym pojedynczym cechom. Przyczyną tego jest redundancja cech, czyli sytuacja, gdy cechy zawierają powtarzające się



Rysunek 3.2: Zmieniający się krajobraz obiektów w różnych podprzestrzeniach cech. Źródło: opracowanie własne.

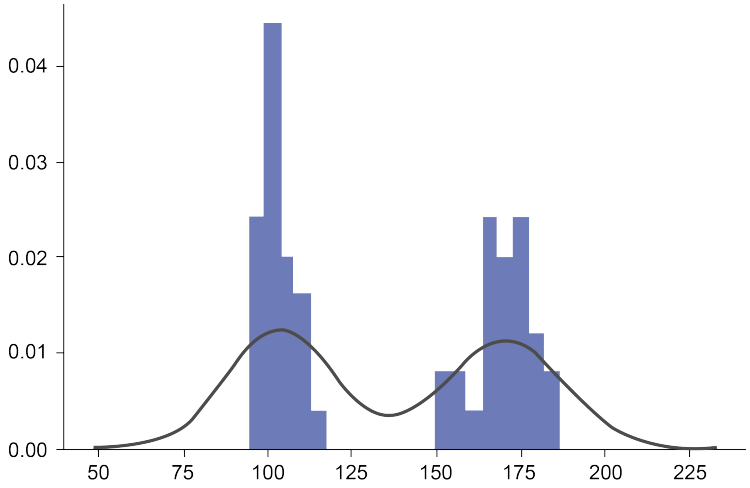
informacje, co może zakłócać proces identyfikacji. Dlatego ważne jest wybieranie takich cech, które minimalizują redundancję, a jednocześnie są istotne dla wykrywania anomalii. Prawidłowy dobór zmiennych jest czynnikiem decydującym o ostatecznej wartości analizy. W przypadku wykrywania anomalii, wybór odpowiednich zmiennych jest bardzo trudnym i złożonym zadaniem, dla którego nie istnieją jednoznaczne kryteria potwierdzające, czy dana zmienna jest istotna dla struktury anomalii czy też nie. W literaturze opisano trzy podejścia do wyboru zmiennych w analizie anomalii [191, 192]. Mimo że te podejścia zostały pierwotnie zaprojektowane dla grupowania i klasyfikacji, są one również istotne dla wykrywania anomalii ze względu na podobne wyzwania związane z wyborem zmiennych:

- **nadanie wag zmiennym:** polega na przypisaniu wag poszczególnym zmiennym, co ma odzwierciedlić ich znaczenie dla wykrywania anomalii,
- **pozostawienie tylko istotnych zmiennych:** w tym podejściu wybiera się tylko te zmienne, które są istotne dla wykrywania anomalii. Jest to szczególny przypadek nadania wag, gdzie wybrane zmienne mają wagę równą 1, a pozostałe są usuwane (waga równa 0),
- **zastąpienie zmiennych nowymi zmiennymi:** czasem używa się nowo stworzonych zmiennych, które lepiej odwzorowują strukturę anomalii lub dane.

Każde z tych podejść ma na celu doskonalenie jakości analizy anomalii poprzez staranny dobór zmiennych, które najlepiej opisują badane zjawisko. Teoretycznie, redukcja wymiarowości danych ułatwia wizualizację oraz zrozumienie danych, zmniejszając efekt klątwy wielowymiarowości, pojęcie szerzej omówiono w podrozdziale 2.2. Selekcja zmiennych sprawia, że pozostają tylko te, które mają zdolność naturalnego podziału zbioru obserwacji na grupy. Wykluczenie zbędnych zmiennych zwykle poprawia wyniki wykrywania ano-

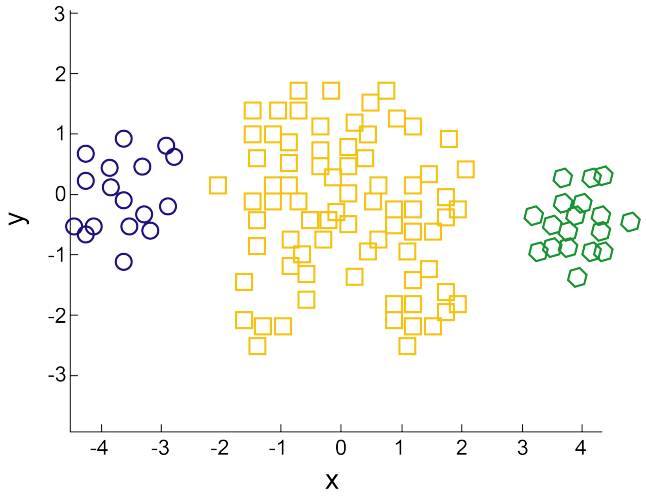
malii, gdyż mniejsza liczba zmiennych może lepiej ujawniać strukturę anomalii. Jednakże w praktyce każde z tych podejść ma swoje wady. Nadanie wag zmiennym nie zmniejsza rozmiaru zbioru danych, co oznacza, że powtarzające się informacje w różnych zmiennych mogą nadal występować. Ponadto, zmienne z przypisanymi wagami są trudniejsze do interpretacji. Z kolei usuwanie zmiennych lub redukcja ich wag może prowadzić do zmiany struktury danych, utrudniając wykrycie innych istotnych zmiennych. W procesie selekcji zmiennych celem jest optymalizacja, co wiąże się z minimalizacją liczby zmiennych niezbędnych do skutecznego wykrywania anomalii. Niemniej jednak, usunięcie jednej zmiennej może znacząco wpłynąć na całą strukturę grup, szczególnie gdy zbiory zmiennych są ze sobą powiązane i dane mogą zawierać wiele różnych struktur anomalii. Zastępowanie zmiennych nowymi, „sztucznymi" zmiennymi może skutkować identyfikacją anomalii trudnych do interpretacji, zwłaszcza gdy silne zależności między zmiennymi nie odpowiadają istniejącej strukturze anomalii. Dodatkowo, redukcja zmiennych może czasami prowadzić do utraty niektórych istotnych anomalii. Z tych powodów, metody analizy anomalii dążą do oceny zdolności poszczególnych zmiennych do wykrywania anomalii w zbiorze danych. Ze względu na praktyczną niemożliwość analizy wszystkich możliwych kombinacji zmiennych, stosuje się różne uproszczone techniki. Szczegółowe analizy porównawcze wielu technik można znaleźć w literaturze, w tym w pracach [193], [192], [194], [195].

Wizualizacja danych wielowymiarowych odgrywa ważną rolę na każdym etapie procesu odkrywania wiedzy. Dzięki wizualizacji możliwe jest zidentyfikowanie struktury danych, co może sugerować istnienie grup lub anomalii. Wizualizacja może być przeprowadzona zarówno na podstawie wartości oryginalnych zmiennych, jak i macierzy odległości między zmiennymi. Nawet proste histogramy mogą okazać się niezwykle przydatne w ujawnieniu struktury danych oraz identyfikacji obiektów odstających. Na rysunku 3.3 zaprezentowano przykładowy histogram, który wyraźnie ukazuje rozkład bimodalny, sugerując obecność dwóch grup w danych oraz potencjalnych anomalii. Graficzną interpretację korelacji stanowi wykres rozrzutu, który sam w sobie może dostarczyć istotnych informacji. Przykładowo, na rysunku 3.4 widzimy wyraźnie trzy naturalne grupy. Dzięki wizualizacji danych możemy dojść do takiego wniosku bez większego wysiłku. Jednakże, nie zawsze obecność grup lub anomalii będzie tak oczywista. W przypadku wykresów rozrzutu w wielu zbiorach danych, tylko część grup lub anomalii będzie wyraźnie widoczna, podczas gdy pozostałe będą ukryte i wykres taki będzie wymagał dodatkowej analizy, aby ujawnić pełną strukturę danych. Często dowody na istnienie skupień obserwacji lub anomalii można uzyskać, poprawiając proste wykresy danych poprzez odpowiednie oszacowanie gęstości. Oprócz podstawowych wykresów dwuwymiarowych, istnieje wiele zaawansowanych technik wizualizacji, które mogą znacząco poprawić zrozumienie danych oraz identyfikację anomalii. Wielowymiarowe wykresy rozrzutu, trójwymiarowe wykresy zmiennych, metody skalowania wielowymiarowego oraz sieci Kohonena to tylko niektóre z narzędzi, które umożliwiają graficzne przedstawienie danych w przestrzeni o mniejszej



Rysunek 3.3: Histogram bimodalny - wyraźnie widoczne dwie grupy. Źródło: opracowanie własne.

liczbie wymiarów. W przypadku zbiorów danych różnego typu, takich jak ilościowe i jakościowe, niezbędne są bardziej wyspecjalizowane metody wizualizacji. Przykładem mogą być dendrogramy, które wymagają zastosowania algorytmów grupowania. Te techniki nie tylko pomagają w identyfikacji ukrytych wzorców, struktur oraz anomalii, ale także umożliwiają lepsze zrozumienie i interpretację złożonych zbiorów danych. W kontekście



Rysunek 3.4: Wykres rozrzutu. Źródło: opracowanie własne.

rozpoznawania wzorców kryterium charakteryzacji jest redukcja błędów klasyfikacyjnych. Aby to osiągnąć, zazwyczaj niezbędne jest zwiększenie statystycznej zależności między docelową klasą a rozkładem danych w wybranej podprzestrzeni. Można to zrealizować

poprzez wybór cech o największej istotności dla zadania identyfikacji anomalii. W przypadku wykrywania odchyleń, klasa docelowa może reprezentować anomalie. Decydującym aspektem jest więc skonstruowanie odpowiednich niskowymiarowych reprezentacji danych poprzez wybór cech, które są istotne dla zadania identyfikacji anomalii. W związku z tym, selekcja cech jest procesem wyboru podzbioru cech na podstawie określonych kryteriów, mającym na celu eliminację cech nieistotnych, redundantnych lub zakłócających. Jest to istotny temat badawczy w eksploracji danych, ponieważ przyspiesza uczenie przez redukcję wymiarowości i usuwanie szumów. W dziedzinie selekcji cech wyróżniamy dwa podejścia: metodę opakowującą (*ang. wrapper*) i metodę filtracyjną (*ang. filter*), jak przedstawiono w tabeli 3.3. Metoda filtrująca zyskała szczególną popularność dzięki połączeniu jej z zastosowaniem informacji wzajemnej MI, która umożliwia precyzyjną ocenę relewancji cech w kontekście klasy docelowej oraz efektywną redukcję redundancji między cechami. W podrozdziale 3.1 przypomniano koncepcję informacji wzajemnej MI, omawiając algorytm k-ANMI.

Tabela 3.3: Opis metod wrapper i filter. Źródło: opracowanie własne na podstawie [31].

Nazwa metody	**Opis działania**
Opakowująca (wrapper)	Metoda opakowująca dla selekcji cech polega na współpracy z klasyfikatorem w celu minimalizacji błędu klasyfikacji konkretnego klasyfikatora. Proces ten można realizować na dwa sposoby: selekcja cech do przodu (dodawanie cech pojedynczo, zaczynając od pustego zbioru) oraz eliminacja cech do tyłu (usuwanie cech pojedynczo, zaczynając od pełnego zbioru). Metody wrapper zazwyczaj osiągają wysoką dokładność klasyfikacji dla konkretnego klasyfikatora, ale są obciążone dużą złożonością obliczeniową i ograniczoną generalizacją wybranych cech dla innych klasyfikatorów.
Filtrująca (filter)	Metoda filtrująca polega na wyborze cech poprzez testowanie, czy spełniają one ustalone warunki w odniesieniu do cech i klasy docelowej. Metody filtracyjne mają znacznie niższą złożoność obliczeniową niż metody opakowujące i są bardziej uniwersalne. Wybrane cechy w metodzie filter powinny zapewniać porównywalne wyniki klasyfikacji dla różnych klasyfikatorów.

Tabela 3.4: Algorytmy selekcji cech analizujące zależnie i niezależnie istotność oraz redundancję cech. Źródło: opracowanie własne na podstawie [31].

Nazwa algorytmu	**Opis działania**
mRMR *Minimal-Redundancy-Maximal-Relevance*	Metoda oparta na kryterium minimalnej redundancji i maksymalnej relewancji, mająca na celu wybór najlepszych atrybutów według maksymalnego kryterium zależności statystycznej, bazującego na informacji wzajemnej. Algorytm wybiera atrybuty, które są najbardziej istotne (*ang. max-relevance*) i jednocześnie minimalizuje redundancję (*ang. min-redundancy*), aby wybrać niezależne atrybuty. Proces selekcji atrybutów jest realizowany stopniowo, wybierając jeden atrybut na raz, co maksymalizuje różnicę między relewancją a redundancją. Po wybraniu zbioru $S_{A_{j-1}}$, składającego się z (j - 1) atrybutów, kolejną wybiera się cechę, która maksymalizuje funkcję celu: $$G = I(A_j; c) - \frac{1}{j-1} \sum_{A_s \in S_{A_{j-1}}} I(A_j; A_s)$$ To wyrażenie oblicza wartość funkcji celu G dla każdego atrybutu A_j, aby określić, który atrybut powinien zostać dodany do zestawu wybranych atrybutów w danym kroku.
NMIFS-OD *Normalized Mutual Information-based Feature Selection for Outlier Detection*	Algorytm NMIFS-OD ma na celu wybór podzbioru atrybutów $A_s \subset A$, które są istotne dla wykrywania anomalii i mają minimalną redundancję. Algorytm filtrujący ocenia istotność i redundancję atrybutów niezależnie, przy czym priorytet nadawany jest istotności. Proces selekcji atrybutów odbywa się przyrostowo, aż średnia redundancja wśród wybranego podzbioru A_s osiągnie ustalony próg lub wszystkie atrybuty zostaną przetworzone. Kroki obliczeniowe obejmują obliczenie entropii $H(A_j)$ dla każdego atrybutu, średniej redundancji $AAR(A)$, przypisanie progu redundancji $thresh$, uporządkowanie atrybutów według wartości entropii, przyrostowe dodawanie atrybutów do A_s oraz zwrócenie zbioru A_s z wybranymi atrybutami.

W tabeli 3.4 przedstawiono zaawansowane algorytmy selekcji cech, które analizują zarówno ich istotność, jak i redundancję, stosując różne podejścia. W algorytmie mRMR ocena relewancji cech opiera się na mierzeniu informacji wzajemnej MI między cechami a klasą docelową, natomiast redundancja jest oceniana na podstawie informacji wzajemnej między samymi cechami.Takie podejście umożliwia wybór cech o niskiej redundancji i wysokiej relewancji, czyli istotności danej cechy dla rozwiązania problemu. Z kolei algorytm NMIFS-OD, wykorzystujący znormalizowaną informację wzajemną NMI (*ang. normalized mutual information*, NMI), ocenia relewancję cech na podstawie ich wartości entropii. Niższa entropia wskazuje na wyższą relewancję dla wykrywania anomalii. Obliczanie redundancji za pomocą NMI pozwala na ocenę stopnia skorelowania cech, co ułatwia wybór cech o niskiej redundancji. Więcej informacji na temat tych metod można znaleźć w literaturze naukowej, w tym w pracach [196, 197, 31].

3.5.2 Metody analizy podprzestrzennej w wykrywaniu anomalii

Analiza podprzestrzenna w wykrywaniu anomalii to zaawansowana metoda, która skupia się na identyfikacji anomalii poprzez eksplorację różnych podprzestrzeni cech. Obiekt może odbiegać od normy w jednej podprzestrzeni, czyli w podzbiorze atrybutów, a jednocześnie wydawać się całkowicie normalny w innych podprzestrzeniach. Głównym elementem tego podejścia jest dokładne porównanie wyników z różnych podprzestrzeni, zwłaszcza gdy mają one różne wymiary i skale odniesienia. Aby to osiągnąć, niezbędne są techniki kwantyfikacji skuteczności poszczególnych podprzestrzeni w ujawnianiu anomalii. Ze względu na różnorodność wyników pochodzących z różnych podprzestrzeni, konieczne jest łączenie wyników z wielu analiz podprzestrzennych. Analiza zespołowa, która polega na agregowaniu wyników z różnych metod, może zwiększyć wiarygodność detekcji anomalii poprzez redukcję błędów wynikających z odmiennych reprezentacji danych.

W analizie podprzestrzennej można zastosować różne metody, takie jak: **Algorytmy genetyczne** (*ang. genetic algorithms*) służące do wykrywania anomalii w niskowymiarowych podprzestrzeniach poprzez identyfikowanie regionów o niskiej gęstości. **Wyszukiwanie rzadkich podprzestrzeni na podstawie odległości** (*ang. distance-based outlying subspace*). Są to metody oparte na odległości, które identyfikują anomalie poprzez analizę odległości w podprzestrzeniach danych. **Feature bagging**, to metoda zespołowa, która łączy wyniki z różnych podprzestrzeni, aby poprawić dokładność wykrywania anomalii. **Grupowanie w podprzestrzeniach** (*ang. projected clustering*) wykorzystuje techniki grupowania do identyfikacji anomalii poprzez rozważanie podzbiorów wymiarów lub projekcji danych do niższych wymiarów w różnych podprzestrzeniach. **Lasy izolacyjne** (*ang. isolation forests*) są metodą opartą na losowych cięciach osiowych, które izolują punkty w podprzestrzeniach, aby zidentyfikować anomalie. **Losowe histogramy w podprzestrzeniach danych** (*ang. random histograms in data subspaces*) używają histogramów do analizy gęstości punktów w podprzestrzeniach w czasie liniowym.

Zalety i wady analizy pełnowymiarowej i podprzestrzennej

Analiza pełnowymiarowa, która bierze pod uwagę wszystkie dostępne cechy, może czasem okazać się korzystna, zależnie od konkretnego przypadku i natury danych. Główne zalety pełnowymiarowej analizy wynikają z faktu, że każda zmienna niesie istotną informację. Wszystkie wymiary są potrzebne do prawidłowego zrozumienia zjawiska, a ignorowanie istotnych cech mogłoby prowadzić do błędnych wniosków. Użycie wszystkich dostępnych zmiennych zmniejsza ryzyko pominięcia cech, które mogą być istotne do wykrywania anomalii. Ponadto, relacje między wieloma wymiarami mogą wspólnie ujawniać wzorce anomalii, które nie byłyby widoczne przy analizie ograniczonej liczby wymiarów. Jednakże, analiza pełnowymiarowa może wprowadzać wiele szumów, które mogą maskować istotne anomalie.

Analiza podprzestrzenna natomiast pozwala skupić się na tych wymiarach, które są najbardziej istotne, redukując wpływ szumów i umożliwiając łatwiejsze wykrycie anomalii. Anomalie mogą być widoczne tylko w niektórych podprzestrzeniach danych, podczas gdy w pełnej wymiarowości mogą być maskowane. Analiza podprzestrzenna umożliwia wykrycie lokalnych anomalii, które byłyby niewidoczne w pełnowymiarowej analizie. Redukcja wymiarowości danych prowadzi do mniejszej liczby obliczeń, co może znacznie przyspieszyć proces analizy, szczególnie w przypadku dużych zbiorów danych. Dodatkowo, analiza podprzestrzenna pozwala na identyfikację istotnych podzbiorów cech, które są najbardziej związane z anomaliami, co jest użyteczne w dalszych analizach i modelowaniu. Analiza ta jest również bardziej odporna na brakujące dane, umożliwiając przeprowadzenie analizy na dostępnych podzbiorach danych, co jest korzystne w wielu rzeczywistych aplikacjach, gdzie pełne dane nie zawsze są dostępne.

Wykrywanie anomalii w podprzestrzeniach z natury zorientowe na zespoły

Pomijanie najważniejszych zmiennych może prowadzić do poważniejszych problemów niż uwzględnianie nieistotnych cech. Wyobraźmy sobie analizę danych zawierających 100 zmiennych, gdzie tylko 5 z nich jest istotne dla wykrywania anomalii. Jeśli te istotne zmienne zostaną pominięte, wyniki analizy mogą być poważnie zniekształcone, zwłaszcza gdy ważne cechy stanowią tylko mały procent całkowitej liczby danych. Częstym błędem jest zakładanie, że metody używane do grupowania danych mogą być bezpośrednio stosowane do wyboru lokalnych podprzestrzeni dla wykrywania anomalii. Metody grupowania łączą podobne obiekty danych, podczas gdy wykrywanie anomalii polega na identyfikacji obiektów odstających. Adaptowanie metod wyboru zmiennych z wcześniejszych badań nad grupowaniem podprzestrzennym, bez uwzględnienia specyfiki analizy podprzestrzeni może prowadzić do przeoczenia istotnych anomalii. Identyfikacja kluczowych podprzestrzeni zmiennych jest niezbędna do wykrywania anomalii. Niektóre anomalie mogą być widoczne tylko w specyficznych kombinacjach zmiennych, a ich identyfikacja wymaga zaawansowanej analizy. Wybór jednej istotnej podprzestrzeni dla każdego punktu danych może prowadzić do nieprzewidywalnych wyników, dlatego lepiej jest łączyć wyniki z wielu podprzestrzeni. Analiza danych na różnych poziomach szczegółowości i łączenie

wyników może przynieść bardziej wiarygodne rezultaty niż skupienie się wyłącznie na jednej kombinacji zmiennych. W skrócie, wykrywanie anomalii w podprzestrzeniach wymaga zespołowego podejścia, które integruje różne metody i perspektywy, aby skutecznie identyfikować istotne anomalie w danych.

Oto krótkie streszczenie wybranych metod detekcji anomalii. Bardziej szczegółowe informacje można znaleźć w literaturze naukowej wskazanej przy każdej z metod:

Algorytmy genetyczne w wykrywaniu anomalii

Algorytmy te identyfikują subprzestrzenne anomalie poprzez znajdowanie lokalnych regionów danych w niskowymiarowych przestrzeniach, które charakteryzują się wyjątkowo niską gęstością [198, 199, 200, 27]. Algorytmy genetyczne są wykorzystywane do odkrywania takich lokalnych regionów subprzestrzennych, naśladując proces ewolucji biologicznej, w którym każda możliwa subprzestrzeń jest traktowana jako indywidualny organizm. Organizmy te konkurują ze sobą, a ich przystosowanie (*ang. fitness*) jest oceniane na podstawie wartości funkcji celu, która mierzy rzadkość (*ang. sparsity*) subprzestrzeni. Algorytmy te nie tylko identyfikują anomalie, ale także wskazują na subprzestrzenie, które są odpowiedzialne za anomalię danego obiektu danych, co zwiększa interpretowalność wyników. Algorytmy genetyczne są szczególnie użyteczne w przypadkach, gdy kombinacje wymiarów, które prowadzą do odkrycia anomalii, są trudne do przewidzenia i wymagają zaawansowanej eksploracji przestrzeni możliwych rozwiązań.

Wyszukiwanie rzadkich podprzestrzeni na podstawie odległości

Metoda HOS-Miner (*ang. high-dimensional outlying subspaces-miner*, HOS-Miner) [201] jest jednym z wczesnych podejść do wykrywania odstających podprzestrzeni na podstawie odległości. Definiuje ona odstającą podprzestrzeń dla danego obiektu danych jako zbiór podprzestrzeni, w których suma odległości k-najbliższych sąsiadów przekracza określoną wartość δ. W tej metodzie wykorzystuje się drzewo X-Tree do efektywnego indeksowania zapytań k-najbliższych sąsiadów w różnych podprzestrzeniach. Aby zwiększyć efektywność procesu wyszukiwania, metoda ta stosuje losową próbkę danych, co pozwala na oszacowanie i dynamiczną aktualizację całkowitego czynnika oszczędności TSF (*ang. total saving factor*, TSF) odstających podprzestrzeni. TSF jest wskaźnikiem, który pomaga w szybkim eliminowaniu podprzestrzeni, które nie zawierają anomalii. W HOS-Miner, zastosowanie drzewa X-Tree do indeksowania wysokowymiarowych danych umożliwia szybkie przeszukiwanie i znajdowanie najbliższych sąsiadów. Algorytm ten dodatkowo wykorzystuje mechanizmy przycinania w górę i w dół, co pozwala na szybkie eliminowanie podprzestrzeni, w których punkt nie może być odstający, znacznie redukując przestrzeń przeszukiwania. Dzięki właściwościom domknięcia, które eliminują nieistotne podprzestrzenie, metoda HOS-Miner skutecznie identyfikuje minimalne, odstające podprzestrzenie, co umożliwia efektywne wykrywanie anomalii w danych o wysokiej wymiarowości. Ponadto, moduł filtracji wyników pomaga usunąć nadmiarowe odstające podprzestrzenie, dostarczając bardziej przejrzyste i zrozumiałe wyniki dla użytkowników. W artykule [202] autorzy definiują nowe zadanie wykrywania anomalii w danych o wysokiej wymiarowości,

polegające na identyfikacji podprzestrzeni, w których konkretne obiekty danych są obserwacjami odstającymi, i przedstawiają nowy algorytm do tego celu, nazwany HighDOD (*ang. high-dimensional outlier detection*, HighDOD).

Feature bagging

Agregacja cech, znana jako *feature bagging*, jest prostą i skuteczną metodą łączenia wyników wykrywania anomalii z różnych podprzestrzeni danych [203, 27]. Technika ta polega na scalaniu wyników wielu algorytmów wykrywania anomalii, które działają na różnych, losowo wybranych podzbiorach cech z oryginalnego zestawu. Każdy algorytm analizuje tylko mały podzbiór cech, co sprawia, że każdy detektor identyfikuje inne anomalie i przypisuje wyniki, które odpowiadają prawdopodobieństwu bycia anomalią dla każdego obiektu. Następnie, wyniki z poszczególnych algorytmów są łączone, aby zidentyfikować anomalie o wyższej jakości. Istnieją dwa proponowane podejścia do łączenia wyników identyfikacji anomalii: podejście szerokie BFS (*ang. breadth-first search*, BFS) oraz podejście z sumą skumulowaną. W podejściu BFS rankingi algorytmów są wykorzystywane do kombinacji wyników, przy czym najlepiej oceniane anomalie z różnych iteracji są umieszczane na pierwszych miejscach. W podejściu z sumą skumulowaną wyniki anomalii z różnych algorytmów są sumowane, a najlepiej oceniane anomalie są identyfikowane na tej podstawie. Chociaż losowe próbkowanie podprzestrzeni nie optymalizuje odkrywania istotnych podprzestrzeni, to jednak jest stosunkowo wydajne i może poprawić odporność na szum poprzez próbkowanie dużej liczby podprzestrzeni. Dzięki temu kombinacja wyników zespołu detektorów może zidentyfikować wszystkie punkty, które są wyróżniające się w wystarczającej liczbie podprzestrzeni.

Ta metoda jest znakomitym przykładem techniki zespołowej, która oferuje skuteczny sposób na przekształcenie globalnej analizy podprzestrzeni w lokalny wybór podprzestrzeni, co zapewnia naturalną przewagę nad pojedynczymi komponentami. Metody zespołowe zostały szczegółowo omówione w rozdziale 5. W artykule [204] wyjaśniono, dlaczego zespoły działają skuteczniej. Ponieważ techniki zespołowe wykorzystano w badaniach w tej rozprawie, przyjrzyjmy się im bliżej. Główne powody tej skuteczności podane przez autorów to: po pierwsze, powód statystyczny: algorytm uczenia przeszukuje przestrzeń hipotez H w celu znalezienia najlepszego przybliżenia funkcji f. Funkcja f to docelowa funkcja, którą algorytm uczenia maszynowego stara się przybliżyć. Przy ograniczonej ilości danych treningowych wiele hipotez może wykazywać podobną dokładność. Tworząc zespół z tych hipotez, algorytm może znacząco zmniejszyć ryzyko wyboru niewłaściwej hipotezy, co poprawia ogólną efektywność modelu. Po drugie, powód obliczeniowy: wiele algorytmów uczenia stosuje lokalne przeszukiwanie, które może utknąć w lokalnych minimach. Zespół utworzony przez uruchomienie lokalnych przeszukiwań z różnych punktów startowych może dostarczyć lepsze przybliżenie funkcji f niż jakikolwiek pojedynczy klasyfikator. Po trzecie, powód reprezentacyjny: funkcja f może nie być dokładnie reprezentowana przez żadną z hipotez w przestrzeni H. Przestrzeń hipotez H obejmuje wszystkie możliwe funkcje h, które algorytm uczenia może wybrać jako swoje przybliżenie funkcji f. Każda

hipoteza h w przestrzeni H jest potencjalnym modelem, który może być trenowany na danych. Tworząc ważone sumy hipotez, można rozszerzyć przestrzeń reprezentowanych funkcji, co pozwala na lepsze modelowanie i skuteczniejszą predykcję. Te trzy powody pokazują, dlaczego techniki zespołowe są potężnym narzędziem w uczeniu maszynowym i mogą znacząco poprawić wyniki w porównaniu do pojedynczych klasyfikatorów.

Grupowanie w podprzestrzeniach

Grupowanie w podprzestrzeniach (*ang. projected clustering*) można zdefiniować jako zaawansowaną technikę analizy danych, która identyfikuje zbiory obiektów w danych wielowymiarowych, jednocześnie wyodrębniając podzbiory wymiarów, w których te obiekty tworzą zwarte grupy. W odróżnieniu od tradycyjnych metod grupowania, które analizują całą przestrzeń atrybutów, grupowanie podprzestrzenne skupia się na odnajdywaniu grup widocznych tylko w specyficznych podprzestrzeniach danych. Dzięki temu możliwe jest skuteczniejsze wykrywanie ukrytych struktur, które mogłyby umknąć przy zastosowaniu konwencjonalnych metod. Proces ten polega na iteracyjnym przeszukiwaniu różnych kombinacji wymiarów, oceniając zwartość grup obiektów w każdej z tych kombinacji. Następnie wyodrębnia się te podzbiory wymiarów, w których obserwowane grupy są najbardziej wyraźne. Takie podejście umożliwia dynamiczne dostosowanie się do charakterystyki danych, co jest szczególnie przydatne przy analizie dużych i złożonych zbiorów danych. Proponowany zestaw algorytmów grupowania podprzestrzennego w różnych wymiarach przestrzeni danych jest szeroki i różnorodny, jak dowodzą liczne prace badawcze [205, 206, 207, 208, 209, 210, 211, 212, 213, 214]. Te metody tworzą solidnie ugruntowany obszar badań, charakteryzujący się wydajnymi schematami przetwarzania oraz różnorodnymi modelami grupowania, które uwzględniają specyficzne wymagania aplikacyjne. Grupowanie podprzestrzenne identyfikuje zestawy istotnych atrybutów dla każdej grupy, co umożliwia analizę danych z wielu perspektyw oraz grupowanie każdego obiektu w różnych podprzestrzeniach.

Chociaż istotne podprzestrzenie dla grup nie zawsze odgrywają decydującą rolę w wykrywaniu anomalii, istnieje między tymi zadaniami pewna zależność. Wykorzystanie zespołów algorytmów może znacznie zwiększyć skuteczność wykrywania różnych typów anomalii. Oto niektóre badania dotyczące wykrywania anomalii w podprzestrzeniach, których jest zdecydowanie mniej niż tych poświęconych samemu grupowaniu [33, 215, 216]. W osobnej pracy [217] zaproponowano algorytm OutRank (*ang. outlier ranking via subspace analysis in multiple views of the data*, OutRank), który wykorzystuje zespoły algorytmów grupowania projekcyjnego jako efektywne narzędzie do wykrywania anomalii w podprzestrzeniach. W OutRank wykorzystuje się algorytm grupowania projekcyjnego, taki jak PROCLUS (*ang. PROjected CLUStering*, PROCLUS) [211], który analizuje zbiór danych i tworzy grupy obiektów, podobnych do siebie w określonych wymiarach. Grupowanie podprzestrzenne oznacza, że każda grupa może być dobrze widoczna tylko w niektórych wymiarach danych, a niekoniecznie we wszystkich naraz. Dla każdego obiektu w danych oblicza się jego podobieństwo do grupy, do której został przypisany. Może to obejmować

różne miary, takie jak odległość punktu od środka grupy, wielkość grupy, czy liczba wymiarów, w których grupa jest dobrze widoczna. Cała procedura, obejmująca grupowanie i ocenę, jest powtarzana wielokrotnie, za każdym razem z elementem losowości w grupowaniu. Każde powtórzenie generuje pewien wynik dla każdego obiektu. Wyniki uzyskane z wielu powtórzeń są następnie uśredniane, co pozwala na określenie, jak bardzo każdy obiekt odstaje od typowych zachowań w danych. Użycie losowego algorytmu grupowania umożliwia odkrycie różnych aspektów struktury danych przy każdym powtórzeniu. Dzięki temu zwiększane są szanse na wykrycie wszystkich możliwych anomalii. Zastosowanie zespołu algorytmów polega na użyciu wielu wersji tego samego algorytmu, aby rozwiązać problem. Powtarzanie procesu wiele razy i uśrednianie wyników pozwala uzyskać bardziej stabilny i dokładny wynik. Przypomina to pytanie wielu ekspertów o opinię i branie pod uwagę wszystkich ich odpowiedzi, aby dojść do najlepszego wniosku.

Lasy izolacyjne

Lasy izolacyjne (*ang. isolation forests*), zaproponowane w pracy [218], to technika wykrywania anomalii, bazująca na koncepcji izolacji obiektów w danych. Choć metoda ta przypomina znane lasy losowe (*ang. random forests*), które są jednymi z najbardziej skutecznych modeli stosowanych w klasyfikacji, co pokazano w artykule [219] pod wymownym tytułem „Do we need hundreds of classifiers to solve real world classification problems?", różni się zasadniczo zarówno w konstrukcji, jak i w sposobie oceny obiektów. Lasy izolacyjne składają się z zespołu drzew izolacyjnych iTrees (*ang. isolation trees*, iTrees), które są budowane poprzez rekurencyjne dzielenie danych za pomocą losowo wybranych punktów podziału w losowo wybranych atrybutach. Proces ten trwa, aż każdy obiekt zostanie odizolowany w osobnym węźle. W odróżnieniu od tradycyjnych metod modelowych, które tworzą profil normalnych instancji, aby zidentyfikować anomalie, lasy izolacyjne koncentrują się na wyraźnym izolowaniu anomalii. Lasy izolacyjne mają silny związek z wykrywaniem anomalii w podprzestrzeniach. Poszczególne gałęzie drzewa odpowiadają różnym lokalnym regionom podprzestrzeni danych, w zależności od tego, które atrybuty zostały wybrane do podziału. Technika ta wykorzystuje fakt, że anomalie są „rzadkie" i „różne", dlatego anomalie łatwiej jest izolować, ponieważ wymagają mniej podziałów, aby zostały oddzielone od reszty danych. Normalne obiekty są trudniejsze do izolowania i zazwyczaj znajdują się głębiej w strukturze drzewa. Lasy izolacyjne opierają się na założeniu, że łatwiej jest izolować obiekty odstające w podprzestrzeniach o niższej wymiarowości, które powstają w wyniku losowych podziałów. Na przykład, w przypadku wykrywania oszustw finansowych, krótka sekwencja podziału, taka jak *Transakcja > 10 000 PLN* oraz *Typ transakcji = Zakupy luksusowe*, prawdopodobnie skutecznie izoluje rzadki przypadek oszukańczej transakcji. Proces dzielenia danych losowymi cięciami wzdłuż osi na przypadkowo wybranych obiektach podziału jest kontynuowany, aż wszystkie instancje zostaną odizolowane w węzłach liści. Lasy izolacyjne tworzą wiele drzew izolacyjnych, aby zwiększyć dokładność i stabilność wyników. Średnia długość ścieżki w różnych drzewach dla danego obiektu jest wykorzystywana do obliczenia jego wskaźnika

anomalii, który wskazuje, jak bardzo dany obiekt odbiega od normy w porównaniu do innych obiektów. Lasy izolacyjne są wydajną metodą wykrywania anomalii, o złożoności czasowej na poziomie $O(n \log n)$ i złożoności przestrzennej $O(n)$ dla każdego drzewa, gdzie n oznacza liczbę obiektów w zbiorze danych. Wykorzystanie podpróbkowania dodatkowo poprawia efektywność obliczeniową i różnorodność modeli. Podejście polegające na próbkowaniu zwiększa różnorodność i korzysta z pewnych zalet, które są wbudowane w zespołowe metody wykrywania anomalii [220].

Losowe histogramy w podprzestrzeniach danych do wykrywania anomalii

Histogramy podprzestrzenne, wprowadzone w pracy [221], jako prosty i szybki algorytm do wykrywania anomalii w podprzestrzeniach danych, bazują na losowym próbkowaniu oraz haszowaniu (*ang. hashing*). Algorytm ma liniową złożoność czasową i stałe wymagania pamięciowe, co czyni go niezwykle szybkim i wydajnym nawet dla bardzo dużych zbiorów danych. Podejście polega na tworzeniu losowych histogramów w podprzestrzeniach danych, a następnie uśrednianiu wyników, co zapewnia wysoką odporność i dokładność.

Dodatkowe techniki wykrywania anomalii w podprzestrzeniach

Oprócz opisanych metod analizy podprzestrzennej w wykrywaniu anomalii w danych wielowymiarowych, można w literaturze naukowej znaleźć jeszcze inne sposoby, takie jak: **wybór podprzestrzeni o wysokim kontraście** (*ang. selecting high-contrast subspaces*) [222], który polega na selekcjonowaniu podprzestrzeni o dużym kontraście, w których anomalie są bardziej widoczne. „Wysoki kontrast" odnosi się do subprzestrzeni danych, w których występuje znaczna niejednorodność rozkładu danych. Oznacza to, że pewne kombinacje wartości atrybutów w tych subprzestrzeniach są znacznie bardziej lub mniej prawdopodobne niż inne. **Lokalna selekcja istotnych podprzestrzeni** (*ang. local selection of subspace projections*) [216] polega na selekcji podprzestrzeni specyficznych dla danych obiektów w celu wykrycia anomalii. **Zbiory referencyjne oparte na odległości** (*ang. distance-based reference sets*) [215] wykorzystują zbiory referencyjne do identyfikacji anomalii w podprzestrzeniach o niskiej wariancji. Dla każdego obiektu X identyfikowany jest zbiór odniesienia obiektów $S(X)$, generowany jako k najbliższych obiektów. Podejście to analizuje lokalne zbiory danych, aby lepiej zrozumieć rozkład obiektów w danej podprzestrzeni.

3.5.3 Metody identyfikacji anomalii w złożonych podprzestrzeniach

W przestrzeniach wielowymiarowych często można spotkać przypadki, w których dane układają się wzdłuż konkretnych kierunków lub wzorców, niekoniecznie równoległych do osi współrzędnych. Na przykład w przestrzeni dwuwymiarowej (2D), gdy punkty są wyrównane wzdłuż jednej z osi (np. osi X), ich współrzędne Y pozostają stałe lub wykazują niewielką wariancję. Taki układ punktów można analizować za pomocą tradycyjnych metod wykrywania anomalii, które identyfikują nietypowe zachowania wzdłuż osi X i Y.

W bardziej złożonych przypadkach dane mogą układać się wzdłuż innych kierunków, np. ukośnych linii lub krzywych. W takich sytuacjach analizowanie danych jedynie wzdłuż osi X lub Y okazuje się niewystarczające. Aby prawidłowo zidentyfikować strukturę danych, niezbędne jest zastosowanie bardziej zaawansowanych metod, które są w stanie wykryć wzorce w bardziej złożonych podprzestrzeniach.

Złożona podprzestrzeń odnosi się do przypadków, gdy dane tworzą skomplikowane wzorce, które nie są prostymi liniami równoległymi do osi współrzędnych. W celu analizy takich złożonych podprzestrzeni stosuje się między innymi modele liniowe oparte na analizie składowych głównych PCA. Modele te identyfikują globalne regiony korelacji w danych, co ułatwia wykrywanie anomalii poprzez określenie dominujących kierunków korelacji.

3.6 Podsumowanie

W tym rozdziale omówiono zaawansowane metody wykrywania anomalii w danych kategorycznych i wielowymiarowych, zwracając uwagę na konieczność doboru odpowiednich technik i narzędzi. Skuteczna identyfikacja anomalii prowadzi do głębszego zrozumienia badanych danych, co ma istotne znaczenie w dziedzinach takich jak finanse, medycyna i analiza sieci społecznościowych. Wysoka wymiarowość danych stanowi wyzwanie dla tradycyjnych metod wykrywania anomalii, dotyczy to zarówno danych numerycznych, jak i kategorycznych. W takich przypadkach stosuje się bardziej zaawansowane techniki analityczne, takie jak modele generatywne i metody oparte na bliskości danych.

Modele generatywne, takie jak mieszaniny gaussowskie oraz algorytmy EM, pozwalają na precyzyjniejsze wykrywanie anomalii. Identyfikacja anomalii w danych kategorycznych często opiera się na analizie częstości wartości atrybutów. Dane wielowymiarowe mogą wymagać zastosowania technik takich jak wizualizacja danych czy analiza podprzestrzenna, które umożliwiają skupienie się na kluczowych wymiarach, redukując wpływ szumów i umożliwiając wykrycie lokalnych anomalii. Podejście zespołowe, integrujące różne metody i perspektywy, odgrywa ważną rolę w efektywnym wykrywaniu anomalii w podprzestrzeniach.

Rozdział ten podkreśla złożoność i różnorodność metod wykrywania anomalii w danych kategorycznych i wielowymiarowych oraz wagę właściwego zarządzania wyzwaniami związanymi z analizą danych o wysokiej wymiarowości. Właściwe techniki mogą znacząco poprawić skuteczność wykrywania anomalii, przyczyniając się do lepszej interpretacji i wykorzystania danych w praktyce.

Omówiono również analizę podprzestrzenną i jej znaczenie w kontekście wykrywania anomalii. Koncentruje się ona na identyfikacji anomalii poprzez eksplorację różnych podprzestrzeni cech, co wymaga zaawansowanych metod kwantyfikacji i agregacji wyników. Opisano techniki takie jak algorytmy genetyczne, wyszukiwanie rzadkich podprzestrzeni na podstawie odległości, feature bagging oraz grupowanie podprzestrzenne. Każda

z tych metod oferuje unikalne podejście, umożliwiając dokładniejsze wykrywanie anomalii w danych o wysokiej wymiarowości. Zwrócono również uwagę na metody wykrywania anomalii w złożonych podprzestrzeniach, gdzie struktury danych nie są prostymi liniami równoległymi do osi współrzędnych. W takich przypadkach konieczne są bardziej zaawansowane metody, zdolne do dostrzeżenia skomplikowanych wzorców.

Omówione techniki pokazują, jak różnorodne i skomplikowane są metody wykrywania anomalii oraz jak ważny jest wybór odpowiednich narzędzi w analizie danych kategorycznych i wielowymiarowych.

Rozdział 4

Uczenie maszynowe i głębokie: Trinity SALT w detekcji anomalii

W ostatnich latach techniki uczenia maszynowego, zwłaszcza głębokiego uczenia, zyskały ogromne uznanie w dziedzinie wykrywania anomalii. Zastosowanie sieci neuronowych, które naśladują działanie ludzkiego mózgu, umożliwia tworzenie zaawansowanych modeli do analizy danych. Historia ich rozwoju, począwszy od badań nad biocybernetyką, przez modele neuronów biologicznych, aż po współczesne sieci wielowarstwowe, ukazuje ewolucję tych technologii i ich rosnącą skuteczność w wykrywaniu anomalii.

W początkowych podrozdziałach 4.1, 4.2, 4.3 i 4.4 zaprezentowano koncepcje i metody związane z ewolucją sieci neuronowych oraz technikami uczenia maszynowego. Przedstawiono podstawy, które opisują rozwój tych rozwiązań, ich zastosowania oraz zalety w kontekście wykrywania anomalii, co stanowi wprowadzenie do systemu Trinity SALT. Pierwsza część opisuje historyczne tło i ewolucję sieci neuronowych, ilustrując rozwój technologii od badań nad biocybernetyką, przez modele neuronów biologicznych, aż po współczesne wielowarstwowe sieci neuronowe. W kolejnym podrozdziale wyjaśniono, jak głębokie uczenie wywodzi się z tradycyjnych metod uczenia maszynowego i jak zdołało rozszerzyć ich możliwości. Przeanalizowano różnice i zalety sieci neuronowych w porównaniu z tradycyjnymi metodami uczenia maszynowego. Podkreślono, że hierarchiczne modele głębokiego uczenia umożliwiają tworzenie bardziej złożonych i dokładnych reprezentacji danych, co jest niezbędne w identyfikacji anomalii. Następnie zaprezentowano prosty model perceptronu oraz pokazano, jak na jego bazie rozwinęły się bardziej złożone techniki, takie jak autoenkodery, które oferują wyższy poziom złożoności i skuteczności w wykrywaniu anomalii. Wskazano również, jak te metody ewoluowały i dlaczego okazały się skuteczne. W ostatniej części przeanalizowano zasady i wyzwania związane z pracą z systemami uczącymi się. Przedstawiono tutaj problemy takie jak nadmierne dopasowanie, optymalizacja, regularyzacja oraz zarządzanie hiperparametrami,

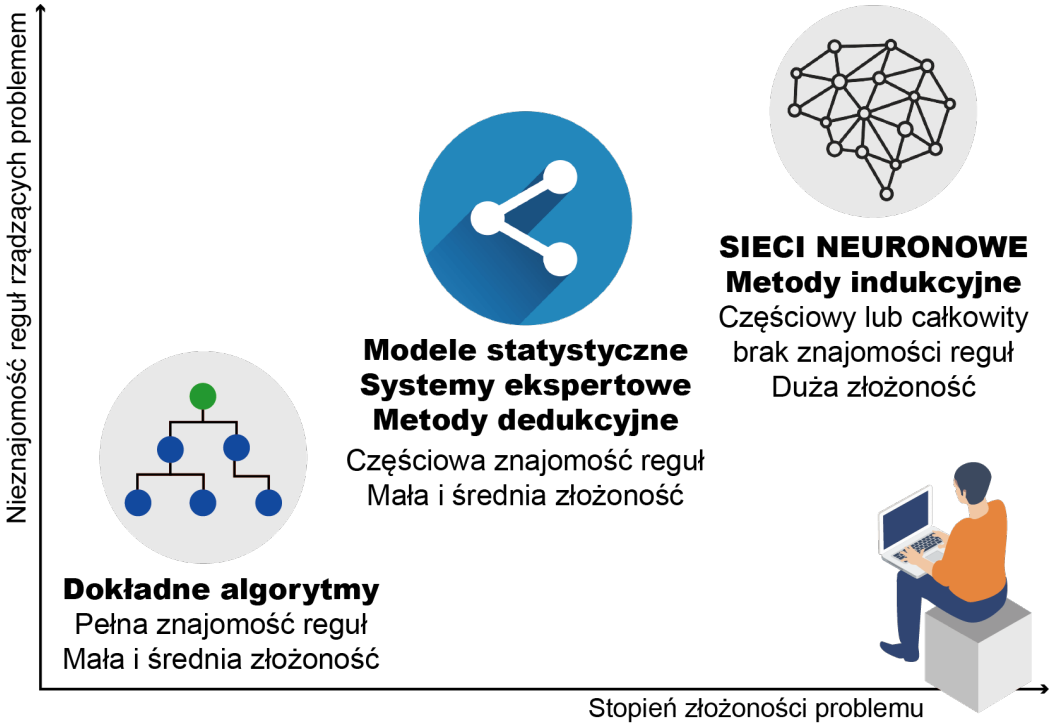
które są ważne dla skutecznego trenowania modeli i ich zdolności do generalizacji na nowych danych.

W dalszej części rozdziału, w podrozdziale 4.5, szczegółowo omówiono, jak wybrane algorytmy współpracują w ramach autorskiego systemu Trinity SALT. Algorytmy te zostały zastosowane w badaniach przedstawionych w tej rozprawie i razem tworzą zespół detekcji anomalii (*ang. outlier ensemble*). W ramach autorskiego rozwiązania połączono tradycyjne metody z zaawansowanymi technikami głębokiego uczenia, co umożliwiło stworzenie wartościowego narzędzia do analizy anomalii.

4.1 Geneza i rozwój sieci neuronowych

Sieci neuronowe, będące nowoczesnymi narzędziami do analizy i przetwarzania informacji, wywodzą się z badań nad biocybernetyką. Historia tej dziedziny zaczyna się w momencie, gdy naukowcy zaczęli intensywnie badać funkcjonowanie ludzkiego mózgu. Zainteresowanie tematem doprowadziło do licznych badań biologicznych, które dostarczyły istotnych informacji na temat struktur i mechanizmów działania mózgu. W wyniku tych odkryć, biocybernetycy stworzyli matematyczne modele, które opisywały działanie systemów nerwowych. W 1943 roku W.S. McCulloch i W. Pitts przedstawili model funkcjonowania pojedynczego biologicznego neuronu. Biologiczny neuron składa się z dendrytów, somy (ciała komórki), aksonów i szczeliny synaptycznej. Poszczególne elementy, mimo różnych nazw, mają swoje odpowiedniki w sztucznym neuronie [223]. Prace doprowadziły do powstania teoretycznych fundamentów, które miały znaczenie dla dalszych badań. Zastosowanie matematycznych opisów umożliwiło inżynierom stworzenie elektronicznych modeli struktur neuropodobnych, odwzorowujących procesy zachodzące w mózgu. Modele te stanowiły pierwszy krok w kierunku opracowania sieci neuronowych, zdolnych do działania na konwencjonalnych komputerach. W wyniku dalszego rozwoju technologicznego powstały sieci neuronowe, które obecnie mają szerokie zastosowanie w różnych dziedzinach, takich jak rozpoznawanie obrazów czy przetwarzanie języka naturalnego. Dzięki nieustannym innowacjom i badaniom sieci neuronowe stają się coraz bardziej zaawansowane i efektywne w rozwiązywaniu złożonych problemów analitycznych. W badaniach wykazano, że istnieje zbiór problemów informatycznych, najlepiej rozwiązywanych za pomocą sieci neuronowych, co ilustruje rysunek 4.1. Zadania te charakteryzują się tym, że jawna i świadoma wiedza badacza okazuje się niewystarczająca do znalezienia odpowiednich rozwiązań. Problemy tego typu często pojawiają się w kontekście identyfikacji anomalii, co czyni sieci neuronowe niezwykle wartościowym narzędziem dla badaczy i praktyków zajmujących się tą dziedziną [224].

W projektach z dziedziny sztucznej inteligencji dążono do stworzenia sformalizowanej bazy wiedzy o świecie, aby komputery mogły automatycznie generować argumenty za pomocą logicznych reguł wnioskowania. Podejście to, znane jako sztuczna inteligencja oparta na wiedzy, nie przyniosło jednak oczekiwanych rezultatów. Wybitnym przykładem jest



Rysunek 4.1: Porównanie sieci neuronowych z innymi metodami obliczeniowymi. Wykres przedstawia zależność między złożonością problemu (oś X) a stopniem nieznajomości reguł rządzących problemem (oś Y). Sieci neuronowe wyróżniają się swoją zdolnością do radzenia sobie z bardziej złożonymi problemami nawet przy braku znajomości reguł. Źródło: opracowanie własne na podstawie [224].

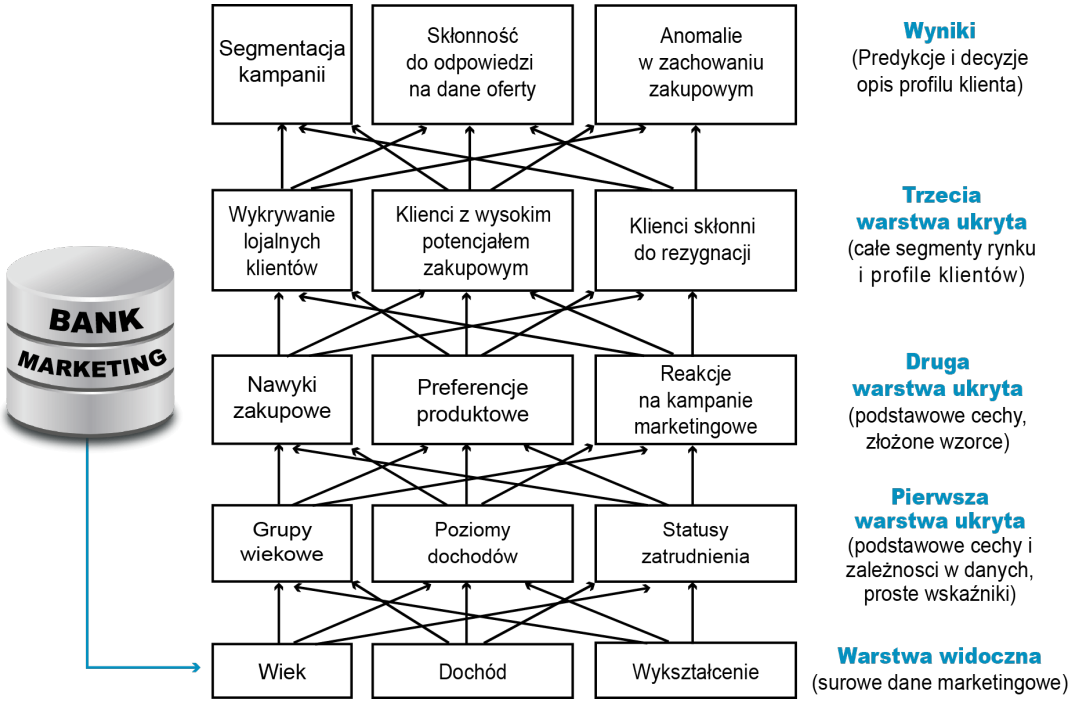
projekt Cyc [225], założony w 1989 roku przez Lenata i Guhę, mający na celu stworzenie rozbudowanej bazy wiedzy o zdrowym rozsądku dla systemów AI. Cyc korzysta z systemu wnioskowania i bazy danych opartej na języku CycL, ręcznie zasilanej przez ludzi. Mimo swojej złożoności, projekt napotkał na liczne trudności, które znacząco wpłynęły na jego realizację. Jednym z głównych wyzwań była skalowalność i spójność bazy wiedzy. Utrzymanie globalnej spójności ogromnej liczby faktów i reguł logicznych okazało się niezwykle trudne, co prowadziło do problemów z zarządzaniem sprzecznościami. Ręczne wprowadzanie wiedzy było czasochłonne i podatne na błędy, co spowalniało postęp. Projekt spotkał się również z krytyką zarówno ze strony zwolenników logiki i symbolicznych reprezentacji wiedzy, jak i z perspektywy innych podejść w AI, takich jak koneksjonizm [226, 50]. Sytuacja ta doskonale ilustruje wyzwania związane z implementacją sztywno zakodowanej wiedzy. Ograniczenia te podkreślają konieczność rozwoju samouczących się systemów AI, które potrafią adaptować się i wyodrębniać wiedzę bezpośrednio z danych, umożliwiając skuteczniejsze rozwiązywanie złożonych problemów. Przykładem prostego algorytmu samouczącego się jest naiwny klasyfikator Bayesa, który skutecznie

oddziela ważne e-maile od spamu. Wiele problemów z zakresu sztucznej inteligencji można rozwiązać, projektując odpowiedni zestaw cech dla danego zadania i dostarczając je prostemu algorytmowi uczącemu się. Jednak dla wielu zadań trudno jest wybrać właściwe cechy. W takich przypadkach automatyczne opanowanie reprezentacji danych może przynieść znacznie lepsze wyniki niż ręczne tworzenie cech. Tworzenie cech ręcznie dla skomplikowanych zadań jest niezwykle pracochłonne, często wymagające dekad wysiłku całych zespołów badawczych. Systemy oparte na wiedzy zmagały się z tym problemem, co ograniczało ich efektywność. Jednym ze sposobów rozwiązania tego problemu jest wdrożenie systemów uczących się, które uwzględniają nie tylko mapowanie reprezentacji na wyniki, ale również sposób, w jaki te reprezentacje są tworzone. Algorytmy uczenia się reprezentacji mogą znaleźć odpowiedni zestaw cech. To podejście znacząco zwiększa wydajność i skuteczność systemów sztucznej inteligencji, umożliwiając im lepsze radzenie sobie z różnorodnymi i skomplikowanymi problemami.

W projektowaniu cech i algorytmów uczących się dążymy do wyodrębnienia czynników wyjaśniających zmienność obserwowanych danych, które często są nieobserwowalnymi abstrakcjami. Głębokie sieci neuronowe wykazują wyjątkową wydajność w wychwytywaniu takich zmiennych, co odzwierciedla fakt, że ich biologiczne odpowiedniki również czerpią swoją siłę z wielowarstwowej struktury. Co więcej, sposób, w jaki biologiczne sieci są połączone, nie jest jeszcze w pełni zrozumiały. Jednak w przypadkach, gdzie osiągnięto pewien poziom zrozumienia ich struktury, zaprojektowanie sztucznych odpowiedników według tych samych zasad doprowadziło do znaczących przełomów.

Głębokie sieci neuronowe to hierarchiczne modele, które uczą się rozpoznawać obiekty na różnych poziomach szczegółowości, co umożliwia bardziej precyzyjne i niezawodne wnioskowanie. Na najniższym poziomie (pierwsze warstwy) sieć uczy się rozpoznawać podstawowe elementy, takie jak linie, krawędzie i kolory – są to fundamentalne elementy, z których zbudowane są bardziej skomplikowane obiekty. Na wyższych poziomach (kolejne warstwy) sieć łączy te proste elementy w bardziej złożone cechy, takie jak fragmenty obiektów – na przykład części twarzy, takie jak oczy czy nos. Najwyższe warstwy sieci koncentrują się na rozpoznawaniu całych obiektów na podstawie tych złożonych cech. Dzięki temu, że każda warstwa uczy się czegoś bardziej abstrakcyjnego niż poprzednia, sieć może rozpoznać obiekt jako całość, nawet jeśli niektóre jego części są zasłonięte lub trudno rozpoznawalne. Przykładem jest praca [227], która koncentruje się na hierarchicznych modelach, które uczą się rozpoznawać różne poziomy szczegółowości w danych. W pracy autorzy pokazali, jak sieć neuronowa rozpoznaje różne kategorie obiektów i zwierząt. Początkowo sieć nauczyła się identyfikować podstawowe cechy, takie jak linie i krawędzie, wspólne dla wszystkich kategorii. W kolejnych warstwach sieć nauczyła się, jak te proste kształty łączą się ze sobą, aby tworzyć bardziej złożone obrazy, na przykład oczy na twarzy lub koła samochodu. Praca autorów ilustruje te koncepcje za pomocą wizualnych przykładów, które przypominają sposób, w jaki ludzka kora wzrokowa przetwarza informacje, rozpoznając coraz bardziej złożone cechy, zaczynając od prostych linii i krawędzi.

Podobnie jest w przypadku innych danych, na przykład marketingowych. Na najniższym poziomie model może identyfikować podstawowe cechy, takie jak wiek, dochód czy wykształcenie klientów. Są to fundamentalne jednostki informacyjne, które tworzą podstawę do dalszej analizy. Na wyższych poziomach model może łączyć te proste cechy w bardziej złożone wzorce, takie jak profil klienta lub segment rynku. Na przykład, może łączyć wiek i rodzaj pracy, aby zidentyfikować grupy klientów o podobnym zachowaniu zakupowym. Najwyższe warstwy modelu mogą następnie skupić się na przewidywaniu zachowań klientów na podstawie tych złożonych wzorców, takich jak skłonność do zakupu konkretnego produktu w ramach kampanii marketingowej. Dzięki hierarchicznej strukturze, model może dokładnie przewidywać wyniki kampanii, nawet w przypadku nowych klientów, których dane nie były wcześniej analizowane, co zostało zilustrowane na rysunku 4.2.



Rysunek 4.2: Ilustracja modelu głębokiego uczenia. Głębokie uczenie radzi sobie z trudnością zrozumienia surowych danych marketingowych poprzez podział skomplikowanego odwzorowania na serię zagnieżdżonych, prostszych odwzorowań, z których każde jest reprezentowane w innej warstwie modelu. Dane wejściowe są wprowadzane do widocznej warstwy, nazwanej tak, ponieważ zawiera obserwowalne zmienne. Następnie przechodzą przez ciąg warstw ukrytych, które wyodrębniają coraz bardziej abstrakcyjne cechy. Warstwy te nazywane są ukrytymi, ponieważ ich wartości nie są bezpośrednio podane w zestawie danych, model musi samodzielnie określić, które koncepcje są przydatne do wyjaśnienia zależności w obserwowanych danych. Źródło: opracowanie własne.

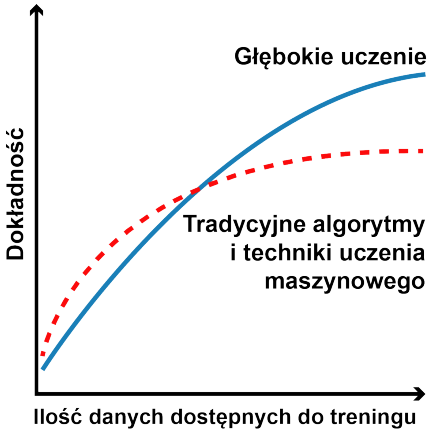
Podstawowa idea głębokiego uczenia polega na wykorzystaniu wielowarstwowych struktur sieci neuronowych, aby zmniejszyć liczbę potrzebnych neuronów. Działa to poprzez kompozycję funkcji, gdzie wyjście jednej warstwy staje się wejściem dla kolejnej. Dzięki temu, mimo zwiększenia liczby warstw, liczba parametrów do nauczenia jest mniejsza, co zwiększa efektywność sieci w uogólnianiu wiedzy na nowe dane. Głębsze sieci lepiej rozpoznają powtarzające się wzorce, co pozwala im skuteczniej rozwiązywać problemy i uczyć się na nowych danych. Te wzorce są zapisywane w wagach sieci jako wektory bazowe, co umożliwia sieci naukę bardziej abstrakcyjnych cech danych w miarę przechodzenia przez kolejne warstwy.

4.2 Od uczenia maszynowego do głębokiego uczenia

Wiele tradycyjnych modeli uczenia maszynowego można postrzegać jako szczególne przypadki uczenia się w ramach sieci neuronowych. Aby w pełni zrozumieć sieci neuronowe, należy najpierw zrozumieć relacje między nimi a klasycznym uczeniem maszynowym. Sieci neuronowe czerpią inspirację z mechanizmów biologicznych, gdzie jednostki obliczeniowe, zwane neuronami, są połączone wagami, naśladując sposób, w jaki neurony w mózgu łączą się i komunikują za pomocą synaps. Tradycyjne modele uczenia maszynowego, takie jak regresja liniowa czy regresja logistyczna, stanowią podstawowe jednostki obliczeniowe w sieciach neuronowych. Sieci neuronowe zyskują swoją moc przez łączenie wielu takich jednostek i uczenie się wag tych jednostek wspólnie, aby zminimalizować błąd predykcji. Gdy sieć neuronowa jest stosowana w swojej najprostszej formie, bez łączenia wielu jednostek, algorytmy uczenia często sprowadzają się do klasycznych modeli uczenia maszynowego. Jednak prawdziwa moc sieci neuronowych ujawnia się, gdy te podstawowe jednostki są połączone w sposób umożliwiający modelowi uczenie się bardziej złożonych funkcji na podstawie danych, co jest trudne do osiągnięcia za pomocą tradycyjnych metod uczenia maszynowego.

Sieci neuronowe oferują kilka istotnych zalet w porównaniu z tradycyjnymi metodami uczenia maszynowego. Po pierwsze, umożliwiają tworzenie bardziej zaawansowanych abstrakcji, co pozwala na głębsze i bardziej precyzyjne zrozumienie analizowanych informacji. Elastyczna architektura sieci neuronowych pozwala na lepsze modelowanie złożonych zależności w badanych zbiorach, co jest trudniejsze do osiągnięcia przy użyciu tradycyjnych algorytmów. Po drugie, sieci neuronowe oferują bardziej zautomatyzowany sposób dostosowywania modeli do skomplikowanych wzorców w badanych danych. Tradycyjne metody często wymagają ręcznego projektowania cech i predefiniowania struktury modelu, podczas gdy sieci neuronowe wykorzystują optymalizację, co pozwala na bardziej efektywne uczenie się i adaptację do skomplikowanych wzorców w zbiorach informacji. Jedną z ważnych zalet sieci neuronowych w porównaniu z tradycyjnymi metodami uczenia maszynowego jest ich elastyczność w dostosowywaniu złożoności modelu. Można to osiągnąć poprzez dodawanie lub usuwanie neuronów w architekturze sieci, w zależności

od dostępności danych szkoleniowych i mocy obliczeniowej. W ostatnich latach sukcesy sieci neuronowych wynikają głównie z rosnącej dostępności dużych zbiorów danych oraz zwiększonej mocy obliczeniowej nowoczesnych komputerów. Te czynniki pozwoliły sieciom neuronowym przewyższyć tradycyjne algorytmy uczenia maszynowego, które nie są w stanie w pełni wykorzystać współczesnych zasobów, co zilustrowano na rysunku 4.3.



Rysunek 4.3: W miarę zwiększania ilości danych dostępnych do trenowania modelu, sieci neuronowe (zwłaszcza głębokie sieci neuronowe) mają tendencję do przewyższania tradycyjnych algorytmów uczenia maszynowego pod względem dokładności. Źródło: opracowanie własne na podstawie [151].

Głębokie uczenie to poddziedzina uczenia maszynowego, która koncentruje się na wykorzystaniu głębokich sieci neuronowych do modelowania skomplikowanych wzorców w dużych zestawach danych. Głębokie sieci neuronowe składają się z wielu warstw przetwarzania, które pozwalają modelowi na hierarchiczne reprezentowanie danych, od prostych cech w niższych warstwach do bardziej złożonych abstrakcji w wyższych warstwach. Tradycyjne modele uczenia maszynowego, takie jak regresja liniowa czy logistyczna, są ograniczone do płaskich struktur, które modelują dane przy użyciu jednej warstwy przetwarzania. Głębokie sieci neuronowe natomiast wykorzystują wiele warstw, co pozwala na tworzenie hierarchicznych reprezentacji. To umożliwia modelom głębokiego uczenia uchwycenie bardziej skomplikowanych wzorców, które są trudne do zidentyfikowania przez tradycyjne metody. W tradycyjnych modelach uczenia maszynowego cechy (*ang. features*) są zazwyczaj ręcznie projektowane przez ekspertów dziedzinowych. W głębokim uczeniu proces ten jest zautomatyzowany - sieci neuronowe same uczą się odpowiednich cech podczas procesu treningu. Na przykład, w zadaniach związanych z przetwarzaniem obrazów, niższe warstwy sieci mogą uczyć się wykrywać krawędzie, a wyższe warstwy mogą rozpoznawać bardziej złożone struktury, takie jak części obiektów czy całe obiekty. Głębokie uczenie wykorzystuje techniki takie jak transfer uczenia, gdzie model trenowany na jednym zadaniu może być dostosowany do innego zadania. Na przykład, sieć neuronowa

przetrenowana na ogromnym zbiorze danych, takim jak ImageNet, który zawiera miliony oznakowanych obrazów przypisanych do tysięcy kategorii, może być następnie dostrojona do mniejszych, bardziej specyficznych zadań. Tradycyjne modele uczenia maszynowego rzadko wykorzystują tego typu transfer wiedzy. Zrozumienie, jak tradycyjne uczenie maszynowe łączy się z sieciami neuronowymi, jest niezbędne, ponieważ liczne klasyczne modele można traktować jako specyficzne warianty sieci neuronowych. Głębokie uczenie poszerza te koncepcje, wprowadzając hierarchiczne reprezentacje, automatyczne uczenie cech i techniki transferu uczenia, co pozwala na lepsze modelowanie skomplikowanych zależności.

Głębokie uczenie stanowi nowoczesne podejście w dziedzinie uczenia maszynowego. W odróżnieniu od klasycznych metod, takich jak regresja liniowa, regresja logistyczna, drzewa decyzyjne, SVM czy algorytmy zespołowe jak Random Forest, które operują na płytkich strukturach, głębokie sieci neuronowe składają się z wielu warstw. Tradycyjne metody skupiają się na ręcznym definiowaniu cech i prostych modelach, które działają na pojedynczych warstwach danych, co ogranicza ich zdolność do automatycznego wykrywania złożonych wzorców. Proces uczenia maszynowego w tradycyjnych metodach polega na trenowaniu modelu na dostępnych danych, aby umożliwić predykcję lub klasyfikację nowych informacji, w oparciu o wzorce odkryte w zestawie szkoleniowym. Głębokie uczenie, będące zaawansowaną formą uczenia maszynowego, wyróżnia się swoją zdolnością do automatycznego tworzenia hierarchicznych reprezentacji danych. Dzięki zwiększeniu liczby warstw i liczby jednostek w każdej warstwie, głęboka sieć neuronowa może modelować funkcje o coraz większej złożoności. W efekcie, głębokie sieci neuronowe są niezwykle efektywne w takich zadaniach jak rozpoznawanie obrazów, przetwarzanie języka naturalnego, analiza dźwięku oraz identyfikacja anomalii. Przykłady technik głębokiego uczenia to autoenkodery, sieci neuronowe LSTM i GAN. Autoenkodery są używane do kompresji informacji i identyfikacji anomalii poprzez analizę rekonstrukcji błędów. LSTM, zaprojektowane specjalnie do analizy sekwencji czasowych, są zdolne do uchwycenia długoterminowych zależności, co czyni je skutecznymi w wykrywaniu nietypowych wzorców w seriach czasowych. GAN natomiast generują nowe próbki danych, które są trudne do odróżnienia od rzeczywistych danych, co jest przydatne w identyfikacji anomalii poprzez modelowanie rozkładu danych.

Na koniec należy uświadomić sobie, iż sukcesy głębokiego uczenia, wynikające z postępów w technologii zbierania danych oraz efektywniejszego przetwarzania zbiorów danych, osiągnięto przy użyciu algorytmów jedynie nieznacznie zmodyfikowanych w porównaniu do wersji sprzed dwóch dekad. Mimo że sztuczne sieci neuronowe osiągnęły imponujące wyniki w wielu zadaniach, wciąż pozostają daleko w tyle za ludzkim mózgiem pod względem elastyczności, efektywności i zdolności do uczenia się z ograniczonych danych. Wciąż istnieje wiele do nauczenia się z biologicznych mechanizmów, które mogą prowadzić do dalszych postępów w dziedzinie sztucznej inteligencji. Mózg ludzki wykazuje wyjątkową wydajność w wielu aspektach uczenia się, przewyższając obecne technologie sztucznej

inteligencji. Ludzie i zwierzęta są zdolni do rozpoznawania obiektów, rozwiązywania problemów i wykonywania skomplikowanych zadań na podstawie zaledwie kilku przykładów. Z kolei sieci neuronowe często wymagają ogromnych ilości oznakowanych danych, aby działać efektywnie. Ponadto, mózg zużywa znacznie mniej energii niż współczesne komputery potrzebnej do trenowania i działania sieci neuronowych. Ludzie potrafią efektywnie wykorzystywać kontekst do rozumienia i przewidywania, co jest wyzwaniem dla obecnych modeli uczenia maszynowego. Mózg ludzki uogólnia na podstawie niewielkiej liczby przykładów, podczas gdy modele uczenia maszynowego mają trudności z uogólnianiem, jeśli nie są dostatecznie przetrenowane na różnorodnych danych.

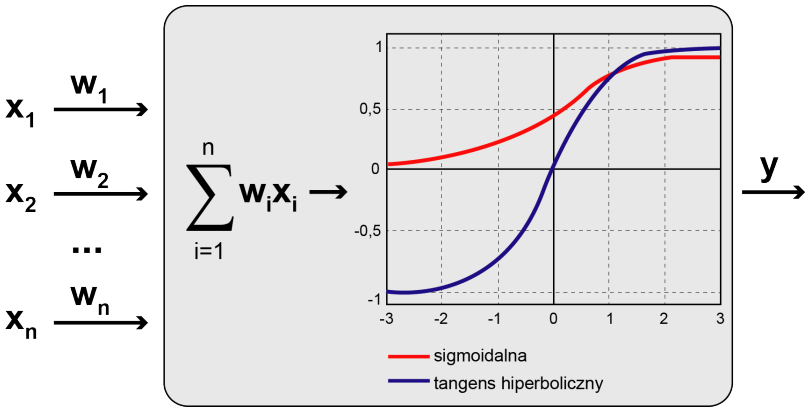
Istnieje wiele badań potwierdzających wyższość ludzkiego mózgu nad sztucznymi sieciami neuronowymi w różnych aspektach uczenia się i przetwarzania informacji. Na przykład, badanie [228] pokazuje, że ludzie mogą nauczyć się nowych pojęć z minimalnej ilości danych, co jest trudne do osiągnięcia dla obecnych algorytmów uczenia maszynowego, które zazwyczaj wymagają dużej ilości danych. Książka [229] opisuje, jak ludzki mózg jest wysoce efektywny energetycznie i zdolny do adaptacji w sposób, który sztuczne systemy wciąż próbują naśladować. Obecne modele głębokiego uczenia mają trudności z efektywnym wykorzystaniem kontekstu w sposób, w jaki robi to ludzki mózg [230]. Artykuł [231] podkreśla, że mimo doskonałego działania na danych treningowych, sieci neuronowe mają poważne problemy z generalizacją na nowe, nieznane rodzaje danych. W innej pracy [232] autorzy poruszają kwestię ograniczeń GPT-4 w porównaniu z ludzkim mózgiem, szczególnie pod względem zdolności do planowania, generalizacji i adaptacji. Wskazują, że choć GPT-4 ma imponujące zdolności w wielu zadaniach, to nadal ma istotne ograniczenia, które sprawiają, że nie dorównuje człowiekowi.

W rozprawie skorzystano z trzech podstawowych detektorów: AE, LOF i SOM. Autoenkoder jest reprezentatywnym przykładem technik głębokiego uczenia. Algorytm LOF, natomiast, jest klasyczną metodą uczenia maszynowego. Metoda ta może być stosowana samodzielnie lub jako część wstępnego przetwarzania danych w głębokim i tradycyjnym uczeniu maszynowym. Z kolei SOM są uznawane za część tradycyjnego uczenia maszynowego i zostały wprowadzone przed pojawieniem się współczesnych technik głębokiego uczenia. W przeciwieństwie do głębokich sieci, SOM zazwyczaj mają jednowarstwową strukturę neuronów ułożonych w kratownicę.

4.3 Anomalie: od perceptronów do autoenkoderów

Perceptron, wynaleziony przez Franka Rosenblatta w 1957 roku [233], jest fundamentalnym elementem sieci neuronowych, działającym na podobnej zasadzie co prosty model regresji liniowej. Model ten został zilustrowany na rysunku 4.4. W swoim artykule Rosenblatt pokazał, że perceptron, mimo swojej prostoty, ma zdolność do realizowania złożonych funkcji poznawczych, takich jak rozpoznawanie wzorców, uczenie się na podstawie prób i błędów oraz analizowanie wzorców czasowych. Jego konstrukcja miała na celu naślado-

wanie procesów zachodzących w mózgu, takich jak percepcja, rozpoznawanie wzorców, generalizacja i pamięć. W artykule Rosenblatt szczegółowo opisał model perceptronu i przeprowadził matematyczną analizę procesu uczenia się. Jego eksperymenty symulacyjne potwierdziły teoretyczne przewidywania dotyczące perceptronu. Algorytm perceptronu, uznawany za fundamentalny kamień węgielny sieci neuronowych, odegrał istotną rolę w rozwoju sztucznej inteligencji i neuroinformatyki. Jego zdolności pokazały, że nawet proste modele mogą być istotne w zrozumieniu i symulacji procesów poznawczych, co stanowiło ważny krok naprzód w tych dziedzinach. Połączenie wielu perceptronów prowadzi do stworzenia sieci wielowarstwowej, która pozwala na modelowanie dowolnych funkcji nieliniowych. Właśnie dlatego sieci neuronowe są często określane jako uniwersalne narzędzia do przybliżania funkcji, zdolne do symulacji skomplikowanych wzorców i procesów.



Rysunek 4.4: Perceptron. Źródło: opracowanie własne.

Perceptron zawiera dwie warstwy węzłów, które odpowiadają węzłom wejściowym i pojedynczemu węzłowi wyjściowemu. Liczba węzłów wejściowych odpowiada wymiarowi d danych. Niech $\mathrm{x} = (x_{i,1}, \ldots, x_{i,d})$ będzie d wejściami, które odpowiadają d wartościom cech rekordów danych. Wyjście perceptronu jest obliczane jako funkcja wejść (liniowa funkcja aktywacji) z odpowiednim wektorem wag $\mathrm{w} = (w_1, \ldots, w_d)$:

$$z = \mathrm{w} \cdot \mathrm{x} = \sum_{j=1}^{d} w_j x_{i,j} \tag{4.1}$$

Ogólnie można użyć dowolnej funkcji aktywacji i uwzględnić bias b:

$$z = \phi(\mathrm{w} \cdot \mathrm{x} + b) \tag{4.2}$$

Funkcja aktywacji ϕ jest często funkcją nieliniową, taką jak funkcje sigmoidalna lub tanh.

Możemy skonstruować jednoklasowe sieci neuronowe, w których wyjście sieci zawsze

wynosi zero, mimo że wagi są różne od zera. W jednoklasowym ustawieniu zakłada się, że wszystkie obiekty treningowe są normalne, więc przewidywanie z z równania 4.1 powinno wynosić 0, co jest zgodne z równaniem 4.3:

$$\sum_{j=1}^{d} w_j \cdot x_{i,j} = 0 \tag{4.3}$$

Dlatego każda niezerowa wartość z przewidziana przez jednoklasową sieć neuronową jest interpretowana jako oznaka anomalii, które nie pasują do modelu danych normalnych. Dla pojedynczego przykładu X_i, gdzie sieć neuronowa przewiduje z_i, błąd kwadratowy SE (*ang. squared error*, SE) dla obiektu i w naszym jednoklasowym podejściu jest określony następująco:

$$SE_i = z_i^2 = (\mathrm{w} \cdot \mathrm{x}_i)^2 \tag{4.4}$$

Aby skorygować ten błąd, konieczna jest aktualizacja wag sieci neuronowej. Proces ten realizuje się za pomocą metody gradientu. Aktualizacja ta może być wyrażona w następujący sposób:

$$\mathrm{w} \leftarrow \mathrm{w} - \eta \nabla SE_i = \mathrm{w} - \eta z_i \mathrm{x}_i, \tag{4.5}$$

gdzie:

- w to wektor wag,
- η to współczynnik uczenia się,
- ∇SE_i to gradient błędu kwadratowego dla punktu i,
- z_i to wyjście perceptronu dla punktu i (czyli $z_i = \mathrm{w} \cdot \mathrm{x}_i$),
- x_i to wektor cech dla punktu i.

Współczynnik uczenia się η jest dodatni ($\eta > 0$). Aby uniknąć trywialnego rozwiązania $\mathrm{w} = 0$, zaktualizowany wektor w jest skalowany do jednostkowej normy. W fazie treningowej sieci neuronowej perceptron przetwarza rekordy $X_1, \ldots, X_N$ o wymiarze d jeden po drugim, wykonując wspomniane aktualizacje wektora w aż do zbieżności. Cały ten proces jest wariantem stochastycznego spadku gradientu. Aby ocenić dany obiekt X_i, używamy wyuczonego modelu do obliczenia jego oceny odchylenia w następujący sposób:

$$\mathrm{Ocena}(X_i) = (\mathrm{w} \cdot \mathrm{x}_i)^2 \tag{4.6}$$

Odchylenia będą miały wyższe oceny. Model ten można również wykorzystać do oceny obiektów, które nie były częścią zbioru treningowego.

Aby umożliwić bardziej ogólne odwzorowania, można zastosować kolejne transformacje odpowiadające sieciom mającym kilka warstw wag adaptacyjnych. Proste modele można składać w bardziej złożone struktury. Wielowarstwowa sieć neuronowa z niewielką liczbą jednostek jest zdolna do modelowania dowolnego jednoklasowego rozkładu danych bez przyjmowania jakichkolwiek założeń na temat kształtu tego rozkładu. Takie sieci składają się z warstw wejściowych, ukrytych i wyjściowych, gdzie każda warstwa ukryta może być połączona w różnych topologiach, najczęściej w formie sieci o propagacji wprzód (*ang. feed-forward*). Sieci te mogą używać różnych nieliniowych funkcji aktywacji, takich jak $\phi(z) = \frac{e^{2z}-1}{e^{2z}+1}$ (funkcja tanh), $\phi(z) = \frac{1}{1+e^{-z}}$ (funkcja sigmoid), czy $\phi(z) = \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)$ (funkcja bazowa Gaussa).

W jednoklasowej sieci neuronowej proces treningowy jest prosty, ponieważ oczekiwana wartość wyjściowa neuronu wynosi 0. W warstwach ukrytych nie znamy jednak wyjść neuronów, co wymaga zastosowania algorytmu wstecznej propagacji błędów (*ang. backpropagation*). Algorytm ten propaguje błędy wstecznie, dostosowując wagi węzłów na podstawie wyjść z późniejszych warstw. Kluczowe jest skalowanie wag węzła wyjściowego do jednostkowej normy, aby uniknąć trywialnych rozwiązań, gdzie wszystkie wagi wynoszą 0. Wsteczna propagacja błędów to potężna i efektywna obliczeniowo metoda dla sieci z różniczkowalnymi funkcjami aktywacji. Umożliwia ona znajdowanie pochodnych funkcji błędu względem wag i biasów w sieci. Jest to ważna cecha takich sieci, ponieważ te pochodne odgrywają bardzo ważną rolę w większości algorytmów treningowych dla sieci wielowarstwowych. Aby uniknąć trywialnych transformacji, narzuca się ograniczenie $\mathrm{W}^T\mathrm{W} = \mathrm{I}$ dla macierzy wag W między warstwami ukrytymi, co wymaga bardziej zaawansowanych metod gradientu prostego z ograniczeniami. Zamiast jednego węzła wyjściowego, można użyć wielu węzłów wyjściowych. Redukcja liczby wymiarów reprezentacji z wieloma wyjściami polega na użyciu kilku węzłów wyjściowych zamiast jednego, aby skuteczniej zmniejszyć liczbę wymiarów reprezentacji danych. Wprowadza się ograniczenia, takie jak prostopadłość i normalizacja wag, aby zapewnić wzajemną ortogonalność i normalizację wektorów wagowych w warstwie wyjściowej. To sprawia, że proces optymalizacji staje się bardziej skomplikowany.

Głębokie sieci neuronowe pozwalają na modelowanie złożonych wzorców nieliniowych, a użycie wielu wyjść może dodatkowo zredukować liczbę wymiarów reprezentacji. Więcej informacji na ten temat można znaleźć w [234]. Poprzez zwiększenie liczby warstw w sieci i użycie różnych typów funkcji aktywacji, możemy modelować dowolnie złożone wzorce nieliniowe. Optymalizacja parametrów w złożonych sieciach neuronowych, które zawierają wiele warstw, wymaga użycia zaawansowanych metod matematycznych i algorytmicznych. Ten zestaw technik, znany jako głębokie uczenie, obejmuje różnorodne podejścia, takie jak propagacja wsteczna oraz metody gradientowe, które są niezbędne do skutecznego trenowania takich sieci [50].

W kontekście sieci neuronowych, możemy użyć specjalnych metod, aby określić, które obiekty są odstające. Jeśli sieć neuronowa ma pojedynczy węzeł wyjściowy (*ang. single*

output node), wynik dla danego obiektu danych X_i to z_i. Aby ocenić, czy X_i jest obiektem odstającym, oblicza się kwadrat tej wartości z_i^2. Im większa wartość z_i^2, tym bardziej obiekt jest uznawany za odstający. Jeśli sieć neuronowa ma r węzłów wyjściowych (*ang. multiple outputs*), dla danego obiektu danych X_i sieć generuje r wyników: $z_i(1), z_i(2), \ldots, z_i(r)$. Aby ocenić, czy X_i jest obiektem odstającym, oblicza się sumę kwadratów tych wartości: $\sum_{j=1}^{r} z_i(j)^2$. Podobnie jak w przypadku pojedynczego wyjścia, im większa jest ta suma, tym bardziej obiekt jest uznawany za odstający. Te metody oceny obiektów odstających są podobne do podejścia stosowanego w faktoryzacji macierzy i PCA, gdzie również analizuje się pewne wartości dla każdego obiektu, aby zidentyfikować obiekty odstające.

Omówione podejście jest logiczne, ale rzadko stosowane z powodu ograniczeń w optymalizacji i trudności w uzyskaniu skompresowanej reprezentacji danych. Lepszym rozwiązaniem są autoenkodery, które są naturalnym wyborem do wykrywania obiektów odstających. Efektywnie redukują one wymiarowość wielowymiarowych zbiorów danych, pełniąc funkcję alternatywną wobec PCA lub faktoryzacji macierzy. Według autorów artykułu z 2017 roku [235], uznanych ekspertów w dziedzinie wykrywania anomalii, takich jak Sathe i Aggarwal (których prace, w tym książki w całości poświęcone wykrywaniu anomalii [236, 27], stanowią istotny wkład w tę dziedzinę), na tamten czas istniało tylko kilka prac dotyczących zastosowania sieci neuronowych do detekcji odchyleń. Artykuł pokazuje, że sieci neuronowe mogą być bardzo konkurencyjną techniką w porównaniu do innych istniejących metod. Mimo że od tego czasu minęło już kilka lat, badania w tej dziedzinie nadal nie są tak liczne, jakby na to zasługiwały, co podkreśla innowacyjność i wciąż niezbadany potencjał tej technologii.

Podstawowe podejście w sieciach neuronowych polega na użyciu wielowarstwowej symetrycznej sieci neuronowej do rekonstrukcji (czyli replikacji) danych. Błąd rekonstrukcji jest używany jako wskaźnik odchylenia. Zastosowanie sieci neuronowych do redukcji wymiarowości zostało szeroko omówione w literaturze [237, 238]. Tradycyjne metody, takie jak PCA czy faktoryzacja macierzy, opierają się na założeniu liniowej kompresji danych [239]. Wielowarstwowe sieci neuronowe oferują jednak bardziej ogólną i elastyczną formę redukcji wymiarowości, umożliwiając nieliniową kompresję dzięki swojej złożonej strukturze. Podział sieci neuronowej replikatora na dwie części w miejscu środkowej warstwy pozwala na wyodrębnienie dwóch funkcjonalnych elementów: kodera i dekodera. Koder (funkcja ϕ) przekształca dane w bardziej zwartą reprezentację, podczas gdy dekoder (funkcja ψ) rekonstruuje oryginalne dane z tej skompresowanej formy. W ten sposób $\phi(D)$ reprezentuje skompresowaną wersję zbioru danych D, a $\psi(\phi(D))$ odtwarza dane jako $D' = (\psi \circ \phi)(D)$. Dane odstające, które różnią się od reszty, są trudniejsze do skompresowania, co powoduje większe różnice między oryginalnymi danymi D a ich rekonstrukcją D'. Te różnice są istotne dla identyfikacji obiektów odstających. Pojawia się tu koncepcja błędu rekonstrukcji (*ang. reconstruction error*), czyli różnica między oryginalną macierzą danych D a jej odszumioną reprezentacją D', gdzie wartości bezwzględne tej różnicy $(D - D')$ stanowią ocenę anomalii.

Główna różnica między autoenkoderami a tradycyjnymi metodami faktoryzacji macierzy polega na większej zdolności autoenkoderów do reprezentowania złożonych, nieliniowych rozkładów danych. W badaniu [238] pokazano, że obraz o rozdzielczości 784 pikseli można zredukować do zaledwie 6 liczb rzeczywistych przy użyciu głębokich autoenkoderów, co nie jest możliwe za pomocą PCA. Głębokie autoenkodery zapewniają również lepszą rekonstrukcję niż faktoryzacja macierzy i PCA [238], co sugeruje, że mogą one być bardziej precyzyjne w wykrywaniu anomalii.

4.4 Istotne aspekty systemów uczących się

W tym rozdziale przedstawiamy krótki opis najważniejszych zasad stosowanych podczas pracy z systemami uczącymi się. Systemy te są zaprojektowane tak, aby mogły adaptować się do nowych danych i doświadczeń, co pozwala im na poprawę działania w miarę upływu czasu. Opisane tutaj zasady obejmują sposoby umożliwiające efektywne trenowanie modeli oraz zapewniające ich zdolność do generalizacji na nowych, nieznanych danych. Obejmują one metody optymalizacji, regularyzacji, przetwarzania oraz zarządzania hiperparametrami i spełniają ważną rolę w procesie uczenia się. Jest to krótki przegląd problemów i podstaw systemów uczących się. Więcej informacji można znaleźć w obszernej literaturze naukowej [122, 111].

4.4.1 Metodologia uczenia się dla identyfikacji anomalii

Algorytmy dla systemów uczących się mają zdolność do nauki na podstawie dostarczonych danych. Proces ten polega na tworzeniu modelu lub algorytmu, który potrafi uczyć się z danych i doświadczenia, zamiast polegać na statycznych regułach programowania. Według Toma Mitchella, znanego ze swojego wkładu w rozwój uczenia maszynowego, sztucznej inteligencji i neuronauki poznawczej [240]:

> „...program komputerowy uczy się na podstawie doświadczenia E (*ang. experience*) w odniesieniu do pewnej klasy zadań T (*ang. task*) i miary wydajności P (*ang. performance*), jeśli jego wydajność w zadaniach z klasy T, mierzona przez P, poprawia się wraz z doświadczeniem E."

Parafrazując te słowa, tak aby odzwierciedlić poruszany w rozprawie problem identyfikacji anomalii, należy uznać, iż program komputerowy uczy się na podstawie doświadczenia D (odpowiednia liczba próbek oznaczonych jako anomalie w zbiorze szkoleniowym. W przypadku wykrywania anomalii najczęściej brakuje oznaczonych etykiet celu, jak w uczeniu nadzorowanym, ponieważ jest to uczenie nienadzorowane, gdzie zbiór zawiera wiele cech, a algorytm uczy się użytecznych właściwości dotyczących struktury tego zbioru bez nauczyciela), w odniesieniu do określonego zbioru zadań Z (wykrywanie anomalii w zbiorze testowym), z wydajnością mierzoną jako W (czułość, precyzja, miara F1 oraz

inne odpowiednio dobrane ilościowe miary wydajności). Jeśli działanie względem zadań Z mierzone przez W poprawia się wraz z doświadczeniem D, uznajemy, że program się uczy. Zatem uczenie się jest narzędziem do nabycia umiejętności realizacji zadania Z.

Herbert Simon, laureat Nagrody Turinga wspólnie z Allenem Newellem ”za wkład w badania nad sztuczną inteligencją, psychologią ludzkiego poznania i procesem ewidencji”, który otrzymał również Nagrodę Nobla w dziedzinie nauk ekonomicznych w 1978 roku „za pionierską pracę nad procesami podejmowania decyzji w organizacjach gospodarczych”, podobnie definiuje w pracy „Why Should Machines Learn?” (Dlaczego maszyny powinny się uczyć?) uczenie się jako „...zmiany w systemie, które są adaptacyjne w tym sensie, że umożliwiają systemowi wykonywanie tego samego zadania lub zadań pochodzących z tej samej populacji bardziej efektywnie i skutecznie następnym razem” [241]. Podsumowując systemy uczące się są zaprojektowane tak, aby mogły samodzielnie poprawiać swoje działanie poprzez analizę dostarczonych danych i doświadczeń, co zwiększa ich elastyczność i efektywność w różnych zadaniach na przykład takich jak identyfikacji anomalii.

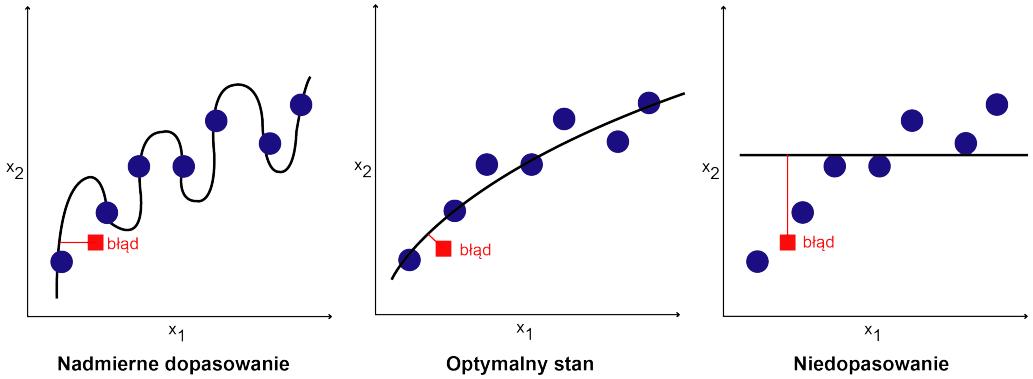
4.4.2 Praktyczne zagadnienia związane z uczeniem się

W praktycznym zastosowaniu systemów uczących się istnieje wiele wyzwań, które mogą wpływać na skuteczność i dokładność modelu. Poniżej wymieniono najważniejsze z nich.

Problem nadmiernego dopasowania i niedopasowania

Nadmierne dopasowanie (*ang. overfitting*) występuje, gdy model jest zbyt złożony, czyli ma zbyt wiele parametrów w stosunku do ilości danych treningowych, przez co zbyt dokładnie odwzorowuje dane treningowe, w tym szumy i przypadkowe wzorce. W rezultacie model osiąga bardzo niski błąd na tych danych, jednak jego wydajność na danych testowych jest znacznie gorsza. Taki model nie radzi sobie dobrze z nowymi danymi, co skutkuje dużym błędem na danych testowych. Nadmierne dopasowanie można opisać jako naruszenie zasady brzytwy Ockhama, która mówi, że model nie powinien być bardziej skomplikowany niż to konieczne. Gdy liczba wolnych parametrów modelu przewyższa ilość informacji zawartych w danych, dobór tych parametrów staje się w dużej mierze losowy. W rezultacie model zaczyna dopasowywać się do przypadkowych zakłóceń obecnych w danych treningowych, co prowadzi do utraty zdolności modelu do uogólniania i skutecznego działania na nowych, nieznanych danych. Aby zminimalizować ryzyko nadmiernego dopasowania, stosuje się różne techniki, takie jak walidacja krzyżowa, regularyzacja i wczesne zatrzymanie procesu uczenia. Zasady stosowane w dzisiejszym uczeniu maszynowym wywodzą się z dawnych idei, takich jak metodologiczna zasada angielskiego filozofa Williama Ockhama, żyjącego w XIV wieku, i zostały one później rozwinięte przez teoretyków statystycznego uczenia [242, 243]. Więcej informacji na temat nadmiernego dopasowania można znaleźć w literaturze naukowej, na przykład w artykułach [244, 245, 246], które omawiają problem nadmiernego dopasowania w modelach statystycznych i matematycznych, przyczyny tego

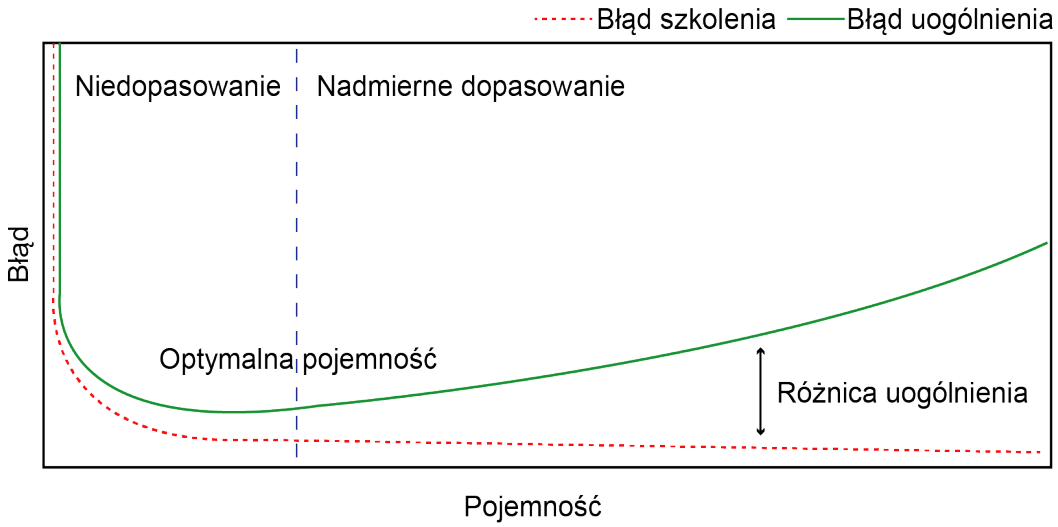
zjawiska, jego konsekwencje oraz metody zapobiegania mu.



Rysunek 4.5: Dopasowanie, niedopasowanie oraz stan optymalny modelu. Źródło: opracowanie własne.

Niedopasowanie (*ang. underfitting*) występuje, gdy model nie jest wystarczająco złożony, aby uchwycić wzorce w danych treningowych. Model o zbyt małej liczbie parametrów lub niewystarczającej strukturze nie jest w stanie dokładnie odzwierciedlić złożoności danych. W rezultacie model nie osiąga niskiego błędu na zbiorze treningowym, co przekłada się na jego niezdolność do skutecznego przewidywania nawet na danych, na których był trenowany. Przyczyny niedopasowania mogą obejmować zbyt krótki czas treningu lub niewystarczające wytrenowanie modelu. Taki model charakteryzuje się wysokim błędem systematycznym (*ang. bias*) i niską wariancją (*ang. variance*). Oznacza to, że model nie jest w stanie uchwycić podstawowych wzorców w danych, co prowadzi do niewłaściwych przewidywań zarówno na danych treningowych, jak i testowych. Aby zapobiec niedopasowaniu, można zwiększyć złożoność modelu, dodać więcej cech lub zastosować bardziej zaawansowany algorytm uczenia. Ważne jest również, aby odpowiednio dostosować czas treningu i zapewnić, że model ma wystarczającą liczbę parametrów, aby uchwycić złożoność. Literatura naukowa zajmuje się również tą tematyką, na przykład [247], gdzie omawiane są kwestie niedopasowania oraz nadmiernego dopasowania w kontekście teoretyki informacji. W artykule [248] podkreślono znaczenie badania niedopasowania w kontekście ogólnych metod eliminacji biasu w uczeniu maszynowym. Autorzy omawiają, jak regularyzacja, choć jest kluczowym narzędziem w zapobieganiu nadmiernemu dopasowaniu, może paradoksalnie prowadzić do niedopasowania, co z kolei może zwiększać bias.

Optymalny stan, zwany uogólnieniem, to sytuacja, w której model działa dobrze zarówno na danych treningowych, jak i na nowych, niewidzianych wcześniej danych. Rysunek 4.5 ilustruje ten stan. Pojemność modelu (*ang. capacity*) wpływa na zdolność modelu do uogólnienia, czyli jego efektywność na nowych, nieznanych danych. Kiedy model ma odpowiednią pojemność, potrafi uchwycić ważne wzorce w danych treningowych i dobrze przewidywać na nowych danych. Oznacza to, że model nie jest ani zbyt prosty



Rysunek 4.6: Związek między pojemnością a błędem. Źródło: opracowanie własne na podstawie [50].

(niedopasowanie), ani zbyt skomplikowany (nadmierne dopasowanie). Taka równowaga pozwala na uzyskanie niskiego błędu zarówno na danych treningowych, jak i testowych. Pojemność klasyfikatora została matematycznie sformalizowana w ramach teorii Vapnika-Chervonenkisa (VC) [242]. Aby pogłębić tematykę, można zajrzeć do książki „Mastering Machine Learning Algorithms" [249]. Sytuację tę ilustruje rysunek 4.6, który pokazuje, że w miarę jak zwiększana jest pojemność modelu, błąd szkoleniowy maleje, ale różnica między błędem szkoleniowym a testowym rośnie. W końcu ta różnica staje się tak duża, że korzyści wynikające z dalszego zmniejszania błędu szkoleniowego są niwelowane przez wzrastający błąd testowy. W ten sposób model wchodzi w obszar nadmiernego dopasowania, gdzie pojemność modelu przekracza optymalną wartość.

Regularyzacja i optymalizacja

Regularyzację można traktować jako formę „stopniowego zapominania". W tym procesie model stopniowo eliminuje mniej istotne lub zakłócające wzorce. To podejście jest podobne do biologicznego zjawiska przycinania synaptycznego, w którym zbędne połączenia nerwowe są z czasem usuwane. Jest to technika stosowana w uczeniu maszynowym w celu zapobiegania nadmiernemu dopasowaniu modelu do danych treningowych. Przykłady regularyzacji to L1 i L2.

Regularyzacja L1, znana także jako Lasso (*ang. least absolute shrinkage and selection operator*) [250], dodaje karę proporcjonalną do wartości bezwzględnych współczynników. Zachęca to do rzadszości, co oznacza, że niektóre współczynniki będą dokładnie równe zero, efektywnie wykonując wybór cech. Wyrażenie kary to: $\lambda \sum |w_i|$. Lasso prowadzi do

bardziej interpretowalnych modeli poprzez eliminację mniej istotnych zmiennych. Wybór cech upraszcza problem uczenia się, dzięki wyborowi podzbioru cech, które powinny być używane. Kara L1 sprawia, że podzbiór wag staje się zerowy, co sugeruje, że odpowiadające im cechy można bezpiecznie odrzucić.

Regularyzacja L2, znana również jako regresja grzbietowa lub regularyzacja Tichonowa, dodaje karę proporcjonalną do kwadratu współczynników. Zachęca to do mniejszych wartości współczynników, ale nie wymusza rzadszości. Wyrażenie kary to: $\lambda \sum w_i^2$. Symbol λ reprezentuje współczynnik regularyzacji. Jest to parametr, który kontroluje wagę kary dodawanej do funkcji straty. Artykuł [251] omawia ciekawą równoważność pomiędzy dodawaniem szumu do danych wejściowych podczas treningu sieci neuronowej a stosowaniem regularyzacji Tikhonova. W szczególności autor pokazuje, że trening z dodanym szumem jest równoważny minimalizacji regularyzowanej funkcji błędu, gdzie człon regularyzacyjny należy do klasy uogólnionych regularyzatorów Tikhonova.

Regularyzację L1 i L2 można również łączyć w jednej technice, znanej jako Elastic Net. W tej technice kara dodawana do funkcji straty obejmuje zarówno składnik L1, jak i składnik L2. Wyrażenie kary dla Elastic Net można zapisać jako:

$$\lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2 \tag{4.7}$$

Elastic Net zapewnia unikalne minimum funkcji straty i obejmuje zarówno regularyzację L1, jak i L2 jako specjalne przypadki (dla odpowiednich wartości λ_1 i λ_2).

Jedną z popularnych metod regularyzacji jest też wczesne zatrzymanie, polegające na przerwaniu procesu gradientowego spadku po określonej liczbie iteracji. Punkt zatrzymania można określić, wydzielając część danych treningowych jako zestaw walidacyjny i monitorując błąd modelu na tym zestawie. Proces uczenia jest przerywany, gdy błąd na zestawie walidacyjnym zaczyna wzrastać. Aby uzyskać więcej informacji na temat wspomnianych wyżej metod, takich jak L1 i L2, oraz innych metod regularyzacji, warto sięgnąć po książkę [50].

Drugim tematem podrozdziału jest optymalizacja. Algorytmy optymalizacyjne pomagają w szybszym i skuteczniejszym uczeniu się modeli. Dzięki nim modele szybciej dostosowują swoje parametry, co przekłada się na lepsze wyniki i krótszy czas treningu. W praktyce oznacza to, że możemy efektywniej wykorzystywać dane i zasoby komputerowe, aby uzyskać dokładniejsze i bardziej niezawodne prognozy czy klasyfikacje. Optymalizacja może obejmować różne konteksty, jednak podstawowy proces polega na dostosowywaniu parametrów sieci, aby minimalizować funkcję kosztów, która obejmuje zarówno miary wydajności na zbiorze treningowym, jak i dodatkowe składniki regularyzacyjne. Algorytmy optymalizacyjne różnią się od tradycyjnych metod optymalizacji, ponieważ często działają pośrednio, zmniejszając funkcję kosztów w celu poprawy miary wydajności na zbiorze testowym. Efektywne algorytmy optymalizacyjne często wykorzystują stochastyczne metody gradientowe, które są bardziej wydajne niż pełne obliczenia gradientu dla całego zbioru treningowego.

Popularne algorytmy optymalizacji to (*ang. stochastic gradient descent*, SGD) oraz (*ang. adaptive moment estimation*, Adam).

SGD to metoda optymalizacji często stosowana w trenowaniu modeli uczenia maszynowego. Polega na aktualizacji parametrów modelu na podstawie obliczeń gradientu funkcji kosztów dla losowo wybranego przykładu lub małej grupy przykładów z zestawu danych. Ta metoda pozwala na szybszą konwergencję w porównaniu do tradycyjnego gradientu prostego, który wykorzystuje cały zbiór danych do każdej aktualizacji. Wzór (4.8) ilustruje metodę stochastycznego spadku gradientu (SGD):

Niech D będzie zbiorem danych, X_i i-tym obiektem w zbiorze danych D, θ parametrami modelu, $J(\theta)$ funkcją kosztu, η współczynnikiem uczenia się, a $\nabla J(\theta; X_i)$ gradientem funkcji kosztu dla przykładu X_i. Aktualizacja parametrów w iteracji t jest dana wzorem:

$$\theta_t = \theta_{t-1} - \eta \nabla J(\theta_{t-1}; X_i), \tag{4.8}$$

gdzie: - θ_{t-1} to wartości parametrów przed iteracją t - θ_t to wartości parametrów po iteracji t - η to współczynnik uczenia się - $\nabla J(\theta_{t-1}; X_i)$ to gradient funkcji kosztu obliczony dla losowo wybranego przykładu X_i z zestawu danych D.

W każdej iteracji losowany jest inny przykład X_i ze zbioru danych D, co sprawia, że aktualizacje parametrów są bardziej zróżnicowane, a proces uczenia się może szybciej konwergować. Więcej informacji na temat SGD można znaleźć w pracy [252].

Adam [253] to adaptacyjny algorytm optymalizacyjny związany z szybkością uczenia się. Nazwa pochodzi od wyrażenia momenty adaptacyjne (*ang. adaptive moments*). Jego główną zaletą jest zdolność do szybkiej i efektywnej optymalizacji parametrów modelu, co przyspiesza proces uczenia. Zasada działania algorytmu jest następująca:

Inicjalizacja:

- Ustawienia początkowe dla parametrów θ (parametry modelu), momentów pierwszego stopnia m i drugiego stopnia v na zero oraz ustawienie liczby iteracji t na zero.

  Moment pierwszego stopnia, oznaczany jako m_t, jest średnią ważoną gradientów. Pomaga wprowadzić „momentum” do aktualizacji parametrów, co oznacza, że uwzględnia zarówno bieżącą wartość gradientu, jak i jego średnie wartości z poprzednich kroków. To podejście pomaga wygładzić ścieżkę optymalizacji, zmniejszając oscylacje i przyspieszając konwergencję.

  Matematycznie, moment pierwszy stopnia jest obliczany w sposób rekurencyjny:

  $$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla J(\theta_t), \tag{4.9}$$

  gdzie β_1 jest parametrem wygładzania, zazwyczaj ustawionym na 0,9.

Moment drugiego stopnia, oznaczany jako v_t, jest średnią ważoną kwadratów gradientów. Uwzględnia zmienność gradientów, co pozwala adaptacyjnie skalować krok uczący. Dzięki temu algorytm Adam może efektywnie radzić sobie z różnymi skalami gradientów, co jest szczególnie przydatne w przypadku niestacjonarnych (tj. zmieniających swoje właściwości statystyczne w czasie) lub hałaśliwych danych.

Moment drugiego stopnia jest obliczany jako:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla J(\theta_t))^2, \tag{4.10}$$

gdzie β_2 jest parametrem wygładzania, zazwyczaj ustawionym na 0,999.

Aktualizacja momentów:

- Dla każdej iteracji, oblicz gradient funkcji kosztu $\nabla J(\theta_t)$ dla aktualnych parametrów θ_t.
- Zaktualizuj moment pierwszego stopnia m_t (wzór 4.9) i drugiego stopnia v_t (wzór 4.10) za pomocą wykładniczego wygładzania.

Korekcja obciążenia:

- Aby poprawić oszacowania momentów w początkowych krokach, gdzie m_0 i v_0 są zainicjowane na zero, Adam wprowadza korekcje obciążenia. Oblicz skorygowane wartości momentów:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4.11}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4.12}$$

Aktualizacja parametrów:

- Zaktualizuj parametry θ_t za pomocą skorygowanych momentów:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{4.13}$$

W algorytmie Adam, ϵ to mała stała, która jest dodawana do mianownika w celu zapewnienia stabilności numerycznej. Dzięki temu unikamy problemów związanych z dzieleniem przez zero lub bardzo małe wartości, które mogłyby prowadzić do niestabilnych i nieprzewidywalnych wyników. Wartość ϵ jest zazwyczaj ustawiana na 10^{-8}, co wystarcza do stabilizacji obliczeń bez znaczącego wpływu na wartości aktualizacji parametrów.

Iteracja:

- Powtórz powyższe kroki do osiągnięcia kryterium stopu (np. określonej liczby iteracji lub konwergencji funkcji kosztu).

Algorytm Adam jest szeroko stosowany w praktyce, szczególnie w obszarze głębokiego uczenia. Jego zdolność do adaptacyjnego dostosowywania kroku uczącego czyni go bardzo efektywnym w przypadku dużych i złożonych sieci neuronowych. Dzięki tej adaptacyjności, Adam jest często wybierany jako domyślny optymalizator w popularnych bibliotekach do głębokiego uczenia, takich jak TensorFlow czy PyTorch.

Poza algorytmami Adam i SGD, istnieją inne powszechnie używane metody optymalizacji, takie jak AdaGrad [254], RMSProp [255], a także RMSProp z momentem Nesterova [50]. Nie ma najlepszego algorytmu optymalizacyjnego, co potwierdza porównanie wielu algorytmów optymalizacyjnych dla szerokiego zakresu zadań uczenia się, zgodnie z wynikami pracy [256].

Podsumowując, optymalizacja to proces znajdowania najlepszych parametrów modelu, które minimalizują funkcję kosztu na danych treningowych. W praktyce oznacza to dostosowywanie parametrów w taki sposób, aby przewidywania modelu były jak najdokładniejsze w porównaniu do rzeczywistych wyników. Regularyzacja, z drugiej strony, wprowadza dodatkowy element do funkcji kosztu, który karze za zbytnią złożoność modelu. Celem jest poprawa zdolności modelu do uogólniania, czyli jego wydajności na nowych, niewidzianych wcześniej danych, co pomaga uniknąć przeuczenia. Przeuczenie występuje, gdy model jest zbyt dopasowany do danych treningowych i nie radzi sobie dobrze z nowymi obiektami. Podczas gdy optymalizacja zajmuje się dopasowaniem parametrów modelu, regularyzacja dba o to, aby model nie był zbyt skomplikowany, zapewniając tym samym lepszą wydajność na nowych obiektach.

Hiperparametry

Większość algorytmów uczenia maszynowego wymaga ustawień znanych jako hiperparametry, które muszą być określone przed rozpoczęciem procesu treningowego. Hiperparametry regulują projektowanie modelu i różnią się od podstawowych parametrów, takich jak wagi połączeń w sieci neuronowej. Hiperparametry te nie są modyfikowane przez sam algorytm w trakcie treningu, lecz są wcześniej zdefiniowane (zazwyczaj ustalane ręcznie lub za pomocą fazy dostrajania). Przykłady takich hiperparametrów obejmują współczynnik uczenia (*ang. learning rate*), liczbę warstw w sieci neuronowej, rozmiar partii danych (*ang. batch size*) i inne. Optymalny dobór tych hiperparametrów ma znaczenie dla osiągnięcia wysokiej skuteczności modelu. Istnieje dwupoziomowa organizacja parametrów w sieci neuronowej, w której podstawowe parametry modelu, takie jak wagi, są optymalizowane za pomocą wstecznej propagacji błędów dopiero po ustaleniu hiperparametrów ręcznie lub w fazie dostrajania.

Hiperparametry są trudne do optymalizacji, dlatego algorytmy nie uczą się ich bezpośrednio. Szczególnie ważne jest, aby nie uczyć hiperparametrów, które wpływają na złożoność modelu, ze zbioru szkoleniowego, ponieważ taki proces zawsze prowadziłby do wyboru maksymalnej możliwej złożoności, co skutkowałoby przeuczeniem modelu. Na przykład, zbiór treningowy można zawsze lepiej dopasować za pomocą wielomianu o bardzo wysokim stopniu i zerowej regularyzacji, niż za pomocą wielomianu o niższym stopniu i dodatniej regularyzacji.

Aby zapobiec nadmiernemu dopasowaniu modelu, kluczowe jest zastosowanie zestawu walidacyjnego, który jest wyraźnie oddzielony od zbioru treningowego. Dane treningowe należy podzielić na dwa niezależne segmenty: jeden służy do nauki parametrów, a drugi do oceny błędu uogólnienia oraz dostrajania hiperparametrów. Zbiór walidacyjny pełni rolę w „trenowaniu" hiperparametrów, choć wynikowy błąd jest zazwyczaj niedoszacowaniem rzeczywistego błędu uogólnienia. Po zakończeniu procesu optymalizacji hiperparametrów, ostateczny błąd uogólnienia można oszacować za pomocą oddzielnego zbioru testowego. W przypadku ograniczonej ilości danych, metody takie jak k-krotna walidacja krzyżowa mogą być używane do dokładniejszego oszacowania średniego błędu testowego, chociaż wiąże się to ze zwiększonymi kosztami obliczeniowymi.

Są dostępne różne metody optymalizacji hiperparametrów. Najbardziej znaną techniką jest wyszukiwanie siatkowe (*ang. grid search*), w której wybiera się zestaw wartości dla każdego hiperparametru i testuje wszystkie kombinacje tych wartości. Liczba kombinacji rośnie wykładniczo wraz z liczbą hiperparametrów, co może być bardzo kosztowne obliczeniowo. W niektórych przypadkach losowe próbkowanie hiperparametrów może być bardziej efektywne. Często zamiast próbkować same hiperparametry, próbkowane są ich logarytmy. Na przykład, zamiast próbkować współczynnik regularyzacji λ między 0,01 a 0,0001, próbkowane jest $\log(\lambda)$ równomiernie między -2 a -4, a następnie podnosi się do potęgi 10. W przypadku dużych zbiorów danych, pełne przetestowanie wszystkich kombinacji hiperparametrów może być niepraktyczne ze względu na długi czas treningu. Dlatego algorytmy często są uruchamiane na krótki czas, aby ocenić ich postępy. Artykuł [257] szczegółowo omawia problem optymalizacji hiperparametrów, koncentrując się na metodach przyspieszania ewaluacji konfiguracji hiperparametrów. Więcej informacji na temat dostrajania hiperparametrów można znaleźć w literaturze, na przykład w pracy [151].

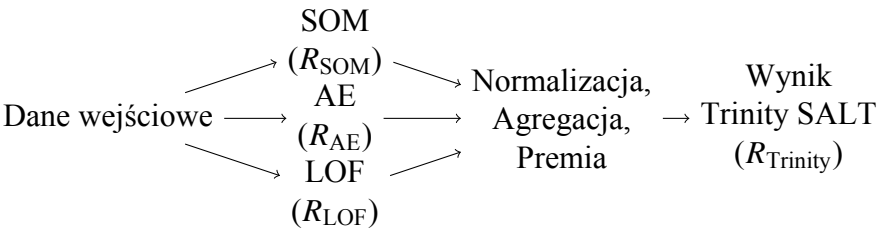
Metody przetwarzania cech i inicjalizacji wag

Metody przetwarzania cech używane w treningu sieci neuronowych nie różnią się znacząco od tych stosowanych w innych algorytmach uczenia maszynowego. Normalizacja cech, czy to poprzez standaryzację, czy normalizację min-max, często zapewnia lepszą wydajność modelu. Relatywne wartości cech mogą różnić się o więcej niż rząd wielkości, co sprawia, że proces uczenia parametrów staje się problematyczny. W takim przypadku funkcja straty jest bardziej czuła na niektóre parametry niż na inne, co wpływa na wydajność algorytmów optymalizacyjnych, takich jak metoda spadku gradientu. Dlatego zaleca się

przeprowadzenie skalowania cech na początku. Więcej informacji na temat wstępnego przetwarzania cech można znaleźć w pracy [151]. Istotne jest także zrozumienie znaczenia prawidłowej inicjalizacji wag w sieciach neuronowych. Odpowiednia inicjalizacja wag jest bardzo ważna dla zapewnienia stabilności gradientów w sieciach neuronowych, co z kolei poprawia wydajność i skuteczność procesu trenowania, zwłaszcza w głębokich sieciach. Przegląd różnych strategii inicjalizacji wag stosowanych w sieciach neuronowych znajduje się w pracy [258].

4.5 Detektory zespołu Trinity SALT: SOM, AE i LOF

Aby opracować system Trinity SALT (**S**OM-**A**E-**L**OF-**T**riDetect), zapewniający stabilne i niezawodne wyniki w różnych zbiorach danych, wykorzystano heterogeniczną mieszankę trzech różnych detektorów bazowych. W kolejnych podrozdziałach omówiono algorytmy SOM, AE i LOF, które stanowią główne komponenty tego rozwiązania. System został zaprojektowany jako zespół detektorów, które łączą różne metody wykrywania anomalii, aby uzyskać wszechstronny i skuteczny mechanizm.



Rysunek 4.7: Schemat działania systemu Trinity SALT. Źródło: Opracowanie własne.

Poniżej przedstawiono krótką charakterystykę trzech głównych komponentów, a w kolejnych podrozdziałach opisano je szczegółowo:

- **Komponent oparty na sieci samouczącej się SOM:**
  SOM jest nienadzorowanym algorytmem uczącym się, który organizuje dane wejściowe w siatkę neuronów. Każdy neuron reprezentuje grupę danych, umożliwiając wykrywanie anomalii poprzez identyfikację punktów danych, które nie pasują do żadnej grupy. W Trinity SALT, SOM identyfikuje anomalie na podstawie błędu kwantyzacji, wskazując obiekty o największych błędach kwantyzacji jako potencjalnie najbardziej odchylone. Algorytm jest uruchamiany wielokrotnie ze zmiennymi hiperparametrami, a najlepsze wyniki są wybierane, tworząc ranking anomalii R_{SOM}.
- **Komponent oparty na autoenkoderze AE:**
  Autoenkoder to rodzaj sieci neuronowej, która uczy się kompresować dane wejściowe do przestrzeni o niższym wymiarze, a następnie rekonstruować je z powrotem

do oryginalnej przestrzeni. Różnica, zwana błędem rekonstrukcji, między obiektami wejściowymi a zrekonstruowanymi obiektami jest wykorzystywana do wykrywania anomalii. W Trinity SALT, autoenkoder jest uruchamiany wielokrotnie ze zmiennymi hiperparametrami, a najlepsze wyniki są wybierane, tworząc ranking anomalii R_{AE}.

- **Komponent oparty na algorytmie gęstościowym LOF:**
  Algorytm LOF identyfikuje anomalie, analizując gęstość lokalną punktu danych w porównaniu z gęstościami jego sąsiadów. W systemie Trinity SALT, LOF jest uruchamiany wielokrotnie, z różnymi wartościami hiperparametru k oraz metrykami odległości. Najlepsze wyniki są następnie wybierane, tworząc ranking anomalii R_{LOF}.

Jak przedstawiono na schemacie działania systemu Trinity SALT, rysunek 4.7, każdy z rankingów anomalii R_{SOM}, R_{AE} i R_{LOF} jest normalizowany na podstawie maksymalnych wartości dla każdej metody. Te znormalizowane rankingi są następnie porównywane, a dla każdej anomalii wybierany jest najwyższy wynik, który staje się częścią ostatecznego wyniku zespołu Trinity SALT, z uwzględnieniem dodatkowych bonusów za zgodność wyników między różnymi algorytmami. Każdy algorytm posiada unikalne podejście do analizy, co pozwala na identyfikację różnych typów anomalii. Detektory bazowe wykorzystują odmienne paradygmaty: SOM operuje w ramach uczenia konkurencyjnego, autoenkoder stosuje wsteczną propagację błędów, a LOF opiera się na analizie gęstości lokalnej. Każdy z tych detektorów odgrywa ważną rolę, co pozwala systemowi Trinity SALT skutecznie wykrywać anomalie w różnych zbiorach danych, zawierających zarówno dane jakościowe, jak i ilościowe.

4.5.1 SOM - algorytm map samoorganizujących się

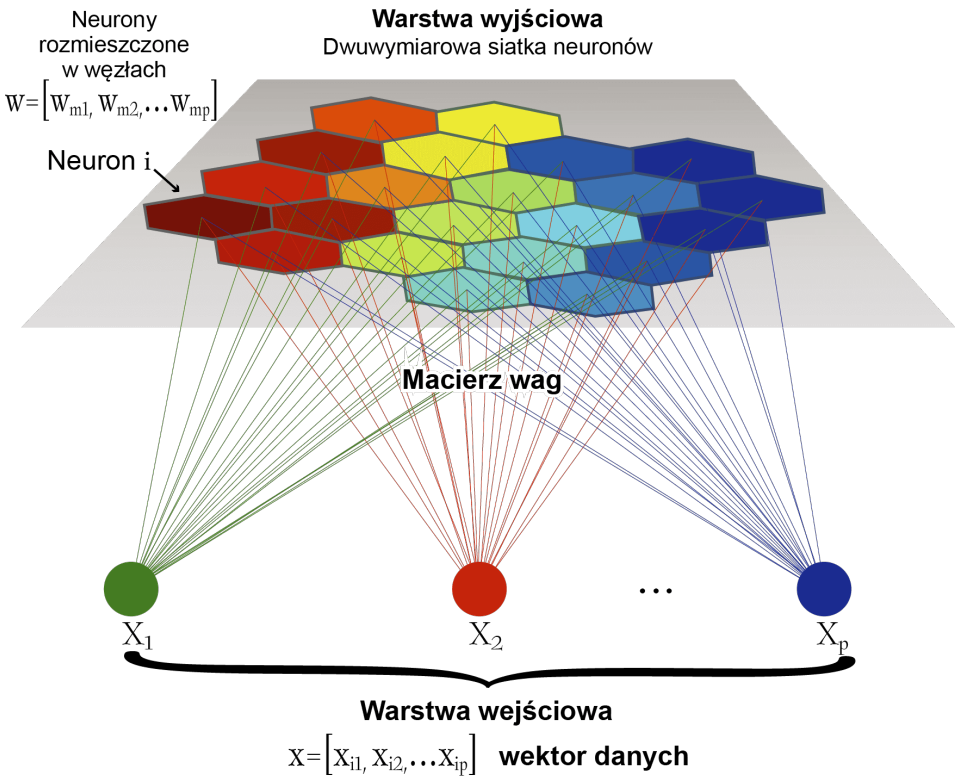
Uczenie nienadzorowane, czasami nazywane samouczeniem lub uczeniem bez nadzoru, odnosi się do metod w ramach sztucznej inteligencji i głębokiego uczenia, gdzie procesy optymalizacji wag w sieciach neuronowych są realizowane przy użyciu danych szkoleniowych, które nie zawierają etykiet ani docelowych wartości wyjściowych. Tego rodzaju podejście pozwala modelom neuronowym na autonomiczne odkrywanie ukrytych struktur i wzorców w danych, co jest przydatne w sytuacjach, gdy brak jest z góry określonej wiedzy o liczbie klas lub ich rozkładzie w kontekście danego zadania klasyfikacyjnego. Uczenie nienadzorowane jest nieodzowne w sytuacjach, gdy potrzebujemy analizować dane bez uprzedniej znajomości ich struktury. Przykłady algorytmów realizujących uczenie nienadzorowane obejmują Winner Takes All (*ang. winner takes all*, WTA) [259, 260] oraz Self-Organizing Feature Map (*ang. Kohonen's self-organizing feature map*, SOFM) opracowaną przez Kohonena [261, 262, 263, 264], częściej nazywaną jako Self Organizing Map (*ang.self-organizing map*, SOM). Sieć SOM jest inspirowana biologicznymi

zjawiskami takimi jak: percepcja wzrokowa (retinotopia), percepcja czuciowa (somatotopia) oraz percepcja słuchowa (tonotopia) [265, 266, 267, 268]. Początki tworzenia modeli samoorganizujących można znaleźć już w 1973 roku w pracy von der Malsburga [269], a także w późniejszych badaniach [270, 271, 272].

Algorytm WTA, po procesie uczenia nienadzorowanego, umożliwia sieci neuronowej oszacowanie liczby klas w danych oraz lokalizacji ich centroidów, które są reprezentowane przez wybrane neurony. W sieciach WTA neurony konkurują o aktywację, przy czym zazwyczaj jedynie neuron o najwyższej aktywacji pozostaje aktywny, a pozostałe są deaktywowane. Ten mechanizm jest szeroko stosowany w kontekście uczenia konkurencyjnego. Większość metod uczenia sieci neuronowych koncentruje się na aktualizacji wag w celu minimalizacji błędów predykcji, zazwyczaj poprzez algorytm wstecznej propagacji. W przeciwieństwie do tego, uczenie konkurencyjne działa na zupełnie innym paradygmacie, gdzie celem nie jest mapowanie danych wejściowych na wyjścia w celu korygowania błędów. Zamiast tego, neurony konkurują o możliwość reprezentowania podzbiorów danych wejściowych, dostosowując swoje wagi, aby lepiej pasować do określonych próbek danych. Proces ten znacząco różni się od wstecznej propagacji błędów, ponieważ skupia się na samodzielnym odkrywaniu strukturalnych wzorców w danych, a nie na bezpośrednim minimalizowaniu błędów.

W odróżnieniu od algorytmu WTA, który modyfikuje tylko wagi jednego neuronu, sieci samoorganizujące się SOM zmieniają wagi także neuronów znajdujących się w topologicznym sąsiedztwie neuronu zwycięskiego. Zasada działania przedstawiona na rysunku 4.8 pokazuje, że algorytm Kohonena nie tylko identyfikuje grupy danych o podobnych cechach, ale także ustala ich topologiczne rozmieszczenie w warstwie neuronów, implementując technikę samoorganizującego się odwzorowania cech. Neurony reprezentujące podobne wartości cech są umieszczane blisko siebie, odzwierciedlając relacje między danymi w przestrzeni. Topologiczne sąsiedztwo może być zdefiniowane w jednym lub dwóch wymiarach, co determinuje ułożenie neuronów. Proces modyfikacji wag opiera się wyłącznie na danych wejściowych, niepowiązanych z żadnymi znanymi wartościami wyjściowymi. Takie podejście umożliwia algorytmowi autonomiczną interpretację i organizację danych, co jest niezwykle przydatne do odkrywania naturalnych struktur i grup. Dzięki temu samoorganizujące się mapy Kohonena stanowią efektywne narzędzie do wizualizacji i analizy skomplikowanych zbiorów danych, pozwalając na lepsze zrozumienie i eksplorację ich strukturalnych właściwości.

W praktyce sieci SOM znajdują szerokie zastosowanie w różnych dziedzinach, od analizy obrazów po eksplorację danych biologicznych. Są one wykorzystywane głównie do grupowania i wizualizacji danych wielowymiarowych, a także do wykrywania anomalii na podstawie błędu kwantyzacji oraz identyfikacji małych, odstających grup w już podzielonym zbiorze danych. Ponadto, SOM jest powszechnie stosowany jako narzędzie do redukcji wymiarów w procesie wizualizacji danych. Na przykład, w artykule [273] opisano, jak zastosowanie SOM może pomóc w radzeniu sobie z problemem przeciążenia infor-



Rysunek 4.8: Schemat działania sieci samoorganizującej się SOM, ilustrujący proces zmiany wag neuronów w topologicznym sąsiedztwie neuronu zwycięskiego. Źródło: opracowanie własne.

macyjnego oraz poprawić efektywność wyszukiwania i organizacji danych w Internecie. Algorytm SOM automatycznie kategoryzuje strony internetowe na podstawie ich treści, co umożliwia bardziej precyzyjne i szybkie wyszukiwanie informacji w specyficznych kategoriach tematycznych, przyczyniając się do lepszego zarządzania dużymi zbiorami danych.

SOM należy do tradycyjnych metod uczenia maszynowego i nie stosuje wielowarstwowej architektury do ekstrakcji cech. Zamiast tego wykorzystuje mechanizm uczenia konkurencyjnego, w którym sieć Kohonena posiada mechanizm aktualizacji wag oparty na zwycięzcy, przekazywany od zwycięskiego neuronu do jego sąsiadów. Mechanizm ten jest kształtowany przez funkcję sąsiedztwa, która definiuje zbiór neuronów wokół danego neuronu na podstawie promienia sąsiedztwa określonego przez metrykę odległości. Najczęściej stosowanymi funkcjami sąsiedztwa w sieci Kohonena są funkcje prostokątne, Gaussowskie i trójkątne. SOM to osobna grupa struktur sieci neuronowych i metod uczenia, w której nie ma warstw neuronów. Sieć uczy się, nie otrzymując informacji zwrotnej co do trafności swoich predykcji lub klasyfikacji. Każda badana jednostka odpowiada w sieci

jednemu neuronowi, który w stosunku do niej jest najbliższy. Ten najbliższy neuron uczy się, modyfikując swoje wagi, aby jeszcze bardziej zmniejszyć tę odległość. Model samodzielnie szuka wzorców i struktur w danych, odzwierciedlając topologię i rozkład danych wejściowych, dzięki czemu może odkryć ukryte wzorce.

Tabela 4.1: Złożoność obliczeniowa i pamięciowa algorytmu SOM. Źródło: opracowanie własne na podstawie [274].

Typ złożoności	**Równanie**	**Opis**
sekwencyjna	$O(nmp)$	złożoność obliczeniowa jednej iteracji szkolenia w trybie sekwencyjnym
wsadowa	$O(3nmp+(2p+4)m^2+(n+m)p)$	złożoność obliczeniowa szkolenia w trybie wsadowym
sekwencyjna	$O((n+m)p)$	zużycie pamięci dla danych i wektorów wag w trybie sekwencyjnym
wsadowa	$O((n+2m)p)$	zużycie pamięci dla danych, wag i sum wektorów w trybie wsadowym
macierzy odległości	$O(m(m-1)/2)$	pamięć dla macierzy odległości między neuronami
n - liczba obiektów, **m** - liczba neuronów, **p** - liczba wymiarów, **t** - liczba iteracji, $t = 1$		

Złożoność algorytmu SOM można określić jako wielomianową złożoność czasową, zależną od trzech czynników: liczby danych (n), liczby neuronów (m) oraz liczby wymiarów (p). Formalnie, złożoność czasowa algorytmu można zapisać jako $O(nmp)$. Oznacza to, że czas wykonania algorytmu rośnie liniowo w zależności od liczby danych, liczby neuronów i liczby wymiarów. Każdy z tych czynników wpływa liniowo na całkowity czas obliczeń, jednak ich kombinacja powoduje, że złożoność algorytmu jest proporcjonalna do iloczynu tych zmiennych. Dlatego złożoność algorytmu SOM klasyfikujemy jako wielomianową. Szczegółowe informacje na temat złożoności obliczeniowej i pamięciowej algorytmu SOM przedstawiono w tabeli 4.1. Istnieje możliwość optymalizacji poprzez dostosowanie parametrów siatki, szybkości uczenia oraz innych parametrów takich jak funkcja sąsiedztwa i liczba iteracji. W algorytmie sekwencyjnym liczba operacji zmiennoprzecinkowych (dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie) wynosi $6nmp+2nm$. W algorytmie wsadowym liczba operacji wynosi $3nmp+(2p+4)m^2+(n+m)p$. Dla przypadków, gdy $n \gg m$ (liczba obiektów znacznie większa niż liczba neuronów), złożoność obliczeniowa jednej iteracji szkolenia sekwencyjnego wynosi $O(nmp)$, natomiast złożoność obliczeniowa algorytmu wsadowego wynosi około połowę złożoności algorytmu sekwencyjnego. Pod względem zużycia pamięci, algorytm sekwencyjny wymaga $(n+m)p$ liczb zmiennoprzecinkowych do przechowywania danych oraz macierzy wektorów wag. W algorytmie wsadowym zużycie pamięci wynosi $(n+2m)p$ liczb zmiennoprzecinkowych, obejmujących dane, wektory wag i sumy wektorów. Dodatkowo, w obu przypadkach obliczane są odległości między jednostkami mapy, co wymaga pamięci na macierz o roz-

miarze m^2 elementów. Ze względu na symetryczność tej macierzy, liczba niezbędnych liczb zmiennoprzecinkowych może zostać zmniejszona do $m(m-1)/2$, zgodnie z [274].

W kontekście dużych zbiorów danych tekstowych, artykuł [275] na temat (*ang. scalable self-organizing map*, SSOM) koncentruje się na konwencjonalnym SOM, gdzie złożoność czasowa jest $O(S^2)$ ze względu na wzrost liczby unikalnych terminów i liczby cykli prezentacji dokumentów wraz z rozmiarem kolekcji. Tabela 4.1 przedstawia ogólne i szczegółowe złożoności obliczeniowe i pamięciowe standardowego SOM, biorąc pod uwagę liczbę obiektów, neuronów i wymiarów, zarówno w trybach sekwencyjnym, jak i wsadowym.

Algorytm 1: Algorytm SOM - Mapa Samoorganizująca się

Input: Zbiór danych D z obiektami $X_i \in \mathbb{R}^p$, liczba neuronów m, liczba iteracji t
Output: Wagi neuronów po zakończeniu uczenia

```
Inicjalizuj wagi neuronów losowo
for każdą iterację i od 1 do t do
    Wybierz losowy obiekt X_i z D
    Znajdź neuron z najbliższymi wagami do X_i, oznacz go jako BMU Najlepiej
      Dopasowany Neuron (Best Matching Unit)
    Zaktualizuj wagi BMU oraz jego sąsiadów na podstawie X_i i funkcji
      sąsiedztwa
return Wagi neuronów po zakończeniu uczenia
```

Budowa sieci SOM obejmuje dwa główne etapy: ustalanie struktury sieci oraz inicjalizację. W pierwszym etapie należy określić liczbę neuronów, topologię i kształt sieci oraz promień sąsiedztwa. W przypadku sieci Kohonena rozmiar sieci (liczba neuronów) musi być ustalony na początku, przy czym najczęściej stosowanymi topologiami są prostokątne lub heksagonalne. Inicjalizacja mapy SOM polega na przypisaniu początkowych wag neuronom oraz określeniu miary odległości między neuronem a odwzorowywaną jednostką. W sieciach SOM można stosować różne miary odległości, takie jak odległość euklidesowa, kosinusowa, Hamminga i inne. Ważne jest również ustalenie kroku uczenia, czyli szybkości zmian wartości wag w kolejnych iteracjach. Aby osiągnąć kompromis między szybkością a dokładnością uczenia się sieci, ważne jest właściwe dostosowanie poziomu współczynnika uczenia. W miarę postępu nauki, współczynnik ten stopniowo się zmniejsza, co powoduje wolniejsze tempo uczenia się sieci. Różne funkcje, takie jak wykładnicza, hiperboliczna, potęgowa i funkcja odwrotna, są stosowane do regulacji zmian współczynnika uczenia w zależności od liczby iteracji. Właściwe ustalenie kroku uczenia jest szczególnie istotne w dużych sieciach, zawierających setki neuronów. W mniejszych sieciach promień sąsiedztwa jest na tyle mały, że większość neuronów uczy się w każdej iteracji, co sprawia, że krok uczenia traci na znaczeniu [263, 262]. Podczas treningu każdy obiekt danych jest prezentowany mapie SOM. Sieć identyfikuje neuron, którego wagi

są najbliższe wektorowi obiektu (zwany neuronem zwycięzcą) na podstawie wybranej metryki odległości. Wagi neuronów są następnie aktualizowane, aby lepiej odwzorować dane wejściowe, z uwzględnieniem funkcji sąsiedztwa, która definiuje, w jakim stopniu zmiany wagi jednego neuronu wpływają na jego sąsiadów.

Kroki SOM przedstawione w Algorytmie 1 obejmują aktualizację wag neuronów w celu lepszego odwzorowania danych wejściowych. Z największą intensywnością aktualizowane są wagi neuronu zwycięzcy, z mniejszą intensywnością neurony w najbliższym sąsiedztwie, a najsłabiej te, które znajdują się najdalej od neuronu zwycięzcy. Odpowiada za to funkcja sąsiedztwa (*ang. neighborhood function*), która definiuje, w jakim stopniu zmiany wag neuronu zwycięzcy wpływają na jego sąsiadów. W sieci SOM są używane różne funkcje sąsiedztwa: prostokątna, gaussowska, ucięta funkcja gaussowska, trójkątna, wykładnicza i wiele innych. Algorytm tworzy uporządkowaną mapę cech. Odwzorowanie wektorów danych przez wektory wag neuronów jest istotą mechanizmu samouczenia się sieci. Neurony przesuwają się w kierunku tej części przestrzeni, w której skupiają się badane jednostki.

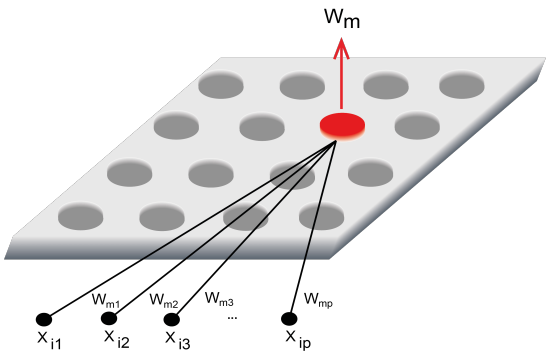
Tabela 4.2: Reprezentacja wektorów wag neuronów w sieci SOM. Źródło: opracowanie własne.

Neuron		Zmienna (cecha, atrybut)			
$(m = 1, ..., m_{max})$		1	2	$\cdots$	p
1	w_1	w_{11}	w_{12}	$\cdots$	w_{1p}
2	w_2	w_{21}	w_{22}	$\cdots$	w_{2p}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
m_{max}	$w_{m_{max}}$	$w_{m_{max}1}$	$w_{m_{max}2}$	$\cdots$	$w_{m_{max}p}$

Zbiór obiektów pogrupowanych przy użyciu sieci SOM jest przedstawiony jako macierz danych D. Zbiór ten zawiera n obiektów X_i $(i = 1, 2, \ldots, n)$, z których każda jest opisana przez p cech A_j $(j = 1, 2, \ldots, p)$. Każdy wektor $x_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]$ w $\mathbb{R}^p$ może być traktowany jako obiekt w przestrzeni p-wymiarowej. Sieć SOM składa się z dwóch warstw: warstwy wejściowej z wektorami danych x_i oraz warstwy wyjściowej, która tworzy m zadeklarowanych neuronów $(m = 1, 2, \ldots, m_{max})$. Neurony umieszczone w węzłach sieci są połączone ze wszystkimi elementami z warstwy wejściowej poprzez wektor wag: $w_m = [w_{m1}, w_{m2}, \ldots, w_{mp}]$. W związku z tym, każdy neuron m-ty traktowany jest jak obiekt w przestrzeni p-wymiarowej. Tabela 4.2 przedstawia reprezentację wektorów wag neuronów w sieci SOM. Rysunek 4.9 pokazuje schemat połączenia wektora danych z neuronem w sieci SOM.

Ocena jakości mapy w sieci SOM

Ocena jakości jednostek mapy w sieci SOM jest ważnym elementem analizy. Tradycyjna metoda Kohonena zaleca użycie błędu kwantyzacji (QE) do oceny dopasowania między



Rysunek 4.9: Diagram przedstawiający połączenie wektora danych z neuronem w sieci SOM. Źródło: opracowanie własne.

węzłami a próbkami danych. QE jest obliczany jako suma odległości między węzłami a próbkami danych, gdzie niższe wartości wskazują na lepsze dopasowanie. Warstwa konkurencyjna sieci generuje przestrzennie uporządkowaną mapę na podstawie podobieństwa analizowanych jednostek, co sprawia, że eksploracja danych w sieci SOM jest bardziej efektywna niż w tradycyjnym grupowaniu. Błąd kwantyzacji jest zdefiniowany jako:

$$QE = d(\mathrm{x}_i, w_m), \tag{4.14}$$

gdzie $d(\mathrm{x}_i, w_m)$ reprezentuje odległość między wektorem danych x_i a wektorem wag zwycięskiego neuronu m.

Wysoki błąd kwantyzacji w sieci SOM wskazuje, że dana jednostka jest bardziej oddalona od swojego zwycięzcy w porównaniu z jednostkami o niższym błędzie kwantyzacji. Obiekty charakteryzujące się wysokim błędem kwantyzacji (czyli te, które są daleko od swoich neuronów zwycięzców) są uznawane za odstające. Średni błąd kwantyzacji (MQE) jest miarą jakości uczenia się i dopasowania sieci SOM do zbioru jednostek. MQE jest obliczany jako średnia odległość między każdą jednostką a jej najbliższym neuronem. Średni błąd kwantyzacji (MQE) jest zdefiniowany następująco [262]:

$$MQE = \frac{\sum_{i=1}^{n} d(\mathrm{x}_i, w_m)}{n}, \tag{4.15}$$

gdzie $d(\mathrm{x}_i, w_m)$ reprezentuje odległość (błąd kwantyzacji dla i-tej jednostki) w wybranej metryce między wektorem danych x_i a wektorem wag zwycięskiego neuronu m. Średni błąd kwantyzacji powinien być jak najmniejszy, aby zapewnić dobre dopasowanie sieci do danych.

Oprócz błędu kwantyzacji, istnieje wiele innych miar oceny jakości sieci SOM. Błąd topograficzny TE (*ang. topographic error*, TE) [276] mierzy topograficzne uporządkowanie neuronów, produkt topograficzny TP (*ang. topographic product*, TP) [277, 278] ocenia, czy rozmiar sieci jest odpowiednio dopasowany do zbioru danych, a błąd dystor-

sji DM (*ang. distortion measure*, DM) [279, 262, 280] informuje o lokalnej zmienności wektorów danych. Współczynnik Kaskiego-Lagusa (*ang. Kaski–Lagus measure*, KLM) [281] ocenia, jak dobrze drugi najbliższy neuron odwzorowuje jednostki, zlogarytmowany współczynnik Nasha-Sutcliffa (*ang. logarithmized Nash–Sutcliffe coefficient of efficiency*, CEEFlog) [282] i indeks Willmotta (*ang. Willmott's index of agreement*, IAg) [283, 284] mierzą dokładność odwzorowania, natomiast średni absolutny błąd procentowy (*ang. mean average percentual error*, MAPE) ocenia procentową różnicę między danymi a neuronami. Każda z tych miar dostarcza unikalnych informacji na temat różnych aspektów jakości odwzorowania sieci SOM. Dalsze szczegóły można znaleźć w literaturze naukowej, w tym w pracach [285, 263, 262].

Tematyka algorytmu SOM była omawiana wraz ze współautorem w artykułach [264, 286]. W pierwszym artykule skupiono się na porównaniu dwóch algorytmów wykrywania anomalii: LOF i SOM oraz analizie wpływu różnych parametrów uczenia na ich skuteczność. Wyniki pokazały, że SOM jest bardziej efektywny czasowo niż LOF, dostarczając jednocześnie zadowalających wyników w wykrywaniu anomalii. W badaniu porównano wyniki wykrywania anomalii za pomocą obu algorytmów, analizując, czy rodzaj danych (jakościowe w porównaniu do ilościowych) wpływa na efektywność wykrywania anomalii. Drugi artykuł dotyczył inteligentnych systemów wspomagania decyzji opartych na bazach wiedzy z regułami. Systemy te wykorzystują reguły „Jeśli warunek, to decyzja" jako formę reprezentacji wiedzy. Proces wnioskowania, który naśladuje ludzki proces rozumowania, polega na identyfikacji reguł potwierdzających fakty, generując tym samym nową wiedzę. W badaniu wykorzystano zarówno algorytm LOF, jak i SOM do wykrywania odstających reguł. Eksperymenty potwierdziły skuteczność obu algorytmów i obejmowały porównanie wyników uzyskanych za pomocą tych metod.

4.5.2 AE - algorytm autoenkodera

Autoenkodery (*ang. autoencoders*, AE) to architektura wykorzystywana w różnych typach uczenia bez nadzoru, takich jak faktoryzacja macierzy, analiza składowych głównych oraz redukcja wymiarów, co oznacza, że nie wymagają oznaczonych danych podczas treningu. W kontekście głębokiego uczenia są one uznawane za potencjalnie skuteczny model uczenia, umożliwiający zaawansowane nieliniowe kodowanie i/lub odporną ekstrakcję cech [239, 287]. Autoenkodery wykorzystują algorytm propagacji wstecznej do uczenia się cech i automatycznie uczą się istotnych cech z danych, eliminując konieczność ręcznej inżynierii cech. To oszczędza znaczną ilość czasu i wysiłku w przetwarzaniu wstępnym [288]. AE wyróżniają się również zdolnością do wykrywania anomalii przez mierzenie błędu rekonstrukcji. Obserwacje z wysokimi błędami rekonstrukcji są identyfikowane jako anomalie, co pomaga w wykrywaniu odstających lub nieprawidłowych danych [289, 290, 291]. Artykuł [292] dostarcza wszechstronnego przeglądu autoenkoderów, zaczynając od omówienia zasad funkcjonowania konwencjonalnych autoenkoderów i ich podstawowych mechanizmów, aż po szczegółowy opis głównych etapów ich ewolucji i zastosowań w uczeniu

maszynowym. W artykule znajduje się także podrozdział poświęcony wykrywaniu anomalii, który nie tylko krótko omawia to zagadnienie, ale również prezentuje odniesienia do współczesnych prac z lat 2020–2023, ukazujących zarówno podejścia nadzorowane, jak i nienadzorowane.

Choć obecnie popularne, autoenkodery mają swoje korzenie w badaniach nad sieciami neuronowymi sięgającymi lat 80. XX wieku i od dziesięcioleci stanowią istotny komponent ewolucji tych sieci. Kilka kluczowych prac i badaczy przyczyniło się do rozwoju tej technologii. Jedną z najważniejszych prac w dziedzinie sztucznej inteligencji i uczenia maszynowego jest ta, która wprowadza i opisuje procedurę zwaną propagacją wsteczną. Procedura ta, opracowana przez Rumelharta, Hintona i Williamsa, stała się podstawowym algorytmem trenowania sieci neuronowych i jest fundamentalna dla trenowania autoenkoderów [157]. Koncepcja AE została pierwotnie wprowadzona w pracy badawczej [293]. W 1987 roku LeCun w swojej pracy [154] przedstawił również wczesne koncepcje autoenkoderów, skupiając się na algorytmach uczenia (propagacji wstecznej), architekturze sieci wielowarstwowych oraz koncepcji redukcji wymiarów i reprezentacji danych. Chociaż praca Hopfielda [294] nie dotyczy bezpośrednio autoenkoderów, zaprezentował on sieci neuronowe zdolne do przechowywania wzorców, co miało wpływ na późniejsze prace nad autoenkoderami. W 1988 roku Bourlard i Kamp w pracy [155] pokazali, że płytki autoenkoder (tj. autoenkoder z tylko jedną w pełni połączoną warstwą ukrytą), z liniową funkcją aktywacji wyjścia i funkcją kosztu MSE, uczy się wag, które obejmują tę samą podprzestrzeń co wektory składowych głównych [295].

Podstawową ideą autoenkodera jest posiadanie warstwy wyjściowej o tej samej liczbie wymiarów co warstwa wejściowa. Celem jest dokładne odtworzenie każdego wymiaru poprzez przepuszczenie go przez sieć. Autoenkoder replikuje dane z wejścia na wyjście, dlatego czasami nazywany jest replikacyjną siecią neuronową, inaczej można powiedzieć, że jest asocjatorem wykonującym projekcję identycznościową, gdzie wyjście jest identyczne z wejściem. Proces adaptacji wag w sieci autoenkodera odbywa się za pomocą metody propagacji wstecznej. Choć na pierwszy rzut oka odtworzenie danych może wydawać się prostym zadaniem, ograniczenie liczby neuronów w warstwach środkowych czyni je znacznie bardziej skomplikowanym. Liczba neuronów w każdej środkowej warstwie jest zazwyczaj mniejsza niż w warstwie wejściowej lub wyjściowej, co prowadzi do uproszczonej reprezentacji danych. W efekcie warstwa końcowa nie jest w stanie dokładnie odtworzyć oryginalnych danych, co skutkuje stratami. Funkcja straty w autoenkoderze, oparta na sumie kwadratów różnic między wejściem a wyjściem, wymusza, aby wyjście było jak najbardziej zbliżone do wejścia. Ponieważ model musi określić, które aspekty wejściowe są najważniejsze do odtworzenia, zmusza to autoenkoder do przechwycenia najistotniejszych cech, co w efekcie prowadzi do nauki przydatnych cech danych szkoleniowych. W warstwie ukrytej są neurony, które uczą się oszczędnie reprezentować dane wejściowe, aby móc je ponownie odtworzyć. Aby to było możliwe, muszą uchwycić wspólne i podobne cechy, co prowadzi do tworzenia reprezentacji najważniejszych właściwości

danych uczących. W rezultacie dochodzi do neuronowej kompresji informacji. Głównym zadaniem jest identyfikacja cech o wysokiej jakości (mocnych, niezmiennych i dobrze dyskryminujących).

Na rysunku 4.10 przedstawiono schemat autoenkodera, w którym wyjściowe jednostki są bezpośrednio związane z jednostkami wejściowymi. Zredukowana reprezentacja danych, określana również jako kod, jest warstwą, której liczba jednostek określa wymiar redukcji. Pierwsza część architektury, znajdująca się przed wąskim gardłem, nazywana jest enkoderem, gdyż generuje zredukowany kod. Ostatnia część architektury to dekoder, który rekonstruuje dane z kodu. Zadaniem autoenkodera jest zakodowanie i-wymiarowych danych wejściowych x_i o p cechach do h-wymiarowej reprezentacji, gdzie $h < i$, a następnie zdekodowanie (odtworzenie) danych wyjściowych $\hat{x}_i$ na warstwie wyjściowej. Proces uczenia polega na iteracyjnym kodowaniu i dekodowaniu, aż błąd rekonstrukcji (*ang. reconstruction error*, RE) zostanie zminimalizowany, co pozwala na uzyskanie optymalnej reprezentacji danych wejściowych. Skutecznie przeszkolony autoenkoder powinien być zdolny do kompresji danych do ukrytej przestrzeni i ich dekodowania do oryginalnej postaci z minimalną utratą informacji.

Koder:

$$h = f(x_i) = \sigma(Wx_i + b), \tag{4.16}$$

gdzie b to wektor obciążeń w warstwie kodera, dodawany do wyniku iloczynu macierzowego wag W i danych wejściowych x_i.

Dekoder:

$$\hat{x}_i = g(h) = \sigma(W'h + b'), \tag{4.17}$$

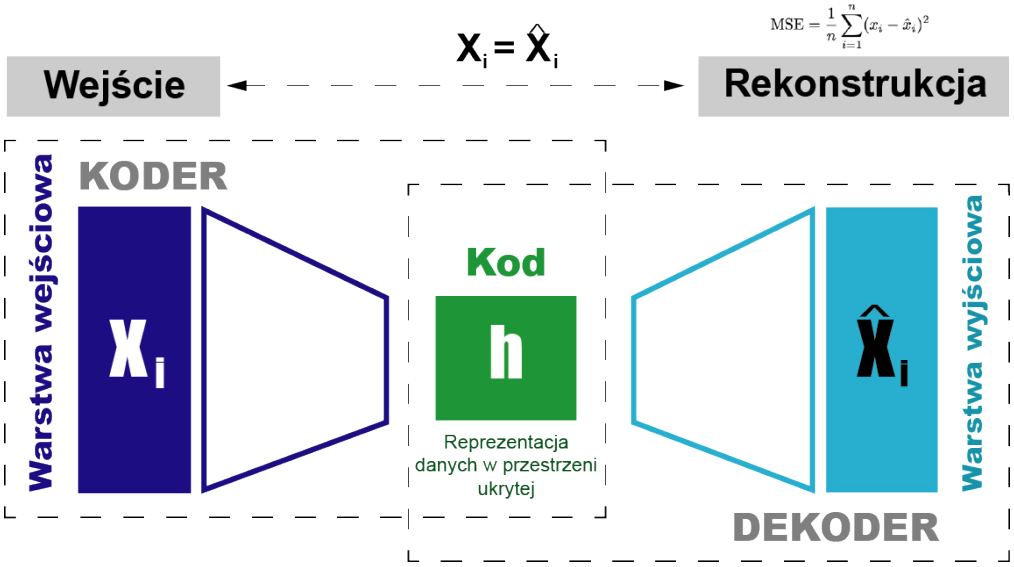
gdzie b' to wektor obciążeń w warstwie dekodera, podobnie dodawany do wyniku iloczynu macierzowego wag W' i zakodowanej reprezentacji h.

Wektory obciążeń w każdej warstwie umożliwiają przesunięcie granicy decyzyjnej sieci neuronowej, zwiększając zdolność modelu do adaptacji i uczenia się różnorodnych wzorców danych. Dzięki temu model może skuteczniej dopasować dane, nawet gdy nie są one liniowo separowalne w oryginalnej przestrzeni danych. W autoenkoderach często implementuje się zasadę współdzielenia wag między koderem a dekoderem. Precyzyjniej, macierz wag dekodera (W') jest transpozycją macierzy wag kodera (W). Stosując tę zasadę, przyjmuje się, że:

$$W' = W^T$$

Macierz W mapuje dane wejściowe x_i na zakodowaną reprezentację h, natomiast W^T odwzorowuje zakodowaną reprezentację h z powrotem na rekonstrukcję danych $\hat{x}_i$.

Koszt rekonstrukcji to miara, która ocenia, jak dobrze autoenkoder odtwarza dane wejściowe na wyjściu. Podstawowe metody są przedstawione na następnej stronie:



Rysunek 4.10: Ogólny schemat podstawowej architektury autoenkodera. Źródło: opracowanie własne.

MSE - funkcja straty dla danych rzeczywistych:

$$L(\mathrm{x}_i, \hat{\mathrm{x}}_i) = \frac{1}{2}\sum_j (x_{i,j} - \hat{x}_{i,j})^2 \tag{4.18}$$

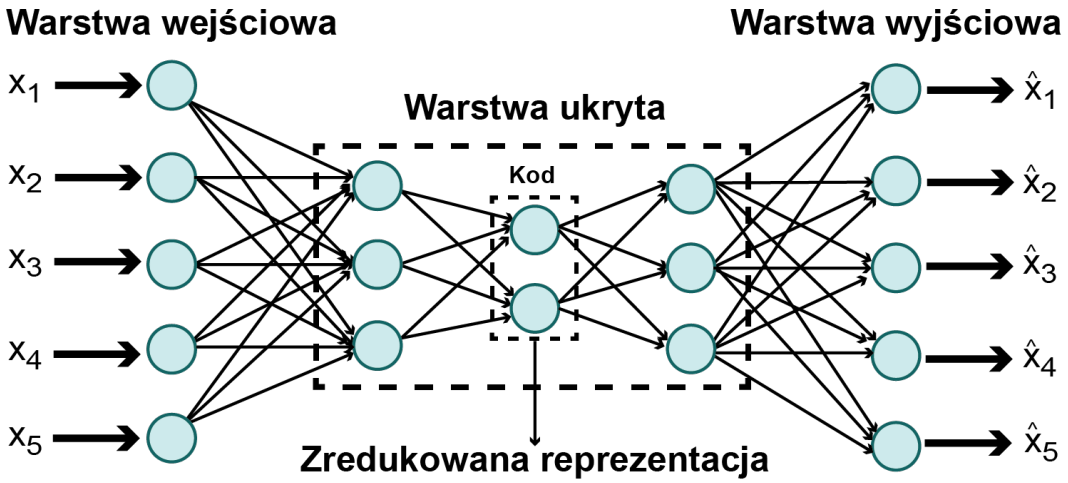
Entropia krzyżowa - funkcja straty dla danych binarnych:

$$L(\mathrm{x}_i, \hat{\mathrm{x}}_i) = -\sum_j (x_{i,j}\log(\hat{x}_{i,j}) + (1 - x_{i,j})\log(1 - \hat{x}_{i,j})), \tag{4.19}$$

gdzie i oznacza indeks obiektu w zbiorze danych D, a j oznacza indeks cechy w danym obiekcie. X_i reprezentuje cały obiekt w zbiorze danych, natomiast x_i to wektor cech tego obiektu, zawierający wartości poszczególnych zmiennych $[x_{i,1}, x_{i,2}, \ldots, x_{i,j}]$.

Autoenkoder może posiadać więcej warstw ukrytych, co pozwala na bardziej złożoną hierarchiczną reprezentację danych, jak pokazano na rysunku 4.11. Reprezentacja najbardziej wewnętrznej warstwy ukrytej jest hierarchicznie powiązana z warstwami zewnętrznymi, co umożliwia skuteczną hierarchiczną redukcję danych. Takie podejście zapewnia stopniowe wyodrębnianie i kompresowanie cech wejściowych na różnych poziomach abstrakcji, co jest przydatne w zadaniach związanych z przetwarzaniem złożonych struktur. Konieczne jest, aby niektóre z warstw głębokiego autoenkodera używały nieliniowej funkcji aktywacji, aby zwiększyć jego zdolność reprezentacyjną, gdyż żadna dodatkowa zdolność nie jest uzyskiwana przez sieć wielowarstwową, jeśli używa się jedynie liniowych aktywacji.

Głębokie sieci neuronowe, dzięki swojej złożonej architekturze i nieliniowym funkcjom aktywacji, wspierają zaawansowaną redukcję wymiarów, przewyższając tradycyjne metody w tej dziedzinie [151]. Redukcja wymiarów jest ściśle związana z wykrywaniem anomalii, ponieważ obiekty odstające trudno jest zakodować i dekodować bez utraty istotnych informacji. W tradycyjnym uczeniu maszynowym, gdzie stosuje się metody takie jak PCA czy SVD (*ang. singular value decomposition*, SVD), istnieją pewne ograniczenia, które sieci neuronowe potrafią przezwyciężyć. Chociaż w tradycyjnym uczeniu maszynowym istnieją różne metody nieliniowej redukcji wymiarów, sieci neuronowe oferują pewne przewagi, takie jak większa elastyczność i możliwość bardziej precyzyjnej kontroli nad właściwościami danych reprezentacji. Algorytm wstecznej propagacji jest niezbędny, ponieważ wspiera efektywne trenowanie głębokich sieci neuronowych, pomagając w obliczaniu skomplikowanych kroków gradientu.



Rysunek 4.11: Schemat podstawowy autoenkodera z trzema warstwami ukrytymi. Wyjście warstwy kodu (*ang. code*) zapewnia zredukowaną reprezentację. Źródło: opracowanie własne na podstawie [151].

Szczegółowy przebieg algorytmu

Kroki AE przedstawione w algorytmie 2 można opisać następująco. Na początku wagi i biasy są inicjalizowane losowo. Dane wejściowe są przekształcane przez warstwę kodera do ukrytej reprezentacji, a następnie przez warstwę dekodera do zrekonstruowanych danych wyjściowych, co stanowi proces propagacji w przód. Różnica między danymi wejściowymi a zrekonstruowanymi danymi, mierzona za pomocą funkcji straty, określa błąd rekonstrukcji. W procesie propagacji wstecznej gradienty błędu są obliczane i używane do aktualizacji wag i biasów, aby zminimalizować błąd rekonstrukcji. Proces ten jest powtarzany przez określoną liczbę epok, aż wagi i biasy zostaną zoptymalizowane. Po zakończeniu procesu

szkolenia optymalizowane wagi i biasy autoenkodera są używane do efektywnej kompresji i rekonstrukcji danych wejściowych. To pozwala autoenkoderowi na odtwarzanie danych z minimalnym błędem, co jest szczególnie przydatne w zadaniach takich jak wykrywanie anomalii, redukcja wymiarów czy usuwanie szumów.

Algorytm 2: Algorytm uczenia autoenkodera

Input: Zbiór danych D z próbkami X_i, liczba neuronów ukrytych h, liczba epok t, współczynnik uczenia α

Output: Wagi neuronów autoenkodera po zakończeniu uczenia

```
Inicjalizuj wagi neuronów losowo
for każdą epokę n od 1 do t do
    for każdą próbkę X_i z D do
        Koder:
        Oblicz wyjście warstwy ukrytej h = σ(W_1 x_i + b_1)
        Dekoder:
        Oblicz wyjście warstwy wyjściowej x̂_i = σ(W_2 h + b_2)
        Oblicz funkcję straty L(x_i, x̂_i) na podstawie x_i i x̂_i
        Zaktualizuj wagi W_1, W_2 oraz biasy b_1, b_2 na podstawie gradientu ∇L
          i współczynnika uczenia α
return Wagi i biasy neuronów autoenkodera po zakończeniu uczenia
```

Zaimplementowana metoda generuje wektor błędów rekonstrukcji (*ang. reconstruction error*, RE) po zakończeniu wszystkich epok treningowych, tworząc ranking anomalii R_{AE}. Proces ten wykorzystuje cały zbiór danych. Algorytm autoenkodera jest trenowany iteracyjnie przez określoną liczbę epok, używając całego zbioru danych do aktualizacji wag. Aby wykryć anomalie, błędy rekonstrukcji RE są najpierw obliczane dla wszystkich próbek danych wejściowych. Następnie błędy te są sortowane malejąco, a odstępstwa są identyfikowane jako największe błędy na podstawie określonego procenta, na przykład 1% rozmiaru zbioru danych. Te odstępstwa to obiekty o największych błędach rekonstrukcji RE, co sugeruje ich odchylenie od wzorca reszty zestawu danych.

Złożoność obliczeniowa i pamięciowa autoenkoderów przedstawiona w tabeli 4.3 zależy od kilku czynników, takich jak liczba warstw, liczba neuronów w każdej warstwie, wielkość danych wejściowych oraz typ zastosowanych funkcji aktywacji i algorytmów optymalizacji. Autoenkodery bez zaawansowanej optymalizacji mają wielomianową złożoność obliczeniową, zazwyczaj $O(n^2)$ lub więcej, w zależności od struktury sieci, gdzie n oznacza liczbę obiektów danych wejściowych. Jednak zastosowanie technik optymalizacyjnych, takich jak L-BFGS, może zmniejszyć tę złożoność do $O(kn)$, co znacząco przyspiesza proces uczenia [296]. W kontekście zaawansowanych technik optymalizacyjnych, k reprezentuje liczbę dopuszczalnych aktualizacji, czyli iteracji, podczas których parametry są korygowane. W przypadku L-BFGS, jest to liczba przechowywanych korekt,

Tabela 4.3: Złożoność obliczeniowa i pamięciowa autoenkoderów uwzględniająca warstwy. Źródło: opracowanie własne.

Typ złożoności	Równanie	Opis
obliczeniowa	$O\left(t \cdot n \cdot \sum_{i=1}^{L} p \cdot m_i\right)$	Złożoność obliczeniowa, gdzie t to liczba iteracji, n to liczba danych wejściowych, p to liczba wymiarów danych, m_i to liczba neuronów w i-tej warstwie, L to liczba warstw
pamięciowa	$O\left(\sum_{i=1}^{L}(p \cdot m_i + m_i)\right)$	Złożoność pamięciowa sieci neuronowej zależy od liczby neuronów i wymiarów danych, obejmując przechowywanie wag, biasów, danych wejściowych, aktywacji neuronów oraz gradientów podczas treningu
n - liczba obiektów, **p** - liczba wymiarów, **L** - liczba warstw, **m** - liczba neuronów, **t** - iteracje		

które są używane do obliczenia nowej aproksymacji macierzy Hessiana. Inne techniki optymalizacyjne, takie jak Stochastic Gradient Descent (SGD) i Adam, również mogą być użyte do redukcji złożoności obliczeniowej. Powyższe wartości są szacunkowe i mogą się różnić w zależności od specyficznej architektury autoenkodera oraz optymalizacji sprzętowej i algorytmicznej.

Autoenkodery mają wiele hiperparametrów, które należy ustalić przed rozpoczęciem procesu uczenia. Wartości tych parametrów mogą znacząco wpłynąć na wydajność modelu. Niektóre hiperparametry są ustalane przed szkoleniem i pozostają stałe, podczas gdy inne mogą być dynamicznie dostrajane w trakcie szkolenia, aby optymalizować działanie modelu. Wybór i regulacja tych hiperparametrów często wymagają eksperymentowania i walidacji, aby uzyskać najlepsze wyniki dla konkretnego zadania.

Hiperparametry ustalane przed szkoleniem:

- **Liczba ukrytych warstw**: Określa głębokość sieci oraz jej zdolność do uchwycenia złożonych wzorców w danych. Większa liczba warstw może zwiększyć zdolności reprezentacyjne sieci, ale jednocześnie komplikuje proces optymalizacji i zwiększa ryzyko nadmiernego dopasowania,
- **Liczba neuronów w każdej warstwie**: Decyduje o zdolności sieci do reprezentowania danych. Większa liczba neuronów zwiększa zdolności reprezentacyjne sieci, ale wprowadza ryzyko nadmiernego dopasowania i komplikacje w optymalizacji,
- **Rozmiar przestrzeni ukrytej**: Określa liczbę neuronów w najwęższej warstwie au-

toenkodera, która kompresuje dane wejściowe do zwartej reprezentacji. Odpowiedni dobór tego parametru jest istotny dla zachowania równowagi między złożonością modelu a jego wydajnością. Zbyt mała przestrzeń może prowadzić do utraty ważnych informacji, natomiast zbyt duża może skutkować nadmiernym dopasowaniem,

- **Funkcja aktywacji**: Stosowana w warstwie wąskiego gardła. Po zsumowaniu danych wejściowych z uwzględnieniem wag powstaje sygnał pobudzenia. Funkcja aktywacji określa sposób obliczania wartości sygnału wyjściowego neuronu na podstawie tego pobudzenia. W kontekście autoenkoderów najczęściej używane funkcje aktywacji to sigmoidalna, tangensoidalna (*ang. tanh*) oraz ReLU (*ang. rectified linear unit*), która stała się bardzo popularna ze względu na swoje właściwości wspomagające szybkie i efektywne trenowanie głębokich sieci neuronowych,

- **Funkcja celu**: Znana również jako funkcja straty lub koszt rekonstrukcji, jest istotna dla trenowania sieci, ponieważ minimalizuje różnicę między danymi wejściowymi a wyjściowymi. Średni błąd kwadratowy (MSE) jest najczęściej stosowaną funkcją straty w autoenkoderach, kwantyfikującą różnice wejścia–wyjścia,

- **Inicjalizacja wag**: Inicjalizacja wag to proces ustalania początkowych wartości wag neuronów przed rozpoczęciem treningu sieci. Właściwa inicjalizacja może znacząco przyspieszyć proces uczenia i poprawić zbieżność modelu. Aby sieć neuronowa mogła efektywnie się uczyć, jej parametry, w tym wagi, muszą mieć odpowiednie wartości początkowe. Ten proces polega na przypisaniu losowych, przeważnie niewielkich wartości bliskich zeru, do wag neuronów. Dzięki temu sieć może rozpocząć proces optymalizacji, umożliwiając efektywne dostrajanie wag podczas treningu,

- **Liczba epok**: Określa liczbę pełnych przejść przez zestaw danych szkoleniowych. Każda epoka składa się z wielu iteracji, podczas których sieć jest szkolona na różnych podzbiorach danych (*ang. batch*). Liczba epok ma istotny wpływ na dokładność szkolenia i zbieżność modelu.

Hiperparametry dynamicznie dostrajane podczas szkolenia:

- **Współczynnik uczenia się**: Hiperparametr określający wielkość kroku podczas optymalizacji, wpływający na aktualizacje wag i odchyleń oraz szybkość zbieżności funkcji celu. Jest istotny w algorytmach uczenia autoenkoderów, takich jak wsteczna propagacja błędu, gdzie decyduje o wielkości zmian wag w odpowiedzi na gradienty. Zbyt niski współczynnik uczenia powoduje, że proces szkolenia trwa długo, ponieważ sieć aktualizuje wagi powoli, co spowalnia konwergencję do optymalnych wartości. Z kolei zbyt wysoki współczynnik uczenia prowadzi do gwałtownych zmian wag, co może skutkować niestabilnym procesem szkolenia i przeskakiwaniem minimalnych wartości funkcji błędu, uniemożliwiając osiągnięcie konwergencji,

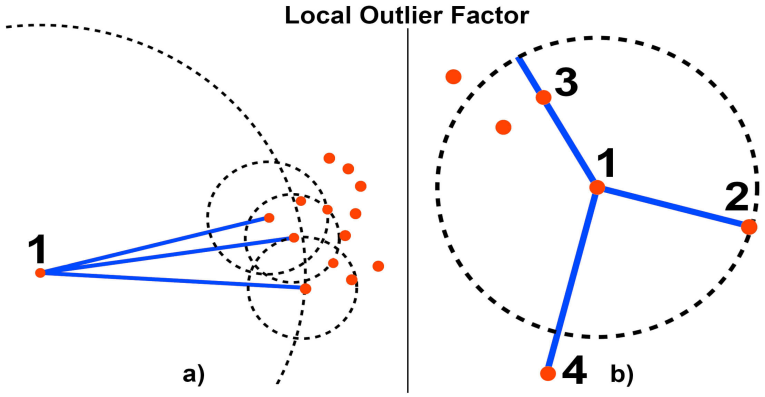
- **Rozmiar partii**: Rozmiar partii (*ang. batch size*) determinuje poziom szumu w gradientach oraz wpływa na efektywność optymalizacji w każdej iteracji. Mniejsze partie wprowadzają większy szum do gradientów, co przyspiesza proces przetwarzania i jest korzystne pod względem wykorzystania pamięci. Większe partie natomiast redukują szum gradientów, co prowadzi do stabilniejszych aktualizacji, ale sprawia, że proces optymalizacji jest wolniejszy i bardziej wymagający pod względem zasobów,
- **Regularyzacja**: Techniki takie jak regularyzacja L1 i L2 są stosowane w celu zapobiegania nadmiernemu dopasowaniu modelu. Regularyzacja wprowadza dodatkowy składnik do funkcji kosztu, który ogranicza złożoność modelu, co pomaga w poprawie jego ogólnej poprawności i stabilności. Te metody zmuszają sieć do nauki bardziej oszczędnych reprezentacji, co prowadzi do lepszej generalizacji oraz redukcji złożoności modelu,
- **Algorytmy optymalizacji**: Do minimalizacji funkcji celu podczas szkolenia wykorzystuje się różne algorytmy optymalizacji, takie jak SGD, Adam czy Adagrad. Te algorytmy są niezbędne do skutecznego trenowania sieci neuronowych, w tym autoenkoderów, umożliwiając szybką i efektywną aktualizację wag w celu osiągnięcia optymalnych wyników.

Te wzajemnie powiązane hiperparametry wymagają starannego doboru, aby osiągnąć optymalną wydajność. Proces ten często wymaga eksperymentowania, ale jest niezbędny dla stworzenia efektywnego modelu autoenkodera. Trzy popularne biblioteki, które są szeroko stosowane do budowania i trenowania modeli autoenkoderów, to TensorFlow, PyTorch i Keras. W tej pracy zdecydowano się jednak na własne implementacje algorytmów bazowych autoenkodera, SOM i LOF, wykorzystując bibliotekę NumPy [297] do operacji na tablicach i macierzach. Taki wybór pozwolił na precyzyjne porównanie podstawowych algorytmów oraz zapewnił pełną kontrolę nad kodem.

4.5.3 LOF - algorytm lokalnego współczynnika osobliwości

Algorytm LOF (*ang. local outlier factor*) jest metodą zaproponowaną przez Markusa M. Breuniga, Hansa-Petera Kriegela, Raymonda T. Nga i Jörga Sandera w 2000 roku [98]. Służy do identyfikowania nietypowych obiektów poprzez mierzenie lokalnego odchylenia obiektu X_i względem jego sąsiadów. Wiele metod wykrywania obserwacji odstających traktuje je jako właściwość binarną, jednakże algorytm LOF przypisuje każdemu obiektowi stopień bycia odstającym. LOF określa, jak izolowany jest obiekt względem swojego otoczenia. Jak sugeruje nazwa algorytmu, LOF mierzy lokalne odchylenie obiektu $X_i \in D$ względem jego k najbliższych sąsiadów. Obiekt X_i jest uważany za anomalię, jeśli jego wartość LOF jest duża. Lokalny współczynnik osobliwości LOF opiera się na koncepcji lokalnej gęstości, gdzie lokalność jest określana przez k najbliższych sąsiadów, których

odległości są używane do szacowania gęstości. Porównując lokalną gęstość obiektu, charakteryzowanego przez jego p cech, z lokalnymi gęstościami jego sąsiadów, można zidentyfikować regiony o podobnej gęstości oraz obiekty, które mają znacznie niższą gęstość niż ich sąsiedzi, co sprawia, że są one uważane za odstające (rysunek 4.12a). Próbka jest uznawana za anomalię, jeśli jej lokalna gęstość jest znacznie mniejsza niż gęstość sąsiadów.



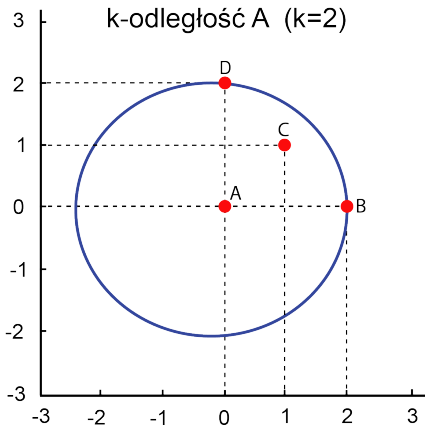
Rysunek 4.12: **a)** Podstawowa idea LOF: porównywanie lokalnej gęstości obiektu z gęstością jego sąsiadów. Obiekt 1 ma znacznie niższą gęstość niż jego sąsiedzi. **b)** Ilustracja odległości osiągalności. Obiekty 2 i 3 mają tę samą odległość osiągalności ($k = 3$), podczas gdy obiekt 4 nie należy do k-najbliższych sąsiadów. Źródło: opracowanie własne.

Załóżmy, że chcemy zidentyfikować odstające obiekty w zbiorze danych. Aby obliczyć LOF dla konkretnego obiektu X_i, należy wykonać kolejne kroki, co ilustruje algorytm 3.

Znajdź k-odległość (*ang. k-distance*) $d_k(X_i)$ pomiędzy X_i a jego k-tym najbliższym sąsiadem. Odległość może być mierzona różnymi miarami, jak Hamminga, kosinusowa, Jaccarda i inne. Chociaż często stosuje się odległość euklidesową, badania koncentrują się na danych jakościowych. Stosuje się kodowanie zero-jedynkowe, przekształcające zmienne kategoryczne w binarne (0 lub 1). Używając odległości Hamminga między dwoma wektorami binarnymi, można określić ich podobieństwo. Im wynik jest bliższy zeru, tym większe podobieństwo. Odległość k-odległość to odległość między X_i a jego k-tym najbliższym sąsiadem, jak pokazano na rysunku 4.13. Jest to miara promienia, w którym znajdują się najbliżsi sąsiedzi X_i. Sąsiedzi oznaczeni jako k-sąsiedzi, obejmują zbiór punktów wewnątrz lub na okręgu o promieniu k-odległości. Liczba k-sąsiadów może być większa lub równa wartości k. Zbiór k-sąsiedzi X_i jest oznaczany jako:

$$N_k(X_i) = \{q \in D - \{X_i\} : d(X_i, q) \leq d_k(X_i)\}, \tag{4.20}$$

gdzie $d(X_i, q)$ oznacza odległość między obiektami X_i i q, a $d_k(X_i)$ to odległość do k-tego najbliższego sąsiada obiektu X_i.



Rysunek 4.13: Na rysunku przedstawiono pojęcie k-odległości, jest to odległość między danym punktem a jego k-tym najbliższym sąsiadem. Sąsiedzi oznaczeni jako k-sąsiedzi, $N_k(A)$, obejmują zbiór punktów leżących wewnątrz lub na okręgu o promieniu równym k-odległości. Jeśli $k = 2$, sąsiadami k punktu A będą C, B i D. Tutaj wartość $k = 2$, ale $|N_2(A)| = 3$. Dlatego $|N_k(A)|$ zawsze będzie większe lub równe k. Źródło: opracowanie własne.

Zdefiniuj odległość osiągalności (*ang. reachability distance*, RD) obiektu q od X_i jako:

$$RD(X_i, q) = \max\{d_k(q), d(X_i, q)\}. \tag{4.21}$$

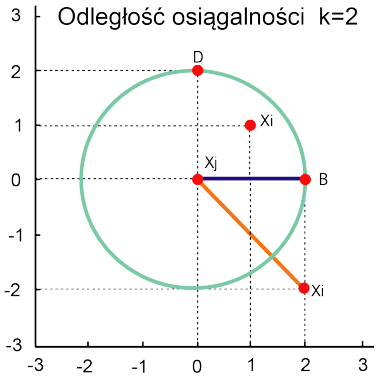
Odległość osiągalności RD jest definiowana jako maksymalna wartość k-odległości X_j i odległości między X_i a X_j, co zilustrowano na rysunku 4.14. Podobnie jak wspomniano we wcześniejszym punkcie, miara odległości zależy od konkretnego problemu (może to być odległość euklidesowa, Manhattan, Gowera itp.).

Oblicz lokalną gęstość osiągalności LRD (*ang. local reachability density*, LRD). LRD określa, jak gęsto otoczony jest dany punkt przez swoje sąsiedztwo. LRD dla punktu X_i jest odwrotnością średniej odległości osiągalności RD jego k-najbliższych sąsiadów. Formalnie, LRD dla punktu X_i jest definiowane jako:

$$LRD(X_i) = \frac{|N_k(X_i)|}{\sum_{q \in N_k(X_i)} RD(X_i, q)}, \tag{4.22}$$

gdzie: $|N_k(X_i)|$ to liczba k-najbliższych sąsiadów punktu X_i, $RD(X_i, q)$ to odległość osiągalności między punktami X_i i q.

Interpretacja LRD jest taka, że punkt X_i o wysokiej wartości LRD jest otoczony gęsto przez inne punkty, co sugeruje, że nie jest on anomalią. Natomiast punkt o niskiej wartości LRD jest bardziej odizolowany, co może wskazywać, że jest anomalią.



Rysunek 4.14: Na rysunku RD obliczana jest jako maksymalna z k-odległości X_j i odległości między X_i a X_j. $RD(X_i, X_j) = \max(\text{k-odległość}(X_j), \text{odległość}(X_i, X_j))$. Jeśli punkt X_i leży w obrębie k-sąsiada X_j, odległość osiągalności to k-odległość X_j (linia niebieska), w przeciwnym razie odległość między X_i a X_j (linia pomarańczowa). Źr.: opr. własne.

Algorytm 3: Algorytm LOF - Local Outlier Factor

Input: Zbiór danych D z próbkami $X_i \in \mathbb{R}^p$, liczba najbliższych sąsiadów k
Output: Wartości LOF dla każdej próbki w D

for *każde* $X_i \in D$ **do**
  Znajdź k najbliższych sąsiadów X_i i oznacz jako $N_k(X_i)$
  for *każde* $q \in N_k(X_i)$ **do**
    Oblicz odległość osiągalności $RD(X_i, q)$
  Oblicz lokalną gęstość osiągalności $LRD(X_i)$
for *każde* $X_i \in D$ **do**
  Oblicz $LOF(X_i)$ na podstawie lokalnych gęstości osiągalności sąsiadów z $N_k(X_i)$
return *Wartości* $LOF(X_i)$ *dla każdej próbki* X_i *w* D

Można teraz obliczyć **lokalny współczynnik osobliwości LOF** (*ang. local outlier factor*). Lokalna gęstość osiągalności LRD jest porównywana z lokalnymi gęstościami osiągalności wszystkich punktów w $N_k(X_i)$, a stosunek tych gęstości jest definiowany jako lokalny współczynnik osobliwości LOF. LOF określa, w jakim stopniu dany punkt X_i jest anomalią w porównaniu do swoich sąsiadów. Formalnie, LOF dla punktu X_i jest (4.23):

$$LOF(X_i) = \frac{\sum_{q \in N_k(X_i)} \frac{LRD(q)}{LRD(X_i)}}{|N_k(X_i)|}, \tag{4.23}$$

gdzie: $LRD(q)$ to lokalna gęstość osiągalności punktu q, $LRD(X_i)$ to lokalna gęstość osiągalności punktu X_i, $|N_k(X_i)|$ to liczba k-najbliższych sąsiadów punktu X_i.

Interpretacja LOF jest taka, że punkt X_i z wartością LOF bliską 1 ma podobną gęstość sąsiedztwa co jego k-najbliżsi sąsiedzi, co sugeruje, że nie jest anomalią. Natomiast punkt X_i z wartością LOF znacznie większą od 1 ma mniejszą gęstość sąsiedztwa niż jego k-najbliżsi sąsiedzi, co może wskazywać, że jest anomalią. Aby uwzględnić k, obliczamy $LOF_k(X_i)$ dla wybranych wartości k i zachowujemy maksymalny $LOF_k(X_i)$. Jeśli $LOF_k(X_i)$ jest duże, to X_i jest uznawane za anomalię. LOF porównuje gęstość danego obiektu z gęstością jego sąsiadów, a ponieważ dane odstające zazwyczaj pochodzą z obszarów o niskiej gęstości, stosunek ten będzie wyższy dla obiektów nietypowych. Zwykle normalny obiekt ma LOF między 1 a 1,5, podczas gdy obserwacje odstające mają znacznie wyższe wartości LOF. Im wyższy LOF, tym większe prawdopodobieństwo, że próbka danych jest anomalią.

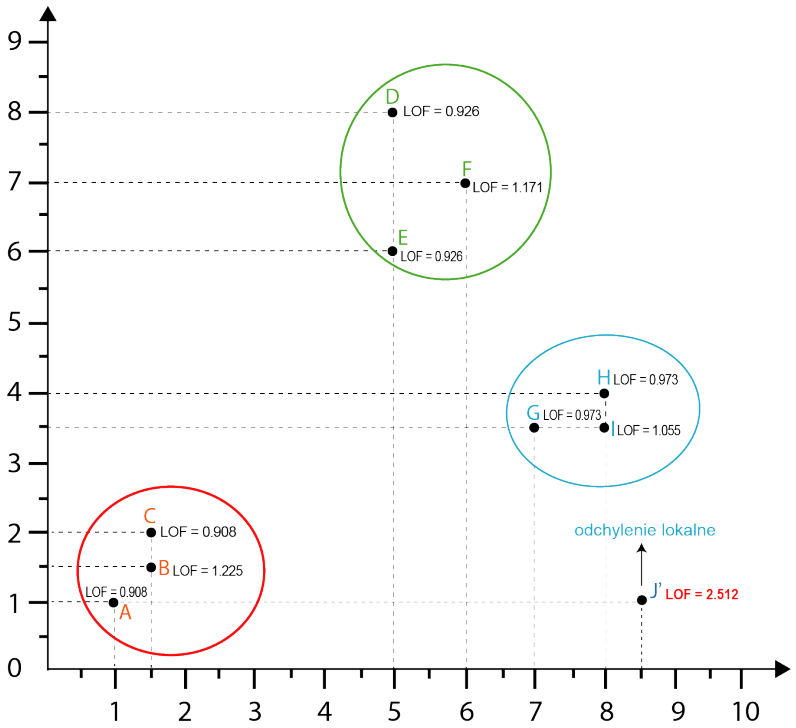
W artykule twórców LOF [98] używane jest oznaczenie MinPts, które oznacza minimalną liczbę punktów potrzebną do określenia lokalnej gęstości osiągalności LRD. W poniższych oznaczeniach skorzystano z k, ponieważ k reprezentuje liczbę najbliższych sąsiadów branych pod uwagę przy obliczaniu LRD i LOF. Wartości k i MinPts są w tym kontekście tożsame i odnoszą się do tej samej liczby punktów wykorzystywanych do analizy lokalnej gęstości. Pojęcie MinPts pochodzi z algorytmu DBSCAN, który również używa tego parametru do definiowania minimalnej liczby punktów w sąsiedztwie, aby móc uznać obszar za gęsty (tj. część grupy). Stosowanie k jest bardziej intuicyjne, ponieważ bezpośrednio odnosi się do liczby najbliższych sąsiadów.

Tabela 4.4: Złożoność obliczeniowa i pamięciowa algorytmu LOF uwzględniająca liczbę wymiarów danych wejściowych. Źródło: opracowanie własne.

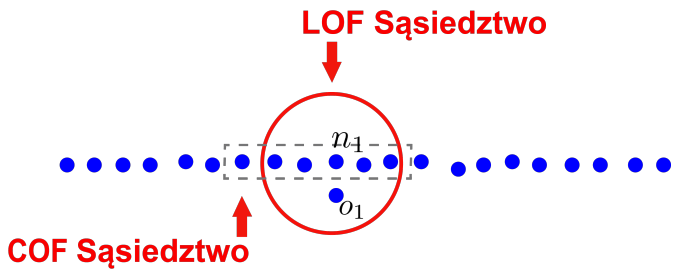
Typ złożoności	**Równanie**	**Opis**
obliczeniowa (indeksowanie)	$O(n \cdot p \cdot \log n + n \cdot k \cdot p)$	Złożoność obliczeniowa dla danych o średniej liczbie wymiarów przy użyciu struktury indeksowej, gdzie n to liczba obiektów w zbiorze danych D, k to liczba najbliższych sąsiadów, p to liczba wymiarów danych
obliczeniowa	$O(n^2 \cdot p)$	Złożoność obliczeniowa bez użycia struktury indeksowej
pamięciowa	$O(n \cdot k + n)$	Złożoność pamięciowa związana z przechowywaniem listy k-najbliższych sąsiadów oraz wartości lokalnej gęstości osiągalności (lrd) dla każdego z n obiektów
D - zbiór danych, **n** - liczba obiektów, **k** - liczba najbliższych sąsiadów, **p** - liczba wymiarów		

Złożoność obliczeniowa i pamięciowa algorytmu LOF przedstawiona w tabeli 4.4 zależy od kilku czynników, takich jak liczba obiektów w zbiorze danych, liczba wymiarów danych oraz liczba najbliższych sąsiadów. Algorytm LOF przy użyciu struktury indeksowej ma złożoność obliczeniową $O(n \cdot p \cdot \log n + n \cdot k \cdot p)$, natomiast bez użycia struktury indeksowej złożoność wynosi $O(n^2 \cdot p)$. Złożoność pamięciowa algorytmu LOF jest rzędu $O(n \cdot k + n)$, co obejmuje przechowywanie listy k-najbliższych sąsiadów oraz wartości lokalnej gęstości osiągalności (LRD) dla każdego z n obiektów. W praktyce, wybór odpowiedniej struktury indeksowej oraz optymalizacji algorytmicznej może znacząco wpłynąć na złożoność obliczeniową i pamięciową algorytmu LOF. Stosowanie struktur indeksowych, takich jak drzewa kd (*ang. k-d trees*) lub drzewa R (*ang. r-trees*), może poprawić efektywność przetwarzania, szczególnie dla dużych zbiorów danych o wysokiej liczbie wymiarów. Dodatkowo, implementacja optymalizacji sprzętowych, takich jak równoległe przetwarzanie lub przyspieszenie GPU, może przyczynić się do dalszego zmniejszenia złożoności obliczeniowej i czasu wykonywania algorytmu LOF. Wybór optymalnej konfiguracji zależy od specyfiki danego zbioru danych oraz wymagań aplikacji.

Rozważmy dwuwymiarowy zbiór danych przedstawiony na rysunku 4.15. W tym zbiorze obiekt J' jest oznaczony jako lokalna anomalia. Rysunek ilustruje wartości współczynników LOF dla poszczególnych obiektów. Dokładne omówienie i przedstawienie tego przypadku znajduje się w artykule opublikowanym we współautorstwie [101]. Artykuł między innymi przedstawia praktyczny przykład działania algorytmów LOF oraz COF (*ang. connectivity-based outlier factor*, COF) [298] na dziesięciu punktach danych, ilustrując proces obliczania wartości LOF i COF dla każdego punktu w sześciu krokach. COF jest algorytmem służącym do wykrywania anomalii w zbiorach danych poprzez analizę lokalnych właściwości obiektów i ich najbliższych sąsiadów. COF ocenia obiekty na podstawie ich łączności w grafie najbliższych sąsiadów, umożliwiając identyfikację obiektów odstających w danych o strukturze liniowej, podczas gdy LOF mierzy różnicę w gęstości lokalnej obiektu w porównaniu do gęstości jego sąsiadów. COF jest w stanie uchwycić obszary, takie jak linie proste, co pokazuje rysunek 4.16. W praktyce wybór między LOF a COF zależy od specyfiki zbioru danych i rodzaju anomalii, które chcemy wykryć. COF może być bardziej odpowiedni w sytuacjach, gdzie połączenia między danymi obiektami odgrywają istotną rolę, podczas gdy LOF sprawdza się lepiej w środowiskach o zróżnicowanej gęstości. Ponadto, artykuł analizuje zastosowanie czterech algorytmów wykrywania anomalii, w tym LOF, do identyfikacji nietypowych reguł w bazach wiedzy. W badaniu wykorzystano metody wykrywania anomalii, aby zidentyfikować określoną liczbę anomalii w regułach, z algorytmem LOF jako jednym z głównych narzędzi do tego celu. Wyniki pokazały, że zarówno LOF, jak i COF są skutecznymi narzędziami do wykrywania anomalii w zbiorach danych i mogą być z powodzeniem stosowane do analizy reguł w bazach wiedzy.



Rysunek 4.15: Wizualizacja wartości współczynników LOF dla dwuwymiarowego zbioru danych: Punkt J' jako lokalna anomalia. Źródło: opracowanie własne.



Rysunek 4.16: Ilustracja różnic między obliczeniami COF i LOF. COF skutecznie identyfikuje obszary, takie jak linie proste. Na rysunku $n1$ reprezentuje normalny punkt danych, a $o1$ reprezentuje obiekt odstający (anomalię). Źródło: opracowanie własne na podstawie [299].

Na temat LOF można znaleźć wiele prac naukowych. Przykładowo, w artykule Alghushairy'ego i współpracowników [300] przedstawiono przegląd algorytmów wykrywania lokalnych anomalii, ze szczególnym uwzględnieniem algorytmu LOF. Zawiera on analizę istniejących algorytmów wykrywania lokalnych anomalii w środowiskach statycznych i strumieniowych oraz omawia ich zalety i ograniczenia. Proponowane są również kierunki

rozwoju ulepszonych metod wykrywania lokalnych anomalii dla strumieni danych. W innym artykule [301] poruszany jest temat różnorodności wyników algorytmów wykrywania odchyleń, takich jak LOF. Główna idea polega na ujednoliceniu wyników tych algorytmów przez przekształcenie ich na skalę od 0 do 1, co ułatwia ich interpretację i porównanie. LOF to jeden z omawianych algorytmów, a artykuł przedstawia metody normalizacji i poprawy interpretacji wyników zarówno tego algorytmu, jak i innych metod, między innymi takich jak kNN, LOCI, ABOD, LDOF. Każdy z tych algorytmów ocenia, w jakim stopniu dany obiekt jest odchyleniem od reszty danych. Większość metod wykrywania anomalii dotyczy statycznych zbiorów danych, gdzie wszystkie dane są dostępne przed rozpoczęciem wykrywania. W rzeczywistości jednak wiele danych napływa w formie strumieniowej, co wymaga analizy w czasie rzeczywistym w dynamicznym i złożonym środowisku. Autorzy [302] zaproponowali algorytm LOF (iLOF), który uwzględnia lokalny wpływ nowych obiektów, dostosowując się do charakterystyki strumieni danych. Publikacja Kriegela i współpracowników [303] przedstawia algorytm LoOP, będący rozwinięciem idei LOF. Algorytm ten wprowadza pojęcie lokalnych prawdopodobieństw odstających, co pozwala na bardziej probabilistyczne podejście do wykrywania anomalii w danych. Zajmuje się problemem interpretacji wyników wskazujących, w jakim stopniu obiekt jest odstający, proponując metodę, która przekłada te wyniki na prawdopodobieństwo.

W kilku artykułach skoncentrowano się na badaniu algorytmu LOF, będącego częścią prowadzonych badań, co przyniosło istotne wyniki w dziedzinie wykrywania anomalii. W artykule [286] przeprowadzono eksperymenty, stosując algorytm LOF do wykrywania reguł odstających w bazach wiedzy. Wyniki jednoznacznie wykazały skuteczność algorytmu LOF w tym kontekście. Dodatkowo, wyniki porównano z algorytmem SOM, co pozwoliło na ocenę skuteczności obu podejść w identyfikacji nietypowych reguł w bazach wiedzy. W kolejnym artykule [304] opisano wykrywanie anomalii w bazach wiedzy dotyczących przypadków COVID-19. Zastosowano dwufazową procedurę. W pierwszej fazie optymalizuje się strukturę grup reguł, uwzględniając obecność reguł odstających w bazie wiedzy. W drugiej fazie, algorytm LOF jest wykorzystywany do identyfikacji reguł odstających. Nietypowe reguły są eliminowane z bazy danych, a wybrane miary jakości grup są sprawdzane pod kątem poprawy po eliminacji tych reguł, co sugeruje, że reguły były słusznie uznane za odstające. Przeprowadzone eksperymenty wykazały skuteczność algorytmu LOF oraz wybranych miar jakości grup w identyfikacji nietypowych reguł. Takie wykrywanie anomalii wspiera inżynierów wiedzy i ekspertów dziedzinowych w eksploracji wiedzy, co przyczynia się do poprawy kompletności bazy wiedzy, która stanowi podstawę systemów wspomagania decyzji. Artykuł [264] zajmuje się problemem wykrywania anomalii przy użyciu algorytmu samoorganizujących się map SOM. W badaniu porównano skuteczność wykrywania anomalii przy użyciu algorytmu LOF z wynikami uzyskanymi za pomocą algorytmu SOM, oceniając skuteczność obu metod w różnych warunkach i na różnych typach danych. Szczególną uwagę poświęcono analizie parametrów uczenia i ich wpływu na dokładność oraz czas szkolenia, co pozwoliło na lepsze zrozumienie, jak te algorytmy

radzą sobie w praktycznych zastosowaniach. W innym artykule [85] przeanalizowano zastosowanie algorytmów LOF i COF do wykrywania anomalii w bazach wiedzy opartych na regułach. W kontekście identyfikacji nietypowych reguł, badano skuteczność tych algorytmów, znajdując określoną liczbę anomalii w regułach (1%, 5%, 10%) oraz oceniając wpływ usunięcia tych anomalii na jakość grup reguł. Eksperymenty przeprowadzone na sześciu różnych bazach wiedzy wykazały, że algorytm COF najczęściej poprawiał jakość grup, wyróżniając się wśród analizowanych wskaźników jakości. W artykule [101] poruszono również problematykę algorytmu LOF. Celem badania była analiza zastosowania czterech algorytmów do wykrywania anomalii w bazach wiedzy opartych na regułach: LOF, COF, K-MEANS oraz autorski algorytm SMALL CLUSTERS. Przeanalizowano siedem różnych wskaźników jakości, które były stosowane do wszystkich reguł, zarówno przed, jak i po usunięciu wybranych anomalii, w celu poprawy jakości grup reguł. W fazie eksperymentalnej wykorzystano sześć różnych baz wiedzy. Najlepsze wyniki, czyli najczęstsze poprawy jakości grup, osiągnięto dla dwóch algorytmów wykrywania anomalii: LOF i COF. W artykule [82] zaproponowano praktyczne rozwiązania w zakresie wykrywania odchyleń w zbiorach danych. W odpowiedzi na brak narzędzi do wykrywania odchyleń przed faktycznym grupowaniem, podjęto zadanie stworzenia w języku R pakietu SOaCRaport. Pakiet ten umożliwia wykrywanie odchyleń w dowolnym zbiorze danych przy użyciu metod LOF i COF, a następnie hierarchiczne grupowanie danych z odchyleniami i bez odchyleń oraz porównanie jakości utworzonych grup. W artykule [305] badania dotyczyły problemu zwiększonego czasu obliczeniowego przy stosowaniu LOF do dużych zbiorów danych. Aby temu zaradzić, zaproponowano optymalizację rozmiaru bloków podczas procesu wykrywania anomalii. Podejście to ma na celu zwiększenie szybkości LOF przy jednoczesnym zachowaniu jego wysokiej skuteczności w identyfikacji anomalii. Ponadto tematyka algorytmu LOF była analizowana w pracach [102, 84].

4.6 Podsumowanie

W tym rozdziale opisano zaawansowane techniki wykorzystywane w uczeniu maszynowym i głębokim w kontekście identyfikacji anomalii, ze szczególnym uwzględnieniem autorskiego systemu Trinity SALT. Omówiono genezę i rozwój sieci neuronowych, co pozwala zrozumieć, jak te metody ewoluowały od podstawowych koncepcji do zaawansowanych modeli, stanowiąc wprowadzenie do prezentowanego systemu. Przedstawiono również ewolucję od tradycyjnego uczenia maszynowego do głębokiego uczenia, podkreślając rozwój zaawansowanych technik przetwarzania danych, które są integralną częścią Trinity SALT. Omówiono wyzwania związane z systemami opartymi na wiedzy oraz zalety samouczących się algorytmów. Na koniec wskazano na rolę głębokich sieci neuronowych w automatycznym wyodrębnianiu cech danych, co znacząco zwiększa efektywność systemów sztucznej inteligencji.

Następnie omówiono ewolucję sieci neuronowych od perceptronów do autoenkoderów, ze szczególnym uwzględnieniem ich zastosowania w wykrywaniu anomalii. Zaczynając od fundamentalnego modelu perceptronu wynalezionego przez Franka Rosenblatta, pokazano, jak jego zasady umożliwiają rozpoznawanie wzorców i uczenie się. Następnie opisano rozwój bardziej złożonych sieci wielowarstwowych, które mogą modelować nieliniowe funkcje, aż do współczesnych autoenkoderów, które efektywnie redukują wymiarowość danych i wykrywają anomalie. Autoenkodery, dzięki swojej zdolności do nieliniowej kompresji danych, są skutecznym narzędziem wykorzystywanym w Trinity SALT do wykrywania anomalii, oferując wyższą skuteczność niż tradycyjne metody takie jak PCA.

W kolejnym podrozdziale podkreślono istotne aspekty systemów uczących się, co pomaga w zrozumieniu ich funkcjonowania, w tym wyzwań związanych ze szkoleniem modeli i ich optymalizacją. Omówiono praktyczne aspekty uczenia maszynowego, koncentrując się na problemach nadmiernego dopasowania i niedopasowania, regularyzacji oraz optymalizacji, a także na znaczeniu hiperparametrów i metod przetwarzania cech. Przedstawiono, jak nadmierne dopasowanie prowadzi do złej generalizacji na nowych danych oraz jak niedopasowanie oznacza, że model nie potrafi uchwycić wzorców w danych. Regularyzacja i optymalizacja są bardzo ważne dla uzyskania stabilnych i efektywnych modeli. Dodatkowo omówiono metody wyboru i dostrajania hiperparametrów oraz podkreślono znaczenie prawidłowej inicjalizacji wag w sieciach neuronowych, co jest istotne dla stabilności i wydajności procesu uczenia oraz skutecznego działania systemu Trinity SALT.

Na koniec omówiono detektory bazowe, takie jak SOM, AE i LOF, które stanowią podstawowe komponenty systemu Trinity SALT. System ten łączy różne techniki wykrywania anomalii, aby uzyskać wszechstronny i skuteczny mechanizm. Każda z metod, SOM, AE i LOF, ma swoje unikalne cechy i zastosowania, co czyni je wartościowymi narzędziami w dziedzinie analizy danych. Detektory bazowe współpracują, co umożliwia skuteczne wykrywanie anomalii w różnych zbiorach danych, zapewniając stabilne i niezawodne wyniki. Omówienie Trinity SALT stanowi istotny wkład w dziedzinę wykrywania anomalii i może być cennym źródłem wiedzy dla badaczy i praktyków zajmujących się analizą anomalii i uczeniem maszynowym.

Rozdział 5

Techniki zespołowe w zaawansowanej identyfikacji anomalii

W poprzednich rozdziałach omówiono różne metody wykrywania anomalii, których efektywność zależy od konkretnego zbioru danych i specyficznego kontekstu użycia. Często okazuje się, że nie istnieje pojedynczy algorytm, który byłby najlepszy we wszystkich przypadkach. W takich sytuacjach analiza zespołowa może być techniką zwiększającą precyzję różnych algorytmów eksploracji danych. Polega ona na łączeniu wyników wielu różnych algorytmów, aby uzyskać jedno spójne rozwiązanie. Główna idea tego podejścia opiera się na założeniu, że pewne algorytmy mogą lepiej radzić sobie z konkretnymi podzbiorami danych, podczas gdy inne będą skuteczniejsze dla innych zestawów danych. Połączenie wyników różnych algorytmów w ramach zespołu pozwala na bardziej niezawodne i wszechstronne działanie, gdyż możliwe jest skorzystanie z zalet różnych metod jednocześnie. Zjawisko to zostało odkryte przez grupę badaczy [306], którzy niezależnie od siebie dążyli do poprawy klasyfikacji, niezależnie od tego, czy używali drzew decyzyjnych [307], sieci neuronowych [308], czy teorii matematycznej [309]. Najbardziej wpływowe wczesne osiągnięcia w tej dziedzinie zostały dokonane przez Breimana [310] z techniką Bagging oraz Freunda i Schapire'a [311] z techniką AdaBoost.

Analiza zespołowa jest powszechnie stosowaną metaheurystyką w wielu problemach eksploracji danych, takich jak klasyfikacja i grupowanie. W literaturze zaproponowano liczne algorytmy oparte na zespołach dla tych zagadnień [306], [312], [313], [314], [306], [315]. W porównaniu do problemów grupowania i klasyfikacji, analiza zespołowa była jednak stosunkowo rzadko badana w kontekście wykrywania anomalii [236]. Wyniki niedawnych metod zespołowych [316], [317], [318], [319] wyraźnie pokazują, że takie metody mogą prowadzić do znacznej poprawy jakości wyników w identyfikacji anomalii. Dlatego analiza zespołowa wydaje się być rozwijającym się obszarem, który może być owocnym kierunkiem badań w celu poprawy jakości algorytmów wykrywania anomalii.

Metody zespołowe trenują wiele modeli bazowych do rozwiązania tego samego problemu. W przeciwieństwie do zwykłych podejść, które starają się stworzyć jeden model bazowy na podstawie danych szkoleniowych, metody zespołowe starają się zbudować zestaw modeli bazowych i połączyć je. Koszt obliczeniowy związany z łączeniem modeli bazowych jest często niewielki, ponieważ większość strategii kombinacji jest prosta. Ten rozdział omawia metody zespołowe, które mają na celu poprawę wydajności poszczególnych modeli bazowych oraz skuteczne radzenie sobie z wysokimi wskaźnikami fałszywie pozytywnych wyników generowanych przez te modele.

5.1 Innowacyjne aspekty analizy zespołowej

Analiza zespołowa w kontekście wykrywania anomalii jest stosunkowo nowym obszarem badań w porównaniu do innych problemów eksploracji danych, takich jak klasyfikacja i grupowanie. Istnieje kilka powodów tej względnej nowości. Choć techniki analizy zespołowej były szeroko stosowane w klasyfikacji i grupowaniu, ich zastosowanie w wykrywaniu anomalii jest znacznie mniej zbadane. Jeśli techniki te były stosowane w algorytmach wykrywania odchyleń, to nie były one wyraźnie rozpoznawane lub deklarowane jako takie. Innymi słowy, były one zintegrowane z algorytmami w sposób ukryty, bez formalnego uznania ich jako niezależnych technik zespołowych. Najwcześniejsze formalne ujęcie analizy zespołowej w identyfikacji anomalii wywodzi się z wykrywania obiektów odstających w danych wysokowymiarowych [203], chociaż nieformalne metody analizy zespołowej dla danych wielowymiarowych były proponowane znacznie wcześniej [33]. Poniżej przedstawiono główne powody, dla których analiza zespołowa w identyfikacji anomlii jest stosunkowo nowym obszarem badań:

- **ograniczone badania w literaturze** - analiza zespołowa, choć intensywnie badana w kontekście klasyfikacji i grupowania, jest znacznie mniej rozwinięta w dziedzinie identyfikacji anomalii. W klasyfikacji i grupowaniu liczne badania i publikacje skupiły się na różnych aspektach analizy zespołowej, takich jak różnorodność modeli, metody łączenia wyników oraz ocena skuteczności zespołów. W przeciwieństwie do tego, w wykrywaniu anomalii istnieje ograniczona liczba badań dotyczących wykorzystania analiz zespołowych. Techniki te były czasami stosowane pośrednio w algorytmach wykrywania obiektów odstających, co oznacza, że były one zintegrowane w ramach algorytmu, ale nie były wyraźnie identyfikowane ani opisywane jako techniki zespołowe. W praktyce oznacza to, że analiza zespołowa była używana w sposób niejawny, bez formalnego uznania jej za metodę zbiorczą ogólnego przeznaczenia. Taki brak formalnego rozpoznania i systematycznego badania tych technik opóźnił rozwój tej dziedziny i ograniczył jej potencjalne zastosowania oraz udoskonalenia w kontekście wykrywania anomalii,

- **brak formalizacji** - wiele użytecznych technik, które mogłyby znacząco poprawić wykrywanie anomalii, nie zostało formalnie uznanych za metody zespołowe. Oznacza to, że techniki te nie były jednoznacznie identyfikowane ani klasyfikowane jako metody zespołowe. Na przykład algorytmy, które łączą wyniki z różnych modeli wykrywania anomalii, mogłyby korzystać z zasad analizy zespołowej, ale ponieważ nie były formalnie rozpoznane jako zespoły, nie były one analizowane pod tym kątem ani rozwijane w sposób systematyczny. Taki brak formalizacji miał kilka negatywnych konsekwencji. Po pierwsze, utrudnił badaczom i praktykom zrozumienie oraz docenienie wartości, jakie mogą przynieść techniki zespołowe w wykrywaniu anomalii. Po drugie, spowodował, że nie powstały spójne ramy teoretyczne ani narzędzia do oceny i porównywania różnych podejść zespołowych. W rezultacie rozwój tej dziedziny został opóźniony, a wiele potencjalnie obiecujących metod nie zostało w pełni zbadanych ani zastosowanych w praktyce. Formalizacja tych technik jako zespoły mogłaby prowadzić do lepszego zrozumienia ich zalet i wad, a także do opracowania nowych, bardziej efektywnych metod wykrywania anomalii. Dalsze badania i rozwój w tym kierunku mogą znacznie przyczynić się do poprawy jakości i skuteczności algorytmów identyfikacji anomalii,

- **kryteria oceny** - techniki zespołowe, aby były skuteczne, muszą być oceniane na podstawie precyzyjnych kryteriów, które pozwalają porównać ich wydajność z bazowymi algorytmami. Jasno zdefiniowane kryteria oceny obejmują metody pomiaru, takie jak na przykład: dokładność, precyzja, czułość i specyficzność, które są niezbędne do obiektywnej oceny skuteczności algorytmów. Jednak w kontekście wykrywania anomalii opracowanie takich kryteriów jest szczególnie trudne z dwóch powodów:

  Po pierwsze, mała liczba próbek stanowi poważne wyzwanie. Zbiory danych używane do wykrywania anomalii często zawierają jedynie niewielką liczbę rzeczywistych obiektów odstających. Ta ograniczona liczba sprawia, że trudno jest uzyskać statystycznie wiarygodne wyniki i ocenić, czy dany algorytm zespołowy rzeczywiście przewyższa algorytm bazowy. Ze względu na małą liczebność próbek nawet niewielkie fluktuacje mogą mieć znaczący wpływ na wynik, co prowadzi do niestabilnych i niewiarygodnych ocen. Również decyzje dotyczące przyszłych kroków algorytmu, takie jak wybór parametrów czy modelu, mogą być oparte na niewystarczających danych, co zwiększa ryzyko nadmiernego dopasowania do nielicznych próbek. Po drugie, brak jasno zdefiniowanych kryteriów oceny wynika z różnorodności technik zespołowych i specyfiki wykrywania anomalii. Każdy zespół algorytmów może różnić się sposobem łączenia wyników, rodzajem wykorzystywanych algorytmów bazowych i typem danych wejściowych. Różne techniki zespołowe mogą mieć odmienne wymagania dotyczące danych, różne metody przetwarzania i integracji wyników oraz różne sposoby radzenia sobie z niepewnością i błędami. To sprawia,

że trudno jest stworzyć uniwersalne kryteria oceny, które byłyby adekwatne dla wszystkich przypadków. Ponadto, same kryteria oceny są często używane w pośrednich etapach algorytmu zespołu (np. boosting lub stacking), aby podejmować przyszłe decyzje dotyczące precyzyjnej konstrukcji zespołu, co utrudnia późniejsze porównywanie ich z algorytmem bazowym.

Ocenianie algorytmów w takich warunkach wymaga opracowania alternatywnych metod oceny, takich jak generowanie syntetycznych danych testowych, które odzwierciedlają różne scenariusze i warunki operacyjne, które mogą pomóc w ocenie skuteczności wykrywania anomalii. Jednak te metody mają swoje własne ograniczenia i mogą nie w pełni odzwierciedlać rzeczywiste warunki. W rezultacie, opracowanie precyzyjnych kryteriów oceny dla technik zespołowych w kontekście wykrywania anomalii jest zadaniem złożonym i wymagającym, co stanowi istotną barierę dla rozwoju i wdrażania tych technik. Aby przezwyciężyć te wyzwania, konieczne są dalsze badania i innowacyjne podejścia, które pozwolą na lepsze zrozumienie i ocenę skuteczności technik zespołowych,

- **brak danych referencyjnych** - nienadzorowany charakter problemu wykrywania anomalii wiąże się z brakiem danych referencyjnych, które są ważne zarówno do oceny jakości komponentów w zespole algorytmów, jak i samego zespołu. W kontekście wykrywania anomalii brak danych referencyjnych, które mogłyby służyć jako punkt odniesienia do oceny wyników algorytmu, oznacza, że nie mamy dostępu do etykiet czy wskazówek określających, które dane są rzeczywistymi anomaliami. Ocenę i walidację algorytmów wykrywających odchylenia znacznie utrudnia brak możliwości jednoznacznego stwierdzenia, czy wykryte anomalie są poprawne.

  Analiza obserwacji odstających jest bardzo trudna do oceny (szczególnie na rzeczywistych zbiorach danych) z powodu połączenia małej liczby anomalii i nienadzorowanego charakteru algorytmów, co dodatkowo komplikuje ocenę, ponieważ brak jest rzeczywistych danych odniesienia do oceny jakości komponentów w zespole. Dodatkowo, brak prawdziwych etykiet uniemożliwia przeprowadzenie standardowych procedur walidacyjnych, takich jak walidacja krzyżowa czy testowanie na zestawach walidacyjnych, które są powszechnie stosowane w problemach nadzorowanych. W takich sytuacjach często polegamy na pośrednich miarach jakości, takich jak stabilność wyników czy zgodność pomiędzy różnymi metodami, ale są one znacznie mniej precyzyjne niż bezpośrednie porównanie z rzeczywistymi anomaliami. Konsekwencją tego jest utrudnione tworzenie złożonych zespołów, ponieważ trudniej jest dokonać optymalnego doboru i kalibracji poszczególnych algorytmów wchodzących w skład zespołu. Wymaga to konstrukcji prostszych zespołów z mniejszą liczbą jakościowych decyzji dotyczących wyboru komponentów w zespole.

W zaawansowanych algorytmach niezbędne jest iteracyjne doskonalenie i ocena poszczególnych komponentów zespołu. W przypadku braku danych referencyjnych trudno jest mierzyć skuteczność i dokładność tych komponentów, co może prowadzić do wyników, które nie w pełni wykorzystują potencjał modeli. Aby przezwyciężyć te wyzwania, konieczne jest rozwijanie nowych metodologii oceny, które mogą lepiej radzić sobie z brakiem danych referencyjnych. Może to obejmować techniki symulacyjne, takie jak syntetyczne generowanie danych, czy metody oparte na analizie jakościowej wyników. Pomimo tych trudności, rozwój takich technik jest ważny dla postępu w dziedzinie wykrywania anomalii i efektywnego wykorzystania zespołów algorytmów w praktyce,

- **subiektywność procesu identyfikacji anomalii** - subiektywność wpływa na decyzje dotyczące wyboru algorytmów, optymalizacji i interpretacji wyników, co utrudnia tworzenie i ocenę skutecznych zespołów. Proces identyfikacji anomalii jest w dużej mierze subiektywny, co oznacza, że sposób definiowania funkcji celu lub modelu dla konkretnego problemu zależy od indywidualnej interpretacji analityka dotyczącej zachowania i struktury danych. Analitycy muszą dokonywać wyborów dotyczących metod i technik na podstawie swojej wiedzy, doświadczenia i zrozumienia danych, co prowadzi do pewnego stopnia subiektywności. Założenia, które analityk przyjmuje na temat generatywnego procesu danych, mogą znacząco wpływać na wyniki analizy. Na przykład, jeden analityk może założyć, że dane są generowane przez proces Gaussowski, podczas gdy inny może przyjąć model oparty na rozkładzie Poissona. Te różnice w założeniach mogą prowadzić do wyboru różnych algorytmów i metod analizy, które będą skuteczne w różnym stopniu w zależności od charakteru danych. Subiektywność ta prowadzi do kilku wyzwań:
  - **ograniczenia algorytmów** - każdy algorytm ma swoje własne założenia dotyczące danych. Na przykład, algorytmy grupowania mogą zakładać, że dane mają określoną strukturę skupień, podczas gdy algorytmy wykrywania odchyleń mogą zakładać, że odchylenia są rzadkie i znacząco różnią się od reszty danych. Jeśli rzeczywiste dane nie spełniają tych założeń, algorytm może działać nieskutecznie,
  - **różnorodność wyników** - różne algorytmy mogą dawać różne wyniki dla tych samych danych, w zależności od przyjętych założeń i metod analizy. To może prowadzić do niejednoznacznych interpretacji i trudności w wyborze najlepszego podejścia,
  - **optymalizacja i dostrojenie** - subiektywne decyzje analityka wpływają na proces optymalizacji i dostrajania modeli. Analityk musi zdecydować, które parametry dostroić i jakie metryki oceny zastosować, co może wpływać na końcową skuteczność modelu. Różni analitycy mogą dojść do różnych wniosków dotyczących najlepszego sposobu dostrojenia modelu,

- **zastosowanie wyników** - wyniki analizy danych muszą być interpretowane w kontekście założeń, które zostały przyjęte. Jeśli założenia są błędne lub nieodpowiednie, wyniki mogą być mylące lub niepoprawne. To wymaga od analityków ostrożności i krytycznego podejścia do interpretacji wyników.

Analiza zespołowa ma duży potencjał w kontekście wykrywania anomalii. Jest to stosunkowo nowy obszar badań ze względu na powyższe przyczyny. Mimo różnic między zespołami do wykrywania anomalii a zespołami klasyfikacyjnymi, badania wykazują, że obie dziedziny dzielą wiele wspólnych cech praktycznych i teoretycznych. Na przykład istnieje możliwość zastosowania zmodyfikowanego kompromisu między błędem systematycznym a wariancją w analizie anomalii, analogicznie jak w klasyfikacji [27].

5.2 Metody zespołowego wykrywania anomalii

Metody zespołowego wykrywania anomalii można kategoryzować według trzech głównych kryteriów:

1. Niezależność komponentów

- **niezależne** - w tej kategorii komponenty zespołu działają niezależnie od siebie. Przykładem może być technika baggingu, gdzie różne modele są trenowane na losowych podzbiorach danych i ich wyniki są łączone w celu uzyskania końcowej decyzji. Niezależność komponentów pozwala na większą różnorodność w wynikach, co może prowadzić do bardziej odpornego systemu wykrywania anomalii,
- **zależne** - komponenty zespołu są od siebie zależne, co oznacza, że wykonanie jednego komponentu wpływa na wyniki kolejnych. Przykładem jest metoda boostingu, gdzie modele są trenowane sekwencyjnie, a każdy kolejny model stara się poprawić błędy poprzedniego. Ta zależność może prowadzić do lepszej dokładności, ale także zwiększa złożoność obliczeniową.

W zespołach niezależnych różne algorytmy lub różne wersje tego samego algorytmu są stosowane na całym zbiorze danych lub jego fragmentach. Wybory dotyczące danych i algorytmów są niezależne od wyników innych algorytmów, a uzyskane wyniki są następnie łączone w celu poprawy dokładności. Kluczową zasadą działania takich zespołów jest to, że różne modele dostarczają unikalne i istotne informacje dotyczące różnych aspektów danych. Połączenie tych informacji prowadzi do bardziej wiarygodnych wyników, które nie są ograniczone specyfiką konkretnego algorytmu czy zestawu danych. Dzięki temu każdy model działa oddzielnie, co pozwala na uzyskanie różnorodnych wyników, które mogą być później łączone w celu poprawy skuteczności i odporności systemu.

W zespołach zależnych, komponenty zespołu wpływają na siebie nawzajem, gdzie wyniki jednego komponentu oddziałują na działanie kolejnych, co prowadzi do bardziej

skoordynowanego procesu uczenia się. Tak jest w zespołach sekwencyjnych, gdzie algorytmy są uruchamiane w określonej kolejności, gdzie każde kolejne wykonanie algorytmu jest uzależnione od wyników uzyskanych przez poprzednie algorytmy. Na przykład, w podejściu dwufazowym, pierwszy algorytm identyfikuje i usuwa oczywiste anomalie, a następnie drugi algorytm tworzy bardziej odporny model na podstawie oczyszczonych danych. Metody te były badane w ramach prac poprzedzających przygotowanie niniejszej rozprawy i znalazły zastosowanie w analizie anomalii opartej na grupach, co umożliwiło budowanie bardziej odpornych grup w późniejszych etapach analizy [102, 82, 84]. Podejście dwufazowe można uznać za metodę zespołową, choć często nie jest tak formalnie nazywane. Takie sekwencyjne zastosowanie algorytmów, gdzie wynik jednego wpływa na działanie kolejnego, wpisuje się w definicję metod zespołowych.

2. Typ komponentu

- **dane** - w tej kategorii komponenty zespołu są definiowane na podstawie wyboru danych. Może to obejmować techniki takie jak boosting czy bagging, gdzie różne podzbiory danych są używane do trenowania modeli. Zespoły zorientowane na dane starają się wykorzystać różnorodność w danych, aby poprawić wykrywanie anomalii,
- **model** - komponenty zespołu są definiowane na podstawie wyboru modeli. Na przykład, różne algorytmy bazowe mogą być używane w podejściu stacking [318], gdzie wyniki różnych modeli są łączone w celu uzyskania końcowej decyzji. Zespoły zorientowane na model koncentrują się na wykorzystaniu różnych algorytmów do uzyskania bardziej wszechstronnego systemu wykrywania anomalii.

W zespołach skoncentrowanych na danych analizowane są różne próbki danych. Każda część dostarcza unikalnych informacji. Poprzez zastosowanie zespołu, który analizuje różne fragmenty danych, można uzyskać różnorodne wyniki, co prowadzi do bardziej kompleksowej i dokładnej analizy. Przykładem zespołu opartego na danych jest praca [203], w której łączy się wyniki wielu algorytmów detekcji anomalii, stosujących różne zestawy cech. Każdy algorytm używa małego podzbioru cech, losowo wybranego z oryginalnego zestawu. Dzięki temu każdy model identyfikuje różne anomalie i przypisuje wszystkim rekordom danych wyniki, które odzwierciedlają ich prawdopodobieństwo bycia anomaliami. Następnie, te wyniki są łączone, aby znaleźć anomalie o lepszej jakości. Innym przykładem takiego podejścia jest praca [320], gdzie zespół modeli jest tworzony poprzez uczenie predyktorów dla każdej cechy na podstawie innych cech. To podejście wykorzystuje różnorodność danych do poprawy wykrywania anomalii, co jest zgodne z ideą różnorodności danych poprzez analizowanie różnych relacji między cechami.

Zespoły oparte na modelach integrują wyniki detekcji anomalii pochodzące z różnych modeli, które operują na tym samym zbiorze danych. Istotnym wyzwaniem jest tutaj porównywalność wyników z różnych modeli, ponieważ wyniki te często mają różne skale. Najczęściej stosujemy znormalizowanie wartości z każdego algorytmu bazowego. Techni-

ka ta zostanie omówiona w kolejnym podrozdziale 5.4. Następnym ważnym wyzwaniem jest dobór odpowiedniej funkcji do łączenia wyników detekcji anomalii z różnych modeli. Ta kwestia zostanie omówiona w późniejszym podrozdziale 5.5 dotyczącym kombinacji funkcji. Zespoły oparte na modelach mogą być skutecznie stosowane przez tworzenie losowych wariantów podstawowego detektora. Na przykład, praca [321] proponuje wykorzystanie RRCF (*ang. robust random cut forest*, RRCF), który jest zbiorem niezależnych drzew RRCT (*ang. robust random cut trees*, RRCT). W tych drzewach losowo wybiera się wymiary i wartości, aby podzielić dane na mniejsze podzbiory, a następnie rekurencyjnie przetwarza te podzbiory. Zasadnicze jest to, że różne drzewa RRCT mają różne losowe podziały, co tworzy losowe warianty detektora bazowego.

W ramach zespołów zorientowanych na model można stosować różne algorytmy, takie jak LOF, autoenkodery i mapy samoorganizujące się jako modele bazowe. To podejście jest przykładem metody opartej na modelu, gdzie różne techniki detekcji anomalii są używane jednocześnie w celu poprawy dokładności wyników. Jedną ze specyficznych form zespołów zorientowanych na model, często stosowaną w analizie anomalii, jest metoda polegająca na wykorzystaniu tego samego modelu z różnymi ustawieniami hiperparametrów bazowych, a następnie łączeniu uzyskanych wyników. Podejście to, polegające na uruchamianiu tego samego modelu z różnymi parametrami i łączeniu wyników, mimo że często jest interpretowane jako strojenie parametrów, w rzeczywistości pełni rolę zespołu modeli. Poprzez systematyczne łączenie wyników z różnych konfiguracji modelu, można uzyskać bardziej dokładne i wiarygodne wyniki detekcji anomalii.

Możliwe jest też połączenie selekcji danych z selekcją modelu, jednak istnieje niewiele metod, które faktycznie to realizują. Zazwyczaj każdy składnik modelu jest definiowany albo jako konkretny model, albo jako specyficzny zestaw danych. W pracy [322] wprowadzono podejście, które łączy zarówno selekcję danych, jak i selekcję modelu. Przedstawiono technikę wyboru losowych podprzestrzeni cech jako formę selekcji danych. Polega to na analizowaniu różnych podzbiorów cech, aby uniknąć problemów związanych z wysoką wymiarowością danych oraz obecnością zaszumionych cech. Jednocześnie wprowadzono selekcję modelu poprzez łączenie różnych metod wykrywania wartości odstających, które mogą być niekompatybilne. Różne techniki detekcji, takie jak metody oparte na odległości czy gęstości, są łączone w celu poprawy dokładności wykrywania wartości odstających.

Oba podejścia - zarówno te skoncentrowane na modelu, jak i na danych - dążą do poprawy wyników przez łączenie różnych perspektyw, czy to poprzez różne modele, czy poprzez różne zestawy danych. W istocie, zespoły skoncentrowane na danych można uważać za podzbiór zespołów skoncentrowanych na modelu, gdzie różnorodność danych jest częścią procesu modelowania. Przedstawiony podział nie jest wyczerpujący, jednak obejmuje znaczną część funkcji zespołowych wykorzystywanych w literaturze. Różne techniki mogą być zorientowane zarówno na dane, jak i na model, co pokazuje ich zdolność do wzajemnego uzupełniania się. Dzięki temu można próbować stworzyć skuteczne hybrydowe podejścia do wykrywania anomalii.

3. Mechanizm redukcji błędu

Kategoryzacja według mechanizmu redukcji błędu odnosi się do metod, które zespoły modeli stosują w celu zmniejszenia całkowitego błędu w identyfikacji anomalii. Mechanizm redukcji błędu jest specyficzny, ponieważ koncentruje się na sposobach zmniejszania błędów w wynikach detektorów anomalii, a nie na strukturze lub charakterze danych czy modeli. Jest to podejście bardziej teoretyczne, które obejmuje analizę składników błędu i ich redukcję poprzez odpowiednie metody zespołowe. W kontekście wykrywania anomalii, całkowity błąd można rozłożyć na składniki biasu i wariancji. Metody redukcji błędu można podzielić na dwa główne podejścia:

- **redukcja obciążenia** (*ang. bias reduction*) - metody w tej kategorii mają na celu zmniejszenie obciążenia modelu. Jest to trudne do osiągnięcia w kontekście wykrywania anomalii, ponieważ często brakuje dostępnych danych referencyjnych (*ang. ground truth*). Przykłady obejmują techniki takie jak SELECT [323] i CARE [324], które wykorzystują heurystyczne podejścia do poprawy dokładności modelu,
- **redukcja wariancji** (*ang. variance reduction*) - te metody skupiają się na zmniejszeniu wariancji modelu, czyniąc go bardziej odpornym na różnice w danych. Techniki takie jak bagging są często stosowane, ponieważ łatwo je dostosować do problemu wykrywania anomalii. Redukcja wariancji jest często bardziej skuteczna w kontekście nienadzorowanego wykrywania anomalii, gdzie etykiety klas są niedostępne.

Równowaga między błędem systematycznym (*ang. bias*) a zmiennością modelu (*ang. variance*) jest ważna nie tylko w uczeniu nadzorowanym, ale również w problemach nienadzorowanych, takich jak detekcja anomalii. Zwykle do oceny tej równowagi niezbędne są etykiety (*ang. labels*), jednak okazuje się, że można to osiągnąć, traktując zmienną zależną jako ukrytą. Chociaż brak jest bezpośredniego dostępu do rzeczywistych wyników (etykiet) w problemach nienadzorowanych, można przyjąć, że taka zmienna zależna istnieje, ale pozostaje ukryta. W kontekście wykrywania anomalii, choć rzeczywiste wyniki nie są dostępne, można założyć, że dla każdego obiektu istnieje pewna idealna wartość (np. idealna ocena anomalii). Podobieństwa w analizie teoretycznej z klasyfikacją ułatwia adaptację wielu typów zespołów klasyfikacyjnych do analizy anomalii. Praktycznym podejściem do zrozumienia związku między wykrywaniem anomalii a klasyfikacją jest traktowanie wykrywania anomalii jako nienadzorowanego odpowiednika problemu identyfikacji rzadkich klas.

Przypuśćmy, że istnieje doskonała funkcja $f(X_i)$, która dla każdego obiektu X_i generuje dokładny wynik y_i, określający, jak bardzo punkt ten jest anomalią. Przykłady takich wartości w algorytmach to wartość wskaźnika LOF w Local Outlier Factor, błąd kwantyzacji w Self-Organizing Maps oraz błąd rekonstrukcji w autoenkoderach, które mogą być użyte do oceny anomalii w danych. Wynik y_i jest nieznany, ale można traktować go jako teoretyczną referencję. Traktowanie zmiennej zależnej y_i jako ukrytej pozwala na

zastosowanie narzędzi teoretycznych do analizy problemów nadzorowanych, mimo braku bezpośrednich etykiet. Wyniki generowane przez algorytmy detekcji anomalii (LOF, SOM, AE) można traktować jako przybliżenia tej nieznanej funkcji, podobnie jak w klasyfikacji. Błąd systematyczny (*ang. bias*) pojawia się, gdy wybrany model nie oddaje dokładnie idealnej funkcji. Zmienność modelu (*ang. variance*) wynika z zależności wyników od konkretnego zbioru danych, a różne zbiory mogą prowadzić do różnych wyników. Całkowity błąd algorytmu detekcji anomalii można przedstawić według wzoru (5.1) jako sumę biasu do kwadratu i wariancji [325]:

$$MSE = Bias^2 + Wariancja \tag{5.1}$$

Oznacza to, że MSE (średni błąd kwadratowy) uwzględnia zarówno systematyczne odchylenie algorytmu od idealnych wyników (bias), jak i jego wrażliwość na zmiany w danych (wariancję). W kontekście wykrywania anomalii, celem jest minimalizacja zarówno biasu, jak i wariancji, aby algorytm był dokładny i stabilny w różnych warunkach.

Kategoryzacja metod zespołowego wykrywania anomalii według powyższych kryteriów pomaga lepiej zrozumieć różne podejścia i strategie stosowane w celu zwiększenia dokładności i niezawodności systemów identyfikacji anomalii. Konstrukcja typowej metody zespołowej do wykrywania anomalii obejmuje różne komponenty, które współpracują ze sobą, aby uzyskać ostateczny wynik. W kolejnych podrozdziałach zademostrowano sposoby projektowania typowej metody zespołowej.

5.3 Wybór modelu bazowego

Aby stworzyć zaawansowanego robota, niezbędne są solidne mechanizmy i komponenty. Podobnie, w kontekście algorytmów bazowych, kluczowe jest posiadanie solidnych podstaw, aby móc zbudować silne i zaawansowane zespoły wykrywania anomalii. Żaden pojedynczy model bazowy nie jest w stanie uchwycić wszystkich cech w zestawie danych, dlatego warto rozważyć kombinację różnych modeli bazowych [326]. Literatura dotycząca łączenia detektorów bazowych pokazuje różnorodne podejścia i techniki mające na celu zwiększenie skuteczności wykrywania anomalii.

Przykładowo, w pracy [322] zaproponowano framework HeDES (*ang. heterogeneous detector ensemble on random subspaces*, HeDES), który integruje różne techniki detekcji anomalii w losowych podprzestrzeniach cech, co pozwala na przezwyciężenie problemów związanych z wielowymiarowością danych oraz szumem. Inne podejścia, takie jak to zaproponowane w artykule [203], koncentrują się na półnadzorowanym wykrywaniu anomalii w podprzestrzeniach cech.

Dziedzina analizy zespołów wykrywania anomalii jest stosunkowo nowa w porównaniu do zespołów klasyfikacyjnych, choć szybko zyskuje na znaczeniu jako ważny obszar badań. W miarę jak badania nad tą dziedziną postępują, wyraźnie widać, że kombinacja różnych

technik bazowych w ramach zespołów nie tylko poprawia ogólną skuteczność wykrywania anomalii, ale również zwiększa stabilność i niezawodność wyników.

Zespoły mogą poprawić wydajność detektorów bazowych na różne sposoby. Pierwsza metoda polega na zastosowaniu pojedynczego detektora bazowego wraz z technikami takimi jak losowe wybieranie podzbiorów cech (*ang. feature bagging*) oraz losowe próbkowanie (*ang. subsampling*). Celem losowego wybierania podzbiorów cech jest zwiększenie różnorodności wyników i zmniejszenie ryzyka nadmiernego dopasowania modelu do danych szkoleniowych, natomiast losowe próbkowanie danych ma na celu zmniejszenie zmienności wyników i poprawę stabilności modelu. Różne zbiory cech często zawierają niezależne informacje, co prowadzi do bardziej dokładnych wyników. Druga metoda polega na integracji wielu różnych detektorów bazowych w celu zwiększenia różnorodności i poprawy ogólnej skuteczności systemu wykrywania anomalii. Obejmuje to zastosowanie różnych algorytmów detekcji anomalii, takich jak lasy losowe, maszyny wektorów nośnych i sieci neuronowe. Różnorodność detektorów bazowych zwiększa szanse na wykrycie różnych typów anomalii, ponieważ różne algorytmy mają swoje unikalne mocne i słabe strony.

Pierwszym krokiem w projektowaniu zespołów do wykrywania anomalii jest wybór modelu bazowego. Ten krok obejmuje wybór algorytmu lub zestawu algorytmów, które będą używane jako komponenty zespołu. Wybór modelu bazowego zależy od celów metody, które mogą być różne:

- **metody wielowymiarowe** - komponenty zespołu mogą obejmować techniki detekcji anomalii w różnych podprzestrzeniach zbioru danych, co oznacza, że jeden algorytm działa na różnych zestawach cech,
- **metody parametryczne** - każdy komponent zespołu może być instancją tego samego algorytmu z różnymi hiperparametrami, co pozwala na lepsze uogólnienie modelu i wykrycie anomalii, które mogłyby zostać przeoczone przy jednej konfiguracji hiperparametrów,
- **różnorodne algorytmy** - bazowe komponenty mogą składać się z różnych algorytmów detekcji anomalii, takich jak lasy losowe, maszyny wektorów nośnych i sieci neuronowe, co zwiększa ogólną skuteczność opracowanego systemu.

Kiedy mówimy o **metodach wielowymiarowych** w kontekście wykrywania anomalii, odnosi się to do sytuacji, w której jeden algorytm wykrywania anomalii działa w różnych podprzestrzeniach zbioru danych. Oznacza to, że algorytm ten analizuje różne zestawy cech, zamiast rozpatrywać cały zbiór danych jako jedną całość. Dzięki temu można wykrywać anomalie, które mogą być niewidoczne w pełnym wymiarze danych, ale stają się bardziej oczywiste, gdy analizowane są w mniejszych, bardziej specyficznych podprzestrzeniach. W **metodach parametrycznych**, na przykład w algorytmie drzewa decyzyjnego, różne komponenty mogą mieć różne głębokości drzewa lub kryteria podziału. Dzięki temu

możliwe jest lepsze uogólnienie modelu oraz wykrycie anomalii, które mogłyby zostać przeoczone przez algorytm z jedną konfiguracją hiperparametrów. **Różnorodne algorytmy** w kontekście zespołowego wykrywania anomalii oznaczają, że komponenty zespołu mogą być oparte na różnych technikach detekcji anomalii. Zamiast używać jednego algorytmu w różnych konfiguracjach, zespół może składać się z różnych algorytmów, takich jak lasy losowe, maszyny wektorów nośnych i sieci neuronowe. Każdy z tych algorytmów może wykrywać różne typy anomalii, co zwiększa ogólną skuteczność systemu. Kombinacja wyników z różnych algorytmów pozwala na bardziej wszechstronne i dokładne wykrywanie anomalii.

Dodatkowo można zastosować technikę adaptacyjnego próbkowania (*ang. adaptive sampling*), która polega na dynamicznym dostosowywaniu próbek danych używanych do trenowania algorytmów bazowych, co pozwala na dokładniejsze wykrywanie anomalii. Przykładem jest algorytm Adaboost (*ang. adaptive boosting*) [327], który zwiększa wagi próbek trudnych do sklasyfikowania, zmuszając algorytmy bazowe do skupienia się na bardziej wymagających przypadkach. W kontekście metod wykrywania anomalii, adaptacyjne próbkowanie współdziała z metodami wysokowymiarowymi, parametrycznymi i różnorodnymi algorytmami, zwiększając ogólną skuteczność systemu wykrywania anomalii. Poprzez dynamiczne dostosowywanie próbek danych, metoda ta skuteczniej radzi sobie z trudnymi do wykrycia anomaliami, co jest szczególnie istotne w heterogenicznych i zmiennych zbiorach danych.

Łączenie różnych detektorów zwiększa odporność zespołu na zmienność wyników. W kontekście ustawień nienadzorowanych, gdzie analityk nie dysponuje wiedzą o optymalnym algorytmie, takie podejście jest szczególnie zalecane. Kompilując wyniki z różnych detektorów, zespół może osiągnąć bardziej stabilną i niezawodną wydajność przy analizie różnych zbiorów danych, co skutkuje wyższą skutecznością w wykrywaniu anomalii. Wybór algorytmu bazowego jest ważny dla skuteczności wykrywania anomalii, a analityk danych powinien kierować się kilkoma ważnymi kryteriami:

- **stabilność algorytmu** - stabilne algorytmy wykazują mniej zmienności w wynikach w zależności od ustawień hiperparametrów, co jest korzystne w praktyce, gdy optymalne ustawienia nie są znane,
- **zakres wartości hiperparametrów** - porównywanie algorytmów w szerokim zakresie wartości hiperparametrów jest istotne, ponieważ optymalne konfiguracje nie zawsze są dostępne. Algorytmy, które działają dobrze w szerokim zakresie ustawień, są bardziej użyteczne w praktyce,
- **charakterystyka danych** - wybór algorytmu powinien uwzględniać charakterystykę danych i typ wykrywanych anomalii. Niektóre algorytmy mogą lepiej wykrywać lokalne anomalia, podczas gdy inne są bardziej skuteczne w identyfikowaniu globalnych obserwacji odstających,

- **redukcja wariancji** - metody zespołowe, takie jak losowe wybieranie podzbiorów cech (*ang. feature bagging*) i losowe próbkowanie (*ang. subsampling*), mogą poprawić stabilność i wydajność detektorów bazowych poprzez redukcję wariancji wyników,
- **liczba hiperparametrów** - algorytmy z mniejszą liczbą hiperparametrów są zazwyczaj bardziej stabilne i mniej podatne na wahania wydajności, co czyni je bardziej odpowiednimi do problemów nienadzorowanej detekcji anomalii.

Niektóre algorytmy, które jako pojedyncze detektory bazowe mają niską wydajność, mogą osiągać znacznie lepsze rezultaty, gdy są używane w zespole. Zachowanie tych algorytmów może znacznie się różnić w zależności od kontekstu ich użycia oraz zastosowanych technik zespołowych. Na przykład metody oparte na grupowaniu i histogramach [328, 329, 221], które same w sobie mogą być mało skuteczne, często przewyższają inne algorytmy o wyższej wydajności bazowej, gdy są zastosowane w konfiguracji zespołowej. W takich przypadkach, zespół może wykorzystać słabości poszczególnych detektorów bazowych na swoją korzyść, osiągając lepszą wydajność w detekcji anomalii.

5.4 Normalizacja wyników w zespołach detekcji anomalii

Głównym wyzwaniem w normalizacji wyników jest fakt, że różne algorytmy stosują odmienne skale odniesienia, co utrudnia ich bezpośrednie porównanie. W pewnych przypadkach wysokie wyniki mogą sygnalizować większą tendencję do bycia anomalią, podczas gdy w innych przypadkach niskie wyniki mogą wskazywać na to samo. To zjawisko komplikuje proces kombinacji wyników. Łączenie wyników z nieporównywalnych skal niesie ze sobą ryzyko nieumyślnego faworyzowania pewnych algorytmów. Dodatkowo, łączenie algorytmów z różnymi konwencjami porządkowania wyników może prowadzić do nieprzewidywalnych rezultatów. Normalizacja wyników w zespołach detekcji anomalii jest istotna z kilku powodów:

- **skale odniesienia** - różne algorytmy mogą generować wyniki na różnych skalach, co utrudnia ich bezpośrednie porównanie. Na przykład, jeden algorytm może zwracać wartości od 0 do 1, podczas gdy inny może generować wartości od -1 do 1. Normalizacja umożliwia sprowadzenie wyników do wspólnej skali, co ułatwia ich łączenie i porównywanie,
- **spójność interpretacji** - wyniki z różnych algorytmów mogą mieć różne znaczenie. W niektórych przypadkach wysokie wartości mogą wskazywać na większą tendencję do bycia odchyleniem, podczas gdy w innych to niskie wartości mogą oznaczać to samo. Normalizacja pozwala ujednolicić interpretację wyników, co jest niezbędne dla poprawnego zrozumienia i analizy danych,

- **redukcja błędów** - bez normalizacji, istnieje ryzyko, że niektóre algorytmy będą nieproporcjonalnie wpływać na wynik końcowy. Na przykład, algorytm generujący wyniki na większej skali może dominować nad innymi algorytmami, nawet jeśli jego wyniki nie są bardziej trafne. Normalizacja zapobiega takiemu zjawisku, zapewniając bardziej zrównoważony wpływ każdego algorytmu na wynik końcowy,
- **kombinacja wyników** - w procesie łączenia wyników z różnych algorytmów, normalizacja jest niezbędna do zastosowania funkcji kombinacyjnych, takich jak średnia czy maksimum. Bez normalizacji wyniki z różnych algorytmów mogłyby być łączone w sposób niespójny, prowadząc do nieprzewidywalnych rezultatów,
- **stabilność i niezawodność** - normalizacja pomaga w zwiększeniu stabilności i niezawodności detekcji anomalii, ponieważ minimalizuje wpływ ekstremalnych wartości i szumów, co prowadzi do bardziej spójnych wyników.

W literaturze zaproponowano kilka metod normalizacji wyników. Jedna z ciekawszych metod została zaproponowana w pracy [330]. Metoda ta polega na przekształceniu wyników w estymacje prawdopodobieństwa, co też umożliwia ich późniejszą kombinację. Szczegóły tej metody omówiono poniżej.

Przekształcenie wyników w estymacje prawdopodobieństwa

Estymacje prawdopodobieństwa są bardziej jednolite i łatwiejsze do porównania niż surowe wyniki różnych algorytmów, prawdopodobieństwa mogą być też łatwiej kombinowane i interpretowane, co pokazano w podrozdziale 5.5.2. Autorzy proponują dwa podejścia:

- **metoda sigmoid**

  Pierwsze podejście zakłada, że estymacje prawdopodobieństwa podążają za funkcją sigmoid i parametry tej funkcji są wyznaczane na podstawie rozkładu wyników detekcji anomalii. Funkcja sigmoid jest szeroko stosowana do przekształcania wyników klasyfikacji na estymacje prawdopodobieństwa.

$$P(O|f_i) = \frac{1}{1 + \exp -(Af_i - B)} \tag{5.2}$$

  gdzie A i B to parametry, które należy wyznaczyć. Parametry A i B są kalibracyjne i są określane za pomocą algorytmu EM (expectation-maximization),

- **metoda mieszaniny rozkładów**

  Alternatywnym podejściem jest metoda modelowania wyników detekcji anomalii przy użyciu mieszaniny rozkładów wykładniczego i normalnego. To podejście zakłada, że wyniki dla klasy normalnej mają tendencję do podążania za rozkładem wykładniczym, podczas gdy dla klasy anomalii (odstającej) wykazują rozkład

normalny. Autorzy przedstawiają teoretyczne uzasadnienie tego założenia [330]. Rozkład wyników jest modelowany jako mieszanina rozkładów wykładniczego i normalnego, a następnie jest obliczana estymacja prawdopodobieństwa za pomocą reguły Bayesa.

$$p_i = p(f_i|O) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(f_i - \mu)^2}{2\sigma^2}\right) \tag{5.3}$$

$$q_i = p(f_i|M) = \lambda \exp(-\lambda f_i), \tag{5.4}$$

gdzie parametrami rozkładów normalnego i wykładniczego są: μ jako średnia rozkładu normalnego dla klasy odstającej O, σ jako odchylenie standardowe rozkładu normalnego dla klasy odstającej O, λ jako parametr rozkładu wykładniczego dla klasy normalnej M.

Prawdopodobieństwo obserwacji t_i można zapisać jako:

$$p(t_i, f_i) = [\alpha p_i]^{t_i} [(1 - \alpha) q_i]^{1 - t_i} \tag{5.5}$$

gdzie α jest wstępnym prawdopodobieństwem przynależności do klasy odstającej, a t_i to zmienna binarna, która wskazuje, czy obserwacja X_i należy do klasy odstającej czy klasy normalnej. Wartość t_i wynosi 1, jeśli X_i jest anomalią, i 0, jeśli jest normalna.

Korzystając z reguły Bayesa:

$$p(t_i|f_i) = \frac{[\alpha p_i]^{t_i} [(1 - \alpha) q_i]^{1 - t_i}}{p(f_i)} \tag{5.6}$$

Parametry modelu $\theta = (\alpha, \mu, \sigma, \lambda)$ są szacowane poprzez minimalizację negatywnej logarytmicznej funkcji wiarygodności (*ang. negative log likelihood*):

$$LL(T|F) = -\sum_{i=1}^{N} [t_i \log(\alpha p_i) + (1 - t_i) \log((1 - \alpha) q_i)] \tag{5.7}$$

Po oszacowaniu parametrów modelu, prawdopodobieństwo a posteriori, że X_i jest anomalią, można obliczyć za pomocą reguły Bayesa:

$$P(O|f_i, \hat{\theta}) = \frac{\alpha p(f_i|O, \hat{\theta})}{\alpha p(f_i|O, \hat{\theta}) + (1 - \alpha) p(f_i|M, \hat{\theta})} \tag{5.8}$$

Podobnie jak w poprzedniej metodzie, należy użyć algorytmu EM do minimalizacji negatywnej logarytmicznej funkcji wiarygodności podanej w równaniu (5.7).

W kontekście wykrywania anomalii, proste metody takie jak standaryzacja z (*ang. z-score*) czy normalizacja min-max mogą okazać się niezwykle skuteczne [220]. Szczegóły tych metod omówiono poniżej.

Standaryzacja z

Najpierw, dla zbioru D o n obiektach, których wyniki detekcji anomalii są oznaczone jako $S = \{s(1), s(2), \ldots, s(n)\}$, obliczamy średnią μ oraz odchylenie standardowe σ. Średnia arytmetyczna μ jest obliczana według wzoru:

$$\mu = \frac{\sum_{i=1}^{n} s(i)}{n} \tag{5.9}$$

Następnie, odchylenie standardowe σ dla tych n wyników jest wyliczane jako:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (s(i) - \mu)^2}{n-1}} \tag{5.10}$$

Dzięki temu można przystąpić do obliczenia wartości z dla każdego X_i w zbiorze D, zgodnie z równaniem:

$$z(i) = \frac{s(i) - \mu}{\sigma} \tag{5.11}$$

Miara z wskazuje, jak daleko i w którym kierunku wynik odchylenia $s(i)$ znajduje się od średniej wartości μ, wyrażonej w jednostkach odchylenia standardowego σ. Wynik z jest dodatni, gdy surowy wynik $s(i)$ jest powyżej średniej, i ujemny, gdy jest poniżej średniej. Matematycznie, jeśli mniejsze wyniki $s(i)$ wskazują na większą anomalię, używamy ujemnej wartości z, co sprawia, że większa wartość z zawsze oznacza większy stopień odchylenia. Standaryzacja z jest przydatna w kontekście zespołów detekcji odchyleń, ponieważ pozwala na ujednolicenie wyników z różnych algorytmów. Jednakże, normalizacja z-score nie jest odporna na wartości odstające. Wartości odstające mogą znacząco wpłynąć na średnią μ i odchylenie standardowe σ, co w rezultacie wpływa na znormalizowane wyniki.

Normalizacja min-max

Jednym z najprostszych i najczęściej stosowanych sposobów normalizacji wyników jest ich przeskalowanie do zakresu od 0 do 1. Ten proces jest znany jako normalizacja min-max i polega na zastosowaniu poniższego wzoru:

$$s'(i) = \frac{s(i) - \min(S)}{\max(S) - \min(S)}, \tag{5.12}$$

gdzie $s(i)$ jest oryginalnym wynikiem dla i-tego obiektu, $\min(S)$ jest najmniejszą war-

tością w zbiorze wyników S, a $\max(S)$ jest największą wartością w tym zbiorze. Chociaż metoda ta jest bardzo prosta, może być niezwykle skuteczna w wielu przypadkach, także w kontekście wykrywania obiektów odstających w zespołach detekcji anomalii. Przeskalowanie do zakresu od 0 do 1 zapewnia, że wszystkie wyniki mieszczą się w tej samej skali, co ułatwia ich łączenie, porównywanie i interpretowanie. Normalizacja min-max również nie jest odporna na wartości odstające. Wartości odstające mogą znacząco wpłynąć na minimalną i maksymalną wartość, co z kolei wpływa na cały zakres normalizacji.

Inne metody normalizacji w zespołach detekcji anomalii

Oprócz wcześniej wymienionych metod normalizacji, istnieje wiele innych technik, które mogą być używane w zespołach detekcji anomalii. Poniżej przedstawiono krótką charakterystykę kilku metod [331]:

- **normalizacja przez medianę** – normalizacja przez wartość mediany. Znormalizowana wartość $s(i)_{norm}$ wyniku detekcji anomalii $s(i)$ może być obliczona za pomocą wzoru:

$$s(i)_{norm} = \frac{s(i)}{\text{median}(S)} \tag{5.13}$$

- **normalizacja sigmoidalna** - użycie funkcji sigmoid do normalizacji. Wartość $s(i)$ może być znormalizowana za pomocą poniższej funkcji sigmoid:

$$s(i)_{norm} = \frac{1}{1 + e^{-s(i)}} \tag{5.14}$$

- **estymator tangensa hiperbolicznego** (*ang. tanh*) - użycie funkcji tangensa hiperbolicznego do normalizacji. Znormalizowana wartość $s(i)_{norm}$ wyniku $s(i)$ może być obliczona za pomocą wzoru:

$$s(i)_{norm} = 0.5\left[\tanh\left(\frac{0.01(s(i) - \mu)}{\sigma}\right) + 1\right], \tag{5.15}$$

  gdzie μ i σ to średnia wartość i odchylenie standardowe wszystkich wyników S,

- **normalizacja logarytmiczna** - zastosowanie transformacji logarytmicznej w celu zmniejszenia skosu i stabilizacji wariancji wyników. Jest przydatna, gdy wyniki mają bardzo szeroki zakres wartości:

$$s(i)_{norm} = \log(s(i) + 1) \tag{5.16}$$

- **normalizacja odporna** (*ang. robust scaler*) - wykorzystuje medianę i zakres międzykwartylowy (IQR) zamiast średniej i odchylenia standardowego, co czyni ją bardziej odporną na wpływ wartości odstających. Jest użyteczna w przypadku danych z dużą

liczbą wartości odstających:

$$s(i)_{norm} = \frac{s(i) - \text{median}(S)}{\text{IQR}} \tag{5.17}$$

- **normalizacja rang** - normalizacja danych polega na użyciu rang z różnych algorytmów analizy anomalii, które następnie są łączone w celu stworzenia znormalizowanego wyniku (jednolitego wyniku wykrywania anomalii). Rangi są przydzielane zgodnie z tendencją od największej do najmniejszej anomalii, co zapewnia spójność w porządkowaniu wyników. W przypadku występowania powtórzeń wartości, można przydzielić średnią rangę.

Spośród wymienionych metod, normalizacja przez medianę oraz normalizacja odporna (*ang. robust scaler*) wykazują największą odporność na wartości odstające. Obie te metody opierają się na medianie, która jest znacznie mniej wrażliwa na skrajne wartości niż średnia arytmetyczna, co czyni je bardziej odpowiednimi do zastosowania w przypadku zestawów danych zawierających dużą ilość wartości odstających. Pozostałe metody, takie jak normalizacja sigmoidalna, estymator tangensa hiperbolicznego oraz normalizacja logarytmiczna, nie są tak odporne na wartości odstające.

Ogólnie, w kontekście analizy anomalii, gdzie istotne jest łączenie wyników z różnych modeli, metody odporne na wartości odstające są bardziej użyteczne, ponieważ minimalizują wpływ ekstremalnych wartości, co prowadzi do bardziej spójnych i porównywalnych wyników. Jednakże, w niektórych przypadkach, gdzie istotne jest na przykład wybranie maksimum z kilku wartości, metody mniej odporne mogą być lepsze. Wartości ekstremalne mogą wskazywać na istotne różnice, które są decydujące dla analizy anomalii. Dlatego też, mniej odporne metody mogą lepiej wykrywać takie anomalie, ponieważ nie redukują one wpływu ekstremalnych wartości, co może być szczególnie istotne przy stosowaniu funkcji kombinacji, takich jak maksymalizacja. Ważne jest, aby wybór metody normalizacji był dostosowany do specyfiki analizowanych danych i celów analizy. Decyzja ta powinna opierać się na doświadczeniu analityka, wynikach eksperymentów oraz badaniach empirycznych.

5.5 Metody łączenia wyników w zespole

Po utworzeniu zestawu bazowych modeli, zamiast dążyć do znalezienia jednego najlepszego modelu, metody zespołowe wykorzystują kombinację wyników w celu poprawy zdolności do uogólnienia, przy czym najważniejsze znaczenie ma metoda łączenia. Istnieją trzy główne korzyści z łączenia modeli [315, 204]:

- **kwestia statystyczna**

  Wykrywanie anomalii często dotyczy pracy z ograniczonymi danymi szkoleniowymi oraz dużą przestrzenią hipotez. W takim przypadku może istnieć wiele różnych hipotez, które na danych treningowych dają podobne wyniki. Łączenie wyników z różnych modeli zmniejsza ryzyko wyboru jednej błędnej hipotezy, która może nie działać dobrze na nowych danych. Przykładem może być niewykrycie oszustwa w systemie bankowym. Wybór jednej hipotezy na podstawie danych szkoleniowych może prowadzić do sytuacji, w której hipoteza okazuje się nieskuteczna w przypadku nowych, rzeczywistych danych. Łączenie wyników z różnych modeli zmniejsza ryzyko polegania na jednej błędnej hipotezie. Na przykład, łączenie wyników z kilku algorytmów wykrywania anomalii zwiększa szanse na wykrycie rzeczywistych anomalii, ponieważ błędy jednego modelu mogą zostać skorygowane przez inne modele w zespole. Dzięki temu ogólna skuteczność wykrywania anomalii jest wyższa, a ryzyko przeoczenia ważnych anomalii jest mniejsze,

- **kwestia obliczeniowa**

  Wykrywanie anomalii często wymaga zastosowania algorytmów, które mogą utknąć w lokalnych minimach podczas procesu optymalizacji. Nawet przy dużej ilości danych szkoleniowych, znalezienie optymalnej hipotezy może być trudne i czasochłonne. Kombinacja wyników z różnych modeli pozwala zredukować ryzyko utknięcia w lokalnym minimum, co znacząco poprawia ogólną skuteczność wykrywania anomalii. Na przykład, w przypadku wykrywania anomalii w sieci komputerowej, pojedynczy algorytm może nie być w stanie znaleźć wszystkich nieprawidłowości, ponieważ może utknąć w lokalnych minimach, analizując tylko część danych. Kombinacja wyników z wielu modeli daje pełniejszy obraz analizowanych danych, co jest bardzo ważne w kontekście bezpieczeństwa sieci i wykrywania potencjalnych zagrożeń,

- **kwestia reprezentacji**

  Często zdarza się, że prawdziwa, nieznana hipoteza nie może być dokładnie reprezentowana przez żadną pojedynczą hipotezę w dostępnej przestrzeni hipotez. Łączenie różnych hipotez pozwala na rozszerzenie przestrzeni funkcji, które mogą być reprezentowane przez modele. W praktyce oznacza to, że dzięki kombinacji wyników z różnych modeli uzyskuje się bardziej precyzyjne i wszechstronne odwzorowanie rzeczywistości. Na przykład, w kontekście wykrywania anomalii w systemach monitorowania zdrowia pacjentów, pojedynczy algorytm może nie być w stanie wychwycić wszystkich możliwych anomalii ze względu na ograniczenia swojej przestrzeni hipotez. Łączenie wyników z różnych algorytmów umożliwia dodanie nowych informacji do wspólnego modelu, co pozwala na dokładniejsze wykrywanie szerokiego zakresu nietypowych przypadków. Przykładowo, jeden algorytm może

być szczególnie dobry w wykrywaniu anomalii związanych z nagłymi skokami ciśnienia krwi, podczas gdy inny może lepiej radzić sobie z identyfikacją anomalii w rytmie serca. Łączenie różnych algorytmów umożliwia stworzenie systemu monitorowania zdrowia, który jest bardziej czuły na różne rodzaje anomalii, zwiększając tym samym szanse na wczesne wykrycie potencjalnych problemów zdrowotnych, które mogłyby zostać przeoczone przez pojedynczy algorytm. Rozszerzenie przestrzeni funkcji dzięki łączeniu modeli pozwala na lepsze modelowanie rzeczywistości i dokładniejsze przybliżanie prawdziwej hipotezy, co jest ważne w zadaniach takich jak identyfikacja anomalii, gdzie różnorodność i nietypowość przypadków wymagają elastycznych i wszechstronnych metod analizy.

Te korzyści sprawiają, że metody zespołowe są bardziej skuteczne w wykrywaniu anomalii, co przekłada się na wyższą dokładność i niezawodność w identyfikowaniu obiektów odstających.

5.5.1 Podstawowe techniki łączenia modeli

Przy wyborze metody kombinacji dla zestawu znormalizowanych wyników detekcji anomalii rozważa się następujące metody:

Prosta średnia arytmetyczna

Średnia wyników jest jedną z najprostszych i najczęściej stosowanych metod łączenia wyników różnych detektorów w zespole wykrywania anomalii [315]. Metoda ta polega na obliczeniu średniej arytmetycznej wyników z różnych detektorów dla każdego obiektu X_i. Metoda ta jest przydatna w sytuacjach, gdy konieczne jest równomierne uwzględnienie wszystkich wartości. Stosowana jest, gdy każda wartość ma podobne znaczenie dla końcowego wyniku. Wynik końcowy jest następnie raportowany jako średnia wyników uzyskanych z różnych komponentów zespołu.

$$H(X_i) = \frac{1}{n} \sum_{j=1}^{n} z_j(X_i), \tag{5.18}$$

gdzie:

- $H(X_i)$ to średnia znormalizowanych wyników dla każdej próbki danych X_i (gdzie i oznacza indeks danego obiektu), będąca wynikiem końcowym metod łączenia,
- n to liczba detektorów anomalii w zespole,
- $z_j(X_i)$ to znormalizowany wynik detektora j dla próbki danych X_i.

Teoretycznie wykazano [315], że oczekiwany błąd całego zespołu modeli będzie mniejszy niż średni błąd poszczególnych modeli. Oznacza to, że łączenie wielu modeli w zespół pozwala na uzyskanie lepszych wyników. Im więcej modeli znajduje się w zespole, tym większa jest potencjalna redukcja błędu. Jednak jest to oparte na założeniu, że błędy poszczególnych modeli nie są skorelowane, co oznacza, że modele popełniają różne błędy. W praktyce modele są często szkolone na tych samych danych i popełniają podobne błędy, co sprawia, że osiągnięcie tak dużej redukcji błędów jest trudne.

Ważone uśrednianie

Jako rozwinięcie metody prostej średniej arytmetycznej, ważone uśrednianie przyznaje indywidualne wagi w_j każdemu detektorowi $z_j(X_i)$ w zespole, co pozwala na większą elastyczność i dostosowanie do specyfiki danych oraz zadania. Ta metoda jest szczególnie przydatna, gdy detektory wykazują różny stopień niezawodności lub efektywności.

$$H(X_i) = \frac{1}{n} \sum_{j=1}^{n} w_j z_j(X_i), \tag{5.19}$$

gdzie:

- $w_j \geq 0$ i $\sum_{j=1}^{n} w_j = 1$, co zapewnia, że wagi są normalizowane i sumują się do jedności,
- $z_j(X_i)$ to wynik j-tego detektora dla próbki danych X_i.

Wagi w_j mogą być dostosowane na podstawie wcześniejszej wydajności detektorów, ich niezawodności, lub innych kryteriów, co pozwala na lepszą adaptację do różnorodnych warunków. Prosta średnia arytmetyczna, rozumiana jako przydzielenie równych wag każdemu z modeli w zbiorze, jest w rzeczywistości specyficznym przypadkiem bardziej ogólnego podejścia, jakim jest średnia ważona. Zastosowanie tej metody jest szczególnie zalecane, gdy modele bazowe prezentują zbliżoną skuteczność. W sytuacji, gdzie poszczególne modele charakteryzują się różnymi poziomami dokładności, zastosowanie średniej ważonej, przydzielającej różne wagi w zależności od efektywności każdego modelu, może znacząco poprawić wyniki końcowe.

Mediana wyników

Mediana wyników jest metodą łączenia, która wykorzystuje medianę zamiast średniej do określenia centralnej wartości wyników poszczególnych detektorów w zespole. Podobnie jak w poprzednich metodach, gdzie prosta średnia i ważona średnia były wykorzystywane do agregacji wyników, mediana służy jako bardziej stabilny reprezentant centralny, mniej podatny na ekstremalne odchylenia w danych.

$$H(X_i) = \text{mediana}\left(\{z_j(X_i) : j = 1, 2, \ldots, n\}\right) \tag{5.20}$$

gdzie:

- $H(X_i)$ to mediana wyników dla próbki danych X_i, będąca wynikiem końcowym metody łączenia,
- $z_j(X_i)$ to wynik detektora j dla próbki danych X_i.

Mediana, wykorzystując centralny punkt zbioru danych, zapewnia odporność na nietypowe wyniki, które mogą wystąpić w poszczególnych detektorach. Jest szczególnie przydatna w przypadkach, gdy oczekuje się stabilności wyników, nawet kosztem utraty informacji o bardziej ekstremalnych przypadkach. Mediana może nie zawsze zapewniać lepsze wyniki niż prosta średnia, ze względu na mniejszą zdolność do odzwierciedlania różnorodności w danych. Mediana może być użyteczna, gdy dane zawierają wartości skrajne lub szum, ponieważ nie jest wrażliwa na ekstremalne wartości. Stosuje się ją w sytuacjach, gdy ważne jest unikanie wpływu wartości odstających na końcowy wynik.

Średnia i mediana rang

W ramach procedury analizy danych, wyniki poszczególnych detektorów są transformowane do postaci rang, gdzie każdy obiekt X_i otrzymuje rangę od każdego z detektorów, tworząc tym samym zbiór rang dla każdego obiektu. Liczba rang przyznanych każdemu obiektowi jest równa liczbie detektorów w zespole. Średnia ranga dla każdego obiektu jest następnie wyznaczana poprzez agregację rang przydzielonych przez różne komponenty zespołu, co opisuje wzór (5.21):

$$\overline{R}(X_i) = \frac{1}{n}\sum_{j=1}^{n} R_j(X_i), \tag{5.21}$$

gdzie $R_j(X_i)$ reprezentuje rangę punktu X_i przyznaną przez detektor j.

Wykorzystanie rang stabilizuje proces analizy, ponieważ zmniejsza wpływ ekstremalnych wartości wykrywanych przez detektory anomalii, które mogą zniekształcać wyniki i prowadzić do błędnych wniosków. W tym przypadku ekstremalne anomalie są często ignorowane, choć mogą dostarczać cennych informacji o stopniu odchylenia badanego obiektu od normy.

Mediana rang, obliczana dla każdego obiektu, uwzględnia wartości przyznane przez wszystkie detektory:

$$M(X_i) = \text{mediana}\left(\{R_j(X_i) : j = 1, 2, \ldots, n\}\right) \tag{5.22}$$

Mimo że rangi są z natury bardziej stabilne niż inne miary, użycie mediany rangi zwykle nie przynosi znacząco różnych rezultatów od średniej. Stosowanie mediany może jednak pomóc

w zniwelowaniu wpływu skrajnych wartości. Zastosowanie metod opartych na rankingu często demonstruje dobrą efektywność w przypadku detektorów charakteryzujących się wysokim poziomem błędów dla specyficznych obiektów. Warto zauważyć, że metody te mogą prowadzić do utraty znaczącej ilości informacji, co jest istotne przy podejmowaniu decyzji o ich stosowaniu.

Maksymalizacja wyników

Metoda maksymalizacji wyników polega na wyborze najwyższego wyniku spośród wszystkich detektorów dla każdej próbki danych X_i. Podejście to może być przydatne w sytuacjach, gdy istotne jest wyłowienie najbardziej ekstremalnych przypadków w danych, nawet kosztem zwiększenia wariancji wyników.

$$H(X_i) = \max_{j=1}^{n} z_j(X_i), \tag{5.23}$$

gdzie:

- $H(X_i)$ to maksymalny wynik uzyskany przez detektory dla próbki danych X_i, będący wynikiem końcowym metody łączenia,
- $z_j(X_i)$ to wynik detektora j dla próbki danych X_i.

W procesie uśredniania, jednorodny rozkład danych może prowadzić do sytuacji, w której trudno jest odróżnić normalne obiekty od anomalii, co może skutkować ukryciem rzeczywistych anomalii przez efekty średniej. W takich okolicznościach, metoda maksymalizacji kieruje uwagę ku najbardziej skrajnym wartościom w zestawie danych, co w teorii pozwala na skuteczniejsze wykrywanie anomalii poprzez podkreślenie ich niekonwencjonalnych wartości.

Mimo że maksymalizacja skoncentrowana jest na ekstremach, paradoksalnie może to zwiększać wariancję wyników, wprowadzając element niestabilności. Jest to znaczące zwłaszcza w zestawach danych o wysokiej zmienności, gdzie stosowanie maksymalizacji może prowadzić do nieprzewidywalnych rezultatów. Z tego powodu, zaleca się integrację tej metody z innymi technikami redukcji wariancji, które zostały zaprezentowane w dalszej części w podrozdziale 5.5.2.

Minimalizacja wyników

Metoda minimalizacji wyników polega na wyborze najniższego wyniku spośród wszystkich detektorów dla każdej próbki danych X_i. Podejście to może być przydatne w sytuacjach, gdy konieczne jest minimalizowanie ryzyka fałszywie pozytywnych wyników, zmniejszając jednocześnie wpływ ekstremalnie wysokich wartości anomalii na końcową decyzję.

$$H(X_i) = \min_{j=1}^{n} z_j(X_i), \tag{5.24}$$

gdzie:

- $H(X_i)$ to minimalny wynik uzyskany przez detektory dla próbki danych X_i, będący wynikiem końcowym metody łączenia,
- $z_j(X_i)$ to wynik detektora j dla próbki danych X_i.

Metoda ta jest szczególnie skuteczna w środowiskach, gdzie ważne jest unikanie fałszywych alarmów i preferowane jest konserwatywne podejście do detekcji anomalii. Minimalizacja może być również użyteczna, gdy oczekuje się, że większość detektorów generuje wyniki bliskie normalności, a ekstremalne wartości, które mogą prowadzić do fałszywych alarmów, są rzadkie. Jednym z ograniczeń tej metody jest to, że może ona przeoczyć rzeczywiste anomalie, jeśli te nie są wykrywane przez większość detektorów. Dlatego minimalizacja wyników powinna być stosowana w połączeniu z innymi metodami detekcji, aby uniknąć pominięcia anomalii.

Iloczyn wyników

Metoda iloczynu wyników polega na pomnożeniu wszystkich wyników detektorów dla każdej próbki danych X_i. Podejście to jest szczególnie przydatne w sytuacjach, gdy konieczne jest, aby wynik końcowy był silnie zależny od najniższych wartości w zestawie danych, co zwiększa wrażliwość metody na potencjalne anomalie ukryte w danych.

$$H(X_i) = \prod_{j=1}^{n} z_j(X_i), \tag{5.25}$$

gdzie:

- $H(X_i)$ to iloczyn wyników uzyskanych przez detektory dla próbki danych X_i, będący wynikiem końcowym metody łączenia,
- $z_j(X_i)$ to wynik detektora j dla próbki danych X_i.

Iloczyn wyników jest szczególnie efektywny, gdy istotne jest uwzględnienie nawet najmniejszych wskazań anomalii, ponieważ nawet jedna bardzo niska wartość w zestawie danych może istotnie obniżyć ogólny wynik. Metoda ta zatem potęguje wpływ niskich wyników detektorów, co czyni ją odpowiednią w aplikacjach wymagających wysokiej czułości na anomalie. Działa również jako filtr przeciwko fałszywym alarmom, jeżeli wszystkie detektory muszą wykazać niską wartość, aby całościowy wynik był uznany za anomalię.

5.5.2 Zaawansowane techniki łączenia modeli

Podrozdział dotyczy zaawansowanych technik łączenia modeli, które mają na celu jednoczesną redukcję błędu systematycznego (bias) i wariancji.

Metoda AOM

Metoda AOM (*ang. average of maximum*, AOM) jest techniką łączenia wyników dwóch różnych metod obróbki danych: maksymalizacji i uśredniania. W tej metodzie komponenty zespołu są najpierw dzielone na grupy (kubełki), gdzie w każdej grupie stosowana jest maksymalizacja. To znaczy, że w każdej grupie wybierany jest największy wynik spośród komponentów w grupie. Następnie wyniki z tych grup są uśredniane, aby uzyskać końcowy wynik.

$$H(X_i) = \frac{1}{m/q} \sum_{b=1}^{m/q} \max_{j \in Q_b} z_j(X_i), \tag{5.26}$$

gdzie:

- $H(X_i)$ to średnia wyników maksymalnych z kubełków dla próbki danych X_i, będąca wynikiem końcowym metody łączenia,
- m/q oznacza liczbę kubełków, na które podzielono komponenty zespołu,
- Q_b to zbiór indeksów komponentów w b-tym kubełku,
- $z_j(X_i)$ to wynik j-tego detektora dla próbki danych X_i.

Podstawowym celem zastosowania maksymalizacji w pierwszej fazie jest redukcja błędu systematycznego. Dzięki temu, że z każdej grupy wybierany jest największy wynik, metoda ta skupia się na najbardziej ekstremalnych przypadkach w danych, co pozwala lepiej identyfikować anomalie lub najbardziej znaczące obserwacje. Druga faza, czyli uśrednianie wyników po wszystkich grupach, służy do redukcji wariancji. Uśrednianie pomaga zniwelować efekt wyolbrzymiania wyników przez maksymalizację, zapewniając bardziej zrównoważony i stabilny wynik końcowy.

Metoda AOM jest przykładem podejścia hybrydowego, które próbuje wykorzystać zalety zarówno maksymalizacji, jak i uśredniania. W praktycznym zastosowaniu, szczególnie przy dużym zestawie danych, AOM może znacząco poprawić dokładność modelu przez skuteczne połączenie dwóch różnych technik. Ponadto, dzięki temu podejściu, możliwe jest osiągnięcie lepszych rezultatów nawet przy użyciu tej samej liczby obiektów.

Metoda progowa

Metoda progowa (*ang. thresh*) jest jedną z metod redukcji wariancji i błędu systematycznego przy łączeniu wyników z wielu detektorów. Polega ona na stosowaniu absolutnego progu t na znormalizowane wyniki detekcji anomalii dla każdego z detektorów. Wyniki poniżej tego progu są odrzucane, co oznacza, że nie przyczyniają się one do ostatecznego wyniku. Wartość progu t często ustawia się na 0, co oznacza, że uwzględniane są tylko te wyniki, które wskazują na obecność anomalii (wyniki większe od zera).

$$H(X_i) = \frac{1}{n} \sum_{j=1}^{n} \max(t, z_j(X_i)), \tag{5.27}$$

gdzie:

- $H(X_i)$ to średni wynik po zastosowaniu progu dla obiektu X_i,
- $z_j(X_i)$ to znormalizowany wynik j-tego detektora dla obiektu X_i,
- t to wartość progu, poniżej którego wyniki są ignorowane (często ustawiana na 0),
- n to liczba detektorów biorących udział w analizie.

Cechy metody progowej to:

- **progowanie** - wyniki poniżej progu t (często $t = 0$) są ignorowane, co pozwala na skupienie się tylko na bardziej znaczących wynikach sugerujących anomalie,
- **zagrożenie remisem** - jeśli wiele wyników osiąga dokładnie wartość graniczną t, może dojść do sytuacji remisu w rankingu. W takich przypadkach rozstrzygnięcie remisu odbywa się przez zastosowanie średniej znormalizowanych wyników tych punktów w komponentach zespołu,
- **praktyczne zastosowanie** - można dostosować metodę, dodając niewielką wartość proporcjonalną do średniego znormalizowanego wyniku, co pomaga rozstrzygnąć remisy. Metoda ta pozwala odpowiednio nagradzać obiekty, które wykazują cechy anomalii.

Metoda progowa jest cenna, gdy istotne jest zminimalizowanie ryzyka fałszywych pozytywów poprzez ignorowanie wyników, które są poniżej ustawionego progu. To podejście jest użyteczne, gdy ważna jest efektywność obliczeniowa, ponieważ pozwala na szybsze łączenie wyników i redukcję błędu systematycznego oraz wariancji, zwłaszcza w kontekście analizy sekwencyjnej, gdzie kolejne komponenty zespołu sukcesywnie doskonalą model.

Metoda MOA

Metoda MOA (*ang. maximum of average*, MOA) jest innym sposobem na połączenie redukcji błędu systematycznego i wariancji w analizie zespołowej. Ten schemat można rozpatrywać jako komplementarny do metody AOM. Główna różnica polega na tym, że w metodzie MOA najpierw wykonuje się uśrednienie wyników detektorów, a następnie stosuje się funkcję maksymalizacji na uśrednionych wynikach.

$$H(X_i) = \max\left(\frac{1}{n} \sum_{j=1}^{n} z_j(X_i)\right), \tag{5.28}$$

gdzie:

- $H(X_i)$ to maksymalna wartość średniej wyników uzyskanych przez detektory dla próbki danych X_i, będąca wynikiem końcowym metody łączenia,
- $z_j(X_i)$ to wynik detektora j dla próbki danych X_i.

Chociaż metoda ta oferuje teoretyczne zalety, takie jak redukcja błędu systematycznego poprzez maksymalizację i redukcja wariancji poprzez uśrednianie, jest ona mniej korzystna, gdy chodzi o badanie redukcji błędu systematycznego jako drugiego kroku. Jest to spowodowane tym, że użyteczność informacji dostarczanych przez poszczególne detektory może być zmniejszona przez wcześniejsze uśrednianie, co obniża skuteczność maksymalizacji w identyfikacji najbardziej ekstremalnych wartości. W praktyce metoda MOA jest użyteczna w sytuacjach, gdy istotne jest wychwycenie drobnych, subtelnych odchyleń od normy, które mogłyby zostać przeoczone przy użyciu tylko jednego detektora do analizy danych.

Agregowanie wyników po estymacji prawdopodobieństwa

Po przekształceniu wyników w estymacje prawdopodobieństwa, co omówiono w podrozdziale 5.4 można zastosować różne metody łączenia. Łączenie wyników z różnych detektorów anomalii w zespole polega na łączeniu ich szacunków prawdopodobieństwa w celu uzyskania bardziej niezawodnych i dokładnych wyników. Poniżej przedstawiono kilka typowych metod łączenia:

- **średnia** (*ang. average*) - uśrednienie wszystkich znormalizowanych wyników.

$$P(O|X_i) = \frac{1}{n}\sum_{j=1}^{n} P_j(O|X_i), \tag{5.29}$$

- **maksimum** (*ang. maximum*) - wybór maksymalnej wartości spośród znormalizowanych wyników.

$$P(O|X_i) = \max_{j=1}^{n} P_j(O|X_i), \tag{5.30}$$

- **konfiguracja szeregowa** (*ang. series*) - przekształcenie łączenia wyników w system szeregowy, gdzie niepowodzenie jednego z komponentów (detektorów) oznacza niepowodzenie całego systemu.

$$P(O|X_i) = \prod_{j=1}^{n} P_j(O|X_i) \tag{5.31}$$

- **konfiguracja równoległa** (*ang. parallel*) - przekształcenie łączenia wyników w system równoległy, gdzie sukces jednego z komponentów oznacza sukces całego systemu.

$$P(O|X_i) = 1 - \prod_{j=1}^{n}(1 - P_j(O|X_i)) \tag{5.32}$$

Analiza modeli detekcji anomalii przez rozkład macierzy

Inną zaawansowaną metodą jest zastosowanie strategii konsensusu, której celem jest zminimalizowanie wpływu detektorów o mniejszej precyzji. Można to osiągnąć poprzez dekompozycję macierzy wyników z różnych detektorów wykrywających anomalie. Detektory, które znacznie odbiegają od pozostałych, mogą wykorzystywać nieodpowiednie modele. Ograniczenie ich wpływu na końcową analizę umożliwia lepszą kontrolę błędów systematycznych oraz redukcję wariancji wyników.

Wyniki każdego detektora są dostosowywane przez macierz V. W ten sposób zmniejszany jest wpływ detektorów, które dają odstające wyniki. Błąd macierzy E jest obliczany jako różnica między faktyczną macierzą wyników S a iloczynem macierzy U i transponowanej macierzy V (UV^T). Pokazuje to stopień, w jakim obecne prognozy różnią się od faktycznych danych. Macierz U, przedstawiająca „rzeczywiste" wyniki dla każdego obiektu, jest inicjowana jako wektor zawierający średnie wyniki anomalii, które są wyliczane jako średnia wartości każdego wiersza w macierzy S. Macierz V jest inicjowana jako wektor jednostkowy, co oznacza, że każdy detektor początkowo wpływa równomiernie na końcowy wynik. Macierze są aktualizowane przy użyciu prostego algorytmu gradientu, gdzie η jest współczynnikiem uczenia się. Końcowe wartości w macierzy U są interpretowane jako wyniki detekcji anomalii dla każdego obiektu.

Dekompozycja macierzy polega na podzieleniu jednej macierzy na dwie, tak aby ich iloczyn jak najdokładniej odwzorował macierz pierwotną. Interpretacja wyników z poszczególnych detektorów wymaga zrozumienia dwóch komponentów: elementów wspólnych i unikalnych. Elementy wspólne to cechy lub wzorce, które są widoczne w większości zestawu danych i są rozpoznawalne niezależnie od specyfiki wykorzystywanego detektora. Z drugiej strony, elementy unikalne dla konkretnego detektora odnoszą się do specyficznych charakterystyk, które mogą różnić się w zależności od metodologii lub technologii użytej w danym detektorze. W analizie detekcji anomalii, każdy wynik z detektora przedstawiany jest jako mieszanka elementów wspólnych oraz elementów unikalnych dla konkretnego detektora. Celem jest ograniczenie wpływu detektorów generujących potencjalnie nieprawidłowe wyniki, co z kolei poprawia wiarygodność detekcji anomalii.

Założenie modelu jest takie, że każdy wynik detektora można dostosować za pomocą mnożnika, który koryguje wynik na podstawie ogólnej charakterystyki tego detektora w porównaniu z innymi. Końcowy wynik dla każdego obiektu jest kombinacją tych skorygowanych wyników. Proces obejmuje optymalizację, podczas której minimalizowane są

różnice (błędy) między rzeczywistymi wynikami a tymi przewidywanymi przez model dekompozycji. Optymalizacja ma na celu dostosowanie mnożników, aby jak najlepiej odwzorowywały rzeczywistą jakość wyników z różnych detektorów. Model jest iteracyjnie ulepszany poprzez modyfikację mnożników i ocenę błędów, co prowadzi do stopniowego udoskonalania dokładności wyników. Po zakończeniu procesu dekompozycji i optymalizacji, końcowe wyniki dla każdego obiektu są przedstawiane jako „rzeczywiste" wyniki, najlepiej oddające obecność anomalii w danych.

5.6 Podsumowanie

W tym rozdziale omówiono podstawowe i zaawansowane techniki zespołowej identyfikacji anomalii. Na początku przedstawiono innowacyjne aspekty analizy zespołowej, które wprowadzają nowe metody identyfikacji obiektów odstających, poprawiające dokładność wykrywania. Zaznaczono, że analiza zespołowa w wykrywaniu anomalii jest stosunkowo nowym obszarem badań, ze względu na ograniczoną ilość istniejącej literatury i formalizacji w tej dziedzinie.

Następnie zaprezentowano różnorodne metody wykrywania anomalii z wykorzystaniem technik zespołowych. Rozdział ten zawiera kategoryzację metod według ich niezależności i rodzaju komponentów, co podkreśla różnorodność podejść oraz korzyści płynące z integracji różnych modeli i danych. Omówiono również zalety i wyzwania związane z zastosowaniem metod zespołowych, zwłaszcza w kontekście poprawy dokładności i odporności systemów detekcji anomalii.

W kolejnej części skupiono się na wyborze odpowiednich modeli bazowych, co jest ważne dla optymalizacji całego procesu. Wybór właściwych modeli bazowych wpływa na ogólną wydajność oraz stabilność wyników, co ma znaczenie dla efektywnego wykrywania anomalii. Omówiono różne kryteria wyboru modelu bazowego, takie jak stabilność algorytmu, zakres wartości hiperparametrów oraz charakterystyka danych, które są ważne dla uzyskania optymalnych wyników detekcji. Wskazano na możliwość zastosowania technik, takich jak losowe wybieranie podzbiorów cech oraz adaptacyjne próbkowanie, zwiększających różnorodność i skuteczność zespołów wykrywania anomalii.

Przedostatni podrozdział poświęcono normalizacji wyników w zespołach detekcji anomalii, aby zapewnić ich spójność i ułatwić interpretację, ponieważ algorytmy generują wyniki na różnych skalach, co utrudnia ich bezpośrednią analizę. Dzięki normalizacji można sprowadzić wyniki do jednej skali i ułatwić ich łączenie i porównywanie. Omówiono różne metody normalizacji, takie jak standaryzacja z, normalizacja min-max oraz normalizacja przez medianę. Każda z tych metod ma swoje zalety i wady w kontekście wykrywania anomalii. Przedstawiono również metody przekształcania wyników w estymacje prawdopodobieństwa za pomocą funkcji sigmoid oraz modelowanie wyników jako mieszaniny rozkładów. Dzięki tym metodom wyniki stają się bardziej jednolite i łatwiejsze do zrozumienia, prawdopodobieństwa mogą być też łatwiej łączone i interpretowane.

Na koniec omówiono metody łączenia wyników, pozwalające uzyskać bardziej wiarygodne i kompleksowe analizy, zwiększające ogólną skuteczność wykrywania anomalii. W szczególności zastosowanie technik takich jak średnia arytmetyczna, ważone uśrednianie, mediana wyników oraz bardziej zaawansowane metody, takie jak metoda AOM i metoda progowa, umożliwia efektywną redukcję błędu systematycznego i wariancji. Metody te, wraz z wykorzystaniem rankingu oraz maksymalizacji i minimalizacji wyników, przyczyniają się do lepszego modelowania rzeczywistości i dokładniejszego aproksymowania prawdziwej hipotezy. Techniki takie jak analiza przez rozkład macierzy pomagają zminimalizować wpływ mniej znaczących detektorów, co poprawia ogólną wiarygodność detekcji anomalii.

Podsumowując omawiane kwestie, wprowadzenie innowacyjnych metod i technik analizy zespołowej w identyfikacji anomalii otwiera nowe horyzonty, jednocześnie stawiając wyzwania związane z brakiem formalizacji i danych referencyjnych. Nowe podejścia wymagają głębszych badań i kreatywnych rozwiązań, aby w pełni wykorzystać potencjał analizy zespołowej. Rozwój tych metod w praktyce może zwiększyć niezawodność i efektywność systemów wykrywania anomalii, co jest ważne w kontekście powstających coraz większych, bardziej złożonych i dynamicznych zbiorów danych.

Rozdział 6

Wskaźniki skuteczności w identyfikacji anomalii

W ocenie skuteczności algorytmów identyfikujących anomalie napotykamy na istotne wyzwania. Anomalie zazwyczaj są rzadkie, a dokładne oznaczenia (*ang. ground-truth*), które jednoznacznie identyfikują anomalie, często nie są dostępne. Te oznaczenia, zwane również etykietami (*ang. labels*), są niezbędne, ponieważ pozwalają na ocenę, czy algorytm rzeczywiście wykrywa anomalie, poprzez porównanie jego wyników z rzeczywistymi danymi. Oznaczenia *ground-truth* to dane uznawane za rzeczywiste i wiarygodne, gdyż pochodzą z bezpośrednich obserwacji i pomiarów, a więc z dowodów empirycznych. W przeciwieństwie do informacji uzyskanych poprzez wnioskowanie, prawdziwe oznaczenia opierają się na bezpośrednio zebranych faktach. Problem ten jest szczególnie trudny w przypadku algorytmów nienadzorowanych, gdzie brak oznaczeń utrudnia precyzyjną ocenę ich wydajności. Wykrywanie anomalii można traktować jako szczególny przypadek klasyfikacji binarnej z nierównomiernie rozłożonymi klasami, gdzie klasa rzadka reprezentuje anomalie. Większość danych to przypadki normalne, co prowadzi do poważnego zaburzenia równowagi klas, szczególnie w kontekście zbiorów danych o ekstremalnej nierównowadze klas, analizowanych w niniejszych badaniach.

Podręczniki i kursy uczenia maszynowego najczęściej omawiają algorytmy, takie jak regresja logistyczna, drzewa decyzyjne, SVM itp., w kontekście równomiernie rozłożonych klas. Klasy są często zakładane jako zrównoważone w standardowych zestawach danych używanych w literaturze i kursach, ponieważ umożliwia to łatwiejsze zrozumienie podstawowych zasad działania tych algorytmów. Liczne prace badawcze wykazują, że tradycyjne algorytmy uczenia maszynowego są bardziej skuteczne, gdy klasy w danych są równomiernie rozłożone, co sprzyja ich lepszemu dopasowaniu i wynikom [332]. Co do nowszych algorytmów, to pomimo ostatnich postępów i rosnącej popularności głębokiego uczenia się, istnieje jednak bardzo niewiele prac empirycznych badających efektywność głębokiego

uczenia się w kontekście braku równowagi klasowej [333, 334, 335]. Klasyfikatory często mają problemy z mniejszymi klasami, co prowadzi do potrzeby stosowania specjalnych technik, takich jak nadpróbkowanie (*ang. oversampling*), podpróbkowanie (*ang. undersampling*) czy modyfikacje algorytmów. Domyślne ustawienia wielu popularnych bibliotek uczenia maszynowego, takich jak *scikit-learn* w Pythonie, są zoptymalizowane pod kątem równomiernie rozłożonych klas.

Metryki, jak dokładność (*ang. accuracy*), są często używane jako domyślne miary skuteczności, co jest odpowiednie dla zbalansowanych klas. Gdy jedna klasa jest znacznie mniejsza (anomalia), algorytmy mogą nauczyć się ignorować tę klasę, ponieważ skupienie się na większościowych klasach daje im lepsze wyniki, jednak nie spełnia celu wykrywania rzadkich obiektów. Standardowe metryki, takie jak dokładność, mogą być mylące w przypadku nierównomiernych klas. Na przykład, jeśli w zestawie danych 95% obiektów to przypadki normalne, a 5% to anomalie, algorytm, który zawsze klasyfikuje każdy obiekt jako normalny, osiągnie dokładność 95%, mimo że nie wykryje żadnych anomalii. Oznacza to, że choć algorytm wydaje się być bardzo skuteczny według metryki dokładności, w rzeczywistości całkowicie zawodzi w wykrywaniu anomalii, które były jego głównym celem. To pokazuje, że dokładność może być mylącą metryką w kontekście identyfikacji anomalii, gdyż nie odzwierciedla rzeczywistej skuteczności algorytmu w tym zadaniu.

W takich przypadkach pojawia się pytanie, jak można ocenić skuteczność algorytmu bez dostępności prawdziwych etykiet (*ang. ground-truth labels*)? Większość algorytmów wykrywania anomalii generuje wynik anomalii (*ang. anomaly score*), a próg tego wyniku jest używany do konwersji wyników na etykiety anomalii. Wyniki generowane przez algorytmy wykrywania anomalii nie mogą być uznane za prawdziwe oznaczenia *ground-truth*, ponieważ, jak wcześniej wspomniano, takie oznaczenia nie są dostępne.

Algorytmy generują wartości wyników anomalii, które następnie mogą być użyte do klasyfikacji danych jako anomalie lub normalne obiekty poprzez zastosowanie odpowiedniego progu. Takie wyniki można traktować jako miary wskaźnika „podejrzliwości" danego obiektu, który wskazuje, jak bardzo dany punkt odstaje od normy w kontekście przyjętego modelu. Algorytmy identyfikacji anomalii analizują dane i przypisują każdemu obiektowi wynik anomalii, który odzwierciedla stopień, w jakim dany punkt można uznać za anomalię. Jest to wartość ciągła, która odzwierciedla „odległość" obiektu od typowych wzorców w zestawie danych, informując o stopniu nietypowości obiektu. W kontekście klasyfikacji, wyniki te są przekształcane na etykiety anomalii po ustaleniu odpowiedniego progu lub przyjęciu innych założeń stosowanych do przypisania etykiet. Przykładowo, w analizie statystycznej często zakłada się, że najbardziej odstające wyniki, czyli pierwszy 1% danych, które przekraczają trzy odchylenia standardowe od średniej w rozkładzie normalnym, mogą być klasyfikowane jako anomalie. Wartość wyniku anomalii powyżej tego progu klasyfikuje obiekt jako anomalię, a wartość poniżej jako normalny przypadek. Proces ten jest bardzo ważny, ponieważ wybór progu bezpośrednio wpływa na wskaźniki fałszywie pozytywnych i fałszywie negatywnych wyników:

- **fałszywe negatywy** - jeśli próg jest wybrany zbyt restrykcyjnie, algorytm może nie wykryć prawdziwych anomalii, co oznacza, że prawdziwe anomalie zostaną zaklasyfikowane jako normalne przypadki,

- **fałszywe pozytywy** - jeśli próg jest ustawiony zbyt nisko, algorytm może zaklasyfikować zbyt wiele normalnych obiektów jako anomalie, co prowadzi do nadmiernej liczby fałszywych alarmów.

Wybór odpowiedniego progu jest bardzo ważny i wymaga starannego wyważenia między czułością (zdolnością do wykrywania prawdziwych anomalii) a swoistością (zdolnością do unikania fałszywych alarmów). Ostateczny wybór progu powinien być uzależniony od specyfiki danego problemu i konsekwencji różnych typów błędów. Ponieważ konwencjonalne metody i metryki mogą nie być wystarczające do skutecznego wykrywania anomalii, konieczne jest zastosowanie technik i podejść, które uwzględniają specyfikę problemu. Zamiast dokładności, należy używać metryk, które lepiej oddają wydajność modelu w przypadku nierównowagi klas, takich jak precyzja (*ang. precision*), czułość (*ang. recall*), miara F1 (*ang. F1-score*), krzywe ROC (*ang. receiver operating characteristic*) czy pole AUC (*ang. area under the curve*). Jednak także takie miary jak AUC wymagają dużej ostrożności w stosowaniu, ponieważ nie wszystkie segmenty krzywej ROC muszą mieć jednakowe znaczenie, zwłaszcza w kontekście identyfikacji anomalii. Powyższe kwestie oraz miary oceny skuteczności identyfikacji anomalii zostaną szczegółowo przedstawione w dalszej części rozdziału.

6.1 Macierz pomyłek

Jak zaznaczono we wstępie możliwość oceny skuteczności algorytmu identyfikującego anomalie istnieje nawet w sytuacji, gdy rzeczywiste etykiety nie są używane w procesie uczenia. Zakłada się, że wynik anomalii, tj. wartość liczbową generowaną przez algorytm, można zestawić z rzeczywistymi etykietami w zestawie danych. W kontekście badań, można przyjąć, że dysponujemy zestawem danych z rzeczywistymi etykietami, które dostarczają niezawodnych informacji na temat tego, które obiekty są anomaliami, a które są normalne. Rzeczywiste etykiety nie są wykorzystywane podczas procesu uczenia algorytmu ani generowania wyników anomalii, służą one wyłącznie do oceny działania algorytmu. Porównując wygenerowane przez algorytm wyniki anomalii z rzeczywistymi etykietami, możemy skonstruować macierz pomyłek (*ang. confusion matrix*), jak pokazano w tabeli 6.1. Macierz ta ilustruje, jak często algorytm poprawnie bądź błędnie klasyfikuje obiekty jako anomalie lub przypadki normalne. Analiza macierzy pomyłek dostarcza nam kompleksowego obrazu jakości i skuteczności algorytmu.

Można rozważyć wykrywanie anomalii jako problem klasyfikacji binarnej, w którym dane dzielą się na dwie klasy: anomalie i normalne obiekty. Macierz pomyłek stanowi

Tabela 6.1: Macierz pomyłek dla klasyfikacji anomalii. Źródło: opracowanie własne.

Rzeczywiste etykiety	Przewidywane etykiety	
	Normalna (C_1)	Anomalia (C_2)
Normalna (C_1)	f_{11} (TN)	f_{12} (FP)
Anomalia (C_2)	f_{21} (FN)	f_{22} (TP)

narzędzie do oceny skuteczności klasyfikatora. Macierz ta jest kwadratową strukturą o wymiarach $m \times m$, gdzie m oznacza liczbę klas. W tym przypadku $m = 2$. Wiersze macierzy reprezentują rzeczywiste etykiety klas obiektów testowych, natomiast kolumny odpowiadają etykietom przypisanym przez klasyfikator. Element f_{ij} w macierzy pomyłek wskazuje liczbę obiektów z klasy C_i, które zostały błędnie przypisane do klasy C_j. Poszczególne symbole w macierzy oznaczają:

- **TP** (*ang. true positive*) - liczba przypadków, w których model prawidłowo identyfikuje pozytywność, trafnie rozpoznając obiekt odstający jako odstający,
- **TN** (*ang. true negative*) - liczba przypadków, w których model słusznie identyfikuje negatywność, właściwie klasyfikując standardowy obiekt jako standardowy,
- **FN** (*ang. false negative*) - liczba przypadków, w których model mylnie sugeruje negatywność, niewłaściwie uznając obiekt odstający za standardowy,
- **FP** (*ang. false positive*) - liczba przypadków, w których model niepoprawnie zakłada pozytywność, błędnie klasyfikując standardowy obiekt jako odstający.

Całkowitą liczbę poprawnie zaklasyfikowanych rekordów testowych można określić dzięki analizie omawianej macierzy. Liczbę przykładów testowych oznaczono jako f_{test}. Poprawnie zaklasyfikowane przypadki to $f_{\text{popr}} = \sum_{i=1}^{m} f_{ii}$, a łączna liczba wszystkich błędnych klasyfikacji to $f_{\text{bl}} = f_{\text{test}} - \sum_{i=1}^{m} f_{ii}$ [86]. Macierz pomyłek dostarcza szczegółowych informacji na temat błędów klasyfikacji w różnych dziedzinach. Przykładem może być system wykrywania awarii w samochodach. Błędne zaklasyfikowanie problemu z układem hamulcowym (klasa C_i) jako problemu z silnikiem (klasa C_j) może prowadzić do poważnych konsekwencji dla bezpieczeństwa pojazdu, ponieważ problemy z hamulcami są krytyczne dla bezpiecznego zatrzymania samochodu. Z kolei błędne zaklasyfikowanie problemu z silnikiem jako problemu z układem hamulcowym może prowadzić do niepotrzebnych napraw i kosztów, ale nie jest tak bezpośrednio groźne dla bezpieczeństwa, zdrowia lub nawet życia. Na podstawie TP, TN, FP i FN zawartych w macierzy można wyprowadzić wiele wskaźników skuteczności, takich jak przedstawione w tabeli 6.2.

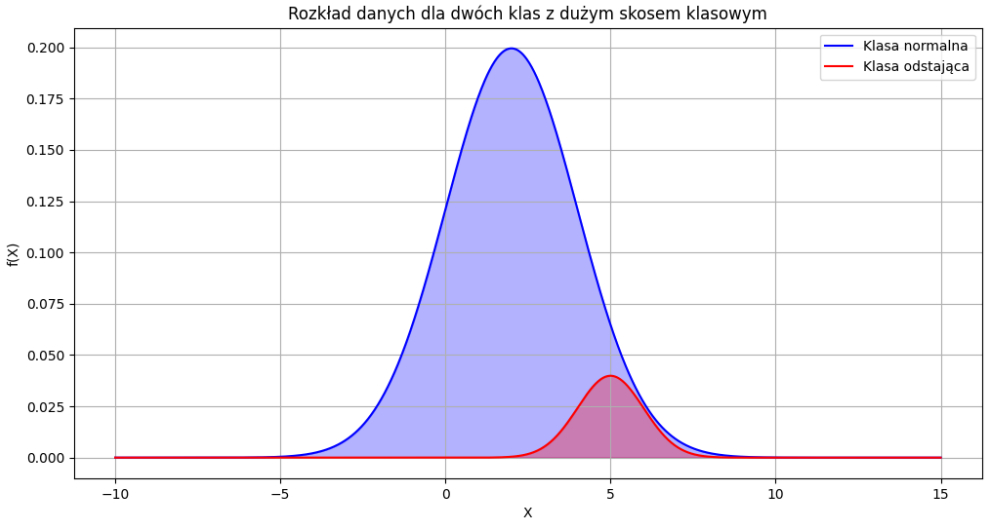
Rozkład normalny, zwany również rozkładem Gaussa, jest jednym z najważniejszych rozkładów w statystyce, charakteryzującym się dzwonowatym kształtem symetrycznym wokół swojej średniej. Wiele naturalnych zjawisk, takich jak wysokość ludzi, ciśnienie krwi czy wyniki testów IQ, można opisać za pomocą tego rozkładu. Na rysunku 6.1 pokazano,

Tabela 6.2: Metryki oceny klasyfikatora binarnego. Źródło: opracowanie własne.

Miara	**Opis**
Specyficzność (Swoistość) $\left(\frac{TN}{TN+FP}\right)$	Odsetek prawidłowo rozpoznanych obiektów normalnych (negatywnych). Ilustruje zdolność modelu do odróżniania prawdziwych przypadków negatywnych od fałszywie pozytywnych.
Dokładność $\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$	Stosunek wszystkich poprawnie zaklasyfikowanych obiektów (zarówno pozytywnych, jak i negatywnych) do całkowitej liczby obiektów. Metryka ta mierzy ogólną dokładność klasyfikacji.
Precyzja $\left(\frac{TP}{TP+FP}\right)$	Odsetek prawidłowo zaklasyfikowanych obiektów odstających (pozytywnych) wśród wszystkich zaklasyfikowanych jako odstające. Miara ta jest istotna dla minimalizacji fałszywych alarmów.
Czułość $\left(\frac{TP}{TP+FN}\right)$	Odsetek prawidłowo wykrytych obiektów odstających (pozytywnych). Jest istotna dla identyfikacji maksymalnej liczby rzeczywistych anomalii.
Miara F1 $\left(\frac{2\cdot\text{precyzja}\cdot\text{czułość}}{\text{precyzja}+\text{czułość}}\right)$	Harmoniczna średnia precyzji i czułości, balansująca te dwie metryki, co pomaga ocenić równowagę między nimi.
Współczynnik FP $\left(\frac{FP}{FP+TN}\right)$	Odsetek błędnie zaklasyfikowanych standardowych obiektów jako odstające. Wskazuje częstość fałszywych alarmów.
Współczynnik FN $\left(\frac{FN}{TP+FN}\right)$	Odsetek rzeczywistych anomalii pominiętych przez model, błędnie uznanych za standardowe.
Wartość predykcyjna negatywna NPV $\left(\frac{TN}{FN+TN}\right)$	Odsetek prawidłowo zidentyfikowanych standardowych obiektów wśród wszystkich zaklasyfikowanych jako standardowe.
Wskaźnik fałszywych odkryć $\left(\frac{FP}{FP+TP}\right)$	Odsetek przypadków, w których model błędnie identyfikuje standardowe obiekty jako anomalie.
Koszt błędnej klasyfikacji (FP+FN)	Suma błędów klasyfikacji, obejmująca zarówno fałszywie pozytywne, jak i fałszywie negatywne wyniki.
Błąd klasyfikacji (1 − Dokładność)	Wskaźnik błędów modelu, pokazujący ogólną częstość niepoprawnych klasyfikacji.

jak niewielka liczba anomalii (klasa mniejszościowa) jest „zakryta” przez ogon rozkładu normalnych obiektów (klasa większościowa). W kontekście znacznego niezbalansowania

klas, większość obiektów należy do jednej klasy (klasa normalna), a tylko niewielki odsetek obiektów stanowią anomalie (klasa mniejszościowa). W rozkładzie normalnym ogon odnosi się do obszarów na końcach krzywej, które zawierają rzadkie wartości. Gdy większość danych jest skupiona wokół średniej (centrum rozkładu), anomalie mogą znajdować się w ogonach rozkładu. Dominacja klasy normalnej sprawia, że jej rozkład jest bardziej widoczny, podczas gdy anomalie, będąc w mniejszości, są mniej widoczne i mogą być interpretowane jako naturalne odchylenia klasy normalnej, szczególnie jeśli znajdują się w jej ogonach. W efekcie algorytmy mogą mieć trudności z odróżnieniem anomalii od ekstremalnych wartości klasy normalnej. Ze względu na to, że anomalie są rzadkie i mogą być mylone z ekstremalnymi wartościami klasy normalnej, algorytmy mogą popełniać błędy. Może to prowadzić do wysokich wskaźników fałszywie pozytywnych (normalne obiekty sklasyfikowane jako anomalie) oraz fałszywie negatywnych (anomalie sklasyfikowane jako normalne obiekty).



Rysunek 6.1: Dystrybucja danych dla dwóch klas z wyraźnym niezbalansowaniem. Źródło: opracowanie własne.

6.2 Metryki oceny klasyfikatorów: Precyzja i Czułość

W kontekście klasyfikacji binarnej, szczególnie w detekcji anomalii, istotne jest precyzyjne określenie skuteczności modelu. Dwie bardzo ważne metryki używane do oceny klasyfikatorów to precyzja i czułość. W niniejszym podrozdziale zdefiniowano te miary oraz omówiono ich zastosowanie w interpretacji wyników identyfikacji anomalii.

Zaczynając od precyzji, dla każdego progu t w detekcji anomalii zbiór anomalii identyfikowanych przy tym progu oznacza się jako $\mathcal{A}(t)$. Wraz ze zmianą wartości t, zmienia

się również rozmiar $\mathcal{A}(t)$. Prawdziwe anomalie, oznaczone jako $\mathcal{A}_{\text{true}}$, reprezentują rzeczywisty zestaw anomalii w analizowanym zbiorze danych. Dla każdego progu t, precyzję definiujemy jako odsetek wykrytych anomalii, które rzeczywiście są anomaliami:

$$\text{Precyzja}(t) = 100 \cdot \frac{|\mathcal{A}(t) \cap \mathcal{A}_{\text{true}}|}{|\mathcal{A}(t)|} \tag{6.1}$$

Wartość precyzji Precyzja(t) nie jest koniecznie monotoniczna względem t, ponieważ zarówno licznik, jak i mianownik mogą zmieniać się w różnym tempie w zależności od t. Z kolei czułość jest definiowana jako odsetek rzeczywistych anomalii, które zostały poprawnie zidentyfikowane przy progu t:

$$\text{Czułość}(t) = 100 \cdot \frac{|\mathcal{A}(t) \cap \mathcal{A}_{\text{true}}|}{|\mathcal{A}_{\text{true}}|} \tag{6.2}$$

Zmieniając wartość progu t, można wygenerować krzywą ukazującą relację między precyzją a czułością. Krzywa ta, określana mianem krzywej precyzji-czułości, nie musi być monotoniczna. Znaczenie tej krzywej oraz powierzchni pod nią, która odzwierciedla ogólną skuteczność modelu klasyfikacyjnego, zostało omówione w dalszej części, w podrozdziale 6.4. Gdy próg t jest zmieniany, liczba wykrytych anomalii ($|\mathcal{A}(t)|$) oraz liczba prawdziwych anomalii wśród wykrytych ($|\mathcal{A}(t) \cap \mathcal{A}_{\text{true}}|$) mogą się zmieniać w różnym tempie. Zmiana progu t może powodować, że precyzja niekoniecznie będzie się regularnie zwiększać lub zmniejszać. Na przykład, zmiana progu może spowodować wzrost liczby fałszywych pozytywnych klasyfikacji (co zmniejszy precyzję) lub wzrost liczby prawdziwych pozytywnych klasyfikacji (co zwiększy precyzję), ale te zmiany nie zawsze będą występować w przewidywalny sposób. Precyzja i czułość różnią się następująco:

- **precyzja** - to miara skuteczności algorytmu, wskazująca, jaki procent wykrytych anomalii rzeczywiście jest anomaliami. Innymi słowy, precyzja koncentruje się na tym, jak dobrze algorytm unika fałszywych alarmów,
- **czułość** - to miara kompletności algorytmu, wskazująca, jaki procent rzeczywistych anomalii został poprawnie wykryty. Czułość mierzy zdolność algorytmu do wykrywania wszystkich prawdziwych anomalii, minimalizując liczbę pominiętych przypadków.

W uczeniu maszynowym i przetwarzaniu języka naturalnego **czułość** może być czasem pomijana lub uśredniana, zwłaszcza gdy większy nacisk kładzie się na pewność co do poprawności klasyfikacji (precyzję). Precyzja jest często mierzona jako procent prognoz pozytywnych, które są rzeczywiście poprawne. Jednak badania wykazały, że w kontekście przetwarzania języka naturalnego i automatycznego tłumaczenia czułość odgrywa fundamentalną rolę w przewidywaniu skuteczności dopasowania słów [336]. W medycynie czułość jest niezwykle istotna, ponieważ celem jest wykrycie wszystkich prawdziwie pozytywnych przypadków. Jest to również jedna z kluczowych miar w analizie ROC, gdzie na osi Y przedstawiana jest jako współczynnik prawdziwie pozytywnych wyników (TPR).

Precyzja, znana również jako trafność w eksploracji danych, to miara proporcji przewidywanych pozytywnych przypadków, które rzeczywiście są prawdziwe. Oznacza to, że precyzja określa, jaki procent przypadków identyfikowanych przez algorytm rzeczywiście odpowiada faktycznym pozytywnym przypadkom. Jest to znacząca miara w ocenie modeli predykcyjnych, istotna w dziedzinach takich jak uczenie maszynowe, eksploracja danych i wyszukiwanie informacji. Jednak w kontekście analizy charakterystyki operacyjnej odbiornika ROC, precyzja nie jest uwzględniana. ROC koncentruje się na czułości TPR oraz współczynniku fałszywie pozytywnych wyników (FPR). Precyzja mierzy dokładność przewidywanych pozytywnych przypadków jako stosunek prawdziwych pozytywnych wyników do wszystkich przewidywanych pozytywnych wyników, w przeciwieństwie do czułości TPR, która ocenia skuteczność w wykrywaniu rzeczywistych pozytywnych przypadków. W praktyce precyzja jest ważna, gdy zależy nam na zminimalizowaniu liczby fałszywie pozytywnych wyników. W aplikacjach takich jak filtrowanie spamu, wykrywanie oszustw czy wyszukiwanie informacji, wysoka precyzja jest niezbędna, aby unikać błędnych alarmów i zapewnić, że większość wykrytych pozytywnych przypadków jest rzeczywiście istotna [337].

W literaturze naukowej można znaleźć wiele analiz dotyczących czułości i precyzji. W jednym z artykułów [338] analizowany jest wybór między metrykami takimi jak czułość i swoistość oraz czułość i precyzja. Podkreślono brak uniwersalnych zasad definiujących wybór par metryk oraz przedstawiono sześć zasad pomagających dobrać odpowiednie miary w zależności od kontekstu i celu klasyfikacji. W innym artykule [339] omówione są wady tradycyjnych metod oceny systemów uczących się, takich jak precyzja i czułość. Zwrócono uwagę na ich stronniczość oraz ryzyko wyciągania błędnych wniosków dotyczących skuteczności modeli. Zaproponowano stosowanie bardziej zaawansowanych miar, takich jak *informedness* (informacyjność) i *markedness* (zaznaczenie), które oferują zrównoważoną ocenę modeli, uwzględniając zarówno przypadki pozytywne, jak i negatywne. *Informedness* to miara oceniająca, jak dobrze model predykcyjny przewiduje rzeczywisty stan i jest zdefiniowana jako:

$$\text{Informedness} = \text{TPR} + \text{TNR} - 1 \tag{6.3}$$

We wzorze (6.3) TNR to swoistość (specyficzność). Miara ta łączy czułość (TPR) i specyficzność (TNR), oferując bardziej zrównoważoną ocenę modelu predykcyjnego. *Markedness* mierzy, jak dobrze wyniki są oznakowane przez model predykcyjny i jest zdefiniowane jako:

$$\text{Markedness} = \text{Precyzja} + \text{Precyzja odwrotna} - 1, \tag{6.4}$$

gdzie:

$$\text{Precyzja odwrotna} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{6.5}$$

Wydajność jest oceniana z perspektywy predykcji, co zapewnia, że przewidywane pozytywne przypadki są rzeczywiście pozytywne, a przewidywane negatywne przypadki są rzeczywiście negatywne. W innym ujęciu analizowana jest relacja między precyzją a czułością w kontekście przeszukiwania dokumentów [340]. Zauważono, że zwiększenie czułości często prowadzi do spadku precyzji, co wskazuje na kompromis między tymi miarami. Proponowane jest dwustopniowe podejście do wyszukiwania, w którym najpierw zwiększana jest czułość poprzez szerokie wyszukiwanie, a następnie poprawiana jest precyzja dzięki bardziej szczegółowemu filtrowaniu wyników.

Miara F1

Miara F1, znana również jako (*ang. F-score*), jest wskaźnikiem wykorzystywanym w analizie danych i klasyfikacji, która harmonizuje precyzję i czułość. Miara F1 jest zdefiniowana jako średnia harmoniczna precyzji i czułości, co można wyrazić wzorem (6.6):

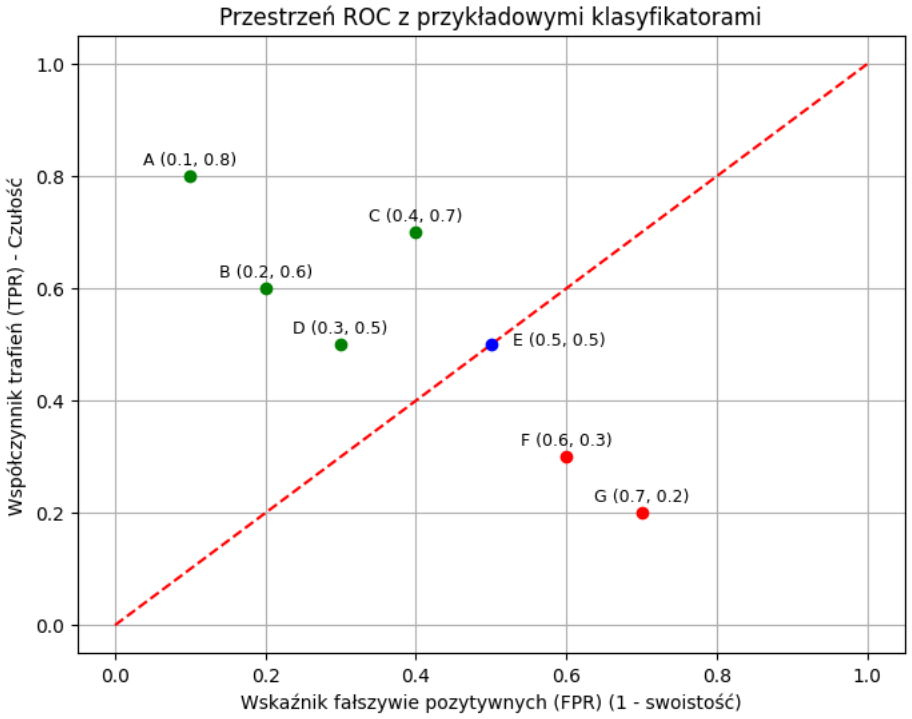
$$\mathrm{F1} = 2 \cdot \frac{\text{precyzja} \cdot \text{czułość}}{\text{precyzja} + \text{czułość}} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{6.6}$$

Średnia harmoniczna jest preferowana nad średnią arytmetyczną, ponieważ skuteczniej uwzględnia wartości obu składowych — precyzji i czułości [341]. Gdy jedna z tych wartości jest znacznie niższa od drugiej, średnia harmoniczna obniża wynik F1, co sprawia, że miara F1 staje się mniej wrażliwa na skrajne różnice między precyzją a czułością w porównaniu do innych średnich. To zjawisko ma szczególne znaczenie przy ocenie wydajności klasyfikatorów, zwłaszcza w sytuacjach, gdzie równowaga między czułością a precyzją jest istotna, na przykład w zastosowaniach medycznych. Brak wykrycia rzeczywistych przypadków pozytywnych może być równie niebezpieczny, jak fałszywe alarmy, które mogą prowadzić do stresu pacjentów, niepotrzebnych badań i obciążenia systemu opieki zdrowotnej. Wysoka wartość miary F1 wskazuje, że zarówno precyzja, jak i czułość są na wysokim poziomie, co sugeruje, że klasyfikator skutecznie identyfikuje pozytywne przypadki, minimalizując fałszywe pozytywy.

Miara F1 jest szeroko stosowana w dziedzinie odzyskiwania informacji, w tym w wyszukiwaniu dokumentów i klasyfikacji zapytań. Jest ona szczególnie przydatna w kontekstach, gdzie duże znaczenie ma klasa pozytywna, a przypadki pozytywne są rzadkie w porównaniu do klas negatywnych. Pomimo swojej popularności, miara F1 bywa krytykowana za równe traktowanie precyzji i czułości [342], co nie zawsze odzwierciedla rzeczywiste koszty różnych typów błędnych klasyfikacji. Alternatywnie można stosować inne miary, takie jak współczynnik korelacji Matthews (MCC) lub miary symetryczne, jak kappa Cohena czy indeks Youdena (J). Te miary uwzględniają zarówno przewidywania prawdziwie pozytywne, jak i prawdziwie negatywne, oferując bardziej zrównoważoną ocenę skuteczności klasyfikatorów. Dodatkowe miary omówiono w podrozdziale 6.5.

6.3 Krzywe w przestrzeni ROC

Jednym z uznanych sposobów ilustrowania efektywności klasyfikatorów jest użycie dwuwymiarowej przestrzeni ROC (*ang. receiver operating characteristics*, ROC). Przestrzeń ROC jest powszechnie stosowana w podejmowaniu decyzji medycznych, a także znajduje zastosowanie w uczeniu maszynowym i badaniach nad eksploracją danych. To narzędzie, stosowane przez dekady w teorii przetwarzania sygnałów [343], doskonale przedstawia zależność pomiędzy współczynnikiem trafień TPR (czułość) a wskaźnikiem fałszywie pozytywnych FPR (1 - swoistość). Przestrzeń ROC rozciąga się w dwóch wymiarach: oś Y odpowiada za wartości współczynnika TPR, natomiast oś X - za wartości współczynnika FPR danego klasyfikatora. Każdy klasyfikator może być zobrazowany jako punkt w tej przestrzeni, którego współrzędne wyznaczają współczynniki TPR oraz FPR. Pozycja punktu reprezentującego klasyfikator w przestrzeni ROC odzwierciedla kompromis między zyskiem, mierzonym wartością współczynnika TPR, a kosztem, mierzonym wartością współczynnika FPR.



Rysunek 6.2: Przestrzeń ROC z przykładowymi klasyfikatorami. Źródło: opracowanie własne.

Rysunek 6.2 przedstawia przykładową przestrzeń ROC z punktami A, B, C, D, E, F i G. Punkt E umieszczono na czerwonej linii reprezentującej klasyfikator losowy. Punkty

F i G znajdują się poniżej czerwonej linii, co wskazuje, że klasyfikator działa gorzej niż losowy. Pozostałe punkty powyżej linii sugerują, że klasyfikatory te działają lepiej niż losowy, wykazując rzeczywistą zdolność do odróżniania przypadków pozytywnych od negatywnych. Przybliżenie osi przestrzeni ROC umożliwia dokładniejsze omówienie ich znaczenia. Dla każdego progu t, czułość TPR definiowana jest jako procent rzeczywistych anomalii poprawnie zgłoszonych przy progu t:

$$\mathrm{TPR}(t) = 100 \cdot \frac{|\mathcal{A}(t) \cap \mathcal{A}_{\mathrm{true}}|}{|\mathcal{A}_{\mathrm{true}}|} \tag{6.7}$$

Wskaźnik fałszywie pozytywnych (FPR) można rozumieć jako „negatywną" czułość, wskazującą procent przypadków negatywnych, które zostały niewłaściwie zidentyfikowane jako anomalie. FPR definiujemy jako procent normalnych przypadków błędnie zidentyfikowanych jako anomalie spośród wszystkich normalnych przypadków dla progu t:

$$\mathrm{FPR}(t) = 100 \cdot \frac{|\mathcal{A}(t) - \mathcal{A}_{\mathrm{true}}|}{|D - \mathcal{A}_{\mathrm{true}}|}, \tag{6.8}$$

gdzie:

- $\mathcal{A}(t)$ oznacza zbiór anomalii identyfikowanych przy progu t,
- $\mathcal{A}_{\mathrm{true}}$ oznacza rzeczywisty zbiór anomalii (*ang. ground-truth*),
- D oznacza cały zbiór danych, na którym przeprowadzana jest analiza.

Idealny klasyfikator byłby reprezentowany w lewym górnym rogu rysunku 6.2, gdzie TPR wynosi 1, a FPR wynosi 0, co wskazuje na maksymalną skuteczność identyfikacji anomalii przy zerowym poziomie fałszywych alarmów. Wykorzystanie wykresu ROC umożliwia nie tylko ocenę, ale także porównanie różnych klasyfikatorów, wyraźnie pokazując, jak każdy z nich balansuje pomiędzy dokładnością a liczbą błędów. Na przykład, klasyfikator charakteryzujący się wysokim TPR, ale również wysokim FPR, może nie być idealny, ponieważ większa liczba fałszywych alarmów może prowadzić do znacznych kosztów. Klasyfikator działa losowo, gdy jego zdolność do klasyfikacji przypadków pozytywnych i negatywnych jest równa, co oznacza, że działa na zasadzie losowego zgadywania. W kontekście przestrzeni ROC, klasyfikator losowy jest reprezentowany przez punkty leżące na przekątnej przestrzeni ROC.

Porównywanie klasyfikatorów może być problematyczne, gdy mają one różne współczynniki TPR i FPR, co oznacza, że każdy z nich ma inne zalety i wady. W takiej sytuacji jeden klasyfikator może mieć wyższy TPR, ale jednocześnie wyższy FPR w porównaniu do innego klasyfikatora. To prowadzi do trudności w jednoznacznym określeniu, który klasyfikator jest lepszy, ponieważ wybór zależy od priorytetów użytkownika. Na przykład w aplikacjach medycznych, gdzie najważniejsze jest wykrywanie chorób, wyższy TPR może być bardziej pożądany, nawet kosztem wyższego FPR. Dodatkowo, koszty związane

z fałszywie pozytywnymi i fałszywie negatywnymi wynikami mogą znacząco wpływać na preferencje użytkownika. Przykładowo, gdy klasyfikator błędnie oznacza negatywny przypadek jako pozytywny. W kontekście medycznym może to oznaczać, że pacjent, który nie ma choroby, zostaje błędnie zdiagnozowany jako chory. Koszty związane z fałszywie pozytywnymi wynikami mogą obejmować:

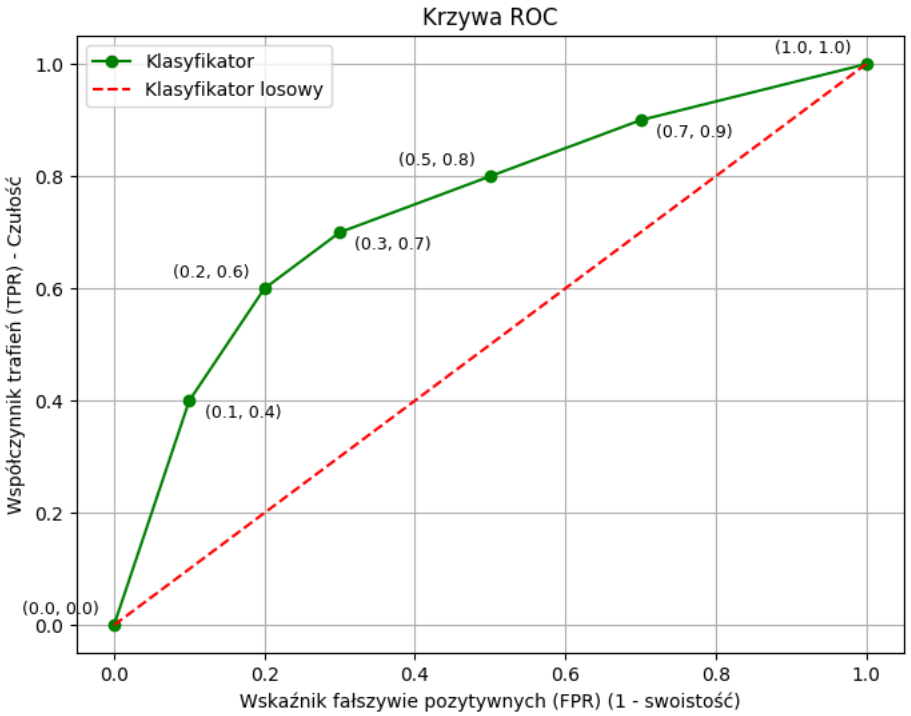
- stres i niepokój pacjenta,
- niepotrzebne leczenie i związane z tym koszty medyczne,
- obciążenie systemu opieki zdrowotnej dodatkowymi, niepotrzebnymi badaniami i konsultacjami.

I odwrotnie, gdy klasyfikator błędnie oznacza pozytywny przypadek jako negatywny, w medycynie oznacza to, że pacjent, który ma chorobę, zostaje błędnie zdiagnozowany jako zdrowy. Koszty związane z fałszywie negatywnymi wynikami mogą obejmować:

- zwiększone ryzyko powikłań i dłuższy czas powrotu do zdrowia,
- wyższe koszty leczenia w przyszłości z powodu zaawansowania choroby,
- opóźnienie w leczeniu, co może prowadzić do pogorszenia stanu zdrowia pacjenta, a nawet śmierci.

Wiele klasyfikatorów, takich jak drzewa decyzyjne czy zestawy reguł, przypisuje każdej instancji jedną z dwóch klas: pozytywną lub negatywną. Po zastosowaniu takiego klasyfikatora do zbioru testowego powstaje pojedyncza macierz pomyłek, co przekłada się na jeden punkt w przestrzeni ROC. Dlatego dyskretne klasyfikatory generują jedynie pojedynczy punkt na wykresie ROC. Z kolei klasyfikatory probabilistyczne, takie jak naiwny Bayes czy sieci neuronowe, przypisują każdej instancji wartość numeryczną, która odzwierciedla prawdopodobieństwo przynależności do danej klasy. Można stworzyć krzywą ROC dla takiego zbioru testowego, wykorzystując zasadę monotoniczności klasyfikacji progowych. Oznacza to, że każda instancja uznana za pozytywną przy określonym progu pozostaje pozytywna również przy wszystkich niższych progach. Dzięki temu instancje testowe można posortować malejąco według ich wyników i przetwarzać kolejno, aktualizując wartości TPR i FPR na bieżąco. Takie podejście umożliwia tworzenie wykresu krzywej ROC, co jest wydajne pod względem obliczeniowym [344].

Krzywe ROC są często wykorzystywane do oceny wydajności modeli. Na przykład, w artykule [345] przedstawiono, jak krzywe ROC i pole AUC służą do oceny i porównania różnych metod wykrywania anomalii na różnych zestawach danych. Podkreślono znaczenie wyboru odpowiedniego progu decyzyjnego, który wpływa na równowagę między wykrywaniem anomalii a minimalizacją fałszywych alarmów. Wykazano, że skuteczność różnych metod zależy od specyfiki zestawu danych, co wskazuje na konieczność porównywania metod na wielu zestawach, aby uzyskać pełniejszy obraz ich mocnych i słabych stron.



Rysunek 6.3: Krzywa ROC z wartościami FPR i TPR dla różnych progów decyzyjnych. Źródło: opracowanie własne.

Rysunek 6.3 przedstawia koncepcję krzywej ROC za pomocą przykładowych danych. Etykiety punktów na wykresie ilustrują wartości FPR i TPR dla różnych progów decyzyjnych. Warto zauważyć, że zarówno „dobra", jak i „zła" czułość rosną monotonicznie wraz ze wzrostem wartości progu t, co prowadzi do wykrywania większej liczby anomalii. Wskaźniki TPR i FPR są obliczane na podstawie stosunków w ramach pozytywnych lub negatywnych przypadków, co oznacza, że zmiana liczby pozytywnych lub negatywnych przypadków w całym zbiorze danych nie wpływa na ich wartości. Z tego powodu krzywe ROC są niewrażliwe na zmiany w rozkładzie klas i pozostają niezależne od proporcji pozytywnych do negatywnych obiektów. Dzięki temu stanowią bardziej uniwersalne narzędzie do oceny skuteczności detektorów identyfikujących anomalie, niezależnie od stopnia zbalansowania zbioru danych. Jednakże, w kontekście wykrywania anomalii, gdzie często mamy do czynienia z silnym przekrzywieniem klas, ta niewrażliwość na zmiany w rozkładzie klas może okazać się wadą.

Aby porównać klasyfikatory, można sprowadzić ich działanie do jednej wartości skalarnej przy użyciu krzywej ROC, reprezentującej oczekiwaną skuteczność. Popularną metodą jest obliczenie pola pod krzywą ROC, znane jako AUC (*ang. area under the curve*, AUC) [346]. Ponieważ AUC jest częścią jednostkowego kwadratu, jego wartość mieści

się w przedziale od 0 do 1. Model generujący przypadkowe decyzje tworzy krzywą ROC w postaci przekątnej od punktu (0,0) do punktu (1,1), której pole powierzchni wynosi 0,5. W związku z tym, żaden realistyczny klasyfikator nie powinien mieć wartości AUC niższej niż 0,5. Do obliczania AUC stosuje się regułę trapezów, która przekształaca schodkową krzywą ROC na gładką aproksymację, ułatwiając tym samym obliczenie pola. Metoda ta dzieli obszar pod krzywą na małe trapezy i sumuje ich pola, aby uzyskać całkowite pole pod krzywą. Estymacja opiera się na przybliżeniu całego obszaru poprzez zsumowanie wszystkich podobszarów, a wzór jest następujący (6.9):

$$AUC_{\text{trapezoid}} = \frac{1}{2}\sum_{i=1}^{n}(f_{i+1} - f_i) \cdot (t_{i+1} + t_i), \tag{6.9}$$

gdzie f to funkcja FPR, a t to TPR, trajektoria jest podzielona na $n - 1$ sekcji. Równanie (6.10)

$$\frac{1}{2} \cdot (f_{i+1} - f_i) \cdot (t_{i+1} + t_i) \tag{6.10}$$

oblicza pole pojedynczego trapezu pod krzywą ROC. Suma tych pól dla wszystkich $n - 1$ segmentów krzywej ROC daje przybliżony obszar pod krzywą, czyli AUC. Jest to liczbowa miara, która podsumowuje ogólną wydajność modelu. Wysoka wartość AUC, bliska 1, wskazuje, że model skutecznie rozróżnia pozytywne i negatywne obiekty. W diagnostyce medycznej, wartość AUC dla testów zazwyczaj mieści się w przedziale od 0,80 do 0,95. Standardowe interpretacje AUC obejmują:

- AUC = 0,9–1,0 – test bardzo dobry,
- AUC = 0,8–0,9 – test dobry,
- AUC = 0,7–0,8 – test satysfakcjonujący,
- AUC = 0,6–0,7 – test średni,
- AUC = 0,5–0,6 – test niedostateczny.

W klasyfikacji medycznej, idealny model powinien być zarówno czuły, czyli skutecznie identyfikować chorych pacjentów, jak i specyficzny, czyli prawidłowo rozpoznawać zdrowych pacjentów. Jednakże, istnieje inherentna sprzeczność między czułością a specyficznością – poprawa jednej zazwyczaj pogarsza drugą. Wybór odpowiedniego klasyfikatora to więc balansowanie między tymi dwoma cechami, mając na uwadze minimalizację kosztów błędnych klasyfikacji [347, 348]. Z perspektywy medycznej, mniej szkodliwym błędem jest zaklasyfikowanie zdrowego pacjenta jako chorego niż odwrotnie [349].

W kontekście identyfikacji anomalii, mając zbiór obiektów ocenionych według ich skłonności do bycia wartościami odstającymi, gdzie wyższe rangi/wyniki wskazują na większe odchylenie, AUC ROC jest równe prawdopodobieństwu, że losowo wybrana para wartości odstającej i nieodstającej będzie prawidłowo uszeregowana, czyli wartość odstająca będzie miała wyższy wynik (*outlier score*) niż wartość nieodstająca [350].

Początkowa część krzywej ROC, znajdująca się blisko osi, jest zazwyczaj bardziej istotna. Ta część odpowiada wysokiej czułości przy niskim odsetku fałszywie pozytywnych wyników. Przykładem może być system wykrywania oszustw finansowych. Istotne jest, aby największe oszustwa (najbardziej odstające przypadki) były wykrywane jak najszybciej, czyli w początkowych analizach transakcji. Klasyfikowanie największych oszustw jako pierwszych (np. na pozycjach 1-100) jest znacznie bardziej użyteczne niż wykrycie ich później (np. na pozycjach 601-701). Poprawne wykrycie anomalii jako jednej z pierwszych (najbardziej odstających) ma większe znaczenie niż jej pozycjonowanie w środkowej części rankingu. AUC jako miara nie rozróżnia istotności tych błędów, ponieważ traktuje wszystkie części krzywej jednakowo. W efekcie, chociaż AUC może być wysokie, klasyfikator może nie być optymalny w kontekście aplikacji, gdzie priorytetem jest szybkie i dokładne wykrywanie najbardziej odstających wartości.

6.4 Krzywa precyzji-czułości kontra ROC

Zróżnicowanie klas jest powszechnym zjawiskiem w praktycznych zastosowaniach, zwłaszcza w dziedzinach intensywnie korzystających z uczenia maszynowego do analizy predykcyjnej, takich jak diagnoza chorób, wykrywanie oszustw, przewidywanie bankructwa czy identyfikacja podejrzanych działań. Klasyfikacja danych z niezrównoważonym rozkładem klas stanowi wyzwanie dla większości uznanych algorytmów klasyfikacyjnych, które zazwyczaj są projektowane z myślą o równomiernie rozłożonych klasach [351]. Krzywa precyzji-czułości (*ang. precision-recall curve*, PRC) przewyższa krzywą ROC w ocenie klasyfikatorów binarnych na niezrównoważonych zestawach danych, szczególnie przy identyfikacji rzadkich, pozytywnych przypadków, takich jak anomalie. PRC dostarcza bardziej szczegółowych informacji o skuteczności modeli, co jest niezbędne w aplikacjach wymagających szybkiego wykrywania, na przykład w medycynie diagnostycznej. Dzięki koncentracji na pozytywnych wynikach i pomijaniu prawdziwie negatywnych klas, PRC są nieocenione w analizie anomalii, co zostało potwierdzone zarówno w symulacjach, jak i badaniach literaturowych [352, 353]. Jak zilustrowano w tabeli 6.3, krzywa ROC oraz krzywa precyzji-czułości PRC mają swoje unikalne zastosowania i zalety w różnych scenariuszach. Wartość AUC dla obu krzywych pokazuje ogólną skuteczność algorytmów w różnicowaniu klas, a szczegółowe porównanie tych metryk pozwala na głębsze zrozumienie ich praktycznej przydatności w wykrywaniu anomalii.

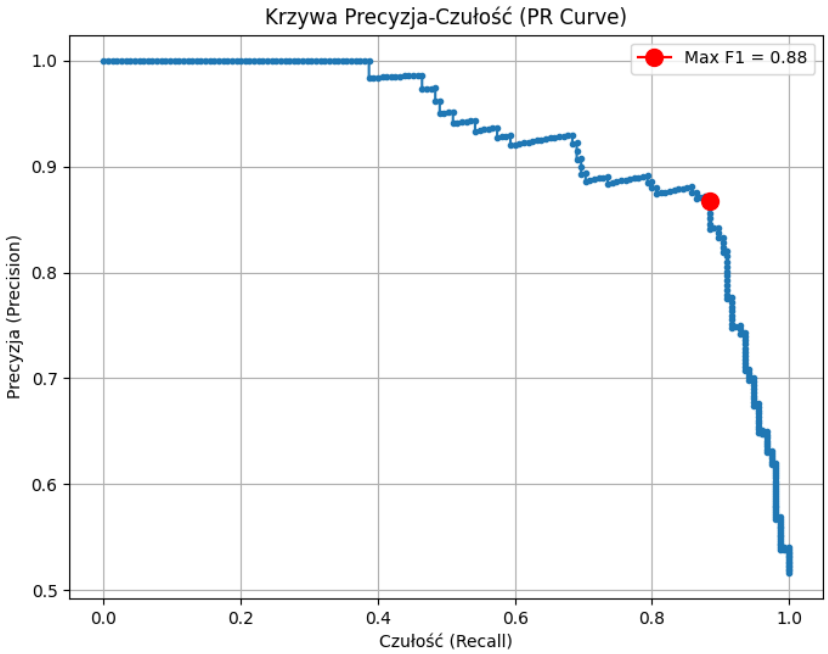
Rysunek 6.4 przedstawia przykładową krzywą precyzji-czułości dla modelu regresji logistycznej wytrenowanego na wygenerowanym zbiorze danych. Oś pozioma (X) reprezentuje czułość, czyli odsetek prawdziwych pozytywnych przypadków, które zostały poprawnie zidentyfikowane przez model. Wartość ta pokazuje, jak skutecznie model wykrywa wartości pozytywne (np. anomalie). Oś pionowa (Y) reprezentuje precyzję, czyli odsetek prawidłowo zidentyfikowanych pozytywnych przypadków wśród wszystkich przypadków zidentyfikowanych jako pozytywne. Wartość ta pokazuje, jak dokładne są

Tabela 6.3: AUC ROC i PR AUC - metryki oceny modeli identyfikujących anomalie. Źródło: opracowanie własne.

Miara	**Opis**
Krzywa ROC (obszar AUC)	Krzywa ROC pomaga zrozumieć, jak czułość zmienia się w zależności od specyficzności. Wartość AUC stanowi kompleksową miarę ogólnej skuteczności algorytmu. Określa zdolność modelu do rozróżniania między różnymi klasami, dostarczając jedną skalarną wartość, która oddaje równowagę między wskaźnikami prawdziwie pozytywnych wyników a wskaźnikami fałszywie pozytywnych wyników przy różnych progach decyzyjnych. AUC nie rozróżnia istotności błędów w różnych częściach krzywej ROC, co może prowadzić do sytuacji, gdzie klasyfikator, mimo wysokiego AUC, nie jest optymalny w aplikacjach wymagających szybkiego i dokładnego wykrywania najbardziej odstających wartości.
Krzywa precyzji-czułości (obszar PR AUC)	Krzywa precyzji-czułości służy do oceny wydajności algorytmów w scenariuszach, gdzie klasy są niezrównoważone i anomalie występują rzadko. Wysoka wartość PR AUC świadczy o zdolności modelu do skutecznego wykrywania takich anomalii. Oceniając algorytmy w kontekście niezrównoważonych danych, krzywa precyzji-czułości dostarcza bardziej precyzyjnego obrazu wydajności, zwłaszcza gdy liczba fałszywie negatywnych wyników jest istotna.

pozytywne klasyfikacje modelu. Krzywa pokazuje związek między precyzją a czułością przy różnych progach klasyfikacji. Każdy punkt na krzywej odpowiada innemu progowi klasyfikacji, który decyduje o tym, kiedy model klasyfikuje przypadki jako pozytywne.

Pole pod krzywą precyzji-czułości (PR AUC) jest ważną metryką uzupełniającą analizę skuteczności modeli. W kontekście niezrównoważonych klas, gdzie pozytywne przypadki są rzadkością, PR AUC dostarcza wartościowych informacji o skuteczności modelu w różnicowaniu rzadkich pozytywnych obiektów od negatywnych. Dzięki zdolności do monitorowania wydajności modelu przy różnych progach klasyfikacji, PR AUC umożliwia szczegółową analizę precyzji i czułości w różnych sytuacjach. Jako że krzywe PRC



Rysunek 6.4: Krzywa precyzji-czułości oraz punkt z optymalnym wynikiem miary F1. Źródło: opracowanie własne.

koncentrują się na jakości identyfikacji pozytywnych przypadków, PR AUC pomaga zidentyfikować najlepszy próg decyzyjny dla modelu, co jest niezbędne przy wykrywaniu rzadkich zdarzeń, takich jak anomalie w danych czy szybka diagnoza w medycynie. W odróżnieniu od ROC AUC, które może prowadzić do mylnych wniosków przy silnie niezrównoważonych klasach, PR AUC zapewnia bardziej wiarygodną ocenę, skupiając się na precyzji i czułości bez uwzględnienia prawdziwie negatywnych wyników.

Algorytmy optymalizujące obszar pod krzywą ROC często nie osiągają podobnych wyników dla krzywej precyzji-czułości PRC. Różnice te wynikają z istotnych rozbieżności między tymi metrykami, szczególnie widocznych w niezrównoważonych danych. W metryce ROC główną rolę odgrywają wskaźniki TPR i FPR, co może nie oddawać pełnego obrazu efektywności klasyfikatora przy dużej nierówności między klasami pozytywnymi a negatywnymi. Natomiast krzywa PRC koncentruje się na precyzji i czułości, pozwalając dokładniej ocenić zdolność algorytmu do rozróżniania prawdziwych pozytywów od licznych negatywnych przypadków. W efekcie algorytm z dobrze optymalizowanym AUC-ROC może nie być tak skuteczny w zakresie AUC-PR, co jest ważne, gdy błędna identyfikacja pozytywnych przypadków może prowadzić do dużych strat. Optymalizacja algorytmów pod kątem PRC jest istotna w warunkach dużej asymetrii między klasami, pozwalając na skuteczniejsze rozróżnianie rzadkich pozytywnych od licznych negatywnych obiektów. Tradycyjne metody oceny, takie jak krzywa ROC, mogą nieadekwatnie

odzwierciedlać rzeczywistą skuteczność algorytmów przy niezrównoważonych danych, prowadząc do błędnych wniosków. Dlatego optymalizacja AUC-PR jest ważna dla dostosowania algorytmów do warunków, gdzie fałszywe pozytywy niosą wysokie ryzyko, jak w medycynie diagnostycznej. Algorytmy zaprojektowane z myślą o PRC poprawiają skuteczność w praktycznych aplikacjach, co jest cenne w sytuacjach wymagających szybkiego i dokładnego diagnozowania [354].

6.5 Alternatywne miary oceny anomalii

W dziedzinie analizy danych, zwłaszcza w kontekście identyfikacji anomalii, standardowe metryki, takie jak dokładność i krzywa ROC, często nie wystarczają do pełnej oceny skuteczności modelu. Tradycyjne podejścia, mimo swojej popularności, mogą prowadzić do niewłaściwych interpretacji. Dlatego badacze i praktycy coraz częściej sięgają po alternatywne miary, które oferują lepszy wgląd w rzeczywistą skuteczność algorytmów w wykrywaniu rzadkich i istotnych zdarzeń. W tym podrozdziale skupiono się na przedstawieniu i omówieniu alternatywnych miar, takich jak na przykład indeks Youdena (J), współczynnik korelacji Matthewsa (MCC), zrównoważona dokładność (BAC) czy precyzja dla pierwszych M wyników (P@m). Każda z tych metryk ma swoje zalety w kontekście wyzwań związanych z detekcją anomalii. Opis tych miar pokazuje, jak mogą one lepiej oceniać modele w różnych scenariuszach, podkreślając ich przydatność poza tradycyjną oceną skuteczności. W tabeli 6.4 przedstawiono alternatywne miary skuteczności, przydatne, gdy konwencjonalne metryki mogą zawieść. Poniżej przedstawiono charakterystyki tych miar.

Zrównoważona dokładność BAC (*ang. balanced accuracy*, BAC) jest miarą, która równo traktuje czułość i swoistość. Jest to szczególnie użyteczne w sytuacjach, gdzie klasy są niezrównoważone, ponieważ zapewnia bardziej zrównoważony obraz skuteczności modelu. BAC unika faworyzowania którejkolwiek z klas, co jest ważne w przypadkach, gdzie koszty błędnej klasyfikacji jednej klasy mogą być równie znaczące co drugiej. Miara ta jest wartościowa w systemach diagnostycznych w medycynie, gdzie równie ważne jest wykrywanie zarówno obecności choroby (czułość), jak i potwierdzanie jej braku (swoistość). BAC jest definiowana jako średnia arytmetyczna czułości i swoistości:

$$\text{BAC} = \frac{\text{Czułość} + \text{Swoistość}}{2} \tag{6.11}$$

Współczynnik Matthewsa MCC (*ang. Matthews correlation coefficient*, MCC) jest cenioną metryką, która dostarcza oceny jakości klasyfikacji binarnej. To, co wyróżnia MCC, to zdolność do efektywnego działania nawet w przypadku bardzo niezbalansowanych klas, co jest typowe dla problemów związanych z wykrywaniem anomalii. MCC uwzględnia wszystkie cztery komórki macierzy pomyłek: prawdziwie pozytywne (TP), fałszywie

Tabela 6.4: Dodatkowe miary wydajności dla klasyfikatora binarnego. Źródło: opracowanie własne.

Miara	**Opis**
Próg rozpowszechnienia (Prevalence Threshold, PT)	$\frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$ Poziom występowania anomalii, przy którym system detekcji staje się użyteczny. Punkt, przy którym korzyści z identyfikacji rzeczywistych anomalii równoważą koszty związane z fałszywymi alarmami.
Indeks Youdena (J)	$TPR + TNR - 1$ Miara zdolności do poprawnego rozróżniania między dwiema klasami, takimi jak obiekty normalne i anomalie. Wartości bliskie 1 wskazują na wysoką skuteczność.
Współczynnik Matthewsa (MCC)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}}$ Miara jakości klasyfikacji binarnej, uwzględniająca wszystkie cztery części macierzy pomyłek: TP, FP, TN, FN. Wartość MCC waha się od -1 do +1.
Wskaźnik Fałszywych Pominięć (FOR)	$\frac{FN}{TN+FN}$ Mierzy, jak wiele rzeczywistych anomalii zostało błędnie zaklasyfikowanych jako standardowe.
Współczynnik Kappa Cohena (κ)	$\frac{2 \times (TP \times TN - FN \times FP)}{(TP+FP) \times (FP+TN) + (TP+FN) \times (FN+TN)}$ Miara zgodności między dwoma zestawami klasyfikacji, szczególnie użyteczna w przypadku niezrównoważonych danych.
Zrównoważona dokładność (BAC)	$\frac{TPR+TNR}{2}$ Średnia z czułości i specyficzności. Miara BAC jest używana, gdy klasy są niezbalansowane.
Precyzja przy M (P@m)	Mierzy precyzję modelu na pierwszych M pozycjach rankingu wyników, pokazując, jak skutecznie model identyfikuje najważniejsze anomalie.

pozytywne (FP), prawdziwie negatywne (TN) oraz fałszywie negatywne (FN) przypadki, co czyni go bardziej wiarygodnym w ocenie wyników niż inne metryki, które mogą skupiać się tylko na jednym aspekcie klasyfikacji.

Wartość MCC waha się od -1 do +1, gdzie:

- +1 oznacza idealną predykcję,
- 0 wskazuje na brak przewidywanej korelacji między predykcjami a rzeczywistymi klasami,
- -1 wskazuje na całkowitą niezgodność między predykcją a rzeczywistością.

MCC jest definiowany jako (6.12):

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{6.12}$$

Interpretacja wzoru:

- licznik to różnica między iloczynem prawdziwie pozytywnych i prawdziwie negatywnych wyników oraz iloczynem fałszywie pozytywnych i fałszywie negatywnych wyników, co pokazuje zgodność pomiędzy obserwacjami a predykcjami,
- mianownik jest to czynnik normalizujący, który uwzględnia liczbę wyników w każdej kategorii decyzyjnej, co pozwala na porównywalność wyników pomiędzy różnymi testami.

Współczynnik kappa Cohena (κ) (*ang. Cohen's Kappa*) jest miarą, która pozwala ocenić zgodność między predykcjami modelu a rzeczywistymi etykietami z uwzględnieniem możliwości przypadkowego osiągnięcia zgodności. Jest to metryka wartościowa w sytuacjach, gdzie dane są niezbalansowane. W tradycyjnej macierzy pomyłek stosowanej w uczeniu maszynowym i statystyce do oceny klasyfikacji binarnych, wzór kappa Cohena można zapisać w następujący sposób (6.13):

$$\kappa = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP}) \times (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) \times (\text{FN} + \text{TN})} \tag{6.13}$$

Wartość kappa waha się od -1 do +1, gdzie:

- +1 oznacza pełną zgodność,
- 0 wskazuje na zgodność równą przypadkowej,
- -1 oznacza całkowitą niezgodność.

Indeks Youdena (J), znany również jako statystyka Youdena, jest miarą oceny zdolności modelu klasyfikacyjnego do różnicowania między dwoma klasami, na przykład między normalnymi i odstającymi obiektami. Indeks Youdena jest wartościowy w kontekście zrównoważonej miary oceny modelu, uwzględniającej jednocześnie zarówno czułość, jak

i specyficzność. Jest to szczególnie ważne w niezbalansowanych zbiorach danych, gdzie jedna z klas jest znacznie rzadsza od drugiej. Dzięki temu, że indeks Youdena koncentruje się zarówno na prawdziwie pozytywnych, jak i na fałszywie pozytywnych wynikach, pozwala na bardziej holistyczne podejście do oceny skuteczności modelu niż czułość lub specyficzność wzięte osobno. Wzór na indeks Youdena jest następujący (6.14):

$$J = \text{TPR} + \text{TNR} - 1 \tag{6.14}$$

Indeks Youdena przyjmuje wartości od -1 do 1, gdzie:

- wartość 0 oznacza, że skuteczność modelu jest losowa,
- wartość 1 oznacza idealną zdolność do rozróżnienia między klasami, z maksymalną czułością (100%) i bez żadnych fałszywie pozytywnych wyników (0% FPR),
- wartość -1 wskazuje, że wszystkie predykcje są błędne w najgorszy możliwy sposób.

Indeks Youdena jest często stosowany w analizach ROC do wyboru optymalnego punktu odcięcia dla wyników diagnostycznych. Może być również użyty jako kryterium wyboru najlepszego modelu spośród kilku konkurencyjnych modeli. W kontekście problemów wieloklasowych, jego zastosowanie rozszerza się poza klasyfikację binarną, gdyż jest on ekwiwalentem wskaźnika Informedness, który omówiono w podrozdziale 6.2 dotyczącym precyzji i czułości.

Precyzja przy M (*ang. precision-at-M*, P@m) jest miarą oceny wydajności modeli klasyfikacyjnych, która koncentruje się na skuteczności identyfikacji najważniejszych przypadków na górnych pozycjach rankingu wyników. Jest to metryka użyteczna w aplikacjach, gdzie ważne jest szybkie identyfikowanie najbardziej istotnych anomalii, na przykład w monitorowaniu transakcji finansowych, systemach zabezpieczeń, czy analizie danych medycznych. P@m mierzy stosunek prawdziwie pozytywnych wyników TP do wszystkich przypadków sklasyfikowanych jako pozytywne wśród pierwszych M pozycji w rankingu modelu. Jest to metryka skoncentrowana na precyzji w określonym, wąskim fragmencie zbioru danych, co jest istotne, gdy najbardziej znaczące są pierwsze wyniki, a nie ogólna dokładność modelu na całym zbiorze danych. Wysoka wartość P@m wskazuje na to, że model efektywnie identyfikuje najważniejsze anomalie, co jest wartościowe w sytuacjach, gdzie pilne zidentyfikowanie i reagowanie na anomalie może zapobiegać znacznym stratom lub ryzykom. Na przykład, w monitorowaniu transakcji finansowych, gdzie ważne jest szybkie wykrycie potencjalnych prób oszustwa, wysoka precyzja w czołowych wynikach jest niezbędna. P@m jest przydatna, gdy ilość danych jest ogromna, a analizowane są tylko te przypadki, które znajdują się na najwyższych pozycjach wyników – te najbardziej podejrzane lub zagrożone. Jest to praktyczne podejście w przypadkach, gdy użytkownik systemu lub analityk nie jest w stanie manualnie przeanalizować wszystkich wyników, a zależy mu na maksymalizacji efektywności wykrywania w ograniczonym, ale istotnym zakresie danych.

Próg rozpowszechnienia PT (*ang. prevalence threshold*, PT) jest użytecznym wskaźnikiem w systemach detekcji anomalii, pozwalającym określić poziom występowania anomalii, przy którym system zaczyna być skuteczny. Ten próg wskazuje na punkt równowagi, w którym korzyści wynikające z prawidłowego wykrycia anomalii równoważą się z kosztami generowania fałszywych alarmów. Próg rozpowszechnienia PT wskazuje, przy jakim minimalnym poziomie występowania rzeczywistych anomalii, system detekcji zaczyna przynosić więcej korzyści niż strat, uwzględniając zarówno prawdziwe pozytywne jak i fałszywe pozytywne wyniki.

6.6 Błędy popełniane przy analizie porównawczej

Model może wydawać się skuteczny w kontrolowanych testach, co nie zawsze przekłada się na jego skuteczność w rzeczywistych zastosowaniach. Dlatego ważne jest, aby podejście do wyboru hiperparametrów było bardziej zrównoważone i odzwierciedlało prawdziwe warunki, w których model będzie używany. Można to osiągnąć, testując model na różnorodnych danych, które lepiej symulują rzeczywiste scenariusze, oraz unikając nadmiernego dopasowania modelu do danych testowych, które mogą być niereprezentatywne dla rzeczywistych zastosowań.

Jeśli dane, na których testowany jest algorytm, nie są wystarczająco różnorodne lub reprezentatywne dla typów danych, które algorytm będzie przetwarzał w rzeczywistym zastosowaniu, wyniki testów mogą nie odzwierciedlać prawdziwej skuteczności algorytmu. Na przykład, jeśli dane testowe zawierają tylko kilka rodzajów anomalii, podczas gdy w rzeczywistości może ich być znacznie więcej, algorytm może być niewystarczająco przygotowany na inne rodzaje anomalii, które pojawią się w praktyce. Testy mogą być przeprowadzane na danych, które mają inny rozkład niż dane, na których algorytm będzie operował w rzeczywistym środowisku. Na przykład, algorytm może być testowany na danych, gdzie anomalie są stosunkowo częste w porównaniu do ich rzeczywistej częstotliwości w aplikacji docelowej. To może prowadzić do nieprawidłowej kalibracji algorytmu, który może stać się zbyt czuły lub zbyt odporny na wykrywanie anomalii. Algorytmy powinny być testowane na danych, które jak najlepiej odzwierciedlają warunki, w jakich będą używane, włączając w to różnorodność przypadków, różnice w częstotliwości występowania anomalii i potencjalne zmiany w danych z czasem. Przykładowo, używanie technik walidacji krzyżowej z różnymi zestawami danych może pomóc zrozumieć, jak algorytm radzi sobie w różnych scenariuszach. W praktyce często wykorzystuje się informacje o etykietach danych (czyli wiedzę o tym, które przypadki są anomaliami) do ustalania, jak dobrze model radzi sobie z identyfikacją anomalii. W sytuacji rzeczywistej, szczególnie w scenariuszach nienadzorowanych, taka wiedza może nie być dostępna. Optymalizacja parametrów, która opiera się na tej wiedzy, może być więc myląca.

Każdy algorytm ma określone hiperparametry, które można dostosować, aby zwiększyć jego skuteczność lub dostosować do specyficznych danych. Przykładowo algorytm k-naj-

bliższych sąsiadów używa hiperparametru k, który określa liczbę sąsiadów branych pod uwagę do klasyfikacji punktu. Maszyna wektorów nośnych jednoklasowa wykorzystuje hiperparametry takie jak margines błędu C oraz rodzaj jądra (kernel). Każdy z tych algorytmów może wymagać zupełnie innego zestawienia i zakresu hiperparametrów do optymalnej pracy, co oznacza, że nie ma jednolitego, uniwersalnego sposobu ich dostosowania. Jeśli różne algorytmy wymagają różnych hiperparametrów, to trudno jest ocenić, który z nich jest „lepszy", bazując wyłącznie na wynikach uzyskanych przy optymalnych ustawieniach dla każdego z nich. Wymaga to szerokiej analizy w różnych scenariuszach z różnymi zestawami hiperparametrów. Znalezienie odpowiednich hiperparametrów dla każdego algorytmu jest procesem, który wymaga czasu i eksperymentów. Co więcej, różne algorytmy mogą różnie reagować na zmiany w swoich hiperparametrach, co utrudnia bezpośrednie porównanie ich skuteczności. Porównując algorytmy, ważne jest, aby uwzględnić różnice w ich naturze i wymaganiach dotyczących hiperparametrów. Porównanie powinno obejmować analizę, jak zmienia się wydajność każdego algorytmu przy różnych ustawieniach, a nie tylko przy jednym, „najlepszym" zestawie hiperparametrów.

Aby sprawiedliwie ocenić różne algorytmy, powinno się przeprowadzić testy przy różnych konfiguracjach hiperparametrów, aby zobaczyć, jak każdy z nich radzi sobie w szerokim zakresie sytuacji. Używanie średnich wyników skuteczności, takich jak mediana AUC z różnych testów, może pomóc zniwelować wpływ ekstremalnych wartości hiperparametrów i dać bardziej zrównoważony obraz skuteczności algorytmu. W kontekście nienadzorowanego uczenia, gdzie nie można stosować standardowych metod walidacji, doświadczenie analityka w doborze hiperparametrów i interpretacji wyników odgrywa bardzo ważną rolę.

6.7 Wybór modelu - testy istotności statystycznej

Każda hipoteza badawcza zaczyna się od obserwacji lub subiektywnej opinii. Aby przekształcić te wstępne spostrzeżenia w uzasadnione twierdzenia, stosujemy metody statystyczne. Hipotezy badawcze wymagają dokładnej weryfikacji, co często oznacza testowanie wielu hipotez statystycznych. Odrzucenie hipotezy zerowej (H_0), która zakłada brak efektu lub zależności, wzmacnia naszą wiarę w hipotezę badawczą (H_1). Testy statystyczne są narzędziami używanymi do analizy danych w celu weryfikacji hipotez badawczych. Wykorzystują metody matematyczne do oceny prawdopodobieństwa, czy zaobserwowane różnice lub zależności są statystycznie istotne, czy też mogą być wynikiem przypadkowej zmienności. Nieistotne wyniki mogą sugerować konieczność dalszych badań lub użycie innych metod analizy [355].

Jednak istnieją różne opinie na temat istotności statystycznej. Jednym z interesujących w ostatnich latach artykułów na ten temat, jest opublikowany w czasopiśmie „Nature", pod intrygującym tytułem „Retire statistical significance" [356]. Artykuł podkreśla różnorodne perspektywy naukowców, którzy twierdzą, że nadmierne poleganie na tradycyjnej istot-

ności statystycznej, zwłaszcza na wartościach p, może prowadzić do mylnych wniosków i niepełnego obrazu wyników badań. Naukowcy sugerują, że bardziej złożone i przejrzyste podejścia statystyczne są niezbędne do uzyskania wiarygodnych wyników badawczych. W artykule przedstawiono argumenty za rezygnacją z podejmowania decyzji badawczych wyłącznie na podstawie tego, czy wartość p jest mniejsza od 0,05. Artykuł nawołuje do zaprzestania klasyfikowania wyników jako „istotnych" lub „nieistotnych" statystycznie, ponieważ taka dychotomia prowadzi do błędnych interpretacji i nadmiernej pewności co do wyników. Autorzy proponują bardziej zniuansowane podejście do wnioskowania statystycznego, kładąc nacisk na pełne oszacowania, przejrzystość oraz prezentację pełnych danych i kontekstu, zamiast podejmowania binarnych decyzji opartych na wartościach p. Artykuł zyskał szerokie poparcie wśród naukowców, co podkreśla, jak powszechny jest problem związany z nadużywaniem istotności statystycznej w badaniach. Ponad 800 sygnatariuszy, w tym statystycy, badacze kliniczni, biolodzy i psycholodzy, poparło propozycje zawarte w artykule. Inny artykuł z czasopisma „The American Statistician" [357] porusza kwestię replikowalności badań naukowych. Autorzy argumentują, że problem replikowalności niekoniecznie wynika z błędów w metodologii, lecz z nadmiernego polegania na statystycznej istotności (szczególnie wartości p) i mylnym postrzeganiu wyników jako definitywnych dowodów na potwierdzenie hipotez.

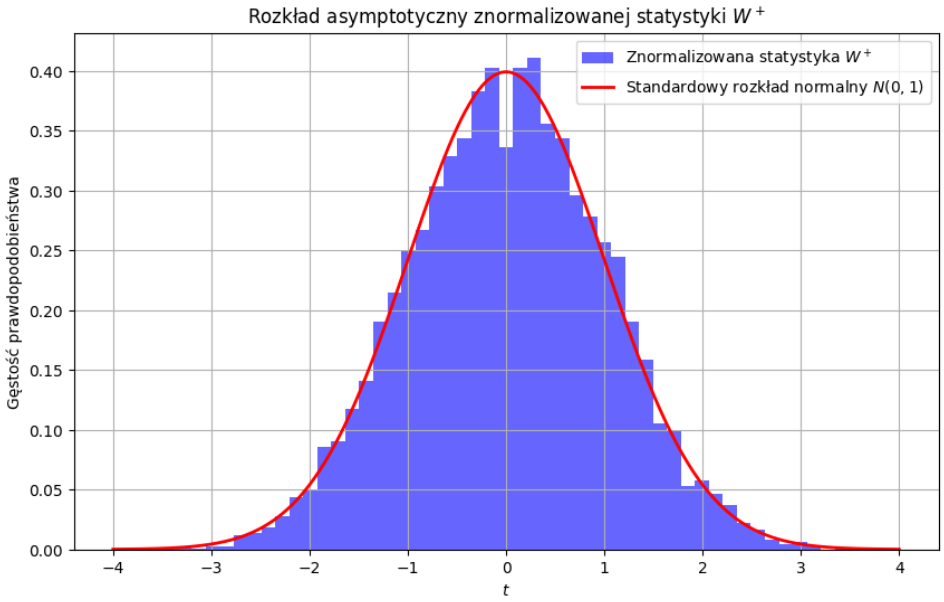
Pomimo tych zastrzeżeń w badaniach, w tej rozprawie odwołano się do istotności statystycznej, aby podkreślić jej znaczenie w kontekście analizy danych. Należy jednak uwrażliwić czytelników, że wyniki testów statystycznych nie są jedynym i ostatecznym kryterium oceny wyników badawczych. Przechodząc do zasad istotności statystycznej, podstawowe pojęcia dotyczące testów statystycznych obejmują:

- **hipoteza zerowa** (H_0) - zakłada brak efektu lub różnicy. Jest to twierdzenie, które testujemy, z nadzieją na jego odrzucenie,
- **hipoteza alternatywna** (H_1) - zakłada istnienie efektu lub różnicy. Jest to twierdzenie, które chcemy udowodnić,
- **poziom istotności** (α) - przyjęty poziom ryzyka odrzucenia hipotezy zerowej, zazwyczaj 0,05 (5%),
- **Wartość** p - prawdopodobieństwo uzyskania wyników co najmniej tak ekstremalnych jak te obserwowane, zakładając, że hipoteza zerowa jest prawdziwa,
- **statystyka testowa** - obliczenie używane do podjęcia decyzji o odrzuceniu lub nieodrzuceniu hipotezy zerowej.

Weryfikacja hipotez statystycznych stanowi fundament badań naukowych, umożliwiając systematyczne i obiektywne ocenianie twierdzeń badawczych. Dzięki temu możemy formułować wiarygodne wnioski i głębiej zrozumieć mechanizmy rządzące badanymi zjawiskami. Istnieje wiele testów statystycznych, które możemy stosować, zakładając normalność rozkładów badanych cech. Jednak w praktyce rozkłady danych często odbiegają od tej idealizacji, co może prowadzić do błędnych wniosków. Aby uniknąć takich pułapek,

warto skupić się na metodach testowania, które są niezależne od założeń dotyczących rozkładu cech przy spełnieniu hipotezy zerowej. Tego rodzaju podejście pozwala na bardziej uniwersalne i niezawodne oceny w różnych kontekstach badawczych. Jedną z takich metod jest test Wilcoxona, szerzej opisany w książce [358].

Test Wilcoxona, zaproponowany w 1945 roku w artykule [359], można zastosować do porównania wyników dwóch algorytmów, które klasyfikują wartości odchyleń od największych do najmniejszych oraz oceniają procent pokrycia prawdziwych odchyleń przez te znalezione przez algorytmy. Stosowanie testu Wilcoxona pozwoli ocenić, czy różnice między wynikami obu algorytmów są statystycznie istotne, niezależnie od rozkładu danych, co ilustruje rysunek 6.5.



Rysunek 6.5: Gdy liczba obserwacji n zmierza do nieskończoności ($n \to \infty$), znormalizowana statystyka W^+ (odjęcie wartości oczekiwanej i podzielenie przez pierwiastek z wariancji) zbiega do standardowego rozkładu normalnego $N(0, 1)$. Oznacza to, że dla dużych prób W^+ będzie rozkładem normalnym z wartością oczekiwaną $\mathbb{E}W^+$ i wariancją $\mathrm{Var}\, W^+$. Źródło: opracowanie własne.

Załóżmy, że zebraliśmy $2n$ obserwacji, po dwie dla każdego z n przypadków. Niech i będzie indeksem danego przypadku, a_i będzie pierwszą, a b_i drugą obserwacją przypadku i. Przypadki mogą obejmować wyniki działania dwóch algorytmów na tych samych danych, gdzie a_i to wynik pierwszego algorytmu, a b_i to wynik drugiego algorytmu.

Niech $d_i = b_i - a_i$ dla $i = 1, \ldots, n$. Zakłada się, że różnice d_i są niezależne. Każda różnica d_i pochodzi z populacji o identycznym ciągłym rozkładzie, symetrycznym względem

wspólnej mediany λ. Testowaną hipotezą zerową jest:

$$H_0 : \lambda = 0 \tag{6.15}$$

Hipoteza zerowa zakłada, że mediana różnic między parami obserwacji a_i i b_i wynosi zero, co sugeruje brak istotnych różnic między wynikami obu algorytmów. Algorytm wyliczania statystyki testu Wilcoxona:

- obliczenie różnic między każdą parą obserwacji d_i,
- uporządkowanie wartości bezwzględnych różnic w kolejności rosnącej $|d_1|, \ldots, |d_n|$,
- zrangowanie tak otrzymanego zbioru i oznaczenie rang przez R_i. Rangi są przypisywane w taki sposób, że najmniejsza wartość bezwzględna różnicy otrzymuje rangę 1, następna rangę 2, i tak dalej. Rangi związane (identyczne) uzyskują wartość średnią,
- ten krok pozwala określić, jak duża część rang pochodzi od dodatnich różnic, co wskazuje, czy jeden z algorytmów systematycznie daje wyższe wyniki. Zdefiniowanie statystyki W^+ jako sumy rang R_i dla których $d_i > 0$. Sumujemy rangi tylko tych różnic, które są dodatnie:

$$W^+ = \sum_{\{i:d_i>0\}} R_i \tag{6.16}$$

Właściwości statystyki Wilcoxona:

- Wartość oczekiwana (średnia) statystyki W^+:

$$E(W^+) = \frac{n(n+1)}{4} \tag{6.17}$$

- Wariancja statystyki W^+:

$$\mathrm{Var}(W^+) = \frac{n(n+1)(2n+1)}{24} \tag{6.18}$$

- Rozkład asymptotyczny: Dla dowolnej liczby t, gdy $n \to \infty$:

$$\mathrm{P}\left(\frac{W^+ - \mathbb{E}W^+}{\sqrt{\mathrm{Var}\, W^+}} \le t\right) \to \Phi(t), \tag{6.19}$$

  gdzie $\Phi(t)$ jest dystrybuantą standardowego rozkładu normalnego $N(0,1)$. Oznacza to, że dla dużych próbek, rozkład statystyki W^+ można przybliżyć rozkładem normalnym o średniej $E(W^+)$ i wariancji $\mathrm{Var}(W^+)$, co pokazano na rysunku 6.5.

Do obliczenia wartości p dla prób o małej liczności (zwykle przyjmuje się $n \le 20$) korzysta się z tablic statystycznych. Natomiast dla dużych prób stosuje się przybliżenie rozkładem normalnym, wykorzystując podane wcześniej parametry.

Alternatywy dla testu Wilcoxona

Test Wilcoxona jest popularnym testem nieparametrycznym do porównywania dwóch skorelowanych próbek. Istnieją jednak inne testy, które mogą być używane w podobnych sytuacjach, w zależności od specyficznych założeń i charakterystyk danych. Więcej informacji można znaleźć w literaturze naukowej, na przykład w znanej pracy Cohena [360]. Inne znane pozycje związane z istotnością statystyczną to [358, 361, 362, 363, 364]. Oto kilka alternatyw podsumowanych w tabeli 6.5:

Tabela 6.5: Przykłady testów statystycznych alternatywnych do testu Wilcoxona. Źródło: opracowanie własne.

Test	**Opis**
Test t-Studenta dla grup zależnych	Test parametryczny t-Studenta [365] dla grup zależnych stosuje się, gdy pomiary danej zmiennej są dokonywane dwukrotnie w różnych warunkach, przy założeniu, że wariancje tej zmiennej w obu pomiarach są zbliżone. Analizowana jest różnica pomiędzy parami pomiarów ($d_i = b_i - a_i$). Różnica służy do testowania hipotezy, że jej średnia w populacji wynosi 0, zakładając jej normalny rozkład.
Test McNemara	Test nieparametryczny McNemara [366] służy do weryfikacji hipotezy o zgodności wyników dwóch pomiarów $A_j^{(1)}$ i $A_j^{(2)}$ cechy A_j dla tego samego obiektu X_i. Badana cecha może mieć tylko dwie kategorie (oznaczone (+) i (−)). Test McNemara zakłada, że różnica wyników pomiarów ma asymptotyczny rozkład chi-kwadrat z jednym stopniem swobody.
Test U Manna-Whitneya	Test nieparametryczny Manna-Whitneya [135] stosuje się, aby sprawdzić, czy różnice pomiędzy medianami badanej zmiennej w dwóch populacjach są nieistotne. Zakłada się, że rozkłady zmiennej są zbliżone, co można potwierdzić za pomocą testu rang Conovera [367]. Dla dużych prób mamy rozkład normalny. Podobny jest test Kruskala-Wallisa [368], lecz stosowany do analizy więcej niż dwóch grup.
Test Friedmana	Test nieparametryczny Friedmana [369] służy do weryfikacji hipotezy o równości median w analizie wariancji powtarzanych pomiarów dla rang. Stosuje się go, gdy pomiary badanej zmiennej są dokonywane wielokrotnie ($k \geq 2$) w różnych warunkach. Podlega rozkładowi chi-kwadrat, a liczba stopni swobody jest o jeden mniejsza od liczby porównywanych grup.

6.8 Podsumowanie

W tym rozdziale przeanalizowano wskaźniki skuteczności stosowane w identyfikacji anomalii. Omówiono metody i narzędzia oceny modeli klasyfikacyjnych, takie jak precyzja, czułość oraz bardziej zaawansowane techniki, jak krzywe ROC i PRC. Uwagę poświęcono zarówno tradycyjnym, jak i alternatywnym wskaźnikom oceny anomalii. Zróżnicowane metryki, w tym indeks Youdena, współczynnik korelacji Matthewsa MCC, zrównoważona dokładność BAC oraz precyzja na M najlepszych wynikach P@m, zostały przedstawione jako ważne narzędzia dostarczające istotnych informacji o wydajności modeli w kontekście niezrównoważonych zbiorów danych.

W rozdziale wskazano, że konwencjonalne metryki, takie jak dokładność i krzywa ROC, mogą nie dostarczać rzetelnych danych o skuteczności algorytmów w sytuacjach z rzadkimi zdarzeniami. W kontekstach medycznych, finansowych czy bezpieczeństwa zaleca się stosowanie bardziej zaawansowanych miar, które lepiej oddają wyzwania związane z niezrównoważonymi zbiorami danych. Porównanie krzywych precyzji-czułości i ROC ukazało ich specyficzne zastosowania oraz ograniczenia, podkreślając znaczenie odpowiedniego wyboru progu decyzyjnego, wpływającego na wydajność modelu oraz balans między wykryciem anomalii a minimalizacją fałszywych alarmów. Wyniki tych analiz mają nie tylko teoretyczne znaczenie, ale także praktyczne zastosowanie w systemach, gdzie decyzje oparte na danych mogą mieć poważne implikacje. Zrozumienie i stosowanie zaawansowanych metryk ma znaczenie dla maksymalizacji skuteczności wykrywania anomalii, co może prowadzić do usprawnień w wielu dziedzinach.

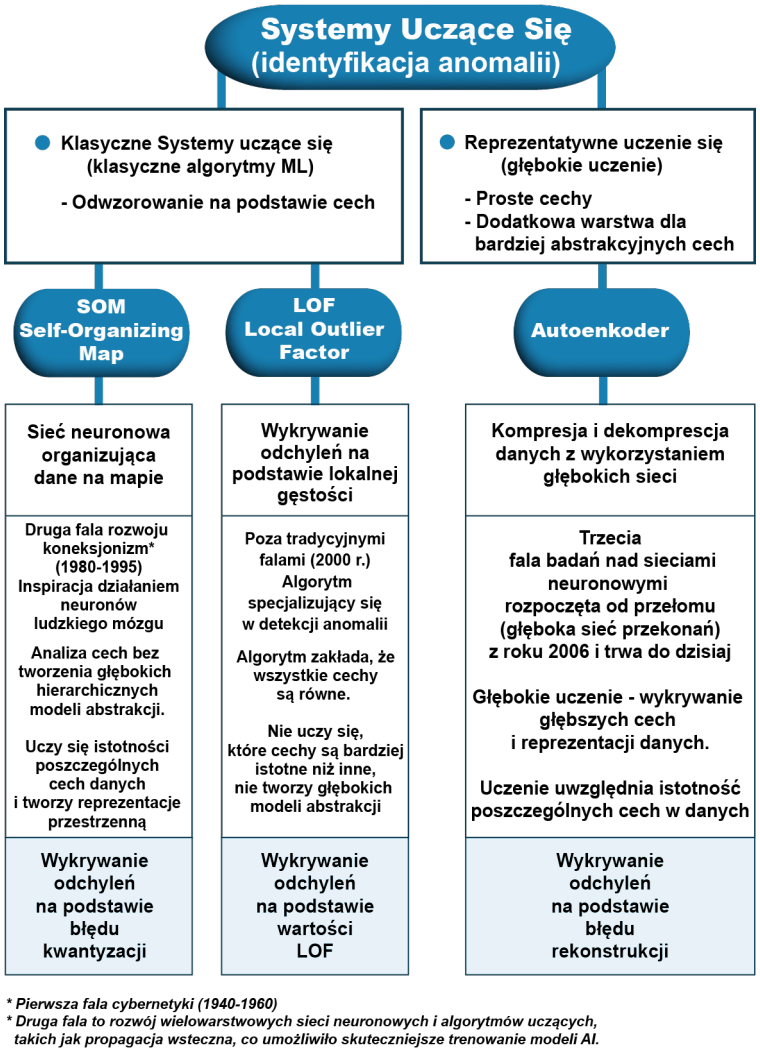
Dodatkowo omówiono błędy popełniane przy analizie porównawczej oraz metody statystyczne służące do oceny istotności różnic w wydajności modeli, zachęcając do skorzystania z odpowiedniej literatury naukowej po dalsze informacje. Testy istotności statystycznej odgrywają ważną rolę w wyborze modelu, umożliwiając ocenę, czy zaobserwowane różnice są statystycznie istotne, co jest niezbędne dla podejmowania świadomych decyzji w analizie danych. Istotne wyniki potwierdzają hipotezę badawczą, podczas gdy nieistotne wyniki mogą sugerować potrzebę dalszych badań lub zastosowania innych metod analizy. Testy statystyczne pomagają określić, czy wyniki badań mają na tyle małe prawdopodobieństwo przypadkowego wystąpienia, że można je uznać za dowód rzeczywistej zależności lub efektu, co jest ważne w naukowym podejściu do weryfikacji hipotez badawczych. W rozdziale wskazano też, że istnieją odmienne zdania w kwestii testów statystycznych.

Podsumowując, rozdział ten stanowi źródło wiedzy dla analityków danych, oferując teoretyczne podstawy oraz praktyczne wskazówki dotyczące identyfikacji i radzenia sobie z anomaliami w danych. Znajomość tych narzędzi i ich stosowanie pomaga unikać błędów interpretacyjnych, zapewniając bardziej wiarygodne wyniki analizy.

Rozdział 7

Projekt systemu Trinity SALT

Zespoły mogą być wykorzystywane do zwiększania efektywności detektorów bazowych na różne sposoby. Pierwsza technika polega na zastosowaniu pojedynczego detektora bazowego w połączeniu z metodami takimi, jak agregacja cech oraz podpróbkowanie. Druga technika opiera się na kombinacji wielu detektorów bazowych, co prowadzi do uzyskania większej różnorodności. W ostatnich latach przeprowadzono wiele badań porównujących algorytmy wykrywania anomalii. Niestety, większość z tych analiz koncentruje się na wynikach osiąganych przez podstawowe detektory, pomijając ich wersje zespołowe. Można to porównać do oceny indywidualnych zawodników bez uwzględnienia współdziałania zespołu. Szczegółowe omówienie w rozdziale 5 wskazuje, że zespołowe podejście do wykrywania anomalii jest stosunkowo nowym obszarem badań w porównaniu do innych zagadnień eksploracji danych, takich jak klasyfikacja czy grupowanie. Powody tej względnej nowości zostały starannie wyjaśnione na początku wspomnianego rozdziału. W rozdziale 4 przedstawiono trzy różne algorytmy wykrywania anomalii, które wybrano do badań: Self-Organizing Maps SOM, Autoenkoder AE oraz Local Outlier Factor LOF. Ich podstawowe cechy i właściwości podsumowano na schemacie przedstawionym na rysunku 7.1. Algorytmy te zostały starannie dobrane ze względu na ich unikalne podejścia do analizy danych. Celowo wybrano algorytmy trzech różnych typów, aby osiągnąć lepszą różnorodność i wyższą wydajność w metodzie zespołowej. W badaniach zastosowano metodę zespołową Trinity SALT, która stanowi heterogeniczne połączenie trzech wybranych algorytmów bazowych. Ta strategia pozwala na lepszą skuteczność identyfikacji anomalii poprzez wykorzystanie mocnych stron każdego z algorytmów. W efekcie uzyskano stabilne i dokładne wyniki, przewyższające te osiągane przez pojedyncze algorytmy bazowe. W kolejnych sekcjach zaprezentowano aplikację webową jako opracowane narzędzie do identyfikacji anomalii oraz szczegóły jej implementacji. Wybrane zbiory danych oraz szczegółowe wyniki badań dotyczące skuteczności poszczególnych algorytmów oraz ich wersji zespołowej zostaną omówione w następnym rozdziale 8.

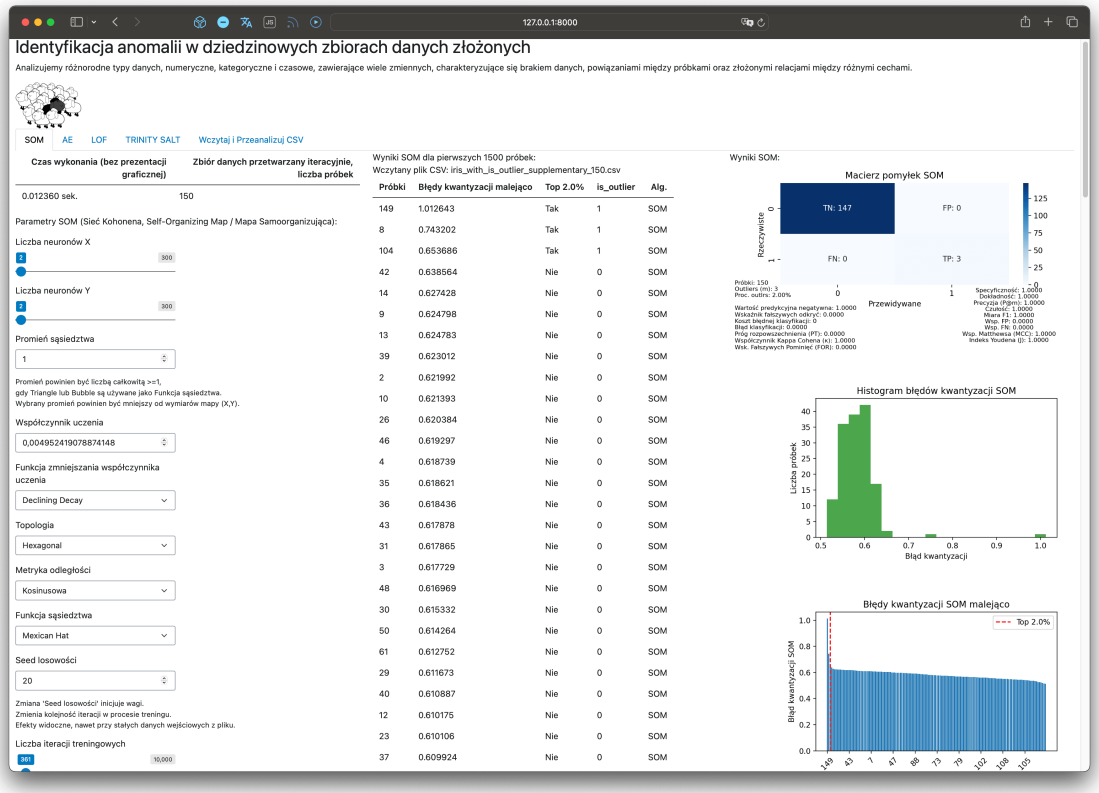


Rysunek 7.1: Analizowane algorytmy i ich kluczowe właściwości w kontekście metod zespołowych. Źródło: opracowanie własne.

7.1 Aplikacja webowa do identyfikacji anomalii

Aplikacja webowa, wyposażona w interaktywne panele nawigacyjne, które wizualizują i analizują dane w czasie rzeczywistym, została stworzona jako narzędzie badawcze w odpowiedzi na potrzeby analiz naukowych dotyczących detekcji anomalii. Pomimo dostępności wielu algorytmów wykrywania anomalii zaimplementowanych w języku Python, konieczne było opracowanie rozwiązania dostosowanego do specyficznych wymagań prowadzonych badań. Analizy wykazały brak narzędzi stosujących metody zespołowe do identyfikacji anomalii. Autorskie oprogramowanie oferuje zaawansowane metody prze-

kształcania danych ilościowych, jakościowych i mieszanych, co umożliwia wszechstronną analizę i identyfikację anomalii metodą zespołową. Fragment głównego okna programu po wczytaniu spreparowanego zbioru Iris zaprezentowano na rysunku 7.2. Plik zamieszczono na płycie DVD i w linkach do zbiorów. Przewijanie w dół ujawnia dodatkowe informacje dostępne w interfejsie. Głównymi komponentami tego interfejsu są wyniki najbardziej odstających obiektów oraz graficzna prezentacja wyników, w tym macierz pomyłek, która dostarcza istotnych informacji na temat skuteczności wykrywania anomalii.



Rysunek 7.2: Zrzut ekranu z aplikacji webowej, przedstawiający działanie narzędzia w ramach pracy naukowej. Źródło: opracowanie własne.

Skorzystano z Shiny dla Python, nowoczesnego narzędzia do tworzenia aplikacji webowych z użyciem programowania reaktywnego, co umożliwia budowanie interaktywnych i dynamicznych aplikacji internetowych. Shiny dla Python został oficjalnie wprowadzony w lipcu 2023 roku, rozszerzając ramy oryginalnego Shiny, które zyskało popularność w języku programowania R, na ekosystem Pythona. Kod aplikacji definiuje interaktywne UI w Shiny, zawierające zakładki dla różnych algorytmów detekcji anomalii (SOM, AE, LOF) oraz analizę wyników metodą zespołową Trinity SALT. Każda zakładka pozwala użytkownikowi na konfigurację parametrów wybranego algorytmu, uruchomienie obliczeń

oraz wizualizację wyników. Dzięki Shiny umożliwiono łatwe łączenie analizy danych z interaktywnymi wykresami, tabelami oraz kontrolkami, które pozwalają użytkownikom na manipulowanie i przeglądanie wyników analizy w czasie rzeczywistym. Stanowi to wsparcie w zaawansowanych analizach naukowych, pozwalając na efektywne przetwarzanie i wizualizację danych w dynamicznym, interaktywnym środowisku.

Aplikacja została umieszczona na serwerze Amazon Web Services (AWS), który dostarcza infrastrukturę serwerową. Wersja pokazowa działa w darmowym środowisku, oferującym tylko 1 GB RAM, co jest wystarczające do obsługi niewielkich zbiorów danych i demonstracji aplikacji. Aktualny adres dostępu (aktualizowany pod DOI) podano w przypisie[1]. Aplikacja w pełnej wersji znajduje się na płycie DVD dołączonej do rozprawy, co umożliwia jej uruchomienie na własnym komputerze przy użyciu odpowiednich zasobów. Szczegóły dotyczące instalacji i uruchamiania aplikacji opisano w podrozdziale 7.4.

7.2 Projekt systemu Trinity SALT

System Trinity SALT został zaprojektowany jako interaktywna aplikacja webowa umożliwiająca użytkownikom wczytywanie danych, przeprowadzanie analiz oraz wizualizację wyników. Kluczowym elementem projektu jest użycie metod zespołowych do detekcji anomalii, w szczególności kombinacji trzech algorytmów: SOM, AE oraz LOF.

Algorytmy zostały zaimplementowane od podstaw, korzystając wyłącznie z biblioteki `numpy` [297]. Jest to popularna biblioteka języka Python, służąca do pracy z tablicami, macierzami i obliczeń numerycznych, zapewniająca wsparcie dla dużych, wielowymiarowych tablic oraz bogaty zbiór funkcji matematycznych. NumPy, to skrót od Numerical Python, biblioteka została stworzona w 2005 roku przez Travisa Oliphanta i jest projektem open source. Jest ona zoptymalizowana pod kątem wydajności, oferując obiekty tablicowe, które są znacznie szybsze niż tradycyjne listy Pythona. NumPy jest częściowo napisany w Pythonie, ale jego istotne części wykorzystują języki C i C++, co dodatkowo przyspiesza obliczenia. Kod źródłowy NumPy znajduje się w repozytorium GitHub (`https://github.com/numpy/numpy`), co umożliwia współpracę wielu programistów nad jego rozwojem. Skorzystano również z bibliotek do wizualizacji danych, takich jak `matplotlib`, `seaborn`, które umożliwiają tworzenie zaawansowanych i interaktywnych wykresów oraz diagramów. Nie zastosowano żadnych gotowych rozwiązań ani bibliotek takich jak TensorFlow czy Keras, które umożliwiają implementację skomplikowanych algorytmów w kilku linijkach kodu. Te biblioteki dostarczają gotowe funkcje i moduły, które upraszczają proces implementacji złożonych modeli. Jednak, aby zachować pełną kontrolę nad implementacją i zapewnić porównywalność wyników, zdecydowano się na użycie `numpy`. Klasy dotyczące SOM, AE czy LOF w każdym zaimplementowanym algorytmie wykorzystują jedynie tablice, macierze i funkcje z biblioteki `numpy`.

[1] http://trinitysalt.pl:8000/, https://doi.org/10.5281/zenodo.13822086

W celu stworzenia systemu zaimplementowano klasy: `LofOutlier`, `SomOutlier` i `AutoencoderOutlier`, z których każda realizuje inny algorytm detekcji anomalii. Szczegółowe wyjaśnienia ich działania znajdują się poniżej, a dodatkowe informacje w obszernych komentarzach opisujących poszczególne elementy kodu w pliku `app.py` na płycie DVD.

Klasa `LofOutlier`

Klasa `LofOutlier` zawiera wszystkie metody potrzebne do obliczenia lokalnego współczynnika osobliwości LOF. Zaimplementowano klasyczny algorytm LOF przedstawiony przez jego autorów [98], posługujący się następującymi metodami:

- `__init__`: inicjalizuje obiekt LofOutlier z danymi wejściowymi, minimalną liczbą punktów w sąsiedztwie oraz metryką odległości:

  ```
  def __init__(self, data, MinPts=2, distance_metric='hamming'),
  ```

- `generate_distance_matrix`: generuje macierz odległości dla danych wejściowych przy użyciu wybranej metryki odległości,
- `k_distance`: oblicza odległość k-tego najbliższego sąsiada dla każdego punktu w macierzy odległości,
- `reachability_distance`: oblicza odległość osiągalności dla każdego punktu na podstawie macierzy odległości i odległości k-tego najbliższego sąsiada,
- `k_nearest_neighbors_for_each_point`: znajduje k najbliższych sąsiadów dla każdego punktu na podstawie macierzy odległości,
- `local_reachability_density`: oblicza lokalną gęstość osiągalności dla każdego punktu,
- `Local_Outlier_Factor`: oblicza wartość lokalnego czynnika odstępstwa dla każdego punktu.

Opisana implementacja klasy `LofOutlier` jest zgodna z podstawowymi założeniami i metodami przedstawionymi przez Breuniga i współautorów [98]. Jednakże, użycie różnych metryk odległości stanowi rozszerzenie oryginalnej koncepcji, które nie zostało opisane w pracy. Dodatkowo, implementacja pozwala na przetwarzanie danych w blokach, co umożliwia analizę dużych zbiorów danych bez nadmiernego obciążania pamięci, co stanowi oryginalne rozszerzenie algorytmu.

Klasa `SomOutlier`

Klasa `SomOutlier` implementuje mapę samoorganizującą się SOM i zawiera następujące metody:

- `__init__`: inicjalizuje mapę SOM z określonymi parametrami, takimi jak liczba neuronów, liczba cech wejściowych, promień sąsiedztwa, współczynnik uczenia,

funkcja zmniejszania współczynnika uczenia, funkcja sąsiedztwa, topologia, metryka i ziarno losowe:

```
def __init__(self, x, y, input_len, radius=4, learning_rate=0.1,
             decay_function=power_decay,
             neighborhood_function='bubble',
             topology='hexagonal',
             metric='cosine', seed=42),
```

- `train`: trenuje mapę SOM na podstawie danych wejściowych przez określoną liczbę iteracji,
- `winner`: zwraca współrzędne wygrywającego neuronu dla danego wejścia,
- `update`: aktualizuje wagi na podstawie wygranego neuronu, współczynnika uczenia i funkcji sąsiedztwa,
- `quantization`: przyporządkowuje dane wejściowe do najbliższych neuronów i zwraca odpowiadające im wagi,
- `quantization_error`: oblicza błąd kwantyzacji dla każdej próbki na podstawie danych wejściowych,
- `calculate_u_matrix`: oblicza macierz U, reprezentującą średnie odległości między neuronami na mapie SOM.

Opisana implementacja jest zgodna z podstawowymi założeniami i metodami przedstawionymi przez Kohonena [262], jednak zawiera pewne rozszerzenia, takie jak różne funkcje zmniejszania współczynnika uczenia, różne funkcje sąsiedztwa oraz różne metryki odległości, aby zwiększyć elastyczność i zastosowanie algorytmu w różnych scenariuszach detekcji anomalii.

Klasa `AutoencoderOutlier`

Klasa `AutoencoderOutlier` implementuje autoenkoder i zawiera następujące metody:

- `__init__`: inicjalizuje model autoenkodera z określonymi parametrami, takimi jak liczba ukrytych warstw, liczba próbek, funkcje aktywacji, ziarno losowe oraz funkcja straty.

```
def __init__(self, hidden_layers, num_samples,
             input_activation='sigmoid',
             output_activation='sigmoid',
             seed=42, loss_function='mean_squared_error'),
```

- `initialize_weights`: inicjalizuje wagi i biasy sieci neuronowej, losując wagi zgodnie z rozkładem normalnym oraz skalując odpowiednio do liczby neuronów w warstwach. Biasy są inicjalizowane jako wektory zerowe,
- `train`: trenuje model autoenkodera na podstawie danych wejściowych przez określoną liczbę epok,
- `forward_pass_with_activations`: przeprowadza propagację w przód przez model sieci neuronowej, obliczając aktywacje dla wszystkich warstw, łącznie z warstwą wyjściową. Proces ten obejmuje obliczanie zsumowanego sygnału wejściowego dla każdej warstwy oraz zastosowanie odpowiedniej funkcji aktywacji do uzyskania wynikowych aktywacji. Metoda ta jest ważna dla generowania prognoz sieci oraz dla treningu modelu [157],
- `sgd_optimizer`: optymalizuje wagi i biasy modelu przy użyciu algorytmu optymalizacyjnego gradientu stochastycznego (*ang. stochastic gradient descent, SGD*). Metoda ta stanowi rozszerzenie algorytmu spadku gradientu, opisanego w sekcji 2.3.2. Więcej na ten temat SGD można znaleźć w literaturze naukowej [50],
- `adam_optimizer`: optymalizuje wagi i biasy modelu za pomocą algorytmu Adam (*ang. adaptive moment estimation*, Adam). Algorytm Adam, zaproponowany przez Kingma i Ba w 2014 roku, jest adaptacyjnym algorytmem optymalizacyjnym. Więcej na ten temat można znaleźć w literaturze naukowej [253],
- `backward_pass`: przeprowadza pasmo wsteczne dla aktualizacji wag i biasów modelu za pomocą metody gradientu prostego. Podczas tego procesu obliczane są gradienty funkcji straty względem wag i biasów na każdej warstwie, a następnie używane do aktualizacji tych parametrów w kierunku minimalizacji błędu modelu. Metoda ta jest zgodna z zasadami przedstawionymi przez Rumelharta i współpracowników [157, 50],
- `backward_pass_l1`: przeprowadza pasmo wsteczne z regularyzacją L1, aby zmniejszyć błąd uogólnienia i zapobiec nadmiernemu dopasowaniu modelu do danych treningowych. Regularyzacja L1 dodaje do funkcji celu sumę wartości bezwzględnych wag $\lambda||w||_1$, gdzie $||w||_1 = \sum_i |w_i|$. Ta metoda minimalizuje sumę kwadratów reszt przy jednoczesnym ograniczeniu sumy wartości bezwzględnych współczynników, co prowadzi do uzyskiwania modeli, w których niektóre współczynniki są dokładnie zerowe. Regularyzacja L1 znana jest także jako (*ang. least absolute shrinkage and selection operator*, LASSO) [50, 250],
- `backward_pass_l2`: przeprowadza pasmo wsteczne z regularyzacją L2, aby zmniejszyć błąd uogólnienia i zapobiec nadmiernemu dopasowaniu modelu do danych treningowych. Najpopularniejszy rodzaj standardowych kar dla parametrów to kara

L2, znana jako zanikanie wagi. Ta strategia regularyzacji przybliża wagi do zera, dodając składnik regularyzacji $\frac{1}{2}\lambda||w||_2^2$ do funkcji celu. Norma L2 wag jest sumą kwadratów wszystkich elementów wektora wag w. Matematycznie jest to wyrażenie $\sum_i w_i^2$. Regularyzacja L2 znana jest także jako regresja grzbietowa lub regularyzacja Tichonowa [50],

- `train`: trenuje model sieci neuronowej na podanych danych wejściowych,
- `reconstruct`: rekonstruuje dane wejściowe za pomocą modelu autoenkodera,
- `compute_reconstruction_error`: oblicza błąd rekonstrukcji danych wejściowych przez model.

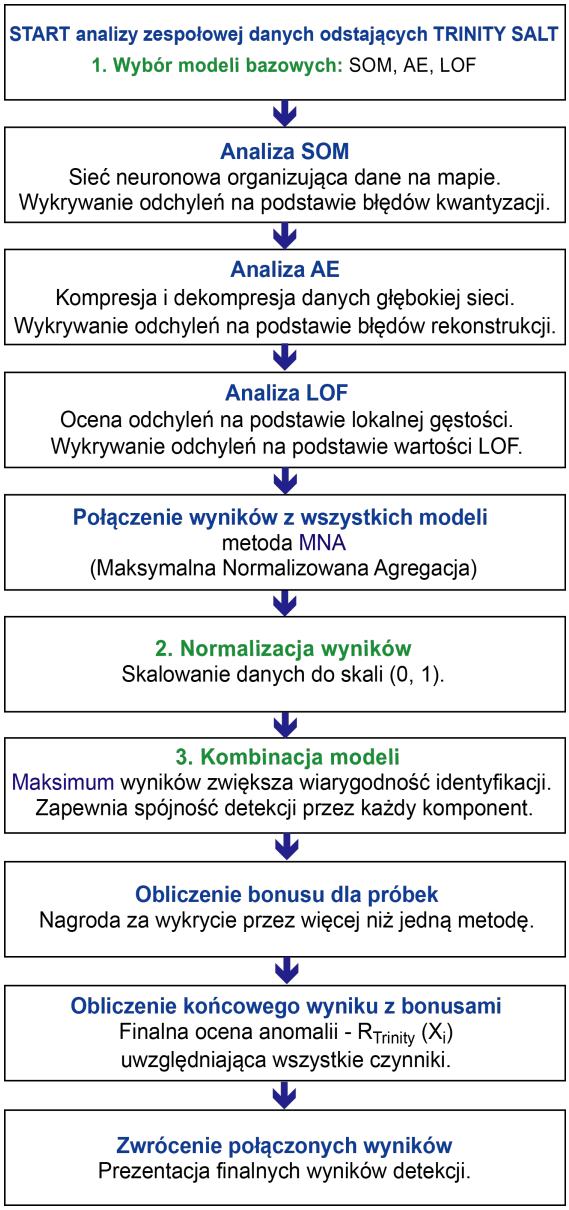
Autoenkodery zostały rozwinięte na przestrzeni lat przez wielu badaczy jako rodzaj sieci neuronowej używanej do nienadzorowanego uczenia. Pierwsze koncepcje autoenkoderów pojawiły się w latach 80. i były dalej rozwijane przez różnych naukowców, takich jak Geoffrey Hinton [370], Yann LeCun [371], Yoshua Bengio [372, 373, 374] i innych. W literaturze naukowej istnieje wiele artykułów na temat autoenkoderów, które omawiają ich architekturę, różne zastosowania i techniki trenowania [50]. Implementacja klasy `AutoencoderOutlier` jest zgodna z ogólnymi zasadami budowy i trenowania autoenkoderów, a jej struktura pozwala na efektywne wykrywanie anomalii poprzez analizę błędów rekonstrukcji.

Jak przedstawiono na rysunku 7.3, zaimplementowane algorytmy SOM, Autoenkoder oraz LOF stanowią kluczowe elementy systemu Trinity SALT (**S**OM-**A**E-**L**OF-**T**riDetect), który łączy ich wyniki w celu skutecznej identyfikacji anomalii. Każdy z tych algorytmów przetwarza dane i zwraca wyniki w postaci listy rankingowej obiektów najbardziej odstających. Autorska technika maksymalnej znormalizowanej agregacji MNA (*ang. maximum normalized aggregation*, MNA) stanowi innowacyjne rozwiązanie, które wyróżnia się spośród tradycyjnych metod dzięki unikalnemu mechanizmowi premiowania konsensusu między modelami. MNA wzmacnia dokładność wykrywania anomalii, premiując obiekty zidentyfikowane przez więcej niż jeden algorytm, co czyni proces bardziej niezawodnym i skutecznym. Każdy z trzech algorytmów przetwarza dane wejściowe i generuje listę rankingową najbardziej odstających obiektów. System Trinity SALT następnie oblicza końcowy wynik dla każdego obiektu $R_{\text{Trinity}}(X_i)$, który jest sumą maksymalnej wartości znormalizowanych wyników z algorytmów SOM, AE i LOF oraz bonusu, zależnego od liczby algorytmów, które zidentyfikowały obiekt jako anomalię:

$$R_{\text{Trinity}}(X_i) = \max\left(R_{\text{SOM}}(X_i), R_{\text{AE}}(X_i), R_{\text{LOF}}(X_i)\right) + B(X_i), \tag{7.1}$$

gdzie:

- $R_{\text{SOM}}(X_i)$, $R_{\text{AE}}(X_i)$, $R_{\text{LOF}}(X_i)$ – znormalizowane wyniki z algorytmów SOM, AE oraz LOF dla obiektu X_i,



Rysunek 7.3: Analiza zespołowa Trinity SALT. Źródło: opracowanie własne.

- $\max\left(R_{\mathrm{SOM}}(X_i), R_{\mathrm{AE}}(X_i), R_{\mathrm{LOF}}(X_i)\right)$ – maksymalna wartość spośród wyników tych trzech algorytmów,
- $B(X_i)$ – wartość bonusu przyznawanego w zależności od liczby algorytmów, które zidentyfikowały obiekt X_i jako anomalię.

Bonus $B(X_i)$ jest funkcją liczby algorytmów l, które wskazały obiekt X_i jako anomalię.

Dla obiektu, który został wykryty przez l algorytmów, wartość bonusu wynosi:

$$B(X_i) = \begin{cases} b_3, & \text{jeśli } l = 3 \\ b_2, & \text{jeśli } l = 2 \\ b_1, & \text{jeśli } l = 1 \\ 0, & \text{jeśli } l = 0 \end{cases} \tag{7.2}$$

,gdzie wartości bonusów to:

$$b_3 = 0.3, \quad b_2 = 0.0002, \quad b_1 = 0.00001.$$

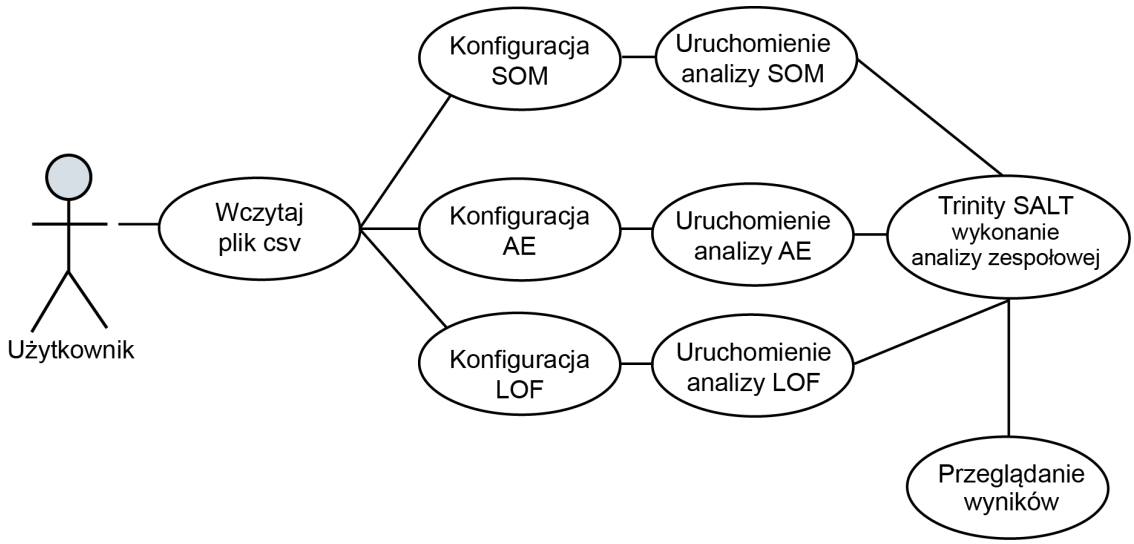
Wyniki każdego z algorytmów są najpierw skalowane do wspólnego przedziału (0, 1), aby zapewnić ich porównywalność. Następnie są łączone za pomocą maksymalnej wartości z tych wyników. Obiekty, które zostały wykryte jako anomalie przez więcej niż jeden algorytm, otrzymują dodatkowy bonus, którego wysokość zależy od liczby modeli wykrywających odstępstwo. W rezultacie końcowy wynik $R_{\text{Trinity}}(X_i)$ jest sumą maksymalnej wartości i bonusu, co pozwala na bardziej precyzyjną identyfikację anomalii.

7.3 Interfejs i funkcjonalność systemu Trinity SALT

Na rysunku 7.4 przedstawiono diagram przypadków użycia, ukazujący funkcje systemu oraz ich wzajemne zależności. Opis diagramu (przypadki użycia) przedstawia się następująco:

- **wczytanie pliku CSV** - użytkownik rozpoczyna proces od wczytania pliku CSV, który będzie poddawany analizie,
- **konfiguracja hiperparametrów** - użytkownik konfiguruje hiperparametry dla trzech algorytmów: SOM, AE i LOF. Każdy z tych algorytmów ma swoje unikalne ustawienia, które można dostosować w zależności od specyfiki danych,
- **uruchomienie analiz**: po konfiguracji użytkownik uruchamia odpowiednie analizy dla każdego z algorytmów: SOM, AE i LOF. Każdy algorytm analizuje dane, generuje wyniki oraz wykresy (macierz pomyłek i inne) detekcji anomalii,
- **analiza zespołowa Trinity SALT** - wyniki z analiz SOM, AE i LOF są następnie łączone w ramach analizy zespołowej Trinity SALT. Metoda ta integruje wyniki trzech algorytmów, skalując je do przedziału (0, 1) i wybierając maksymalną wartość. Obiekty wykryte jako anomalie przez więcej niż jeden algorytm otrzymują dodatkowy bonus, zależny od liczby algorytmów, które zidentyfikowały dany obiekt jako anomalię,

- **przeglądanie wyników** - użytkownik przegląda zintegrowane wyniki w zakładce Trinity SALT. System generuje wykresy i podsumowania, które użytkownik może analizować, oraz prezentuje ostateczne wskaźniki, zapewniając klarowny i wiarygodny obraz analizowanych danych.



Rysunek 7.4: Diagram przypadków użycia. Źródło: opracowanie własne.

Funkcjonalności systemu Trinity SALT

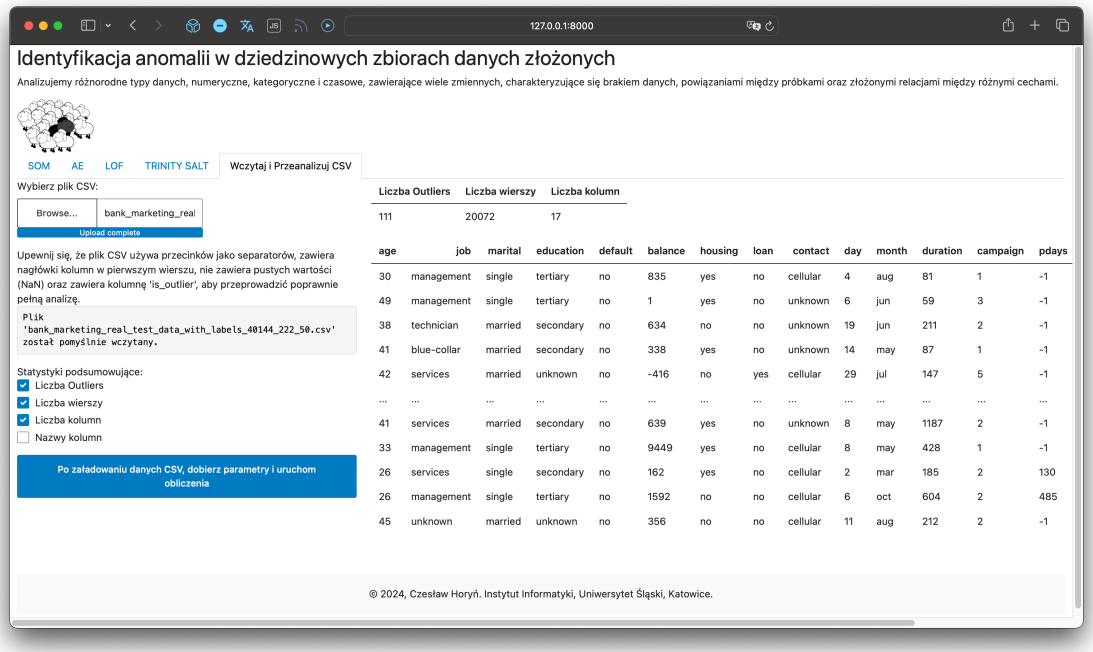
System Trinity SALT to zaawansowana aplikacja webowa stworzona do analizy danych odstających (anomalii) za pomocą trzech bazowych modeli: Self-Organizing Map SOM, autoenkoder AE oraz Local Outlier Factor LOF. Po uruchomieniu aplikacji użytkownik ma dostęp do interfejsu składającego się z kilku głównych zakładek, jak pokazano na rysunku 7.2, oferujących różnorodne funkcje analityczne i konfiguracyjne:

Zakładka Wczytaj i Przeanalizuj CSV

Rysunek 7.5 przedstawia zakładkę, która umożliwia użytkownikom wczytywanie plików CSV zawierających dane do analizy i generuje szczegółowe statystyki na temat wczytanych danych.

Zakładka SOM

W zakładce dedykowanej algorytmowi SOM można dostosować hiperparametry sieci Kohonena, takie jak liczba neuronów (rozmiar sieci w osiach X i Y), promień sąsiedztwa, współczynnik uczenia, funkcję zmniejszającą współczynnik uczenia oraz topologię sieci. Dodatkowo, możliwe jest wybranie metryki odległości i funkcji sąsiedztwa, ustawienie ziarna losowości, liczby iteracji treningowych oraz opcji usuwania wysoko skorelowanych

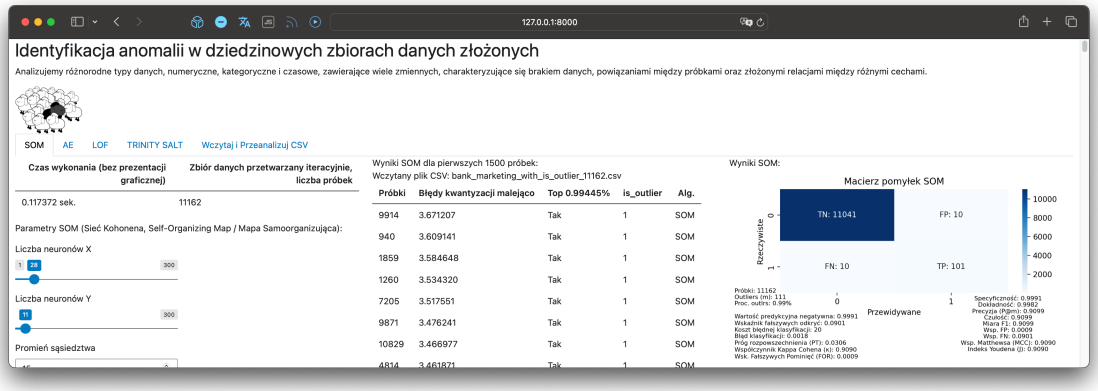


Rysunek 7.5: Zakładka Wczytaj i Przeanalizuj CSV. Źródło: opracowanie własne.

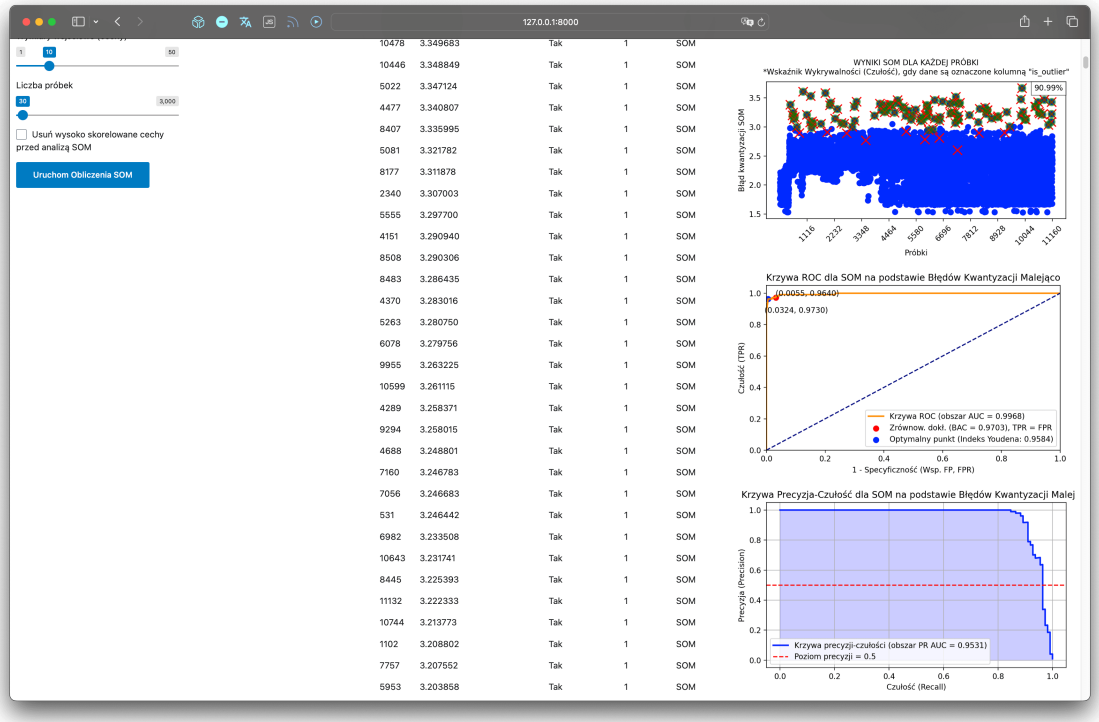
cech przed analizą SOM. Wyniki SOM są prezentowane w formie tabel i wykresów, takich jak macierz pomyłek, histogram, posortowane obiekty odstające, krzywa ROC, krzywa precyzji-czułości oraz macierz U. Na rysunku 7.6 przedstawione są dwa widoki 7.6a oraz 7.6b przedstawiające wyniki analizy przy użyciu metody SOM dla przykładowych danych. Ponieważ cała zawartość nie mieści się na jednym ekranie, przedstawiono ją w dwóch częściach. Dodatkowo, zakładka SOM umożliwia wizualizację macierzy U, co pokazano na rysunku 7.7. Macierz U przedstawia odległości między sąsiednimi neuronami w sieci Kohonena i jest użyteczna do identyfikacji struktur grupowania w danych.

Zakładka AE

W zakładce dedykowanej algorytmowi AE, pokazanej na rysunku 7.8, możliwa jest konfiguracja hiperparametrów autoenkodera, takich jak liczba epok, współczynnik uczenia, funkcje aktywacji dla wejścia i wyjścia, funkcja straty, optymalizator oraz regularyzacja. Możliwe jest również dostosowanie rozmiaru partii (batch) oraz wybór spośród różnych konfiguracji warstw autoenkodera. Dodatkowo dostępna jest opcja usuwania wysoko skorelowanych cech przed analizą autoenkodera. Wyniki AE są prezentowane w formie tabel i wykresów, umożliwiając szczegółową analizę błędów rekonstrukcji, kluczową dla wykrywania odchyleń. Podobnie jak dla SOM, wyniki pokazują macierz pomyłek, histogram, posortowane obiekty odstające, krzywą ROC oraz krzywą precyzji-czułości.



(a) Widok Początkowy

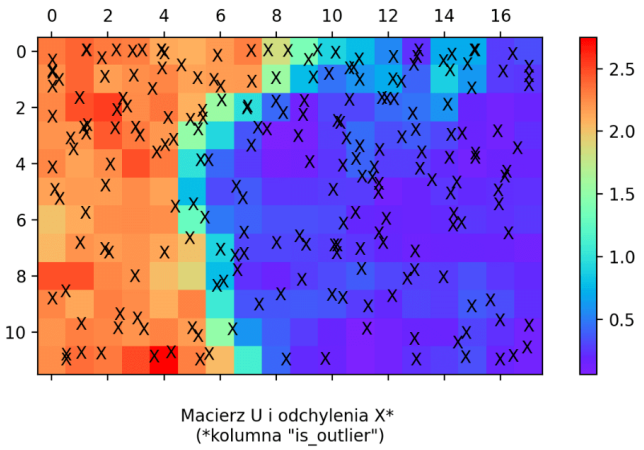


(b) Widok Końcowy

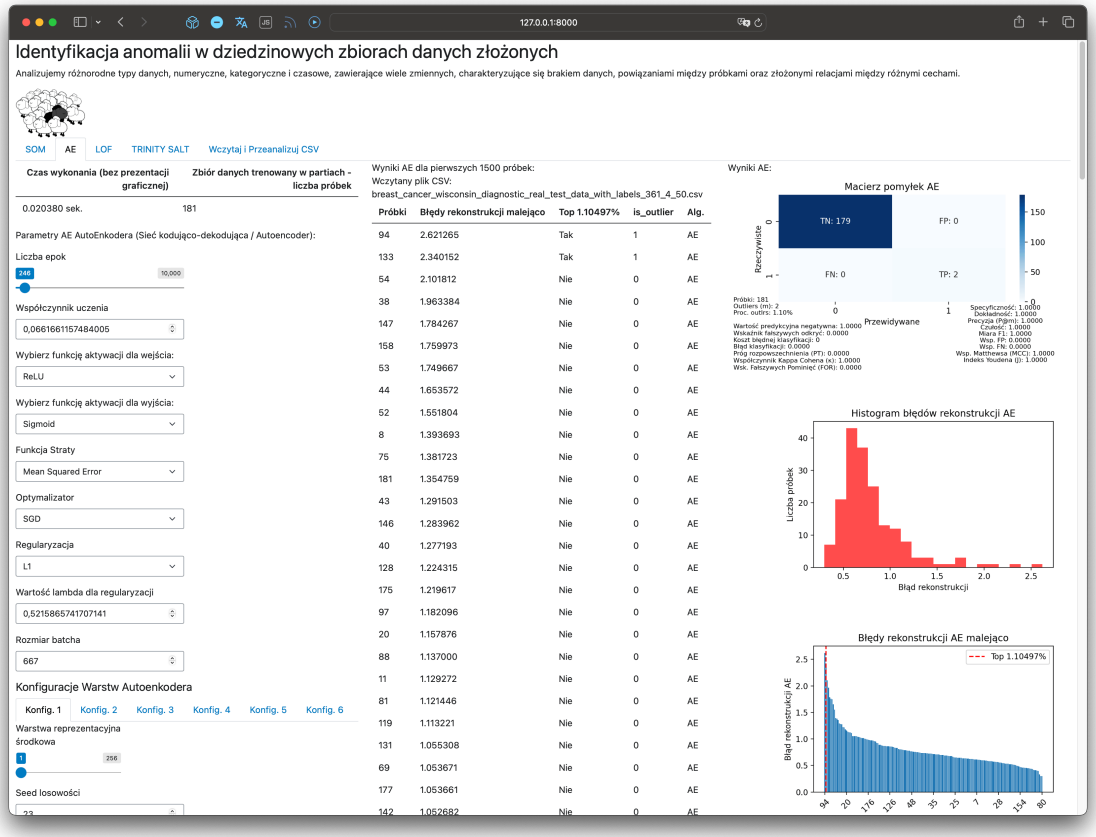
Rysunek 7.6: Zakładka SOM - Widok Początkowy i Końcowy. Źr.: opracowanie własne.

Zakładka LOF

W zakładce dedykowanej algorytmowi LOF, pokazanej na rysunku 7.9, możliwa jest konfiguracja hiperparametrów związanych z lokalnym współczynnikiem osobliwości.

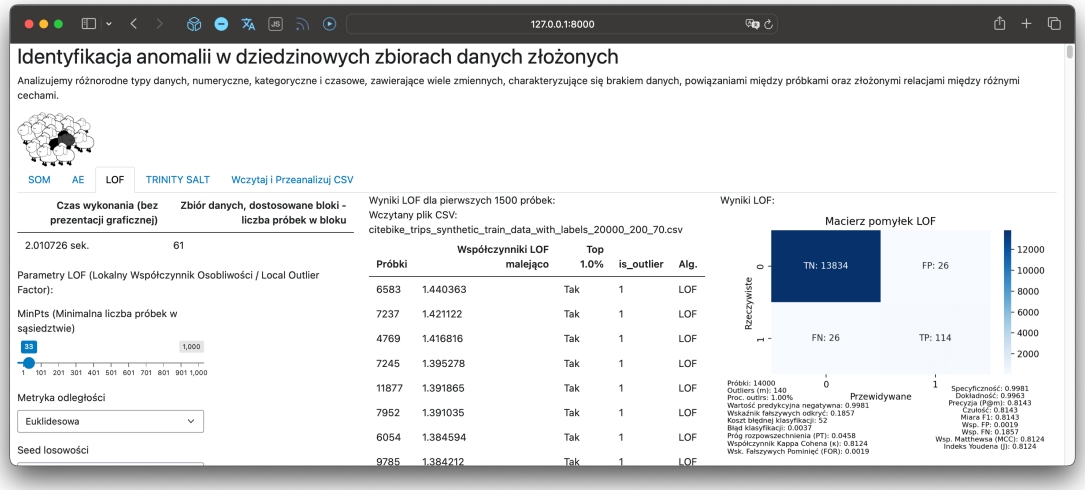


Rysunek 7.7: Zakładka SOM - Widok Końcowy, ciąg dalszy. Źr.: opracowanie własne.



Rysunek 7.8: Zakładka AE - Widok Początkowy. Źródło: opracowanie własne.

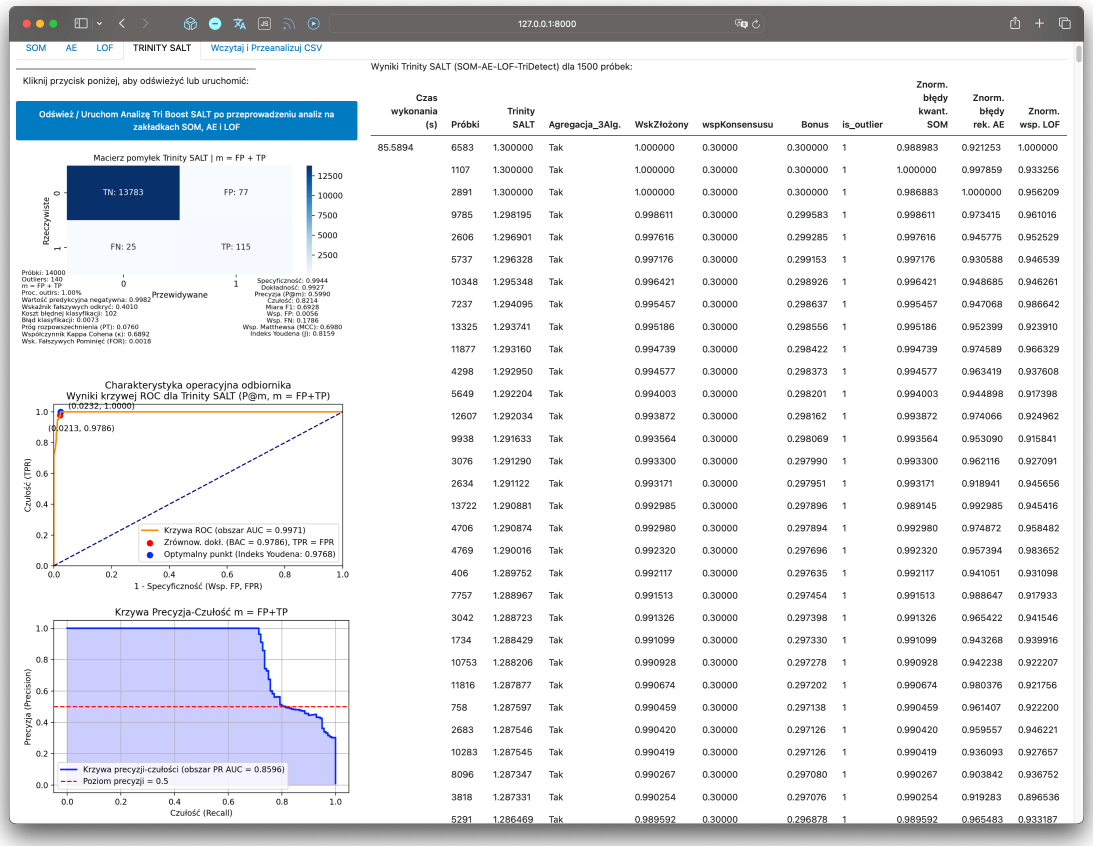
Można ustawić minimalną liczbę próbek w sąsiedztwie (MinPts) oraz wybrać metrykę odległości. Dodatkowo, możliwe jest ręczne ustawienie rozmiaru bloku. Dostępna jest także opcja usuwania wysoko skorelowanych cech przed analizą LOF. Wyniki LOF są prezentowane w formie tabel i wykresów, podobnie jak w przypadku SOM i AE.



Rysunek 7.9: Zakładka LOF - Widok Początkowy. Źródło: opracowanie własne.

Zakładka Trinity SALT

Najważniejszym elementem systemu jest zakładka dedykowana analizie zespołowej Trinity SALT, gdzie przeprowadzana jest analiza danych odstających. Proces rozpoczyna się od wyboru modeli bazowych: SOM, AE i LOF. Każdy z modeli wykonuje swoją analizę, a wyniki są następnie łączone za pomocą metody maksymalnej normalizowanej agregacji (MNA), uwzględniając premie za konsensus modeli. Ostateczna ocena uwzględnia wszystkie czynniki i prezentuje połączone wyniki, które są przedstawiane użytkownikowi w formie tabeli i wykresów, co pokazano na rysunku 7.10. System Trinity SALT wyposażony jest w szczegółowe wyjaśnienia dotyczące wskaźników używanych do oceny wydajności modelu detekcji anomalii, takich jak specyficzność, dokładność, precyzja, czułość, miara F1, współczynnik fałszywie pozytywnych i negatywnych, wartość predykcyjna negatywna, wskaźnik fałszywych odkryć, koszt błędnej klasyfikacji, błąd klasyfikacji, próg rozpowszechnienia, indeks Youdena, współczynnik korelacji Matthewsa, wskaźnik fałszywych pominięć, współczynnik Kappa Cohena, krzywa ROC, obszar AUC, krzywa precyzji-czułości oraz zrównoważona dokładność. Jak zademonstrowano, Trinity SALT oferuje kompleksową analizę danych, wykorzystując różne techniki i metryki do oceny wydajności modeli detekcji anomalii. Interfejs jest intuicyjny i łatwy w użyciu, a bogata funkcjonalność umożliwia przeprowadzenie szczegółowych analiz, co czyni go wszechstronnym narzędziem analitycznym.



Rysunek 7.10: Zakładka Trinity SALT - Widok Początkowy. Źr.: opracowanie własne.

Podsumowując, najważniejsze komponenty systemu to:

- **moduł wczytywania danych** - umożliwia użytkownikom wczytywanie plików CSV zawierających dane do analizy,
- **moduł przetwarzania danych** - realizuje funkcje przekształcania i normalizacji danych, dostosowując je do wymagań poszczególnych algorytmów,
- **moduł analizy anomalii** - zawiera implementację algorytmów SOM, AE oraz LOF, a także metody zespołowe do ich kombinacji,
- **moduł wizualizacji** - odpowiada za generowanie wykresów, takich jak krzywe ROC, krzywe precyzji-czułości oraz macierze pomyłek, które pomagają w interpretacji wyników.

7.4 Instalacja, uruchamianie i wymagania sprzętowe

Poniżej przedstawiono sposób instalacji i uruchamiania aplikacji oraz minimalne wymagania sprzętowe. Proces instalacji systemu jest prosty i intuicyjny. Instalacja aplikacji polega na skopiowaniu folderu z plikiem `app.py` (znajdującego się na dołączonej do pracy płycie DVD) oraz uruchomieniu tego pliku. W tym celu należy wykonać kilka kroków w wierszu poleceń:

```
# Instalacja biblioteki Shiny for Python
pip install shiny

# Instalacja innych wymaganych bibliotek
pip install numpy pandas matplotlib
```

Na początku pliku `app.py`, który został zamieszczony na płycie DVD znajdują się importy dla wszystkich niezbędnych bibliotek, które należy zainstalować za pomocą powyższych poleceń. Poniżej przedstawiono przykładowe importy:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from shiny import App, ui, reactive, render
```

Dodatkowe informacje dotyczące instalacji i uruchamiania aplikacji Shiny dla Python można znaleźć w dokumentacji dostępnej na stronie: [Shiny for Python Documentation][2]. Przed przystąpieniem do instalacji należy upewnić się, że zainstalowano odpowiednią wersję Python. Python można pobrać ze strony: [Python.org][3].

1. Otworzyć terminal (na Windowsie może to być PowerShell, CMD lub terminal w środowisku takim jak Anaconda; na macOS i Linuxie należy użyć Terminala),

2. Przejść do katalogu, w którym znajduje się plik `app.py`, używając komendy `cd`:

   ```
   cd /ścieżka/do/pliku/app.py
   ```

3. Uruchomić plik `app.py` za pomocą Pythona:

   ```
   python app.py
   ```

[2] https://shiny.posit.co/py/docs/deploy-on-prem.html
[3] https://www.python.org/downloads/

Po wykonaniu tych kroków aplikacja powinna być dostępna w przeglądarce pod adresem `http://127.0.0.1:8000`. Tam będzie można wybrać metodę detekcji anomalii, wczytać badany plik CSV i wygenerować wyniki. Podczas uruchamiania aplikacji, jeśli zapomnimy o zainstalowaniu którejś z bibliotek, w wierszu poleceń pojawi się komunikat informujący o brakującej bibliotece. Przykład komunikatu:

```
ModuleNotFoundError: No module named 'numpy'
```

W takim przypadku należy doinstalować brakującą bibliotekę za pomocą komendy `pip`. Na przykład, jeśli brakuje biblioteki `numpy`, należy użyć poniższego polecenia:

```
pip install numpy
```

Analogicznie, należy postępować w przypadku innych brakujących bibliotek, zastępując `numpy` nazwą brakującej biblioteki. Dołączona do rozprawy płyta DVD zawiera cyfrową wersję pracy, plik aplikacji `app.py` oraz dodatkowe zasoby, takie jak zbiory danych używane w badaniach. Struktura katalogów została szczegółowo opisana w pliku struktura.pdf, umieszczonym w głównym folderze płyty.

Minimalne wymagania sprzętowe

Aplikacja webowa działa poprawnie na standardowych komputerach osobistych. Niemniej jednak, aby zapewnić optymalną wydajność, zalecane jest spełnienie poniższych wymagań sprzętowych. Dla małych plików CSV (do około 30 tysięcy obiektów i 20 atrybutów), zaleca się konfigurację pokazaną w tabeli 7.1. Należy pamiętać, że wraz ze wzrostem

Tabela 7.1: Minimalne wymagania sprzętowe. Źródło: opracowanie własne.

Komponent	**Minimalne wymagania i zalecenia dodatkowe**
Procesor	co najmniej Intel Core i5 lub równoważny, preferowane procesory wielordzeniowe (co najmniej 4 rdzenie); dla lepszej wydajności zaleca się procesory z większą liczbą rdzeni
Pamięć RAM	minimum 8 GB (dla bardzo dużych plików CSV, zawierających na przykład 500 tysięcy obiektów lub więcej i kilkadziesiąt zmiennych, zaleca się co najmniej 64 GB RAM lub więcej)
Dysk twardy	co najmniej 50 GB wolnej przestrzeni
System operacyjny	Windows 10, macOS 10.15 lub nowszy, Linux z jądrem 4.15 lub nowszym
Przeglądarka internetowa	najnowsza wersja Google Chrome, Mozilla Firefox, Safari lub Microsoft Edge

rozmiaru plików CSV wymagania sprzętowe mogą również wzrosnąć. Dla większych plików danych (powyżej 500 tysięcy obiektów i kilkudziesięciu zmiennych opisujących każdy obiekt), zaleca się zwiększenie pamięci RAM oraz wykorzystanie procesorów z większą liczbą rdzeni, aby zapewnić płynne działanie. Badania przeprowadzono na

Tabela 7.2: Konfiguracja sprzętowa użyta do przeprowadzenia badań oraz dodatkowa konfiguracja użyta do weryfikacji aplikacji. Źródło: opracowanie własne.

Komponent	**MacBook Pro**
Procesor	Apple M3 Pro z 11-rdzeniowym CPU, 14-rdzeniowym GPU i 16--rdzeniowym systemem Neural Engine
System operacyjny	macOS Sonoma 14.5
Pamięć RAM	36 GB zunifikowanej pamięci RAM
Dysk twardy	512 GB pamięci masowej SSD
Komponent	**Dell Latitude 5540**
Procesor	Intel® Core™ i7-1370P vPro trzynastej generacji (24 MB pamięci podręcznej, 14 rdzeni, do 5,2 GHz)
System operacyjny	Windows 11 Pro, wersja 64-bitowa
Pamięć RAM	64 GB, 2 x 32 GB, DDR4 3200 MHz, pamięć dwukanałowa
Dysk twardy	M.2 2230, 1 TB, PCIe NVMe x4 czwartej generacji, dysk SSD, Class 35

MacBooku Pro z konfiguracją sprzętową pokazaną w tabeli 7.2. Aplikacja została również zweryfikowana na komputerze Dell Latitude 5540 z systemem Windows Pro, gdzie działała bez zarzutu. Dodatkowa konfiguracja również znajduje się w tabeli 7.2.

7.5 Wymagania i struktura plików CSV dla systemu

Struktura przykładowego pliku CSV, rozpoznawanego przez system Trinity SALT, prezentuje się następująco:

```
outlook,temperature,humidity,windy,play,is_outlier
cloudy,75,80,TRUE,no,0
sunny,78,88,FALSE,yes,0
rainy,82,92,TRUE,no,1
cloudy,69,85,FALSE,yes,0
sunny,65,78,TRUE,yes,1
rainy,72,84,FALSE,no,0
cloudy,66,68,TRUE,yes,0
sunny,73,91,FALSE,no,0
cloudy,70,75,FALSE,yes,0
rainy,76,89,FALSE,yes,0
sunny,77,80,TRUE,yes,0
cloudy,68,86,TRUE,yes,0
sunny,80,83,FALSE,yes,0
rainy,67,90,TRUE,no,0
cloudy,26,68,FALSE,yes,1
```

Aby przeprowadzić analizę Trinity SALT, niezbędne jest wczytanie pliku CSV, który spełnia następujące wymagania:

- plik CSV musi używać przecinków jako separatorów,
- pierwszy wiersz pliku CSV musi zawierać nagłówki kolumn,
- plik nie może zawierać pustych wartości (NaN),
- plik musi zawierać kolumnę `is_outlier` z wartościami 0 lub 1.

Uruchomienie analizy bez wczytania odpowiedniego pliku lub bez wykonania wstępnych analiz SOM, AE i LOF na danych z pliku CSV będzie skutkować użyciem losowych danych. Podejście to ma na celu jedynie zilustrowanie potencjalnych wyników. Jeśli plik CSV nie spełnia powyższych wymagań, system postępuje w następujący sposób:

- **brak kolumny `is_outlier`** - system wykona niepełną analizę i wyświetli komunikat ostrzegawczy, informując użytkownika o konieczności dodania tej kolumny do pliku CSV,
- **Brakujące wartości (NaN) lub puste wartości** - System automatycznie zidentyfikuje brakujące lub puste wartości w danych. W takim przypadku system wyświetli komunikat ostrzegawczy i usunie wiersze zawierające puste wartości, co w skrajnych przypadkach może skutkować brakiem statystyk podsumowujących,
- **nieprawidłowy separator lub brak nagłówków** - system wyświetli komunikat o błędzie, wskazując na problem z separatorem lub brakującymi nagłówkami kolumn. Użytkownik zostanie poproszony o poprawienie formatu pliku CSV.

7.6 Podsumowanie

Rozdział przedstawia motywacje stojące za stworzeniem autorskiego systemu w postaci aplikacji webowej do identyfikacji anomalii w złożonych danych, nazwanego Trinity SALT, oraz jego projekt i szczegóły implementacyjne. Opisuje wszystkie funkcjonalności programu, umożliwiając użytkownikowi końcowemu pełne zrozumienie jego obsługi. W rozdziale wyszczególniono również wymagania sprzętowe aplikacji oraz sposób jej instalacji.

Rozdział 8

Eksperymenty obliczeniowe

W ostatnich latach uczenie maszynowe zyskało ogromne znaczenie w wielu branżach, co jest wynikiem dynamicznego rozwoju algorytmów i spadających cen sprzętu, umożliwiając automatyzację wielu zadań, które wcześniej były niemożliwe do zrealizowania. Dzięki temu ludzie mogą teraz skupiać się na bardziej twórczych i innowacyjnych działaniach, a problemy, które jeszcze dekadę temu były nie do pokonania, są rozwiązywane przez standardowe komputery osobiste. Pomimo tego postępu, rosnące zainteresowanie analizą dużych zbiorów danych generuje coraz większe zapotrzebowanie na wydajne i skuteczne metody, zwłaszcza w zakresie wykrywania anomalii.

W odpowiedzi na te potrzeby, przeprowadzone w ramach niniejszej rozprawy eksperymenty dotyczą zaimplementowanego zespołu detektorów Trinity SALT (SOM-AE--LOF-TriDetect), który integruje heterogeniczny zespół algorytmów: sieć samouczącą się SOM, autoenkoder AE oraz algorytm gęstościowy LOF. W uczeniu maszynowym zespoły klasyfikatorów [375] są stosowane, aby zwiększyć wydajność, stabilność i moc predykcyjną pojedynczych modeli. Zespoły te są szczególnie skuteczne w redukcji nadmiernego dopasowania, zwiększaniu odporności na szumy i poprawie zdolności generalizacji modeli [376, 375, 377]. Algorytm LOF, ceniony za zdolność do wykrywania lokalnych anomalii, napotyka jednak na problemy wydajnościowe wraz ze wzrostem wielkości przetwarzanych zbiorów danych, zwłaszcza gdy dane zawierają zarówno cechy numeryczne, jak i kategoryczne. Obecne metody wykrywania lokalnych odchyleń wymagają przeszukiwania wszystkich obiektów w zbiorze danych w celu obliczenia lokalnego współczynnika osobliwości, co jest bardzo czasochłonnym procesem. Może to prowadzić do wydłużonych czasów przetwarzania, szczególnie gdy liczba obiektów jest duża, a złożoność obliczeniowa osiąga poziom $O(n^2)$. Analiza skuteczności i efektywności algorytmu LOF w wykrywaniu anomalii w zbiorach danych z różnych dziedzin, zawierających zarówno dane kategoryczne, jak i mieszane, jest ważnym aspektem przedstawianych badań. Aby sprostać tym wyzwaniom, podjęto próbę opracowania metody optymalizacji algorytmu LOF. To podejście

zostało również wdrożone w zespole Trinity SALT. Skuteczność odnosi się do zdolności metody do wykrywania anomalii, natomiast efektywność dotyczy zarówno czasu wymaganego na przeprowadzenie procesu, jak i zasobów pamięci RAM, oceniając szybkość działania metody oraz jej ogólną wydajność w przetwarzaniu danych. Istnieją dwie główne metody poprawy wydajności algorytmu LOF. Pierwsza polega na uproszczeniu obliczeń lokalnego współczynnika osobliwości [378, 103, 379, 380, 381, 300, 382, 383], co może jednak wpływać na skuteczność algorytmów oraz w istotny sposób zmienia sposób działania oryginalnego algorytmu, czego starano się uniknąć. Druga wykorzystuje struktury danych, takie jak R-drzewo [384], KD-drzewo [385], drzewo pokrywające (*ang. cover tree*) [386] itp., do bardziej efektywnego wyszukiwania najbliższych sąsiadów [387]. Te struktury mają również swoje wady: R-drzewa i KD-drzewa mogą doświadczać spadku wydajności przy danych o wysokiej liczbie wymiarów, a drzewa pokrywające są skomplikowane w implementacji i wymagają znacznych zasobów pamięci przy bardzo dużych zbiorach danych. Dodatkowo, drzewa pokrywające charakteryzują się wysokimi kosztami konstrukcji oraz zależnością wydajności od współczynnika ekspansji, co oznacza, że ich wydajność spada, gdy liczba obiektów rośnie w miarę zwiększania się promienia wokół dowolnego obiektu. Podobnie jak R-drzewa i KD-drzewa, mogą być mniej efektywne dla danych o dużym wymiarze. Ponadto, R-drzewa mogą mieć zmienną wydajność wyszukiwania, która w najgorszych scenariuszach może być bardzo niska, a proces dzielenia węzłów może być kosztowny. Natomiast KD-drzewa mogą wymagać kosztownej optymalizacji, aby zapewnić równomierny podział danych i utrzymać efektywność operacji wyszukiwania.

Ze względu na te ograniczenia, opracowano nowe podejście, które utrzyma wysoką skuteczność wykrywania anomalii, przyspieszy proces i zachowa istotne cechy oryginalnego algorytmu. Jako rozwiązanie problemu zaproponowano autorską metodę optymalizacji rozmiaru bloku, mającą na celu skrócenie czasu obliczeń przy zachowaniu wysokiej skuteczności w wykrywaniu anomalii. W pierwszym etapie badania i eksperymenty skoncentrowały się na proponowanej metodzie optymalizacji rozmiaru bloku, a szczegółowa metodologia oraz uzyskane wyniki zostały przedstawione w następnym podrozdziale 8.1.

W ramach badań i eksperymentów powstał artykuł „*Improving Detection Efficiency: Optimizing Block Size in the Local Outlier Factor (LOF) Algorithm*" opublikowany w prestiżowej serii Lecture Notes in Computer Science (LNCS)[1] [305]. Artykuł zwrócił uwagę innego czasopisma Applied Soft Computing[2], które wyraziło zainteresowanie rozszerzoną wersją artykułu zawierającą dodatkowe dowody eksperymentalne, udoskonalenie metody oraz pogłębienie tematu. Po pozytywnych trzech recenzjach i wskazaniu drobnych kwestii redakcyjnych do poprawy, oczekiwana jest publikacja rozszerzonych wyników badań również w tym renomowanym czasopiśmie (CiteScore 15,8; IF 7,2) do końca 2024 roku lub początkowych miesiącach 2025 roku.

[1] https://www.springer.com/series/0558

[2] https://www.sciencedirect.com/journal/applied-soft-computing

W kolejnym etapie, szczegółowo omówionym w dalszej części w podrozdziale 8.2, badania i eksperymenty koncentrują się na uczeniu zespołowym i systemie Trinity SALT. System ten jest heterogenicznym zespołem detektorów anomalii, składającym się z trzech różnych algorytmów bazowych: sieci samouczącej się SOM, autoenkodera AE oraz algorytmu gęstościowego LOF. Szczegółowy opis tych algorytmów znajduje się w podrozdziale 4.5. Analiza zespołowa polega na połączeniu wyników z różnych modeli, co prowadzi do lepszej wydajności, stabilności oraz odporności modelu końcowego w porównaniu do zastosowania pojedynczego rozwiązania. Przykładem skuteczności takich podejść jest zwycięski model w konkursie Netflix Prize, który zawierał setki różnych modeli filtrowania współpracującego [388]. Metody analizy zespołowej regularnie wygrywają konkursy z zakresu analizy danych w tak różnych dziedzinach jak grupowanie, klasyfikacja czy systemy rekomendacyjne. Jak wskazują uznani autorzy Aggarwal i Sathe, zajmujący się tematyką wykrywania anomalii od wielu lat, metody analizy zespołowej były dotychczas głównie stosowane w grupowaniu i klasyfikacji. Zastosowanie tych metod w wykrywaniu anomalii to stosunkowo nowa dziedzina, której badanie może przynieść znaczące korzyści [236]. Szczegółowe uzasadnienie tego stwierdzenia znajduje się w rozdziale 5, który w całości poświęcono metodom zespołowym. Dokładnie w podrozdziale 5.1 temat ten został szczegółowo rozwinięty.

W uczeniu zespołowym podstawowym założeniem jest podział algorytmów na modele silne (*ang. strong learners*) i słabe (*ang. weak learners*). Modele silne, takie jak drzewa decyzyjne, maszyny wektorów nośnych czy sieci neuronowe, mogą osiągnąć wysoką dokładność poprzez minimalizację zarówno błędu systematycznego, jak i wariancji. Modele słabe, takie jak k-najbliżsi sąsiedzi, regresja liniowa czy proste klasyfikatory bayesowskie, osiągają nieco lepszą dokładność od losowego wyboru, ale są prostsze i szybkie w uczeniu. Chociaż ta definicja jest teoretyczna, ponieważ obecne modele są znacznie skuteczniejsze niż losowy wybór, wprowadzenie wielu słabych modeli, które są w rzeczywistości całkiem silne, pozwala na przypisanie im ograniczonych podprzestrzeni. Dzięki temu osiągają wysoką dokładność lokalną przy małej wariancji. Wynik końcowy w zespole tych modeli uzyskuje się przez uśrednienie prognoz lub głosowanie większościowe. Używanie umiarkowanie silnych modeli jako grupy pomaga zwiększyć ogólną dokładność i zmniejszyć wariancję. Silne klasyfikatory mogą ulec przetrenowaniu, co utrudnia utrzymanie wysokiej dokładności dla całej podprzestrzeni obiektów. Aby temu zapobiec, konieczne jest znalezienie odpowiedniego kompromisu, co prowadzi do uproszczenia hiperpłaszczyzny decyzyjnej. W procesie uczenia zespołowego każdy model oddaje „głos” na swoją przewidywaną klasyfikację lub prognozę. Skuteczny proces uczenia zapewnia, że większość modeli w zespole podejmuje trafne decyzje, co sprawia, że decyzja większości jest najbardziej dokładna.

Jedną ze strategii uczenia zespołowego jest agregacja (*ang. bagging*). Technika ta polega na tworzeniu wielu modeli na różnych zestawach danych pochodzących z oryginalnego zbioru. Na każdym z tych zestawów trenuje się często słabe modele. Po przeszkoleniu

wszystkich modeli, ich wyniki są łączone. Można to zrobić na dwa sposoby: przez uśrednianie wyników lub poprzez głosowanie większościowe. Innym podejściem w uczeniu zespołowym jest wzmacnianie (*ang. boosting*), w którym zespół modeli jest budowany stopniowo, począwszy od jednego słabego modelu, dodając kolejne modele jeden po drugim. Metoda systematycznie dodaje nowe modele do zespołu, koncentrując się na trudnych obiektach, aby poprawić dokładność, ale wymaga ostrożności, aby uniknąć przetrenowania. Na początku trenowany jest jeden model na całym zestawie danych. Po przeszkoleniu pierwszego modelu sprawdzane są obiekty sklasyfikowane nieprawidłowo. Następnie tworzony jest drugi model, w którym kładzie się większy nacisk (większe wagi) na obiekty błędnie sklasyfikowane przez pierwszy model. W ten sposób drugi model uczy się trudniejszych przypadków. Proces powtarza się dla kolejnych modeli, gdzie każdy nowy model stara się poprawić błędy poprzednich modeli. To iteracyjne podejście pozwala zespołowi na osiągnięcie coraz lepszej dokładności, ponieważ każdy nowy model uczy się na poprzednich błędach. Jednakże, ze względu na zwiększone ryzyko przetrenowania, konieczne jest stosowanie prostszych modeli lub wprowadzenie technik regularyzacji, aby zrównoważyć dokładność i stabilność zespołu.

W tej rozprawie eksperymenty zademonstrowane w rozdziale 8.2 przeprowadzono na systemie Trinity SALT, który wykorzystuje inną strategię niż wyżej wymienione, mianowicie technikę kontaminacji (*ang. stacking*). Jest to kolejny prostszy, lecz równie skuteczny sposób tworzenia zespołu polegający na zastosowaniu ograniczonej liczby silnych modeli, których specyficzne cechy pozwalają na osiągnięcie lepszej skuteczności w określonych obszarach przestrzeni danych. Ta strategia jest szczególnie skuteczna, gdy zestaw danych ma złożoną strukturę, którą można lepiej zarządzać za pomocą różnych typów algorytmów. Technika ta polega na trenowaniu różnych modeli na tym samym zestawie danych i łączeniu ich wyników poprzez meta-klasyfikator, który w przypadku Trinity SALT polega na normalizacji wyników, a następnie wyborze maksimum spośród znormalizowanych wyników każdego modelu, dodatkowo przyznawany jest bonus za konsensus z modeli. Eksperymenty przeprowadzone w ramach niniejszej rozprawy mają na celu ukazanie efektywności i skuteczności metody optymalizacji rozmiaru bloku w LOF oraz systemu Trinity SALT, który również wykorzystuje optymalizację rozmiaru bloku dla efektywnej analizy. Celem jest także potwierdzenie zasadności zastosowania tych technik. W kolejnych podrozdziałach zostaną szczegółowo przedstawione eksperymenty i ich wyniki, a także omówione zbiory danych, na których te eksperymenty przeprowadzono.

8.1 Optymalizacja rozmiaru bloku w LOF

Skuteczność mierzy zdolność metody do wykrywania anomalii, podczas gdy wydajność odnosi się do czasu potrzebnego na przeprowadzenie tego procesu, oceniając szybkość działania, efektywność przetwarzania danych oraz wykorzystanie zasobów sprzętowych, takich jak pamięć RAM. Algorytm LOF często napotyka problem długiego czasu obliczeń,

zwłaszcza przy dużych zbiorach danych z cechami numerycznymi i kategorycznymi. Obecna metoda wymaga przeszukiwania wszystkich obiektów, co jest czasochłonne i prowadzi do wydłużenia przetwarzania, szczególnie gdy liczba obiektów jest duża, a złożoność obliczeniowa osiąga poziom $O(n^2)$. Aby uniknąć tych problemów, przeprowadzono eksperymenty z nową metodą, która obiecuje poprawić wydajność algorytmu, jednocześnie zachowując jego skuteczność.

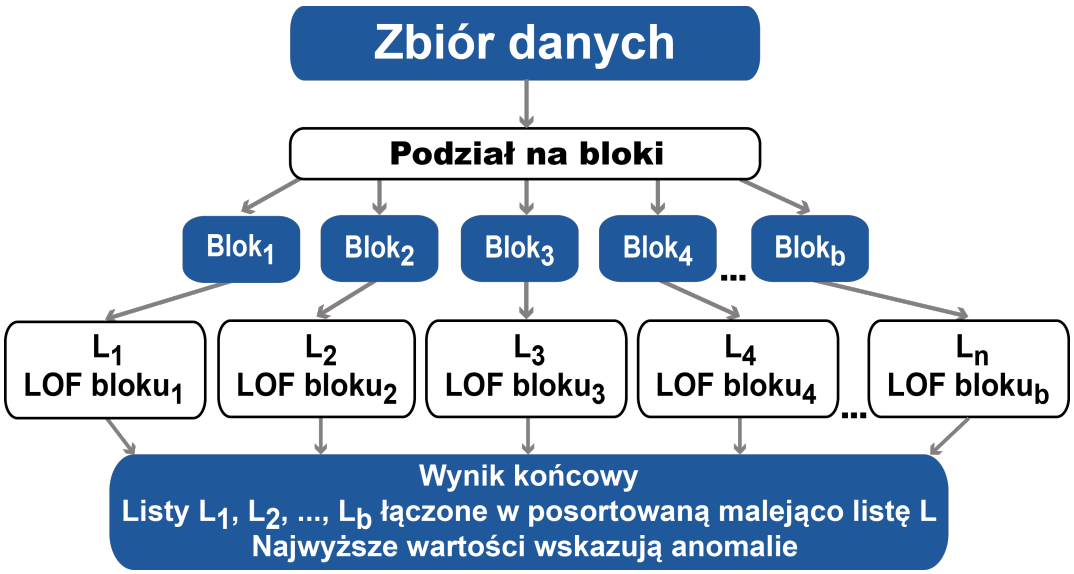
8.1.1 Eksperyment 1: empiryczna optymalizacja rozmiaru bloku

Wprowadzono innowacyjne podejście, które optymalizuje rozmiar bloku w algorytmie LOF, aby przyspieszyć wykrywanie anomalii przy zachowaniu wysokiej skuteczności. Założono, że rozmiar bloku można dostosować empirycznie poprzez eksperymenty, co pozwala na szybsze i bardziej efektywne przetwarzanie danych. Optymalny rozmiar zależy od konkretnego zestawu danych. Standardowy algorytm LOF wymaga intensywnych obliczeniowo kalkulacji odległości między obiektami, co jest szczególnie uciążliwe w przypadku dużych zbiorów danych. Optymalizacja rozmiaru bloku koncentruje te obliczenia na obiektach w obrębie tego samego bloku, minimalizując nakłady obliczeniowe. Każdy zestaw danych ma unikalną strukturę i rozkład obiektów, a dostosowanie rozmiaru bloku uwzględnia lokalne struktury danych. Algorytm LOF identyfikuje anomalie w lokalnych grupach danych, a optymalizacja rozmiaru bloku umożliwia bardziej precyzyjną analizę tych struktur. Poniższe matematyczne wyrażenie procesu optymalizacji rozmiaru bloków (8.1) opisuje sposób minimalizacji sumy odległości między obiektami wewnątrz bloków. Wskaźnikowa funkcja $I_{ik}(p)$ określa, czy dwa obiekty X_i i X_k należą do tego samego bloku, a funkcja $\text{dist}(X_i, X_k)$ mierzy odległość między obiektami X_i i X_k. Minimalizując tę sumę, można osiągnąć bardziej efektywne grupowanie obiektów. Celem jest dobranie rozmiaru bloku (p) w taki sposób, aby suma odległości między obiektami w tym samym bloku była minimalna. Teoretycznie zakłada się, że optymalny rozmiar bloku to taki, który minimalizuje tę sumę. W praktyce, w tym eksperymencie, rozmiar bloku dobierany jest doświadczalnie tak, aby uzyskać największy stopień pokrycia rzeczywistych, etykietowanych anomalii, mając na uwadze, że minimalizacja tej sumy jest założeniem teoretycznym:

$$p = \arg\min_{p} \left(\sum_{i=1}^{n} \sum_{k=1}^{n} I_{ik}(p) \cdot \text{dist}(X_i, X_k) \right), \tag{8.1}$$

gdzie:

- p - liczba obiektów na blok,
- $I_{ik}(p)$ - funkcja wskaźnikowa, która wynosi 1, jeśli obiekty X_i i X_k należą do tego samego bloku o rozmiarze p i 0 w przeciwnym razie,
- $\sum_{i=1}^{n} \sum_{k=1}^{n} I_{ik}(p) \cdot \text{dist}(X_i, X_k)$ - suma odległości między obiektami X_i i X_k tylko dla par obiektów w tym samym bloku o rozmiarze p.



Rysunek 8.1: Podział danych na bloki w algorytmie LOF pozwala na bardziej efektywne przetwarzanie dużych zbiorów danych, redukując obciążenie pamięci i poprawiając wydajność obliczeń. Źródło: opracowanie własne.

Zastosowana technika dekompozycji, zademonstrowana na rysunku 8.1, polega na podziale danych na mniejsze bloki, obliczeniu wartości LOF dla każdego z tych bloków oraz połączeniu wyników. Technika jest użyteczna w przypadku bardzo dużych zbiorów danych, gdzie tradycyjna kwadratowa złożoność obliczeniowa algorytmu LOF staje się problematyczna. Poniżej przedstawiono wpływ tego podejścia na złożoność:

- redukcja złożoności - złożoność obliczeniowa algorytmu LOF dla każdego bloku wynosi $O(p^2 \times m)$, gdzie p to liczba obiektów w bloku, a m to liczba zmiennych. Ponieważ p jest znacznie mniejsze niż w przypadku pełnego zbioru danych, czas obliczeń dla każdego bloku jest odpowiednio krótszy,
- złożoność całkowita – w przypadku podzielenia dużego zbioru danych na (b) bloków, gdzie każdy blok ma rozmiar $p = \frac{n}{b}$, złożoność obliczeniowa dla każdego bloku wynosi $O\left(\left(\frac{n}{b}\right)^2 \times m\right)$. Sumując złożoność obliczeniową wszystkich bloków, całkowita złożoność wynosi $O\left(b \times \left(\frac{n}{b}\right)^2 \times m\right)$, co upraszcza się do $O\left(\frac{n^2}{b} \times m\right)$. Wskazuje to, że złożoność obliczeniowa na blok jest mniejsza, a całkowity czas obliczeń może być krótszy przy odpowiednio dużej liczbie bloków b.

Obliczenia pokazują, że podział danych na mniejsze bloki zwiększa efektywność, ponieważ bloki o mniejszej liczbie obiektów charakteryzują się niższą złożonością obliczeniową. Większa liczba bloków oznacza, że każdy blok zawiera mniej danych, co przyczynia się do skrócenia całkowitego czasu obliczeń. Przetwarzanie mniejszych bloków jest szybsze ze

względu na zmniejszoną wielkość macierzy do obliczeń i mniejszą liczbę porównań między parami. Zarządzanie pamięcią jest bardziej efektywne, co zmniejsza ryzyko przeciążeń pamięci i pozwala na wykorzystanie szybszych algorytmów cache. Dekompozycja danych umożliwia równoległe przetwarzanie bloków na różnych procesorach lub serwerach, co znacząco przyspiesza całkowity czas przetwarzania. Przykładowy rozmiar bloku jest ustawiany w następujący sposób:

```
block_size = 20000  # rozmiar bloku
```

Bloki danych są generowane przy użyciu poniższego kodu w języku Python:

```
data_blocks = [data[i: i + block_size] for i in range(0,
    len(data), block_size)]
```

Podsumowując wstęp teoretyczny, kroki algorytmu LOF z podziałem na bloki są przedstawione poniżej jako algorytm 4 (por. rysunek 8.1). Algorytm ten przetwarza duże zbiory danych poprzez ich podział na mniejsze fragmenty, analizuje te fragmenty osobno, a następnie sortuje i łączy wyniki LOF, co pozwala na uzyskanie końcowej uporządkowanej listy wartości LOF.

Algorytm 4: Algorytm LOF z podziałem na bloki

Input: Zbiór danych D z obiektami $X_i \in \mathbb{R}^m$, liczba bloków b, liczba sąsiadów k
Output: Uporządkowana lista wartości LOF, gdzie najwyższe wartości wskazują anomalie

1 Podziel dane D na b bloków $B_1, B_2, \ldots, B_b$, gdzie większość bloków ma równą wielkość, a ostatni blok może być mniejszy
2 **for** *każdy blok jest przetwarzany indywidualnie* B_j *w* $B_1, B_2, \ldots, B_b$ **do**
3 Oblicz macierz odległości dla B_j
4 Znajdź k-najbliższych sąsiadów dla każdego obiektu w B_j
5 Oblicz odległość osiągalności dla każdego obiektu w B_j
6 Oblicz lokalną gęstość osiągalności dla każdego obiektu w B_j
7 Oblicz LOF dla każdego obiektu w B_j
8 Zapisz wartości LOF dla wszystkich obiektów w B_j w liście L_j
9 Połącz listy $L_1, L_2, \ldots, L_b$ w jedną listę L
10 Posortuj listę L w kolejności malejącej
11 **return** *Lista L, gdzie najwyższe wartości wskazują anomalie*

Eksperymenty przeprowadzono zgodnie z metodologią przedstawioną na rysunku 8.2, opiera się ona na podejściu porównawczym. Celem badania była ocena skuteczności i wydajności algorytmu LOF w identyfikacji anomalii w danych kategorycznych i mieszanych, zarówno z podziałem na bloki, jak i bez tego podziału. W celu przeprowadzenia badań

wygenerowano syntetyczne anomalie dla badanych zestawów danych. Wartości odstające, stanowiące 1% zbioru, zostały stworzone, aby symulować różnorodne obserwacje odstające, niezbędne do testowania algorytmu. Te anomalie otrzymały losowe wartości różnych cech i były losowo wprowadzane w miejsce rzeczywistych danych. Metoda ta nie obejmowała zbiorów danych *„credit card”* i *„p53 mutants”*, które już zawierały oznaczone etykiety. Inspiracją dla tego podejścia była zasada trzech sigm w statystyce. Zgodnie z tą zasadą, wprowadzone anomalie odpowiadają wartościom leżącym poza trzema odchyleniami standardowymi, co jest zgodne z koncepcją wartości ekstremalnych w rozkładzie normalnym. Ta zasada mówi, że około 99,7% danych znajduje się w obrębie trzech odchyleń standardowych od średniej. W związku z tym, wybrano próg 1%, aby dokładnie reprezentować wartości ekstremalne.

Rysunek 8.2: Schemat metodologii badawczej. Źródło: opracowanie własne.

Algorytm LOF został zaimplementowany w języku Python w celu obliczenia wartości współczynników LOF dla każdego obiektu w badanych zbiorach. Wykorzystano bibliotekę `NumPy` [297] oraz moduł `time` [389] do pomiaru czasu wykonania algorytmu w celu oceny jego efektywności. Implementacja uwzględniała także moduł `cdist` [390]

z biblioteki `SciPy` [391], doceniono zalety zoptymalizowanych wersji w C i bibliotek niskopoziomowych. Implementacja pozwala przetwarzać dane z podziałem na bloki, umożliwiając dostosowanie rozmiaru bloku, wybór wartości hiperparametru k (MinPts) oraz zastosowanie różnych popularnych metryk, takich jak euklidesowa, kosinusowa, Hamminga, Minkowskiego, Canberra, Jaccarda itp. Przed zastosowaniem algorytmu, dane zostały poddane przetwarzaniu wstępnemu. Dane kategoryczne przekształcono w wektory binarne za pomocą kodowania zero-jedynkowego. Brakujące wartości uzupełniono średnimi wartościami kolumn, a dane numeryczne przeskalowano do zakresu [0, 1]. Zbiór danych *„p53 mutants”* użyty w tej analizie zawierał początkowo 5 408 cech i 16 772 obiekty, w tym brakujące dane, co sprawiało, że był trudny do przetworzenia z powodu swojej ogromnej wymiarowości oraz silnych korelacji między cechami. Aby rozwiązać te problemy, usunięto 180 obiektów z brakującymi wartościami w jakiejkolwiek kolumnie. Dodatkowo przeprowadzono selekcję cech, tworząc macierz korelacji i usuwając cechy o współczynnikach korelacji większych niż 0,5, aby zmniejszyć wzajemne powiązania między cechami. W wyniku tego procesu pozostało 16 591 obiektów, a liczba cech została zredukowana do 444. Badane zbiory danych, zaprezentowane na rysunku 8.3, znajdują się

Rysunek 8.3: Zbiory danych użyte w badaniu. Źródło: opracowanie własne.

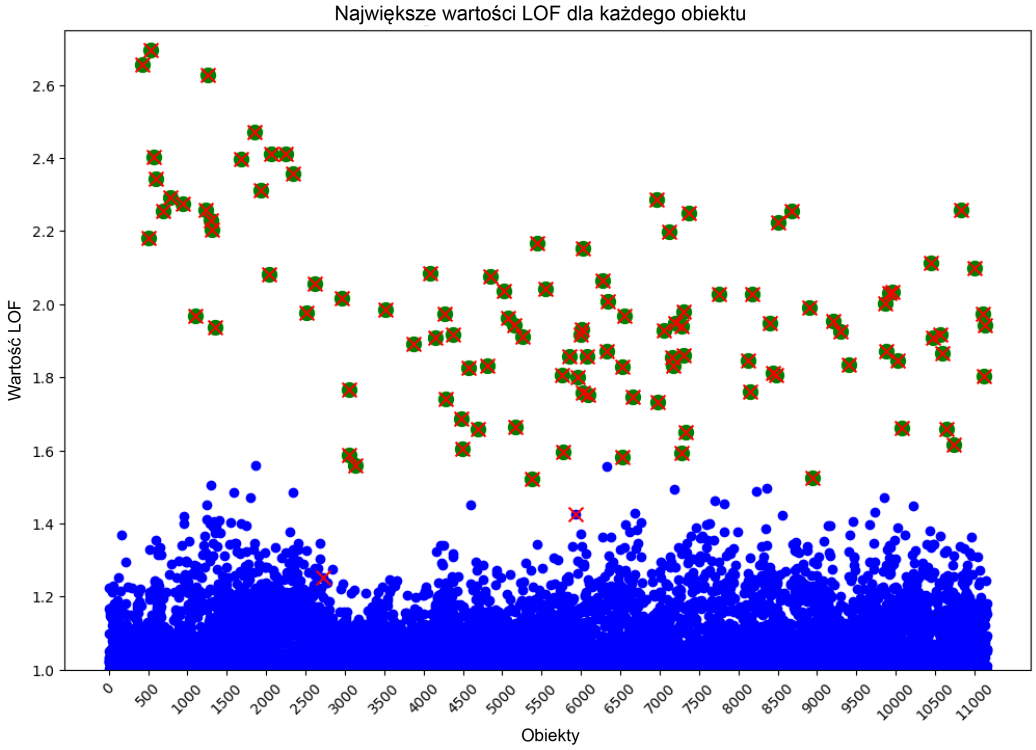
pod następującym linkiem[3]. Wszystkie zbiory danych, z wyjątkiem *„credit card”* i *„p53 mutants”*, zawierają cechy kategoryczne. Poniżej znajduje się ich krótka charakterystyka:

[3]https://drive.proton.me/urls/KP6QTFJ05W#Vud5diXWs295

- *„car evaluation”* - zawiera informacje oceniające samochody na podstawie takich kryteriów jak cena zakupu, koszty utrzymania, liczba drzwi, pojemność samochodu, wielkość bagażnika oraz poziom bezpieczeństwa. Składa się z 1 728 obiektów i 6 cech, wszystkie cechy są kategoryczne i nie zawierają brakujących wartości [392],
- *„mushroom”* - zawiera 8 124 obiekty z 22 cechami kategorycznymi i jest używany do klasyfikacji grzybów na podstawie ich właściwości fizycznych. Dane pochodzą z przewodnika, każdy gatunek jest oznaczony jako jadalny, trujący lub niezalecany do spożycia. Zbiór danych obejmuje opisy hipotetycznych próbek odpowiadających 23 gatunkom grzybów z rodzin Agaricus i Lepiota [393],
- *„bank marketing”* - zawiera informacje dotyczące kampanii marketingowych przeprowadzanych przez portugalską instytucję bankową. Kampanie te były oparte na rozmowach telefonicznych, których celem było sprawdzenie, czy klient zdecyduje się na założenie lokaty terminowej. Dane zawierają 11 162 obiekty i 17 cech, w tym cechy kategoryczne, takie jak rodzaj pracy, stan cywilny, poziom wykształcenia, typ kontaktu, dzień tygodnia ostatniego kontaktu oraz wynik poprzedniej kampanii. Klasyfikacja ma na celu przewidzenie, czy klient założy lokatę terminową [394],
- *„citibike”* - zawiera informacje o wynajmach rowerów w Nowym Jorku, dostarczone przez system Citi Bike. System został uruchomiony w maju 2013 roku i stanowi ważną część sieci transportowej miasta. Dane obejmują 577 703 obiekty i 15 cech, w tym czas trwania podróży, czas rozpoczęcia i zakończenia, identyfikatory i nazwy stacji początkowych i końcowych, współrzędne geograficzne stacji, identyfikatory rowerów oraz dane demograficzne użytkowników, takie jak rok urodzenia, typ użytkownika (klient jednorazowy lub abonamentowy) oraz płeć. Wszystkie te cechy są kategoryczne. Do analizy i eksperymentów użyto pierwszych 20 000 obiektów [395],
- *„adult”* - zawiera informacje pochodzące z amerykańskiego spisu ludności z 1994 roku. Celem jest przewidzenie, czy dochód osoby przekracza 50 000 dolarów rocznie. Zbiór danych zawiera 48 842 obiekty i 14 cech, w tym cechy kategoryczne, takie jak rodzaj pracy, poziom wykształcenia, stan cywilny, zawód, relacja w rodzinie, rasa, płeć, a także cechy ciągłe, takie jak wiek, liczba godzin pracy w tygodniu, kapitał zysków i strat oraz kraj pochodzenia [396],
- *„credit card”* - zawiera informacje o transakcjach kartami kredytowymi dokonanych we wrześniu 2013 roku przez europejskich posiadaczy kart. Dane obejmują 284 807 obiektów (transakcji), z czego 492 są oznaczone jako oszustwa (co stanowi 0,172% wszystkich transakcji). Zbiór danych jest mocno niezrównoważony. Wszystkie zmienne wejściowe są ilościowe i są wynikiem transformacji PCA, z wyjątkiem cech *Time* i *Amount*. Zbiór danych zawiera 31 cech, w tym *Time*, *Amount* oraz 28 składowych głównych oznaczonych jako *V1* do *V28*. Cecha decyzyjna przyjmuje wartość 1 w przypadku oszustwa i 0 w przeciwnym razie [397],

- *„p53 mutants”* - zawiera 16 772 obiekty, informacje o mutacjach białka p53 wyekstrahowane z symulacji biofizycznych. Początkowo zbiór zawierał 5 408 cech reprezentujących właściwości mutantów białka p53 uzyskanych z symulacji. Po optymalizacji pozostały 444 cechy poprzez usunięcie cech o wysokich współczynnikach korelacji oraz 180 obiektów, które miały brakujące wartości w jakiejkolwiek kolumnie, co pozwoliło na zachowanie 16 591 obiektów. Celem analizy jest modelowanie transkrypcyjnej aktywności mutanta p53 (aktywny w przeciwieństwie do nieaktywnego). Cechą klasyfikacyjną jest aktywność, oznaczona jako *active* (transkrypcyjnie kompetentne, aktywne p53) lub *inactive* (rakowe, nieaktywne p53). Więcej informacji na temat zbioru można znaleźć w artykule [398].

Na rysunku 8.4 zaprezentowano wyniki analizy dla jednego ze zbiorów danych *„bank marketing”*. Wykres przedstawia wartości współczynnika LOF dla każdego obiektu w zbiorze. Wykryte anomalie są oznaczone jako zielone punkty z czerwonym krzyżykiem, podczas gdy pozostałe wartości LOF są reprezentowane jako niebieskie punkty. Anomalie, które nie zostały wykryte, są oznaczone jedynie czerwonym krzyżykiem bez zielonego punktu.



Rysunek 8.4: Najwyższe wartości LOF: Przykład zbioru danych *„bank marketing”*. Wspólne obiekty: 109, Pokrycie: 98.2%, anomalie bez wspólnych obiektów: {2730, 5933}, Liczba anomalii bez wspólnych obiektów: 2, Parametry: MinPts (k): 24, Metryka odległości: Hamming. Źródło: opracowanie własne.

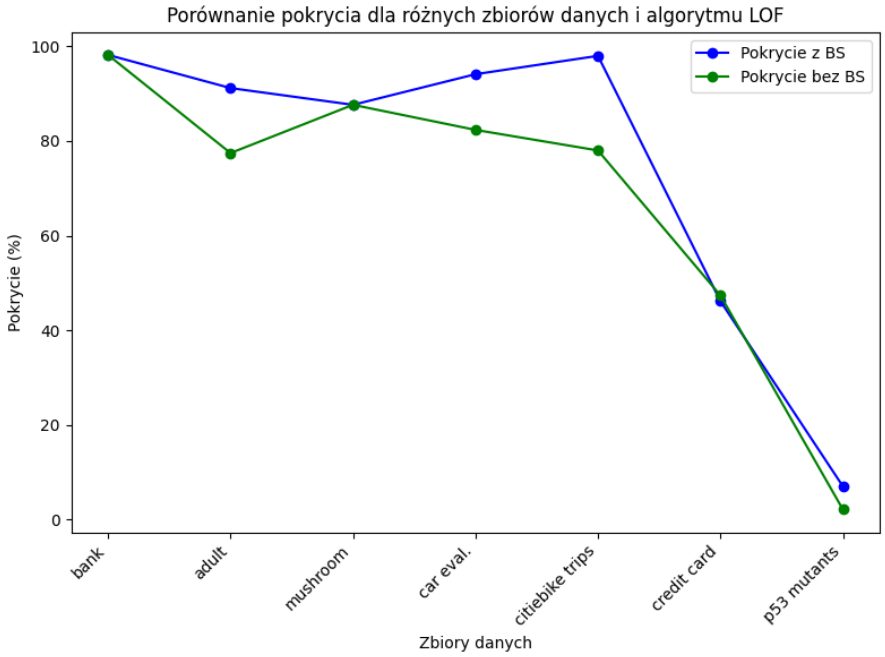
Tabela 8.1: Wyniki dla różnych zbiorów danych i algorytmu LOF, gdzie BS to rozmiar bloku (*ang. block size*). Źródło: opracowanie własne.

Zbiór danych [obiekty;anom.;cechy]	Pokrycie z BS|Pokrycie bez BS %	Czas z BS|Czas bez BS sek.	Param. z BS|bez BS
bank [11162;111;17]	98,20 | 98,20	0,62 | 39,92	BS: 200 | 11162 k: 24 | 74 metryka: hamming
adult [48842;488;14]	91,19 | 77,45	1,57 | 186,07	BS: 100 | 24421[1] k: 27 | 104 metryka: hamming
mushroom [8124;81;22]	87,65 | 87,65	0,23 | 8,94	BS: 200 | 8124 k: 10 | 97 metryka: cosine
car eval. [1728;17;6]	94,12 | 82,35	0,025 | 0,234	BS: 200 | 1728 k: 8 | 6 metryka: euclidean
citiebike trips [20000;200;15]	98 | 78	10,33 | 3913,43	BS: 200 | 20000 k: 28 | 107 metryka: hamming
credit card [284807;492;31]	46,14 | 47,35	252,6 | 1497,18	BS: 6500 | 35600[2] k: 250 | 250 metryka: euclidean
p53 mutants [16591;143;444]	6,99 | 2,1	0,84 | 382,97	BS: 100 | 16591 k: 9 | 46 metryka: euclidean

[1] BS: 48842 po 255 sek., brak dostępnej pamięci RAM (52 GB niewystarczające)
[2] BS: 284807 po 201 sek., brak dostępnej pamięci RAM (52 GB niewystarczające)

Implementacja zapewniła analizę zbioru danych z wykorzystaniem hiperparametrów algorytmu LOF, takich jak k (liczba sąsiadów), miary odległości oraz dobór rozmiaru bloku. Badanie składało się z dwóch faz. **W pierwszej fazie eksperymentu** cały zbiór danych traktowano jako jeden wspólny blok. Algorytm LOF był uruchomiony z wcześniej zdefiniowanymi hiperparametrami, takimi jak parametr k (liczba sąsiadów) oraz miara odległości. Dla wybieranych hiperparametrów obliczano wspólne obiekty oraz procentowe pokrycie między rzeczywistymi anomaliami (*ang. ground truth*) a największymi wartościami współczynnika osobliwości LOF, a także mierzono czas wykonania, starając się uzyskać jak najlepszy rezultat. Wyniki zanotowano, aby ocenić skuteczność algorytmu w identyfikacji anomalii oraz czas przetwarzania, i porównać je z wynikami drugiej fazy. **W drugiej fazie eksperymentu** dostosowywano rozmiar bloku dla każdego zbioru danych. Celem było znalezienie optymalnego rozmiaru bloku, który zapewni wysoką skuteczność wykrywania anomalii oraz efektywność czasową. Dane zostały podzielone na bloki o różnych rozmiarach (np. 200 obiektów na blok), a algorytm LOF był uruchamiany dla każdego bloku z wcześniej określonymi hiperparametrami, takimi jak parametr k (liczba sąsiadów) oraz metryka odległości. Wyniki dla wszystkich bloków były następnie agregowane, aby uzyskać wspólny wynik dla całego zbioru. Notowano skuteczność algorytmu

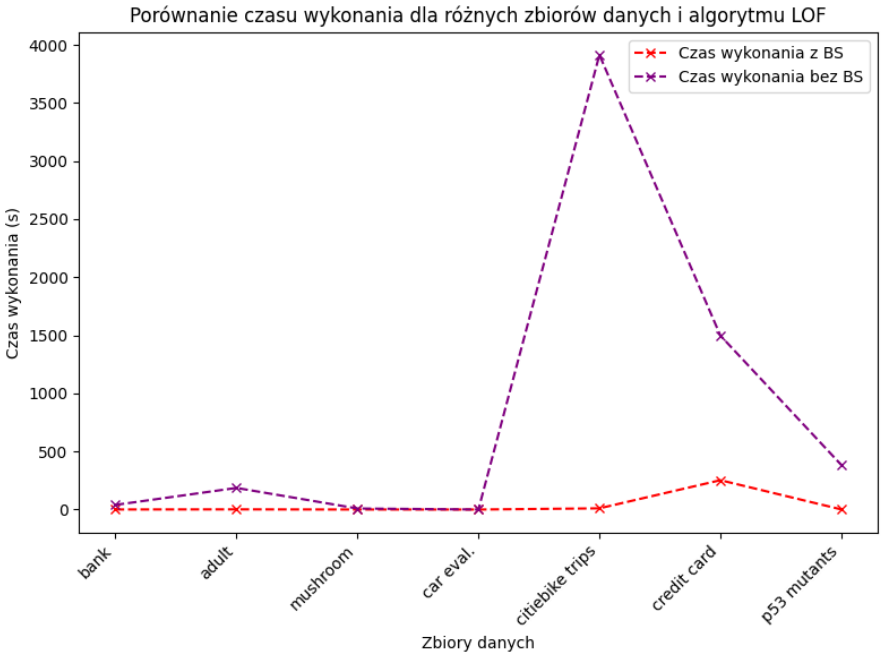
w identyfikacji anomalii oraz zmierzony czas przetwarzania dla całego procesu. Dla każdego ustawienia rozmiaru bloku obliczano procentowe pokrycie między rzeczywistymi anomaliami a największymi wartościami LOF. Eksperymenty pozwoliły na określenie optymalnego rozmiaru bloku oraz dostosowanie wartości k i miary odległości, które zapewniały najlepszą zgodność między wykrytymi anomaliami a rzeczywistymi. Szukano równowagi między jakością detekcji a jak najkrótszym czasem przetwarzania.



Rysunek 8.5: Porównanie zgodności między rzeczywistymi anomaliami a najwyższymi wartościami LOF dla różnych zbiorów danych z podziałem na bloki i bez podziału. Źródło: opracowanie własne.

Tabela 8.1 oraz rysunki 8.5 i 8.6 przedstawiają wyniki eksperymentów na różnych zbiorach. Na podstawie tych eksperymentów można wyciągnąć następujące wnioski:

- **skuteczność identyfikacji anomalii** - tabela 8.1 porównuje *Pokrycie z BS* i *Pokrycie bez BS* dla każdego zbioru danych, używając algorytmu z optymalizacją rozmiaru bloku i bez niej. Odsetek pokrycia mierzy skuteczność wykrywania, wskazując zgodność między rzeczywistymi anomaliami a najwyższymi wartościami LOF. Optymalizacja rozmiaru bloku zazwyczaj zapewnia podobne lub nawet nieco wyższe pokrycie w porównaniu do algorytmu bez optymalizacji, co sugeruje, że zmiana rozmiaru bloku nie tylko nie ma negatywnego wpływu na skuteczność algorytmu, ale może ją nawet zwiększyć. Nie jest to regułą, jak pokazuje przykład zbioru danych *„credit card”*, pokrycie było o 1,21% gorsze z podziałem na bloki. Szczegóły ilustruje rysunek 8.5,



Rysunek 8.6: Porównanie czasu wykonania dla różnych zbiorów danych, z podziałem na bloki i bez tego podziału. Źródło: opracowanie własne.

- **czas przetwarzania** - tabela 8.1 porównuje również czasy wykonania *Czas z BS* i *Czas bez BS* dla każdego zbioru danych przy użyciu algorytmu LOF z optymalizacją rozmiaru bloku i bez niej. Optymalizacja rozmiaru bloku często prowadzi do znaczącej redukcji czasu wykonania, co czyni algorytm bardziej efektywnym w przetwarzaniu dużych zbiorów danych, co doskonale ilustruje rysunek 8.6,
- **hiperparametry** - tabela 8.1 zawiera informacje o optymalnych hiperparametrach używanych w optymalizacji rozmiaru bloku dla każdego zbioru danych, takich jak rozmiar bloku (BS), liczba najbliższych sąsiadów (k) oraz metryka odległości. Wybór tych parametrów może znacząco wpływać na wydajność algorytmu. Na przykład, dla zbioru danych *„bank marketing”*, zastosowanie rozmiaru bloku wynoszącego 200, liczby sąsiadów $k = 24$ oraz metryki odległości hamming prowadzi do wysokiego pokrycia rzeczywistych anomalii z wykrytymi oraz niskiego czasu wykonania,
- **ograniczenia zasobów** - dla niektórych przypadków zbadanie całych zbiorów było niemożliwe ze względu na ograniczenia zasobów sprzętowych, co wskazano w przypisie do tabeli 8.1. Ograniczenia uniemożliwiały wykorzystanie całych zbiorów jako jednego bloku, co wymuszało dobór największego możliwego rozmiaru dostosowanego do dostępnej pamięci RAM. Dla wersji bez optymalizacji maksymalny rozmiar bloku wynosił 24 421 dla zbioru *„adult”* i 35 600 dla zbioru *„credit card”*. Wyższe wartości powodowały błędy z powodu niewystarczającej pamięci RAM (52 GB).

Eksperymenty wykazały, że optymalizacja rozmiaru bloku w algorytmie LOF rzadko obniża skuteczność wykrywania anomalii, a często ją poprawia, co jest bardzo obiecujące. Jednocześnie podział na bloki znacząco skraca czas przetwarzania, co jest szczególnie istotne dla dużych zbiorów danych. Ograniczenia sprzętowe, takie jak pamięć RAM, w niektórych przypadkach wymusiły redukcję rozmiaru badanego zbioru. W praktyce optymalizacja rozmiaru bloku była ograniczana przez dostępne zasoby pamięci, nawet w płatnym środowisku Colab Pro+ [399], gdzie podczas badań maksymalna dostępna pamięć wynosiła 52 GB. Na bardziej wydajnych komputerach, przy odpowiednich zasobach pamięci, dostosowanie większego rozmiaru bloku (cały zbiór) jest zapewne możliwe. Wnioski te podkreślają potrzebę dostosowania parametrów algorytmu do specyfiki zbioru danych i dostępnych zasobów obliczeniowych, aby uzyskać optymalne wyniki. W kolejnym eksperymencie podjęto próbę zautomatyzowania doboru rozmiaru bloku i innych hiperparametrów, biorąc pod uwagę również zasoby sprzętowe.

8.1.2 Eksperyment 2: adaptacyjna optymalizacja rozmiaru bloku

We wcześniejszych eksperymentach zbadano innowacyjne podejście, które wykorzystuje optymalizację rozmiaru bloku do przyspieszenia procesu wykrywania anomalii przy jednoczesnym utrzymaniu wysokiej skuteczności. Obecne eksperymenty kontynuowano w tej linii badań, wprowadzając nowe podejście, które udoskonala poprzednią metodę. Rozmiar bloku oraz inne hiperparametry nie będą już wybierane ręcznie. Zamiast tego proces ten został zautomatyzowany za pomocą algorytmu, który dynamicznie oblicza ich optymalne wartości, dostosowując się do liczby obiektów w zbiorze, jego struktury oraz dostępnej pamięci RAM. Celem jest adaptacyjna i automatyczna optymalizacja rozmiaru bloku i innych hiperparametrów dla efektywnego wykorzystania zasobów i skutecznego przetwarzania danych.

W fazie wstępnej zastosowano specjalnie stworzoną metodę opartą na dynamicznym dostosowywaniu rozmiaru bloku przetwarzanego przez algorytm, co powinno umożliwić efektywne zarządzanie zasobami sprzętowymi i skrócić czas obliczeń, przy zachowaniu wysokiej skuteczności wykrywania anomalii. Ważnym elementem proponowanego rozwiązania jest funkcja celu *wynik* (*ang. score*), która łączy czas wykonania, procent pokrycia (stosunek rzeczywistych anomalii do wykrytych) oraz sumę odległości w analizowanym bloku (szczegóły we wzorze (8.1)). Celem funkcji jest minimalizacja czasu i sumy odległości przy maksymalizacji pokrycia. Wyzwanie podjęte w tym eksperymencie wynika z ograniczeń algorytmu LOF w kontekście analizy dużych zbiorów danych, gdzie tradycyjna metoda często zawodzi ze względu na rosnącą złożoność obliczeniową i wymagania pamięciowe [98, 400, 401, 402]. Dodatkowo, celem było zautomatyzowanie oraz algorytmiczny dobór rozmiaru bloku i innych hiperparametrów na podstawie precyzyjnie określonych warunków, mających istotne znaczenie dla tego procesu.

Zastosowano technikę optymalizacji bayesowskiej [403, 404, 405, 406], która pozwala na dostosowanie hiperparametrów algorytmu w celu osiągnięcia jak najlepszych

wyników. Optymalizacja bayesowska, wykorzystująca funkcję `gp_minimize` z biblioteki `scikit-optimize` [407], przeprowadza `n_calls` iteracji procesu optymalizacji, eksplorując przestrzeń hiperparametrów w poszukiwaniu optymalnej konfiguracji. Przestrzeń ta jest zdefiniowana jako `param_space` i obejmuje zakresy dla liczby najbliższych sąsiadów (k), współczynników skali (a i b), stałej (c) oraz metryki odległości. Hiperparametry algorytmu są dostosowywane w celu maksymalizacji skuteczności wykrywania anomalii przy minimalizacji czasu wykonania i sumy odległości między obiektami. Wybór optymalizacji bayesowskiej jako pomocnej metody dla badań był motywowany jej zdolnością do efektywnego eksplorowania przestrzeni hiperparametrów, co jest potrzebne do identyfikacji najbardziej efektywnych konfiguracji dla wybranych zbiorów danych. Funkcja celu *wynik* składa się z trzech kluczowych komponentów, zdefiniowano ją jako (8.2):

$$\begin{aligned} \text{wynik} = {} & \text{waga}_{\text{pokrycia}} \times \text{procent\_pokrycia} \\ & - \text{waga}_{\text{czasu\_wykonania}} \times \text{czas\_wykonania} \\ & - \text{waga}_{\text{sumy\_odległości}} \times \text{całkowita\_suma\_odległości} \end{aligned} \tag{8.2}$$

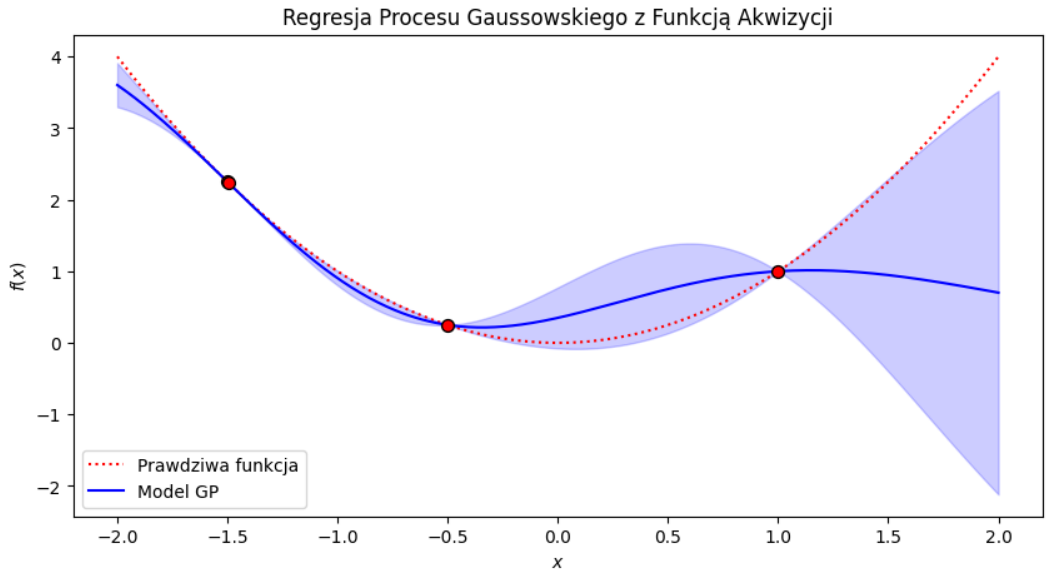
Wagi poszczególnych składowych funkcji celu zostały starannie dobrane, aby dokładnie odzwierciedlać istotę analizy i znaczenie różnych czynników. Pokrycie (*waga_pokrycia* (*WtCov*)) jest kluczowym elementem, ponieważ pokazuje, jak skutecznie algorytm identyfikuje anomalie w danych. Dlatego jego waga (*WtCov*) została ustalona na najwyższą wartość (*1,0*), co oznacza, że ma największy wpływ na ocenę. Z kolei czas wykonania (*waga_czasu_wykonania* (*WtETm = 0,001*)) został uwzględniony w celu zapewnienia odpowiedniej efektywności algorytmu. Mimo że czas wykonania jest istotny, nie powinien przeważać nad pokryciem. Dlatego jego waga jest znacznie niższa. Na koniec, suma odległości (*waga_sumy_odległości* (*WtSumD = 0,000000001*)) ma mniejsze znaczenie w porównaniu do pokrycia i czasu wykonania. Jej niższa waga pomaga utrzymać równowagę między szybkością działania a precyzją identyfikacji anomalii, nie wpływając nadmiernie na ocenę. Uwzględniono ten czynnik, ponieważ analiza bardziej zwartych bloków danych może poprawić efektywność i skuteczność algorytmów wykrywania anomalii. Algorytmy oparte na gęstości, takie jak LOF, lepiej rozróżniają między normalnymi przypadkami a anomaliami, gdy operują na bardziej zwartych grupach. Zwartość obiektów, czyli ich bliskość, pozwala skuteczniej identyfikować obiekty wyróżniające się spośród głównych grup. Suma odległości jest istotna, ponieważ analiza zwartych bloków przyspiesza przetwarzanie. Obiekty zebrane blisko siebie są mniej prawdopodobne do uznania za szum, co poprawia dokładność wykrywania anomalii i minimalizuje wpływ szumu na analizę. Ta część funkcji celu pochodzi z teorii grupowania, gdzie algorytmy takie jak k-means czy DBSCAN wykorzystują zwartość obiektów do tworzenia spójnych grup, co ułatwia interpretację wyników i poprawia jakość grupowania, pozwalając na identyfikację obiektów odstających.

Opisane podejście i wykonywane eksperymenty uzasadnia obszerna literatura naukowa dotycząca optymalizacji bayesowskiej. Przykładowo, artykuł [408] skupia się na zastosowaniu optymalizacji bayesowskiej do strojenia hiperparametrów w różnych modelach uczenia maszynowego, co jest analogiczne do wykorzystania tej metody w innych kontekstach, takich jak optymalizacja rozmiaru bloku czy hiperparametrów algorytmu wykrywania anomalii. Teoretyczne podstawy tego procesu, opisane w artykule, obejmują zastosowanie procesów gaussowskich do modelowania rozkładu funkcji celu oraz wybierania kolejnych hiperparametrów do oceny za pomocą funkcji akwizycji, które wskazują potencjalnie lepsze rozwiązania.

Funkcja z biblioteki `scikit-optimize` [409] wykorzystuje procesy gaussowskie do efektywnej optymalizacji funkcji celu, szczególnie gdy ocena tej funkcji jest kosztowna. Minimalizuje ona liczbę koniecznych wywołań funkcji celu poprzez iteracyjne oceny hiperparametrów w przestrzeni poszukiwań, korzystając z modeli probabilistycznych i funkcji akwizycji do wyboru najbardziej obiecujących hiperparametrów. Funkcja celu jest wywoływana dla hiperparametrów wybranych przez funkcję akwizycji, a wyniki tych ocen są używane do aktualizacji modelu gaussowskiego, co prowadzi do bardziej precyzyjnych przewidywań. Proces optymalizacji kontynuuje iteracyjne wybieranie nowych hiperparametrów do oceny, bazując na zaktualizowanym modelu, aż do osiągnięcia określonej liczby wywołań `n_calls` funkcji celu. Wynikiem końcowym jest zestaw najlepszych znalezionych hiperparametrów oraz odpowiadająca im wartość funkcji celu. Jeśli celem jest maksymalizacja wartości funkcji celu, można łatwo dostosować problem maksymalizacji do formy minimalizacji, zwracając ujemną wartość funkcji celu w funkcji obliczeniowej. W ten sposób proces wyszukiwania minimalnej wartości ujemnej skutecznie maksymalizuje oryginalną wartość funkcji celu.

Rysunek 8.7 ilustruje, jak model procesu gaussowskiego (GP) może być używany do optymalizacji funkcji celu $f(x) = x^2$ przy użyciu funkcji akwizycji. Funkcja celu w przedstawionym przykładzie jest zdefiniowana jako $f(x) = x^2$. Celem jest znalezienie minimalnej wartości funkcji $f(x)$. W kontekście optymalizacji dążymy do znalezienia wartości x, dla której funkcja celu osiąga swoje minimum. Poszczególne elementy rysunku 8.7 oznaczają:

- **prawdziwą funkcję** (czerwona linia przerywana) - jest to funkcja celu, która jest oceniana w ramach optymalizacji, w przykładzie opisana równaniem $f(x) = x^2$,
- **model GP** (niebieska linia) - pokazuje przewidywania modelu procesu gaussowskiego (GP). Model GP szacuje wartości funkcji celu na podstawie próbki danych,
- **przewidywana niepewność modelu GP** (niebieskie cieniowanie) - reprezentuje zakres przewidywanej niepewności modelu. Jest to przedział ufności 95%, pokazujący, jak bardzo model jest pewny swoich przewidywań w przestrzeni poszukiwań,
- **punkty próbkowania** (czerwone kropki) - są to miejsca, w których oceniono funkcję celu. Punkty te są używane do budowy modelu GP i do oceny funkcji akwizycji w celu wyboru nowych punktów do próbkowania.



Rysunek 8.7: Przykład regresji procesu gaussowskiego z funkcją akwizycji. Źródło: opracowanie własne.

Implementacja przygotowana do eksperymentów stosuje optymalizację bayesowską do dostosowania hiperparametrów algorytmu, takich jak k, a, b, c oraz wybór metryki odległości. Celem jest maksymalizacja funkcji celu (8.2). Hiperparametry te służą do obliczania optymalnego rozmiaru bloku, który według wzoru (8.3) dostosowuje się do liczby obiektów n oraz dostępnej pamięci RAM. Współczynniki a i b wpływają na skalowanie rozmiaru bloku, a stała c determinuje wpływ n na rozmiar bloku, podczas gdy k określa optymalną liczbę najbliższych sąsiadów, wprowadzając elastyczność do procesu wykrywania anomalii.

Nowe podejście proponowane w eksperymentach automatyzuje proces wyboru rozmiaru bloku oraz innych hiperparametrów. Dzięki zastosowanej metodzie możliwa jest efektywna eksploracja przestrzeni rozwiązań, umożliwiająca identyfikację optymalnych wartości a, b, c, k oraz wybór najodpowiedniejszej metryki (euklidesowej, kosinusowej, Hamminga itp.). Takie podejście zapewnia maksymalizację wartości funkcji celu *wynik*. Nowością w porównaniu do poprzednich eksperymentów jest również wprowadzony wzór (8.3), który przyczynia się do usprawnienia procesu optymalizacji. Wartość rozmiaru bloku (*block_size*) jest zaokrąglana do najbliższej liczby całkowitej przed zastosowaniem:

$$\text{block\_size} = \max\left(1, \min\left(n, \frac{n}{a \cdot \sqrt{k} \cdot b} \cdot \frac{\text{dostępny\_RAM}}{n \cdot c}\right)\right) \tag{8.3}$$

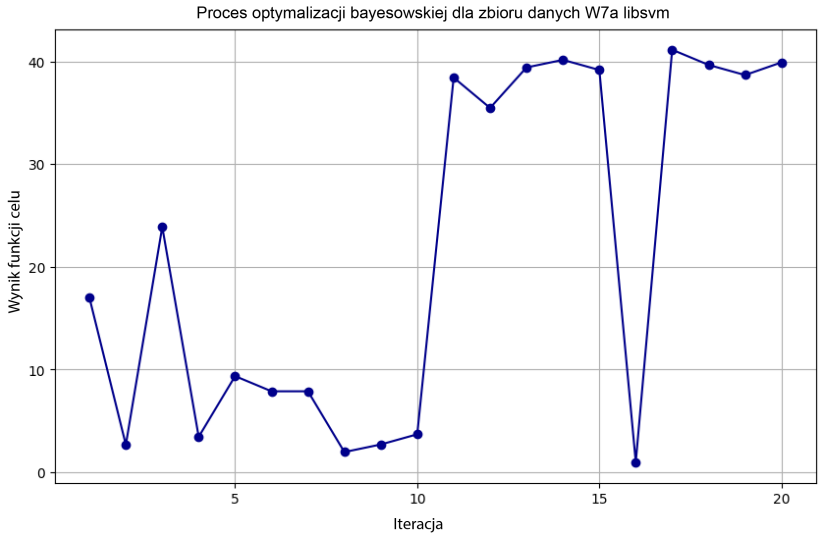
Wyrażenie *max(1, min(n, ...))* zapewnia, że *block_size* wynosi co najmniej 1 i nie przekroczy n. Rozmiar bloku jest obliczany na podstawie znalezionych optymalnych parametrów:

- n – liczba obiektów,
- a, b – parametry wpływające na skalowanie,
- c – stała, im większa, tym większy wpływ n na rozmiar bloku,
- k – optymalna liczba najbliższych sąsiadów,
- *dostępny_RAM* – ilość dostępnej pamięci RAM.

Zastosowanie dwóch niezależnych parametrów a i b zamiast jednego współczynnika skalowania d (co mogłoby wydawać się bardziej intuicyjne) pozwala na bardziej precyzyjne i zróżnicowane dostosowanie algorytmu do specyficznych wymagań zestawów danych, co prowadzi do wyższej wydajności i skuteczności w praktycznych zastosowaniach. Możliwość niezależnej modyfikacji tych parametrów zapewnia większą kontrolę nad dostosowaniem rozmiaru bloku do zmian charakterystyki zestawu danych (n) i dostępnej pamięci RAM. Zakładając stałą $c = 1$, równanie (8.3) upraszcza się do równania (8.4). Wartość *block_size* jest zaokrąglana do najbliższej liczby całkowitej przed zastosowaniem:

$$\text{block\_size} = \max\left(1, \min\left(n, \frac{\text{dostępny\_RAM}}{a \cdot \sqrt{k} \cdot b}\right)\right) \tag{8.4}$$

Wartość n służy jako ograniczenie, aby rozmiar bloku nie przekroczył liczby obiektów. Rozmiar bloku zależy głównie od dostępnej pamięci RAM oraz parametrów a, b i k.



Rysunek 8.8: Iteracyjny proces optymalizacji bayesowskiej dla zbioru danych *„w7a libsvm”*. Oś X przedstawia numer iteracji, natomiast oś Y pokazuje wartość funkcji celu oznaczonej jako *wynik*. Z każdą iteracją wartość funkcji celu stopniowo zbliża się do optymalnego rozwiązania, maksymalizując *wynik* poprzez inteligentne próbkowanie przestrzeni hiperparametrów. Źródło: opracowanie własne.

Rysunek 8.8 przedstawia iteracyjny proces optymalizacji bayesowskiej na przykładzie zbioru danych *„w7a libsvm”*. Oś X oznacza numer iteracji, natomiast oś Y pokazuje wartość funkcji celu, określanej jako *wynik*. Z każdą iteracją wartość funkcji celu zbliża się do optymalnego rozwiązania. Optymalizacja bayesowska ma na celu maksymalizację *wyniku* poprzez inteligentne próbkowanie przestrzeni hiperparametrów. Informacje z poprzednich iteracji są wykorzystywane do wyboru kolejnych próbek hiperparametrów. Jest to iteracyjna metoda, w której rozkład prawdopodobieństwa hiperparametrów jest aktualizowany na każdym etapie, aby wskazać obszar, gdzie funkcja celu może osiągnąć swoje maksymalne wartości.

Różnorodność i specyfika badanych zbiorów

Eksperymenty zostały przeprowadzone na nowych, zróżnicowanych zbiorach danych, aby potwierdzić skuteczność zaproponowanej metody. Skupiono się na zbiorach takich jak *„wine quality (white)”*, *„vehicle claims”*, *„chess krkopt (king-rook vs. king)”*, *„kdd cup 1999”*, *„w7a libsvm”*, *„covid-19 case surveillance”* oraz *„mushroom”*. Z wcześniejszych badań pozostawiono jedynie dwa zbiory danych numerycznych: *„credit card fraud”* i *„p53 mutants”*. W przypadku zbioru *„mushroom”*, zamiast generować syntetyczne anomalie jak w poprzednich eksperymentach, przyjęto, że trujące grzyby stanowią rzeczywiste anomalie. W związku z tym, wybrano 0,988% trujących grzybów z całego zbioru jako anomalie, a resztę usunięto. W efekcie powstał mniejszy, ale bardziej realistyczny zbiór anomalii, nazwany *„mushroom real”*.

Zbiory danych, zawierające oznaczone wartości odstające w kolumnie *is_outlier*, użyte w eksperymentach, można znaleźć pod linkiem[4]. Wszystkie zbiory danych użyte w tym badaniu są publicznie dostępne w repozytorium UCI Machine Learning Repository [410], na platformie Kaggle [411] oraz z innych źródeł. Tabela 8.2 przedstawia te zbiory, podkreślając ich zróżnicowanie w opisie. Trzy z nich koncentrują się na danych numerycznych: *„Credit Card Fraud”*, *„Wine Quality (White)”* oraz *„p53 Mutants”*, które obejmują odpowiednio transakcje kartami kredytowymi, pomiary fizykochemiczne win oraz badania mutacji białka p53. Pozostałe zbiory, takie jak *„Vehicle Claims labeled”*, *„Chess King-Rook vs. King (KRKOPT - zero vs. all)”*, *„KDD Cup 1999”*, *„w7a libsvm”*, *„Covid-19 Case Surveillance”* i *„Mushroom”*, zawierają dane kategoryczne lub mieszane, odzwierciedlając szeroki zakres zastosowań – od ubezpieczeń, przez cyberbezpieczeństwo, po epidemiologię.

Poniżej przedstawiono szczegółowe informacje dotyczące każdego z omówionych zbiorów danych oraz opis procesu ich przygotowania. Dla danych kategorycznych we wszystkich zbiorach zastosowano metodę kodowania one-hot, z wyjątkiem zbioru danych *„Covid-19 Case Surveillance”*, gdzie bardziej efektywne okazało się użycie metody `LabelEncoder` z biblioteki `scikit-learn` [412].

[4]https://drive.proton.me/urls/33D2V4MPB8#5CLG49nM8Mwf

Tabela 8.2: Charakterystyka analizowanych zbiorów danych. Źródło: opracowanie własne.

Zbiór danych	Rozmiar	Cechy	Anom.	Anom. %	Opis
1. Credit Card	284 807	30	492	0,173%	Analiza transakcji kartą kredytową w celu wykrycia oszustw.
2. Wine Quality	4 898	11	25	0,51%	Ocena jakości białego wina na podstawie pomiarów fizykochemicznych.
3. p53 Mutants	16 591	444	143	0,86%	Badanie mutacji białka p53 w kontekście aktywności transkrypcyjnej i raka.
4. Vehicle Claims	212 994	19	1488	0,699%	Analiza roszczeń ubezpieczeniowych pojazdów w celu identyfikacji potencjalnych oszustw.
5. Chess KRKOPT	28 056	6	27	0,096%	Analiza końcowych pozycji w szachach w celu wykrycia rzadkich przypadków.
6. KDD Cup 1999	97 577	41	299	0,306%	Wykorzystanie danych o ruchu sieciowym do trenowania modeli wykrywania intruzów.
7. w7a libsvm	48 676	300	406	0,834%	Analiza anomalii w danych kategorycznych z niezrównoważonym rozkładem klas.
8. Covid-19 Case	33 911	10	332	0,924%	Monitorowanie przypadków Covid-19, w tym analiza zgonów.
9. Mushroom Real	4 250	22	42	0,988%	Klasyfikacja grzybów jako jadalnych lub trujących na podstawie ich cech.

„Credit Card Fraud” [397] – zbiór danych dotyczący transakcji kartami kredytowymi, szczegółowo omówiono w podrozdziale 8.1.1. Dlatego w tym miejscu przedstawiono jedynie proces wstępnego przygotowania danych, jaki został zastosowany. Zbiór ten stanowi wyzwanie dla badaczy, gdyż dane są mocno niezrównoważone. Kolumny numeryczne w zbiorze danych zostały znormalizowane przy użyciu metody `MinMaxScaler` [413], co jest standardową procedurą przygotowania danych dla wielu algorytmów uczenia maszynowego. Proces ten polega na przekształceniu każdej cechy tak, aby jej wartości mieściły się w jednolitym zakresie od 0 do 1. Operacja normalizacji dla cechy A_j jest zdefiniowana przez wzór (8.5):

$$x'_{i,j} = \frac{x_{i,j} - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{8.5}$$

gdzie $x_{i,j}$ jest oryginalną wartością cechy A_j dla obiektu X_i, $\min(x_j)$ jest minimalną

wartością cechy A_j w zbiorze danych, $\max(x_j)$ jest maksymalną wartością cechy A_j w zbiorze danych, a $x'_{i,j}$ jest znormalizowaną wartością cechy A_j dla obiektu X_i.

„Wine Quality (White)" [414] – dane dotyczą pomiarów fizykochemicznych oraz ocen sensorycznych próbek białego wina z regionu Vinho Verde w Portugalii. Próbki ocenione na 3 (najgorsza jakość) i 9 (najlepsza jakość) zostały oznaczone jako obiekty odstające. Zbiór danych jest wielowymiarowy i zawiera 4 898 próbek, z 11 cechami: kwasowość stała, kwasowość lotna, kwas cytrynowy, cukier resztkowy, chlorki, dwutlenek siarki wolny, dwutlenek siarki całkowity, gęstość, pH, siarczany i alkohol. Celem jest modelowanie jakości wina na podstawie wyników testów fizykochemicznych. Zbiór danych nie zawiera brakujących wartości. Kolumny numeryczne zostały znormalizowane przy użyciu metody `MinMaxScaler` [413]. Szczegółowe wprowadzenie do tego zbioru danych można znaleźć w artykule [415].

„p53 Mutants" [416] – podobnie jak *„Credit Card Fraud"* - zbiór ten został omówiony w podrozdziale 8.1.1. Składa się z cech uzyskanych z symulacji biofizycznych mutantów białka p53, mających na celu modelowanie ich aktywności transkrypcyjnej. Mutanty białka p53 zostały sklasyfikowane jako „rakotwórcze" lub „aktywnie transkrypcyjne", gdzie tylko 0,86% z 16 591 próbek, czyli 143 przypadki, przypisano do kategorii aktywnych. Przed analizą usunięto wiersze z brakującymi danymi, co zmniejszyło liczbę próbek z początkowych 16 772. Aby zoptymalizować zbiór danych do analizy, obliczono macierz korelacji, eliminując cechy o wysokiej korelacji (powyżej 0,5), co zredukowało liczbę kolumn z 5 408 do 444. Kolumny numeryczne zostały znormalizowane przy użyciu metody `MinMaxScaler` [413]. W artykule [398] autorzy wprowadzają nowatorską metodę aktywnego uczenia się MIP (*ang. most informative positive*, MIP), która ma na celu identyfikację regionów w białku p53 zdolnych do odwrócenia jego rakotwórczych mutacji. Badanie to wnosi istotny wkład w pogłębienie wiedzy na temat mechanizmów działania p53, jednego z supresorów nowotworów, którego mutacje są obecne w niemal połowie wszystkich ludzkich przypadków raka. Odkrycia te otwierają nowe horyzonty w rozwoju precyzyjnych terapii nowotworowych, ukierunkowanych na przywrócenie normalnej funkcji p53.

„Labeled Vehicle Claims" [417] - zbiór koncentruje się na roszczeniach ubezpieczeniowych związanych z pojazdami. Celem jest identyfikacja potencjalnych oszustw, błędów lub nieścisłości w procesie ubezpieczeniowym. Zbiór danych szczegółowo analizuje roszczenia ubezpieczeniowe dotyczące pojazdów, łącząc dane kategoryczne i numeryczne, koncentrując się na identyfikacji anomalii. Kolumna *Label* oznacza, czy obiekt jest anomalią, z wartością 1 wskazującą na nietypowe cechy, takie jak godziny pracy, koszty naprawy czy brakujące dane. Kolumna *Category_anomaly* identyfikuje anomalie w danych kategorycznych, takich jak niespodziewane wartości dla koloru, typu nadwozia czy roku rejestracji, również z wartością 1 oznaczającą anomalię. Zbiór został przefiltrowany, by skupić się na istotnych anomaliach. Z 268 255 obiektów wybrano 24 968 z anomaliami w kolumnach *Label* i *Category_anomaly*, a następnie ograniczono do 1 488 kluczowych

przypadków, pozostawiając 212 994 obiekty do analizy. Zbiór danych jest mocno niezbalansowany, z anomaliami stanowiącymi tylko 0,699% wszystkich obiektów. Mniej istotne kolumny zostały usunięte, pozostawiając 19 kluczowych dla analizy. Dane obejmują kategorie, liczby całkowite i zmiennoprzecinkowe, z dużą różnorodnością, np. 65 marek pojazdów, 23 kolory i 17 892 unikalne ceny. Kolumny numeryczne w zbiorze danych zostały znormalizowane przy użyciu metody *MinMaxScaler* [413]. Zbiór danych został opracowany na potrzeby pracy „Unsupervised Anomaly Detection for Auditing Data and Impact of Categorical Encodings", zaprezentowanej na NeurIPS 2022, co podkreśla jego wartość i potencjalne zastosowania w branży ubezpieczeniowej i motoryzacyjnej [418].

„Chess King-Rook vs. King" *(KRKOPT - zero vs. all)* [419] - to popularny zbiór danych dotyczący końcówek szachowych, w których białe mają króla i wieżę, a czarne tylko króla. Celem jest klasyfikacja sytuacji na szachownicy, aby określić, czy białe mogą wymusić mata na czarnych w następnym ruchu. Zbiór danych jest szeroko stosowany w dziedzinie uczenia maszynowego i sztucznej inteligencji jako przykład problemu klasyfikacyjnego w grach strategicznych. Zbiór z 1994 roku zawiera 28 056 obiektów i sześć cech opisujących pozycje szachowe, w tym pozycje białego króla, wieży oraz czarnego króla. Cechy są kategoryczne i liczbowe, obejmują optymalną głębokość zwycięstwa białych, od 0 do 16 ruchów lub remis. Zbiór nie zawiera brakujących wartości. Klasyfikacja wyników obejmuje remisy i zwycięstwa białych, najczęściej w 12, 13 i 14 ruchach. Występuje 27 przypadków, gdzie optymalna głębokość zwycięstwa wynosi zero, co traktowane jest jako anomalia ze względu na rzadkość i trudność w identyfikacji. Kolumny numeryczne zostały znormalizowane metodą *MinMaxScaler* [413]. Istotne jest zrozumienie, że termin „anomalie" w tym kontekście nie odnosi się do tradycyjnych obiektów odstających lub błędów w danych, jak to bywa w klasycznej analizie danych. Zamiast tego, „anomalie" opisują unikalne i rzadkie konfiguracje w problemie szachowym, które różnią się od typowych wzorców. Przypadki, w których białe nie mogą wymusić zwycięstwa w kolejnym ruchu (oznaczone jako „zero"), mają wyjątkowe znaczenie strategiczne i teoretyczne. Te konfiguracje prowadzą do remisu niezależnie od wykonanych ruchów, co odróżnia je od pozostałych, gdzie określona jest optymalna liczba ruchów do zwycięstwa. Taka charakterystyka zbioru danych umożliwia dogłębne badanie scenariuszy końcowych w szachach, dostarczając danych dla rozwoju algorytmów w sztucznej inteligencji i uczeniu maszynowym.

„KDD Cup 1999" [420] - zbiór danych używany w Trzecim Międzynarodowym Konkursie Narzędzi Odkrywania Wiedzy z Danych i Uczenia Maszynowego (KDD-99) [421], który odbył się w ramach Piątej Międzynarodowej Konferencji na temat Odkrywania Wiedzy z Danych i Uczenia Maszynowego [422], jest ważnym źródłem do budowy detektorów ataków sieciowych. Zadaniem konkursu było stworzenie modelu predykcyjnego zdolnego do rozróżnienia między niebezpiecznymi połączeniami, klasyfikowanymi jako włamania lub ataki, a normalnymi, bezpiecznymi połączeniami. Zbiór danych zawiera standardowy zestaw danych audytowych, obejmujący szeroki zakres symulowanych włamań w środowi-

skach sieciowych, zarówno wojskowych, jak i komercyjnych. Zredukowana wersja zbioru (10 proc.), stosowana w badaniach, jest często używana w pracach naukowych. Mimo mniejszej wielkości, nadal stawia wyzwania związane z niezrównoważonymi klasami i powtarzającymi się obiektami, które mogą wpływać na skuteczność modeli wykrywających włamania. Pomimo redukcji, zbiór zachowuje reprezentatywność pełnego zestawu, umożliwiając skuteczne szkolenie i testowanie modeli w dziedzinie wykrywania włamań. Zbiór danych zawiera takie cechy jak rodzaje protokołów, typy usług, wartości flag i inne. Anomalie w tym zbiorze reprezentują różne typy ataków sieciowych. Dane z 494 020 obiektami zostały przefiltrowane, pozostawiając tylko normalne obiekty oraz ataki U2R i R2L, co zredukowało zbiór do 97 577 obiektów. W sumie 97 278 to obiekty normalne, a 299 to ataki (0,306% zbioru). Kolumny numeryczne w zbiorze danych zostały znormalizowane przy użyciu metody *MinMaxScaler* [413]. Ataki U2R (*ang. user to root*), takie jak przepełnienie bufora czy skrypty Perl, umożliwiają przejęcie kontroli przez napastnika nad systemem z uprawnieniami superużytkownika, co może prowadzić do poważnych konsekwencji, takich jak przejęcie systemu czy instalacja złośliwego oprogramowania. Ataki R2L (*ang. remote to local*), takie jak *ftp_write*, *guess_passwd*, *imap*, czy *warezclient*, umożliwiają zdalnym użytkownikom uzyskanie nieautoryzowanego dostępu do systemu, co stanowi ryzyko kradzieży danych lub manipulacji systemem.

„w7a libsvm” [423, 424] - to zbiór danych pochodzący z repozytorium LIBSVM, zawierający 49 749 obiektów, obejmujących dane zarówno testowe, jak i treningowe, z 300 cechami. Zbiór ten został opracowany w wyniku prac Johna Platta z 1998 roku, który wprowadził szybki algorytm trenowania maszyn wektorów nośnych SVM, znany jako SMO (*ang. sequential minimal optimization*, SMO) [425]. Dane te są silnie niezrównoważone pod względem rozkładu klas, gdzie rzadkie klasy są traktowane jako anomalie, a pozostałe jako normalne. Zbiór jest szeroko stosowany w badaniach nad wykrywaniem anomalii w danych kategorycznych oraz w analizie danych o wysokiej liczbie wymiarów, takich jak dane tekstowe. Wykorzystywany jest także w zadaniach klasyfikacji binarnej, gdzie celem jest przewidzenie przynależności próbki do jednej z dwóch klas na podstawie jej cech. Zastosowania tego zbioru danych obejmują dziedziny uczenia maszynowego i sztucznej inteligencji, zwłaszcza badania nad algorytmami klasyfikacyjnymi, takimi jak SVM, oraz zadania związane z przetwarzaniem języka naturalnego i analizą tekstu, gdzie liczba cech jest duża, a ich rzadkość stanowi wyzwanie. Dane w zbiorze są kompletne, bez brakujących wartości czy cech, co pozwala na ich bezpośrednie zastosowanie w eksperymentach uczenia maszynowego, bez potrzeby stosowania imputacji. Mimo że zbiór zawiera tylko wartości 0 i 1, jest klasyfikowany jako kategoryczny, ponieważ te wartości mogą reprezentować różne kategorie. Dane są zapisane w formacie rzadkim, co oznacza, że większość wartości to zera, a zapisywane są jedynie wartości niezerowe. Taki format jest typowy dla zbiorów danych wykorzystywanych w zadaniach klasyfikacji tekstu lub innych danych o wysokiej liczbie wymiarów, gdzie wiele cech nie pojawia się w każdym rekordzie. Zbiór danych początkowo zawierał 48 270 próbek oznaczonych jako 0 (klasyfikowane jako normalne)

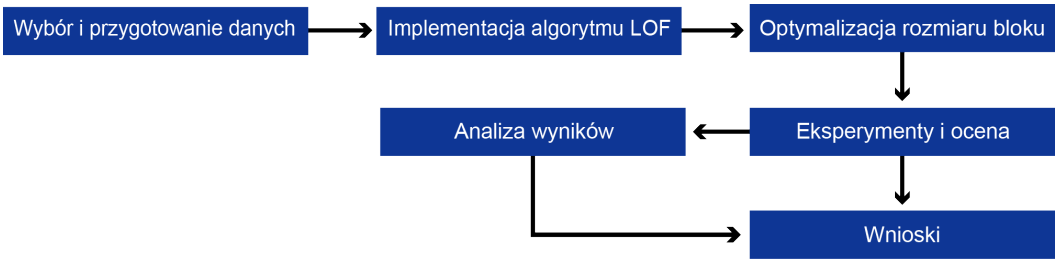
oraz 1 479 próbek oznaczonych jako 1 (klasyfikowane jako anomalie). Po usunięciu 1 073 próbek anomalii, pozostało 406 obserwacji odstających, co stanowi 0,834% całego zbioru. Ze względu na to, że wszystkie rekordy mają wartości binarne, nie było potrzeby stosowania normalizacji przy użyciu metody *MinMaxScaler* [413].

„Covid-19 Case Surveillance” [426, 264, 304] - zbiór danych zawiera szczegółowe informacje dotyczące przypadków Covid-19 zgłoszonych w Stanach Zjednoczonych. Obejmuje demografię, historię narażenia, wskaźniki nasilenia choroby, wyniki kliniczne, dane z badań laboratoryjnych oraz informacje o współistniejących chorobach. Wszystkie te elementy są zgodne z danymi zbieranymi za pomocą formularza raportu przypadków COVID-19, dostępnego na stronie CDC. Anomalie zostały zdefiniowane na podstawie atrybutu *death_yn*, który określa, czy choroba była przyczyną zgonu pacjenta. Początkowy zestaw danych składał się z 8 405 079 obiektów, z których losowo wybrano próbkę 1 miliona obiektów przy użyciu narzędzia *RapidMiner* [427]. Po oczyszczeniu zbioru z brakujących lub pustych danych do analizy wykorzystano zredukowany zestaw 35 212 obiektów. Spośród nich 1 633 obiekty, co stanowi 4,64% zbioru, zostały zidentyfikowane jako anomalie na podstawie cechy *death_yn*. Ostatecznie zachowano 332 anomalie z wartością *death_yn* równą 1, co stanowi 0,924% całego zbioru, czyli około jeden procent wszystkich anomalii. Zbiór zawiera 33 911 obiektów, 10 cech i został poddany przygotowaniu, które rozpoczęło się od konwersji kolumn z datami na odpowiedni format, umożliwiający przeprowadzanie analiz czasowych. Następnie obliczono różnice w dniach między kluczowymi datami, takimi jak data wystąpienia pierwszych objawów a data pozytywnego testu oraz data zgłoszenia przypadku do CDC. Te różnice czasowe pozwoliły lepiej zrozumieć sekwencję wydarzeń. Kolejnym istotnym krokiem było kodowanie danych kategorycznych, gdzie kolumny zostały przekształcone na format numeryczny przy użyciu `LabelEncoder` [428], co umożliwia ich wykorzystanie w dalszej analizie. `LabelEncoder` z biblioteki `scikit-learn` służy do zamiany etykiet tekstowych na liczby całkowite, co upraszcza przetwarzanie danych. Kolumny liczbowe zostały poddane normalizacji przy użyciu `MinMaxScaler` [413].

„Mushroom” [393] - podobnie jak *„Credit Card Fraud”* i *„p53 Mutants”*, zbiór został omówiony w podrozdziale 8.1.1. Jest to całkowicie kategoryczny zbiór, zawierający 8 124 obiekty oraz 22 cechy, które umożliwiają klasyfikację grzybów jako jadalnych lub trujących. W dalszej części analizy skupiono się na przypadkach anomalii, gdzie liczba trujących grzybów została zredukowana do 0,988% próbek, co stworzyło zbiór *„Mushroom Real”*, zawierający 42 rzeczywiste anomalie. Dzięki zastosowaniu one-hot encoding [429], nie było potrzeby normalizacji, co uprościło proces przygotowania danych do analizy. Rzeczywiste anomalie, jak w *„Mushroom Real”*, oferują bardziej autentyczne testy algorytmów, co przekłada się na dokładniejsze wyniki, lepsze dopasowanie modeli do rzeczywistych danych i bardziej wiarygodne prognozy.

Metodyka i proces badań

Badanie składa się z kilku etapów, zaczynając od wstępnej obróbki danych, która obejmuje przekształcenie danych kategorycznych na wektory binarne, normalizację danych liczbowych, uzupełnianie brakujących wartości oraz eliminację silnie skorelowanych cech. Implementacja algorytmu LOF koncentruje się na precyzyjnym dostosowaniu hiperparametrów, takich jak liczba najbliższych sąsiadów (k), wybór odpowiedniej metryki odległości (euklidesowej, kosinusowej, Hamminga) oraz optymalizacji rozmiaru bloku. Metodologia obejmuje zastosowanie technik optymalizacji bayesowskiej (za pomocą `gp_minimize` [409] z biblioteki `scikit-optimize` [407]) w celu eksploracji przestrzeni hiperparametrów i znalezienia optymalnej konfiguracji. Obejmuje to automatyczne dostosowanie rozmiaru bloku danych oraz innych znaczących hiperparametrów algorytmu. Seria eksperymentów przeprowadzonych przy różnych konfiguracjach rozmiaru bloku i hiperparametrach koncentruje się na analizie czasu wykonania algorytmu, skuteczności w identyfikacji anomalii oraz obliczaniu sumy odległości w celu oceny zwartej struktury bloku danych. Badanie wykorzystuje specjalnie zaprojektowaną funkcję celu, dążącą do jednoczesnej optymalizacji tych trzech aspektów.



Rysunek 8.9: Schemat blokowy metodologii optymalizującej wykrywanie anomalii. Źródło: opracowanie własne.

Analiza wyników uzyskanych dla różnych zbiorów danych pozwala na wyciągnięcie wniosków dotyczących uniwersalności i ograniczeń proponowanego podejścia. Schemat blokowy przedstawiający przebieg całego procesu zilustrowano na rysunku 8.9. Implementacja algorytmu została wykonana w języku Python z wykorzystaniem bibliotek `NumPy` [297], `SciPy` [391] oraz `scikit-optimize` [407]. Z biblioteki `SciPy` wykorzystano funkcję `cdist` [390] do szybkiego wyznaczania odległości w przestrzeni wielowymiarowej. Zamiast tworzyć podobne mechanizmy niezależnie, zdecydowano się na użycie gotowej, wydajnej metody `cdist` opartej na optymalizacjach w języku C. Dzięki modułowi `time` [430] możliwe było precyzyjne śledzenie czasu wykonania algorytmu. W badaniu wykorzystano standardową wersję algorytmu bez żadnych modyfikacji, umożliwiając jedynie dodatkowy podział na bloki. Eksperymenty przeprowadzono w środowisku Google Colab Pro+ z 52 GB pamięci RAM.

Eksperymenty i analiza wyników

Optymalizacja bayesowska wykorzystuje teorię prawdopodobieństwa do modelowania niepewności przy ocenie funkcji celu. W kontekście optymalizacji bayesowskiej, równanie (8.2) opisujące wartość funkcji celu *wynik* jest ważne dla zrozumienia, jak różne aspekty algorytmu są oceniane i równoważone w celu osiągnięcia optymalnej konfiguracji. W ramach badań zoptymalizowano funkcję celu dla wszystkich dziewięciu zestawów danych, wykorzystując zakres parametrów przedstawiony w tabeli 8.3. Ten zakres parametrów był konsekwentnie stosowany we wszystkich eksperymentach. Zaproponowane równanie (8.3) automatycznie określa rozmiar bloku, dostosowując go do specyfiki analizowanych danych i dostępnych zasobów sprzętowych.

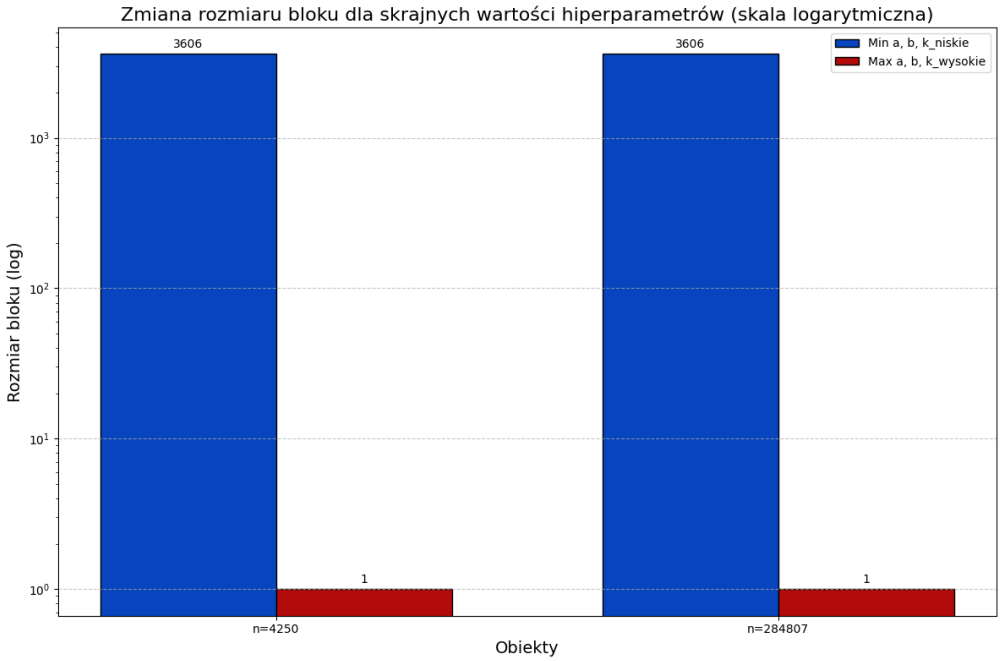
Tabela 8.3: Parametry optymalizacji funkcji celu dla badanych zbiorów (seed = 3). Źródło: opracowanie własne.

WtCov	WtETm	WtSumD	Hiperparametry	n_calls
1,0	0,001	0,000000001	[(1 - 250), (0,1 - 1,9), (0,1 - 1,9), (0 - 2), (0,9 - 1,0 - 1,1)]	do 200

W kolumnie Hiperparametry lista reprezentuje kolejno (k, a, b, indeks metryki, c). Hiperparametr k określa liczbę najbliższych sąsiadów. Parametry a i b są specyficzne dla modelu i wpływają na skalowanie danych. Indeks metryki wskazuje metodę obliczania odległości, gdzie 0 oznacza metodę Euklidesa, 1 - kosinusową, 2 - Hamminga. Parametr c wpływa na rozmiar bloku danych - im wyższa wartość, tym większy wpływ liczności danych n na ostateczny rozmiar bloku. Natomiast n_calls to parametr używany w optymalizacji bayesowskiej, określający liczbę wywołań funkcji celu. WtCov, WtETm i WtSumD reprezentują wagi składników funkcji celu, takich jak procentowe pokrycie, czas wykonania i suma odległości.

Rysunek 8.10 ilustruje wpływ skrajnych wartości hiperparametrów na rozmiar bloku w zależności od liczby obiektów (n). W przypadku minimalnej liczby obiektów ($n = 4250$) oraz maksymalnej ($n = 284807$) widać wyraźne różnice w rozmiarach bloków. Niebieskie słupki odzwierciedlają scenariusz, w którym wartości hiperparametrów a i b są minimalne, a wartość k jest niska, co prowadzi do większych rozmiarów bloków. Te hiperparametry ograniczają redukcję rozmiaru bloku, skutkując większą przestrzenią przeznaczoną na każdy blok. Z kolei czerwone słupki ukazują sytuację, w której wartości hiperparametrów a i b są maksymalne, a wartość k wysoka, co skutkuje mniejszymi rozmiarami bloków danych. Takie ustawienie hiperparametrów sprzyja znacznemu zmniejszeniu rozmiaru bloku. Różnice między rozmiarami bloków w obu scenariuszach są znaczące, co podkreśla, jak bardzo rozmiar bloku może być dostosowany do specyficznych potrzeb analizy,

w zależności od wybranych hiperparametrów. To dostosowanie pokazuje elastyczność algorytmu, który może być precyzyjnie skonfigurowany, aby optymalnie zarządzać zasobami obliczeniowymi i dostosować się do różnych danych. Jest to ważne w praktycznych zastosowaniach, gdzie efektywność przetwarzania danych ma duże znaczenie.



Rysunek 8.10: Zmiana rozmiaru bloków na skrajnych wartościach hiperparametrów. Źródło: opracowanie własne.

Tabela 8.4 prezentuje wyniki oszacowania optymalnych rozmiarów bloków dla różnych zbiorów danych, analizowanych w ramach badania. Każdy wiersz odnosi się do konkretnego zbioru danych, a specyfikacje obejmują:

- nazwa zbioru danych - określa analizowany zbiór,
- rozmiar (n) - liczba obiektów w zbiorze danych,
- hiperparametry a, b, k, c - wartości użyte w procesie optymalizacji,
- RAM - ilość pamięci operacyjnej użytej do przetwarzania danych,
- rozmiar bloku (BS) - oszacowany optymalny rozmiar bloku, ważny dla efektywności pamięciowej i obliczeniowej.

Na przykład, dla zbioru danych *„Credit Card”*, składającego się z 284 807 obiektów, przy dostępnych 52 GB pamięci RAM, hiperparametry a, b, k i c zostały dobrane zgodnie z przedstawionymi wartościami, a optymalny rozmiar bloku został oszacowany na 95.

Tabela 8.4: Optymalna estymacja rozmiaru bloku. Źródło: opracowanie własne.

Zbiór	Rozm. (n)	a	b	k	c	RAM	BS
Credit Card	284 807	0.1086	0.6983	50	1.0	52GB	95
Wine Quality	4 898	0.0858	0.0907	16	1.0	52GB	1638
p53 Mutants	16 591	0.2388	0.5805	1	1.0	52GB	368
Vehicle Claims	212 994	0.1000	0.3000	10	0.9	52GB	597
Chess KRvsK.	28 056	0.0112	0.0107	18	1.0	52GB	28056 (n)
KDD Cup ’99	97 577	0.4870	0.1484	29	1.1	52GB	119
w7a libsvm	48 676	0.2115	0.2103	15	1.1	52GB	269
Covid-19	33 911	0.1000	0.1158	210	1.0	52GB	304
Mushroom (R)	4 250	0.1000	0.1000	8	1.0	52GB	1803

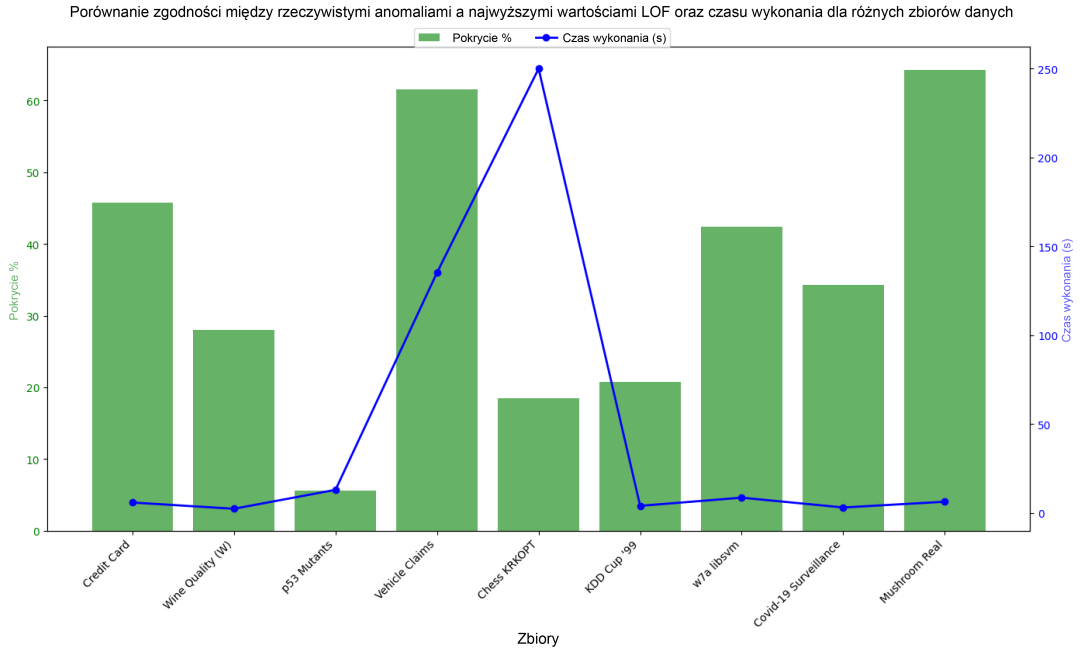
Analiza porównawcza procentowego pokrycia oraz czasu wykonania, przedstawiona na rysunku 8.11, dostarcza istotnych spostrzeżeń na temat skuteczności i wydajności algorytmu LOF z optymalizowanym rozmiarem bloku oraz innymi hiperparametrami w wykrywaniu anomalii w różnych zbiorach danych. Szczegółowe wizualizacje dla zbiorów *„Vehicle Claims”* (rysunek 8.13), *„w7a libsvm”* (rysunek 8.14) oraz *„Mushroom Real”* (rysunek 8.15 skrótowo *„Mushroom (R)”* w tabelach), w połączeniu z wynikami zawartymi w tabeli 8.5, ujawniają pewne prawidłowości. Podział zbioru danych na mniejsze bloki znacząco przyspiesza przetwarzanie, co jest istotne przy pracy z dużymi zbiorami danych. Chociaż przetwarzanie całego zbioru danych w jednym przebiegu mogłoby teoretycznie

Tabela 8.5: Podsumowanie wyników eksperymentów. Źródło: opracowanie własne.

Zbiór	Rozm.	Anom.	Anom. %	Pokry.%	Czas (s)	k	Metr.	BS
Credit Card	284 807	492	0,172%	45,73%	5,77	50	eucl.	95
Wine Quality	4 898	25	0,51%	28,00%	2,26	16	cosine	1638
p53 Mutants	16 591	143	0,86%	5,59%	12,91	1	hamm.	368
Vehicle Claims	212 994	1488	0,699%	61,49%	135,19	10	hamm.	597
Chess KRvsK.	28 056	27	0,096%	18,51%	249,90	18	cosine	28 056
KDD Cup ’99	97 577	299	0,306%	20,73%	3,90	29	cosine	119
w7a libsvm	48 676	406	0,834%	42,36%	8,50	15	cosine	269
Covid-19	33 911	332	0,924%	34,33%	2,98	210	hamm.	304
Mushroom (R)	4 250	42	0,988%	64,28%	6,21	8	hamm.	1803

prowadzić do lepszych wyników w wykrywaniu anomalii, praktyczne ograniczenia czasowe i pamięciowe czynią takie podejście nieefektywnym. W niektórych przypadkach przetwarzanie całego zbioru danych naraz byłoby wręcz niemożliwe z powodu dostępnych zasobów pamięci lub długiego czasu oczekiwania na wynik, nawet jeśli te zasoby byłyby dostępne. Optymalizacja rozmiaru bloku pozwala na znalezienie optymalnej równowagi między czasem przetwarzania a skutecznością, co sprawia, że jest to bardziej praktyczna

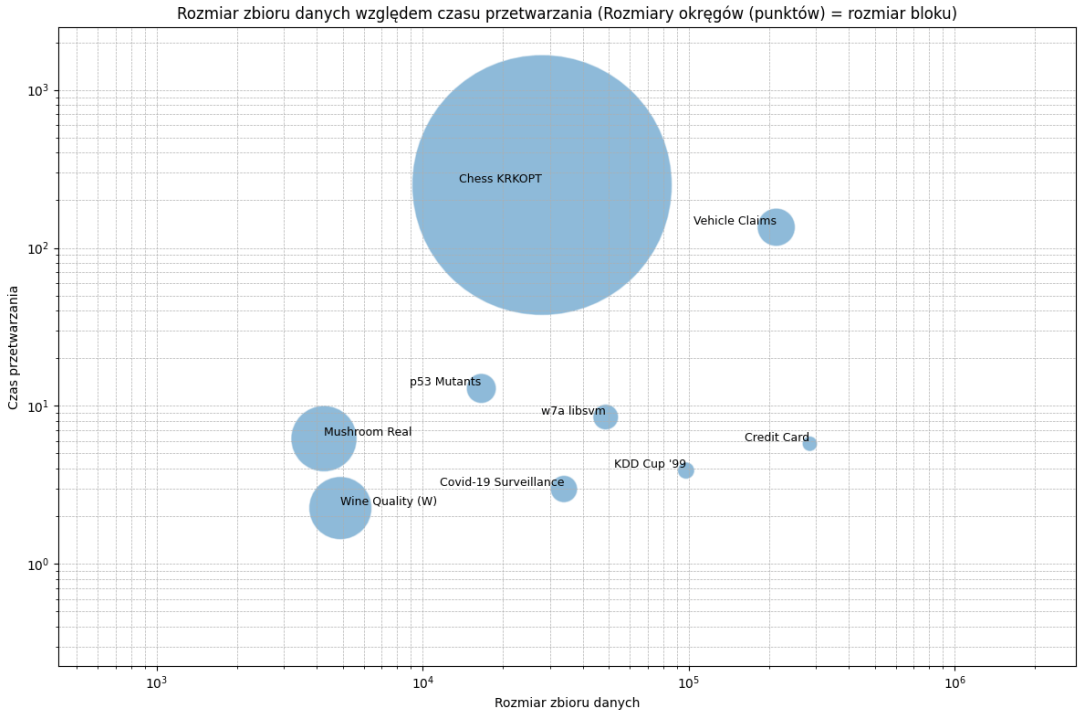
i efektywna strategia w wielu rzeczywistych zastosowaniach. Choć algorytm LOF został pierwotnie zaprojektowany z myślą o analizie danych numerycznych, badania wykazały, że dzięki technice kodowania one-hot encoding [429], może on również skutecznie przetwarzać dane kategoryczne. Ta zdolność adaptacji rozszerza zakres potencjalnych zastosowań algorytmu na różnorodne typy danych.



Rysunek 8.11: Porównanie pokrycia i czasu wykonania dla różnych zbiorów danych. Źródło: opracowanie własne.

Dalsze wnioski wyciągnięte z podsumowania eksperymentalnego przedstawionego w tabeli 8.5 są zgodne z początkowymi obserwacjami, podkreślają zdolność adaptacyjną algorytmu LOF do różnorodnych danych oraz znaczenie precyzyjnego doboru parametrów w celu optymalizacji wydajności. Rysunek 8.11 przedstawia zależność między procentem pokrycia a czasem wykonania dla różnych zbiorów danych analizowanych za pomocą algorytmu LOF. Zielone słupki ilustrują procent pokrycia (Pokrycie%) dla każdego zbioru, pokazując, jak skutecznie algorytm identyfikuje anomalie w poszczególnych przypadkach. Niebieska linia ukazuje czas wykonania, obrazując, jak wydajność algorytmu zmienia się w zależności od rozmiaru i złożoności zbiorów danych. Z analizy wynika, że czas przetwarzania dla większości zbiorów pozostaje niski, niezależnie od ich wielkości, szczególnie w przypadku zbiorów podzielonych na mniejsze bloki. Ciekawym zjawiskiem jest to, że tak duży zbiór danych jak *„Credit Card”* (284 807 obiektów) ma szybszy czas przetwarzania niż niektóre mniejsze zbiory, co można przypisać efektywnemu podziałowi danych na mniejsze bloki. Wyjątek stanowi zbiór *„Chess KRKOPT”* (skrócony w tabelach

jako *„Chess KRvsK.”*), gdzie czas wykonania jest znacznie dłuższy. W tym przypadku blok danych nie został podzielony, ponieważ optymalizacja bayesowska nie znalazła lepszego pokrycia przy podziale na mniejsze segmenty, co spowodowało konieczność analizy całego zbioru jako jednego bloku, znacząco wydłużając czas przetwarzania. Zauważalna korzyść z podziału zbiorów na mniejsze segmenty podkreśla znaczenie optymalizacji rozmiaru bloku w praktycznych zastosowaniach. Brak podziału, jak w przypadku *„Chess KRKOPT”*, prowadzi do nieefektywnego wydłużenia czasu wykonania, co może być nieopłacalne.



Rysunek 8.12: Rozmiar zbioru danych a czas przetwarzania (Rozmiar punktów = Rozmiar bloku). Źródło: opracowanie własne.

Rysunek 8.12 uzupełnia te obserwacje, ilustrując wpływ rozmiaru bloku na czas przetwarzania. Oś X przedstawia rozmiar zbioru (n) w liczbie obiektów, a oś Y czas przetwarzania w sekundach, obie w skali logarytmicznej, co umożliwia porównanie zbiorów o różnych wielkościach. Niebieskie kółka reprezentują poszczególne zbiory danych, a ich wielkość odpowiada rozmiarowi bloku (BS) użytemu w analizie. Rysunek wyraźnie pokazuje, że choć zbiory takie jak *„Credit Card”* i *„Vehicle Claims”* są duże, ich czas przetwarzania pozostaje stosunkowo krótki dzięki efektywnemu podziałowi na mniejsze bloki. W przeciwieństwie do nich, zbiór *„Chess KRKOPT”*, mimo mniejszego rozmiaru, wymaga znacznie dłuższego czasu przetwarzania z powodu braku podziału na bloki, co wymusza analizę całego zbioru jako jednego bloku. To podkreśla znaczenie rozmiaru bloku w optymalizacji czasu przetwarzania. Z kolei mniejsze zbiory, takie jak *„Wine Quality”*

i *„Mushroom Real”*, mimo większych bloków, przetwarzane są szybko dzięki swoim niewielkim rozmiarom. Pokazuje to, że czas przetwarzania zależy zarówno od rozmiaru bloku, jak i wielkości zbioru. Jednak nawet małe zbiory mogą wymagać znacznego czasu na przetwarzanie, gdy przypisane im zostaną duże bloki lub gdy są analizowane w całości.

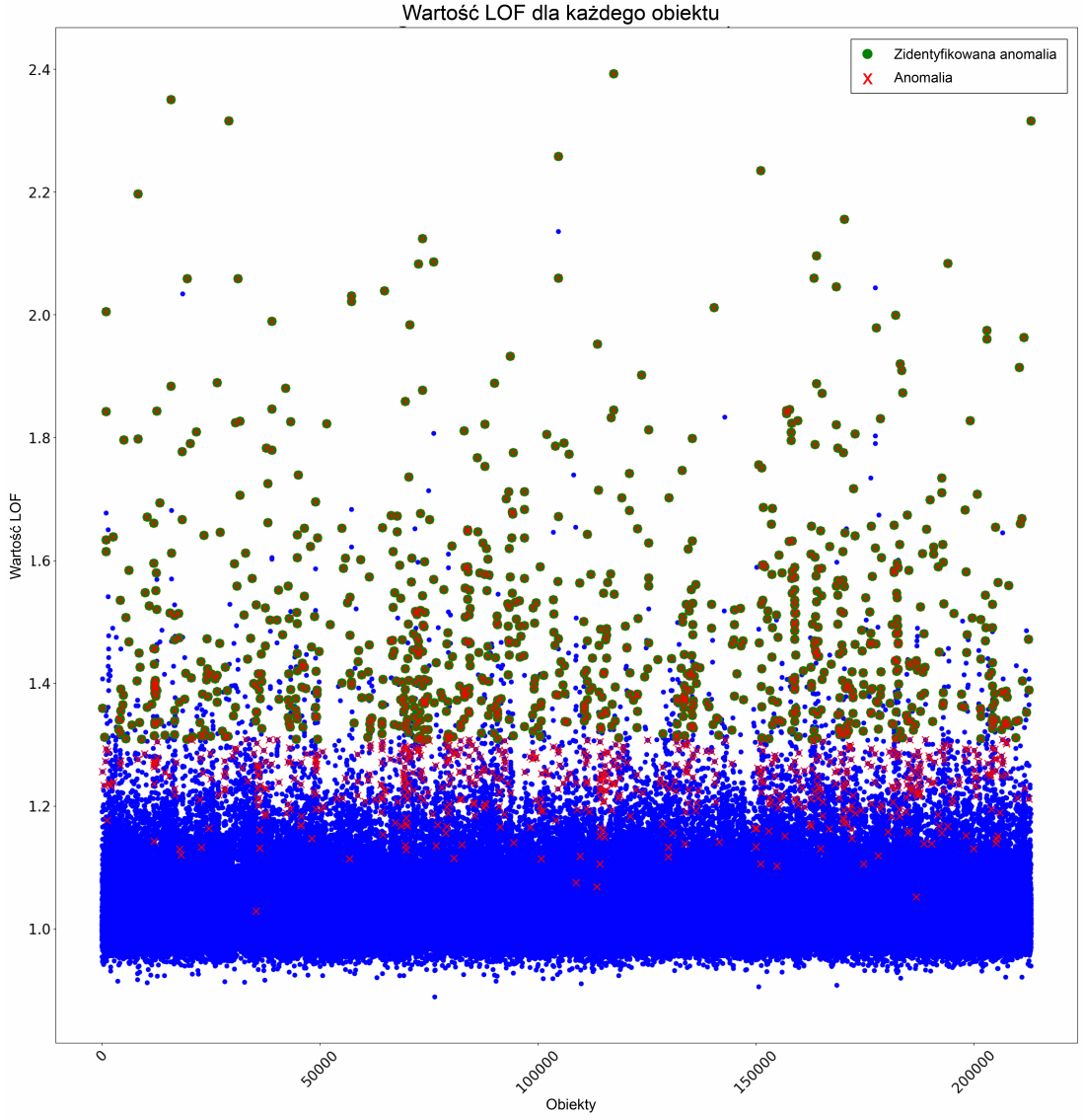
Wizualizacje na rysunkach 8.13, 8.14 oraz 8.15 prezentują szczegółowe wyniki działania algorytmu LOF dla wybranych zbiorów danych, umożliwiając dokładną ocenę jego skuteczności i wydajności. Wykryte anomalie są oznaczone zielonymi kropkami z czerwonymi krzyżykami, natomiast obiekty uznane za „normalne” przedstawiono jako niebieskie kropki. Niewykryte anomalie zostały oznaczone jedynie czerwonym krzyżykiem, co wyraźnie pokazuje obszary, w których algorytm nie zdołał poprawnie zidentyfikować anomalii. Na rysunku 8.13 przedstawiono wyniki dla zbioru danych *„Vehicle Claims”*, gdzie spośród 1 488 anomalii algorytm LOF zidentyfikował 915, co przekłada się na pokrycie na poziomie 61,49%. Optymalizacja przeprowadzona z parametrami $k = 10$, metryką odległości Hamminga oraz rozmiarem bloku 597 wykazała czas wykonania wynoszący 135,19 sekundy. Rysunek 8.14 przedstawia wyniki dla zbioru danych *„w7a libsvm”*, w którym algorytm LOF zidentyfikował 172 z 406 anomalii, co przekłada się na pokrycie rzędu 42,36%. Optymalizacja algorytmu została przeprowadzona przy użyciu parametrów $k = 15$, metryki odległości kosinusowej oraz rozmiaru bloku wynoszącego 269, co pozwoliło osiągnąć krótki czas wykonania, wynoszący 8,5047 sekundy. Rysunek 8.15 przedstawia wyniki dla zbioru danych *„Mushroom Real”*, zawierającego 42 anomalie, z których algorytm zidentyfikował 27, osiągając pokrycie na poziomie 64,28%. Zastosowanie parametrów $k = 8$, metryki odległości Hamminga oraz rozmiaru bloku 1 803 skutkowało czasem wykonania wynoszącym 6,2080 sekundy. Te wizualizacje ukazują wpływ doboru parametrów algorytmu, takich jak liczba najbliższych sąsiadów, metryka odległości i rozmiar bloku, na skuteczność oraz czas wykonania algorytmu LOF. Różnice w procentowym pokryciu i czasie wykonania dla różnych zbiorów danych podkreślają znaczenie precyzyjnego dostosowania tych parametrów do specyfiki każdego zbioru, aby zoptymalizować wydajność algorytmu w detekcji anomalii.

Podsumowując wyniki eksperymentów i analizując wpływ optymalizacji rozmiaru bloku na wydajność algorytmu LOF, można sformułować kilka istotnych wniosków, które mają znaczenie dla procesu wykrywania anomalii w dużych zbiorach danych. Optymalizacja rozmiaru bloku, jako hiperparametr, odgrywa znaczącą rolę w zwiększaniu efektywności i skuteczności algorytmu LOF. Odpowiednio dobrany rozmiar bloku pozwala na znaczące skrócenie czasu wykonania, jednocześnie zachowując wysoką precyzję w identyfikacji anomalii. Optymalny rozmiar bloku umożliwia algorytmowi efektywne skupienie obliczeń na mniejszych, łatwiejszych do przetworzenia segmentach danych, co prowadzi do przyspieszenia przetwarzania oraz bardziej oszczędnego wykorzystania zasobów pamięciowych. Eksperymenty wyraźnie pokazują, że dynamiczne dostosowanie rozmiaru bloku do charakterystyki danych oraz dostępnych zasobów sprzętowych stanowi ważny element w elastycznym dostosowywaniu procesu wykrywania anomalii, co ma szcze-

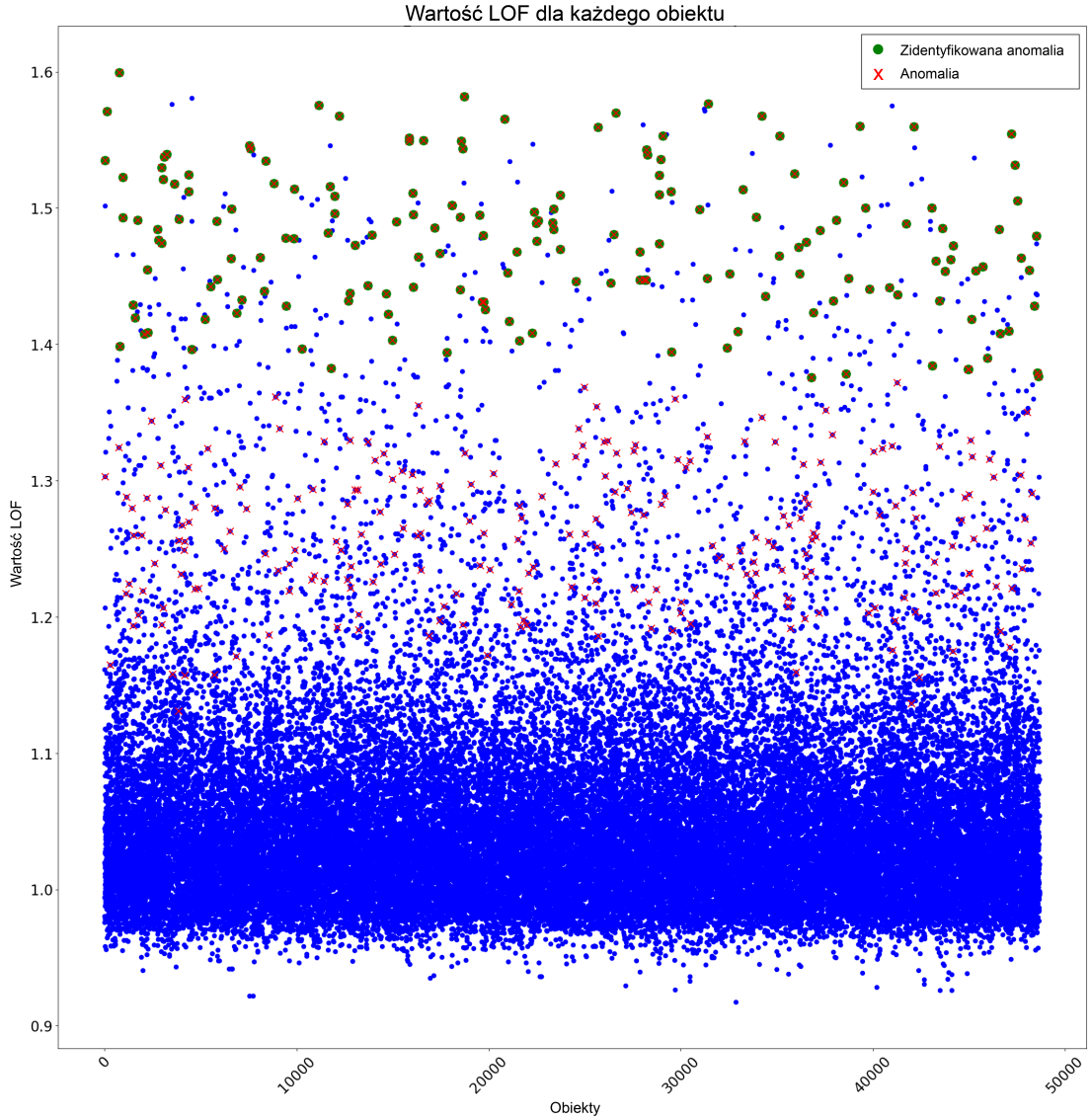
gólne znaczenie przy pracy z dużymi, zróżnicowanymi zbiorami danych. Zastosowanie optymalizacji rozmiaru bloku wykazało zdolność do znacznego skrócenia czasu działania algorytmu LOF, bez istotnego obniżenia jego skuteczności w wykrywaniu anomalii. Wynik ten podkreśla potencjał optymalizacji rozmiaru bloku jako efektywnego rozwiązania dla problemów związanych z analizą dużych zbiorów danych. Badania również potwierdzają, że po optymalizacji algorytm LOF zachowuje swoją wszechstronność, skutecznie działając zarówno na danych numerycznych, jak i kategorycznych. Różnice w pokryciu procentowym i czasie wykonania pomiędzy różnymi zbiorami danych wskazują na konieczność indywidualnego dostosowania parametrów, w tym rozmiaru bloku, co podkreśla wagę elastyczności w procesie optymalizacji. Mimo że optymalizacja rozmiaru bloku znacząco zwiększa wydajność algorytmu LOF, istnieje potrzeba dalszych badań i eksperymentów, aby dokładniej zbadać wpływ tej techniki na różne aspekty wykrywania anomalii, w tym jej zastosowanie w kontekście innych algorytmów i scenariuszy.

Ostatecznie, optymalizacja rozmiaru bloku w algorytmie LOF okazuje się być potężnym narzędziem, które pozwala znacząco przyspieszyć wykrywanie anomalii, zapewniając jednocześnie efektywne wykorzystanie zasobów obliczeniowych oraz minimalizację czasu przetwarzania. Badania jednoznacznie wskazują na istotność tej techniki w analizie dużych zbiorów danych, otwierając tym samym nowe możliwości dla przyszłych badań i rozwoju w dziedzinie detekcji anomalii. Szczególnym wyzwaniem pozostaje znalezienie optymalnej równowagi pomiędzy dokładnością detekcji anomalii a szybkością przetwarzania. Optymalizacja rozmiaru bloku, z wykorzystaniem funkcji celu *wynik* (8.2), umożliwia precyzyjne dostosowanie działania algorytmu do specyficznych wymagań zadania, maksymalizując skuteczność przy jednoczesnym ograniczeniu zużycia zasobów obliczeniowych. Dynamika procesu optymalizacji bayesowskiej, przedstawiona na przykładzie zbioru danych *„w7a libsvm”* (rysunek 8.4), ukazuje, jak algorytm adaptuje się do zmieniających się warunków. Tego rodzaju dostosowanie nie tylko zwiększa wydajność, ale także poprawia jakość wykrywania anomalii, dostarczając bardziej wiarygodnych wyników. Wybór wag dla poszczególnych komponentów funkcji celu ma decydujący wpływ na ukierunkowanie algorytmu na określone priorytety, takie jak szybkość działania czy precyzja wykrywania. Implementacja równania (8.3), które automatycznie oblicza rozmiar bloku na podstawie dostępnej pamięci RAM, pokazuje nowoczesne podejście do zarządzania zasobami, umożliwiające efektywne wykorzystanie sprzętu i minimalizujące ryzyko przeciążenia systemu. Odkrycia dotyczące tej optymalizacji oraz zastosowania funkcji celu *wynik* stwarzają nowe możliwości do dalszych badań, w tym eksploracji alternatywnych metod optymalizacji, eksperymentowania z różnymi algorytmami oraz dostosowywania parametrów funkcji celu, co może prowadzić do jeszcze większych usprawnień w przetwarzaniu danych. Wykorzystanie optymalizacji bayesowskiej w tym kontekście podkreśla, jak zaawansowane techniki matematyczne i statystyczne mogą być zastosowane do rozwiązywania rzeczywistych problemów w nauce o danych. Metoda ta nie tylko ulepsza istniejące procesy, ale także oferuje nowe perspektywy dla przyszłych badań i rozwoju w tej dziedzinie. Te wnioski

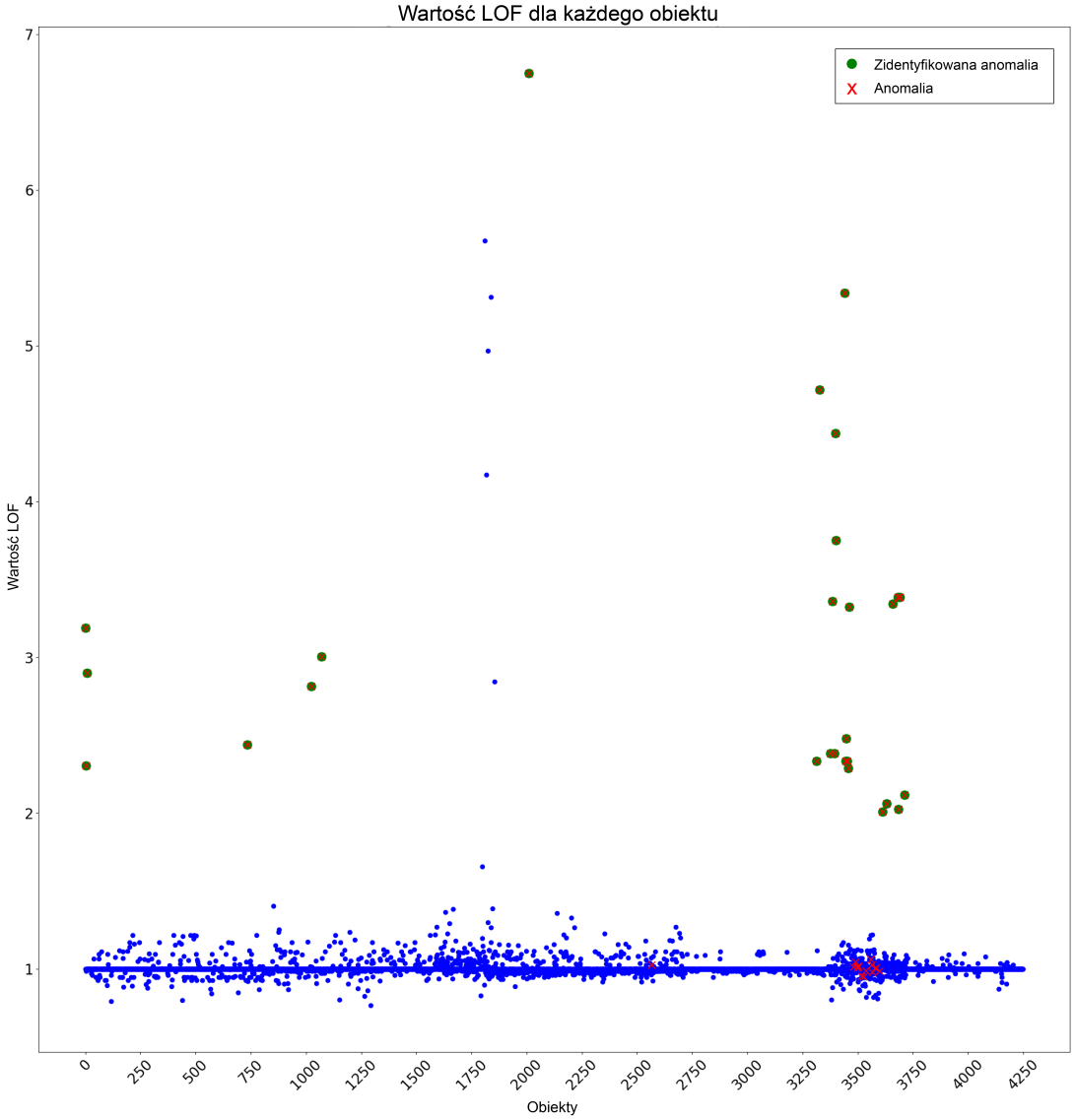
ukazują znaczenie zaawansowanych metod optymalizacji i statystyki w analizie danych, co może znacząco wpłynąć na skuteczność i efektywność wykrywania anomalii.



Rysunek 8.13: Najwyższe wartości LOF dla przykładowego zbioru danych *„Vehicle Claims”*, obejmującego 212 994 obiektów, w tym 1 488 anomalii, co odpowiada odsetkowi anomalii wynoszącemu 0,699%. Spośród nich z powodzeniem zidentyfikowano 915 obiektów odstających, co daje **pokrycie na poziomie 61,49%**. Liczba niezidentyfikowanych anomalii wyniosła 573. Optymalizację przeprowadzono z następującymi parametrami: $k = 10$, metryka odległości: Hamming oraz **rozmiar bloku wynoszący 597**, co dało **czas wykonania równy 135,19 sekundy**. Źródło: opracowanie własne.



Rysunek 8.14: Dla zbioru danych *„w7a libsvm”*, składającego się z 48 676 obiektów, w tym 406 anomalii (0,834%), algorytm LOF zidentyfikował 172 anomalie, co przekłada się na **pokrycie na poziomie 42,36%**. Pozostałe 234 anomalii nie zostały wykryte. Optymalizację wykonano z parametrami: $k = 15$, metryka odległości: kosinusowa oraz **rozmiar bloku: 269**, co skutkowało **czasem wykonania wynoszącym 8,5047 sekundy**. Źródło: opracowanie własne.



Rysunek 8.15: Dla zbioru danych *„Mushroom Real”*, zawierającego 4 250 obiektów, z czego 42 to anomalie (0,988%), algorytm LOF zidentyfikował 27 anomalii, osiągając tym samym **pokrycie na poziomie 64,28%**. Pozostałe 15 anomalii nie zostało wykryte. Proces optymalizacji został przeprowadzony przy parametrach: $k = 8$, metryka odległości Hamminga oraz **rozmiar bloku: 1 803**, co zaowocowało **czasem wykonania wynoszącym 6,2080 sekundy**. Źródło: opracowanie własne.

8.2 Eksperymenty i ocena zespołu Trinity SALT

Zespół detekcji anomalii to metaalgorytm, którego celem jest zwiększenie stabilności i poprawności identyfikacji anomalii poprzez łączenie wyników różnych algorytmów. Stabilność oznacza, że wyniki detekcji anomalii pozostają spójne i powtarzalne, nawet w przypadku pewnych zmian w danych wejściowych. System powinien być w stanie wykrywać anomalie w podobny sposób, niezależnie od zmienności tych danych. W analizie anomalii można wyróżnić dwa główne typy zespołów [27]:

- **zespoły sekwencyjne** - w tych zespołach algorytmy są uruchamiane jeden po drugim, a wyniki wcześniejszych uruchomień wpływają na kolejne kroki. Oznacza to, że wybory dotyczące danych lub algorytmów mogą być modyfikowane w oparciu o wyniki poprzednich uruchomień,
- **niezależne zespoły** - w przypadku niezależnych zespołów, jak pokazano w pseudokodzie 5, różne algorytmy lub różne instancje tego samego algorytmu działają na całych danych lub ich częściach, niezależnie od wyników poprzednich uruchomień. Każdy algorytm niezależnie identyfikuje anomalie, a następnie wyniki są łączone. Można to zrobić na różne sposoby, takie jak obliczenie średniej wyników, sumowanie, wybieranie wartości maksymalnych lub zastosowanie innych technik agregacji. W przeprowadzonych eksperymentach wykorzystano właśnie ten typ zespołu, dlatego ogólny schemat jego działania zademonstrowano w algorytmie 5.

Wybrane podejście powinno umożliwić uzyskanie bardziej wiarygodnych i stabilnych wyników, ponieważ błędy jednego algorytmu mogą zostać skompensowane przez inne w zespole. Techniki zespołowe zostały szczegółowo omówione w osobnym rozdziale 5.

Algorytm 5: Niezależne zastosowanie algorytmów do analizy anomalii na zbiorze danych

Input: Zbiór danych D, lista algorytmów {A1, A2, ..., Ar}, liczba iteracji n
Output: Wykryte anomalie, wyniki na zbiorze testowym

```
Podziel D na zbiór treningowy D_train (70%) i zbiór testowy D_test (30%)
for każdy algorytm m z listy {A1, A2, ..., Ar} do
    for iterację j od 1 do n do
        Utwórz nowy zbiór danych f_j(D_train) jako 70% danych szkoleniowych
          z D_train
        Zastosuj algorytm m do f_j(D_train)
        ▷ Ocena wyników algorytmu m na zbiorze testowym. Oceń algorytm m na
        zbiorze testowym D_test i zapisz wyniki
return Wyniki wykrytych anomalii na zbiorze treningowym oraz wyniki na zbiorze
  testowym
```

Celem podrozdziału jest przedstawienie wyników eksperymentów przeprowadzonych na zespole detekcji anomalii Trinity SALT. Celowo dobrano trzy różne typy algorytmów: klasyczny algorytm SOM oparty na błędzie kwantyzacji, autoenkoder AE wykorzystujący metodę błędu rekonstrukcji oraz LOF, który opiera się na wartości współczynnika osobliwości. Łączenie algorytmów tego samego typu nie jest optymalne, dlatego najlepsze rezultaty powinno się osiągnąć, łącząc algorytmy różnych typów.

Opis zastosowanych algorytmów bazowych można znaleźć w podrozdziale 4.5. Algorytm LOF, który jest jedną z metod bazowych, wykorzystuje podział na bloki, co zostało obszernie omówione we wcześniejszych eksperymentach w podrozdziale 8.1. W rozdziale 7 opisano aplikację webową, zaimplementowaną w celu wsparcia realizacji eksperymentów oraz w przyszłości przydatną dla innych użytkowników do wykorzystania w identyfikacji anomalii. Aplikacja webowa Trinity SALT wspiera proces analizy, weryfikując wyniki algorytmów na zbiorach treningowym i testowym oraz oceniając model na podstawie wskaźników takich jak miara F1, czułość, precyzja i dokładność. Wyniki są przedstawiane użytkownikowi w formie tabel i wykresów. Aplikacja umożliwia także konfigurację hiperparametrów dla każdego z bazowych algorytmów (SOM, AE, LOF) oraz przedstawia wyniki analizy zarówno dla poszczególnych algorytmów, jak i dla całego zespołu. Użytkownik ma możliwość ręcznego wskazania wartości rozmiaru bloku dla algorytmu LOF, a w przypadku braku takiej wartości, aplikacja automatycznie proponuje optymalną. Dzięki wbudowanym narzędziom, takim jak macierz pomyłek, krzywe ROC oraz krzywe precyzji-czułości, użytkownik może przeprowadzić kompleksową ocenę.

Wybór optymalnych hiperparametrów jest czasochłonnym procesem, przeprowadzanym w oddzielnym środowisku Google Colab Pro+, przy użyciu specjalnie przygotowanego kodu, wykorzystującego metody optymalizacji bayesowskiej. Optymalizacja bayesowska oraz biblioteka `gp_minimize` [409], z której korzystano, zostały omówione w ramach eksperymentów opisanych we wcześniejszym podrozdziale 8.1, związanym z optymalizacją rozmiaru bloku i algorytmem LOF.

Każdy detektor bazowy można połączyć z technikami zespołowymi, takimi jak zmienne próbkowanie, które umożliwia eksplorację szerszej przestrzeni parametrów poprzez tworzenie różnych podzbiorów danych oraz eksplorację podprzestrzeni poprzez trenowanie modelu na różnych kombinacjach cech. Możliwe jest także połączenie obu tych technik, aby uzyskać jeszcze lepsze rezultaty. W niniejszych eksperymentach zastosowano jednak inne podejście.

Strategia zastosowana w systemie Trinity SALT to forma stacking-u, zwana kontaminacją. Polega ona na zastosowaniu ograniczonej liczby silnych modeli, z których każdy posiada unikalne właściwości pozwalające dostosować się do złożonej struktury danych. Zamiast próbkowania, przeanalizowano pełne zbiory danych (14 różnych zestawów), a wyniki analizowano za pomocą aplikacji webowej Trinity SALT, której nazwa, podobnie jak nazwa zespołu detektorów, pochodzi od akronimu **S**OM-**A**E-**L**OF-**T**riDetect. Aby ocenić skuteczność algorytmów LOF, AE i SOM na szerokim zakresie parametrów, zastosowa-

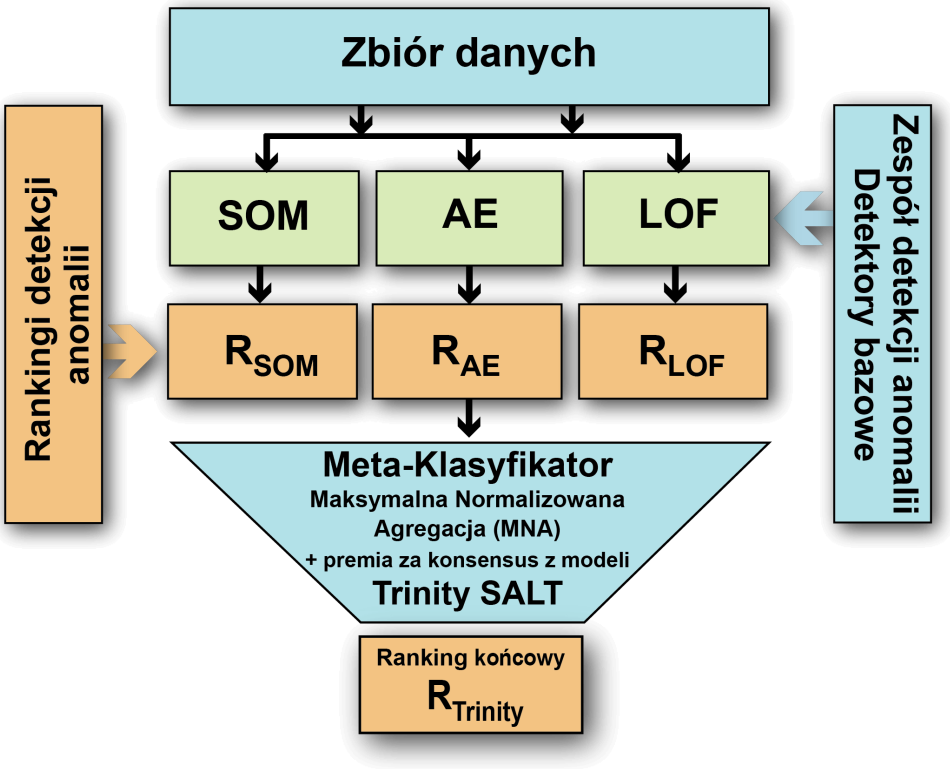
no algorytm 6. Podstawowe cechy i właściwości algorytmów zostały podsumowane na schemacie przedstawionym na rysunku 7.1 w rozdziale 7.

Algorytm 6: Ocena algorytmów bazowych SOM, AE i LOF na szerokim zakresie hiperparametrów z użyciem miary F1 i mediany

Input: Zbiór danych D, lista modeli bazowych {SOM, AE, LOF}, lista hiperparametrów dla każdego modelu, liczba iteracji $n\_calls = 301$
Output: Optymalne hiperparametry dla każdego modelu, wyniki na zbiorze testowym

Podziel D na zbiór treningowy D_{train} (70%) i zbiór testowy D_{test} (30%)
for *każdy model m z listy {SOM, AE, LOF}* **do**
  for *każdą kombinację hiperparametrów param dla modelu m przy użyciu optymalizacji bayesowskiej (gp_minimize) z n_calls = 301* **do**
    ▷ Optymalizacja przeprowadzana jest w środowisku Google Colab Trenuj model m z parametrami $param$ na D_{train}
    Oblicz wskaźnik F1 na D_{train}
    Zapisz wynik F1 dla kombinacji hiperparametrów $param$
  Posortuj wyniki F1 dla wszystkich kombinacji hiperparametrów z 301 iteracji
  Oblicz medianę wyników F1 z 301 iteracji
  Wybierz zestaw hiperparametrów, który odpowiada medianie F1
  Użyj aplikacji webowej Trinity SALT do oceny modelu m z wybranymi hiperparametrami na pełnym zbiorze treningowym D_{train}
  Zapisz różne wskaźniki wydajności (F1, czułość, precyzja, dokładność, itd.) na D_{train} dla wybranych hiperparametrów
  Użyj aplikacji webowej Trinity SALT do oceny modelu m na zbiorze testowym D_{test} z wybranymi hiperparametrami i oblicz wskaźnik F1 oraz inne wskaźniki wydajności
  Zapisz różne wskaźniki wydajności (F1, czułość, precyzja, dokładność, itd.) na D_{test} dla wybranych hiperparametrów
return *Optymalne hiperparametry dla każdego modelu, wyniki na zbiorze testowym*

Algorytm 6 opisuje proces optymalizacji hiperparametrów dla trzech algorytmów bazowych: SOM, AE i LOF. Każdy model jest trenowany na zbiorze treningowym, a następnie oceniany na zbiorze testowym, przy czym optymalne hiperparametry są wybierane na podstawie mediany wyników miary F1. Choć wyniki te są istotne, nie stanowią końcowego etapu analizy. Jak pokazano na rysunku 7.3 w rozdziale 7 dotyczącym projektu aplikacji webowej oraz na schemacie działania systemu Trinity SALT na rysunku 8.16, a także w podrozdziale 4.5 na rysunku 4.7, zaimplementowane algorytmy są następnie wykorzystywane w metodzie zespołowej Trinity SALT (SOM-AE-LOF-TriDetect).



Rysunek 8.16: Zasada działania Trinity SALT. Źródło: opracowanie własne.

Wyniki z algorytmów SOM, AE i LOF są łączone w celu uzyskania skonsolidowanego rezultatu detekcji anomalii. Do połączenia wyników z wszystkich modeli zastosowano autorską metodę nazwaną MNA (maksymalna normalizowana agregacja, *ang. maximum normalized aggregation*). MNA nie jest powszechnie rozpoznawana w literaturze naukowej, ale stanowi unikalne zastosowanie znanych technik kombinacji wyników modeli, takich jak normalizacja i wybór maksymalnej wartości. Można ją uznać za nową wariację istniejących zasad, co czyni ją innowacyjną i efektywną w kontekście analizy zespołowej. Proces MNA obejmuje trzy główne kroki: normalizację wyników do przedziału (0,1), połączenie wyników poprzez wybór maksimum z znormalizowanych wartości oraz obliczenie premii za konsensus z modeli, co ilustruje wzór (7.1) i algorytm 7. Każdy model wnosi coś unikalnego, co pozwala na bardziej precyzyjne prognozowanie. Jeśli jeden model popełnia błąd, inny model może go skorygować, co prowadzi do bardziej stabilnych i dokładnych wyników. Dzięki temu podejściu system skutecznie wykorzystuje różne aspekty danych, co poprawia ogólną skuteczność analizy. Metodę Trinity SALT można postrzegać jako zaawansowaną formę stacking-u, w której zamiast tradycyjnego meta-modelu uczącego się na wynikach bazowych algorytmów, wykorzystuje się operacje takie jak normalizacja, wybór maksimum oraz przyznawanie bonusów za zgodność wyników. Dzięki temu podejściu,

meta-klasyfikator Trinity SALT skutecznie integruje wyniki identyfikacji anomalii z kilku modeli. Algorytm 7 przedstawia końcowy etap analizy w systemie Trinity SALT, który integruje wyniki z trzech różnych algorytmów (SOM, AE, LOF) po ich normalizacji. Wybrana jest maksymalna wartość spośród znormalizowanych rankingów $R_{\text{SOM}}, R_{\text{AE}}, R_{\text{LOF}}$, do której dodawany jest bonus, jeśli obiekt został wykryty jako anomalia przez więcej niż jeden algorytm. Ostateczny wynik R_{TRINITY} jest zwracany jako końcowy wynik detekcji anomalii.

Algorytm 7: Końcowy etap analizy wyników w systemie Trinity SALT z wykorzystaniem MNA (Maksymalna Normalizowana Agregacja) i bonusu za zgodność

Input: Znormalizowane rankingi algorytmów $R_{\text{SOM}}, R_{\text{AE}}, R_{\text{LOF}}$
Output: Ostateczny ranking R_{TRINITY} dla detekcji anomalii

▷ Po normalizacji wyników wszystkich algorytmów: **for** *każdy obiekt X_i w zbiorze danych D* **do**
  Wybierz maksymalną wartość spośród rankingów $R_{\text{SOM}}, R_{\text{AE}}, R_{\text{LOF}}$ dla obiektu X_i (tj. $\max(R_{\text{SOM}}, R_{\text{AE}}, R_{\text{LOF}})$)
  if *obiekt X_i został wykryty jako anomalia przez więcej niż jeden algorytm* **then**
    Przyznaj dodatkowy bonus dla obiektu X_i, proporcjonalny do liczby algorytmów wykrywających X_i jako anomalię
  Zsumuj maksymalną wartość i bonus, aby uzyskać końcowy wynik R_{TRINITY} dla obiektu X_i
return *Końcowe wyniki $R_{TRINITY}$ dla wszystkich obiektów, które umożliwią detekcję anomalii*

Mediana odgrywa ważną rolę w ocenie systemu Trinity SALT, ponieważ w kontekście detekcji anomalii algorytmy często wykazują niestabilność wobec różnych ustawień hiperparametrów. Skupienie się wyłącznie na najlepszych wynikach może prowadzić do błędnych wniosków, gdyż optymalne hiperparametry zazwyczaj nie są znane w rzeczywistych, nienadzorowanych problemach. Analiza wydajności algorytmu na podstawie mediany pozwala uzyskać bardziej realistyczny obraz jego działania w praktycznych warunkach, gdzie idealne hiperparametry mogą być trudne do określenia. Mechanizm wyboru optymalnych hiperparametrów dla algorytmów SOM, AE i LOF, opisany w algorytmie 6, został dokładniej omówiony w algorytmie 8.

Kolejnym uzasadnieniem wyboru mediany w algorytmie 6 do oceny algorytmów SOM, AE i LOF jest ograniczenie miary AUC w kontekście detekcji anomalii. Miara AUC jest popularna, jednak może prowadzić do błędnych wniosków, na przykład przez to, że traktuje wszystkie części krzywej ROC jednakowo. Nie zawsze jest to praktyczne w identyfikacji anomalii, gdzie najczęściej znaczenie mają pierwsze miejsca na górze rankingu. W detekcji anomalii ważne jest szybkie wykrycie rzeczywistych anomalii, a miara AUC, uwzględniając całą krzywą ROC, nie zawsze oddaje znaczenie początkowych wartości, które są

Algorytm 8: Optymalizacja hiperparametrów dla SOM, AE i LOF

Input: Zbiór danych D, lista modeli bazowych {SOM, AE, LOF}, lista hiperparametrów dla każdego modelu, liczba iteracji $n\_calls = 301$
Output: Optymalne hiperparametry dla każdego modelu, wyniki na zbiorze testowym

```
Importuj niezbędne moduły, takie jak gp_minimize z biblioteki skopt
Inicjalizuj strukturę danych do przechowywania wyników miary F1 dla każdej iteracji optymalizacji
Zdefiniuj przestrzeń hiperparametrów param_space_config, określając zakresy wartości dla kombinacji hiperparametrów specyficznych dla każdego algorytmu (SOM, AE, LOF)
for każdy model m z listy {SOM, AE, LOF} do
    Funkcja celu trenuje model m na danych treningowych i zwraca odpowiednią miarę F1 dla różnych kombinacji hiperparametrów (minimalizacja negatywnej wartości miary F1, co odpowiada maksymalizacji wartości F1)
    Optymalizacja hiperparametrów za pomocą funkcji gp_minimize przez 301 iteracji (n_calls=301)
    Zapisz wyniki miary F1 dla każdej kombinacji hiperparametrów w trakcie optymalizacji
Oblicz medianę wyników miary F1 z każdej iteracji, aby uzyskać stabilne hiperparametry
Wybierz hiperparametry zgodnie z medianą miary F1 dla każdego modelu
return Optymalne hiperparametry dla każdego modelu
```

najbardziej istotne w identyfikacji anomalii. Aby uzyskać bardziej wyważoną ocenę, zastosowano miarę F1, która harmonizuje precyzję i czułość. Starano się unikać błędów w wykrywaniu anomalii, takich jak nadmierne dopasowanie do specyficznych hiperparametrów czy nadmierne poleganie na wiedzy o etykietach anomalii, stąd zastosowanie mediany.

Zmodyfikowane zbiory danych, z oznaczonymi obiektami odstającymi w kolumnie *is_outlier*, używane w tych eksperymentach, są dostępne pod linkiem[5]. Szczegółowy opis dwunastu zbiorów znajduje się w poprzednich sekcjach 8.1.1 i 8.1.2, poświęconych algorytmowi LOF oraz eksperymentalnemu podziałowi na bloki. W trakcie tych badań wprowadzono dwa nowe zbiory danych: *„Thyroid Disease”* oraz *„Breast Cancer Wisconsin (Diagnostic)”*, zwiększając łączną liczbę analizowanych zbiorów do czternastu. Te dwa dodatkowe zbiory zostały celowo zmodyfikowane, aby zawierały rzeczywiste anomalie w granicach około 1%, co podkreśla ich wartość w kontekście detekcji rzeczywistych obiektów odstających. Ponadto zbiory *„Car”*, *„Mushroom”*, *„Bank”* oraz *„Adult”* zo-

[5]https://drive.proton.me/urls/D4V54KVHE4#UJvY8lvWsPQE

stały przekształcone z syntetycznych na rzeczywiste, w odróżnieniu od wcześniejszych eksperymentów opisanych w podrozdziale 8.1.1. W związku z tym liczba anomalii w tych zbiorach zmieniła się, wynosząc 3,76%, 0,99%, 0,55% oraz 0,58%, jak pokazano w tabeli 8.6, gdzie również szczegółowo przedstawiono podział zbiorów danych, najczęściej w proporcji 70% do 30%, z wyjątkiem zbiorów *„Bank”* oraz *„Breast Cancer”*, które zostały podzielone niemal po równo (50% na 50%). Szczegóły dotyczące nowych zbiorów oraz wprowadzonych w nich modyfikacji przedstawiono poniżej:

„Thyroid Disease” [431] - to zbiór danych pochodzący z Garavan Institute, zawierający informacje niezbędne do diagnozowania chorób tarczycy. Zbiór ten obejmuje 7 200 obiektów (osób) oraz 21 cech, które obejmują zarówno dane ciągłe, jak i binarne. Przypadki zostały sklasyfikowane w trzech kategoriach: zdrowi, osoby z nadczynnością tarczycy oraz osoby z niedoczynnością tarczycy. Analiza skupia się na identyfikacji anomalii, które mogą sugerować błędne diagnozy lub nietypowe objawy, szczególnie w kontekście nadczynności i niedoczynności tarczycy. W zbiorze danych nie ma brakujących wartości. Ze zbioru wyodrębniono 6 783 obiekty, z czego 117 rzeczywistych obiektów dotyczy nadczynności lub niedoczynności tarczycy i zostały one zaklasyfikowane jako anomalie (1,73%), pozostawiając 6 666 obiektów (osób) zdrowych.

„Breast Cancer Wisconsin (Diagnostic)” [432] - to zbiór danych medycznych zawierający 569 obiektów oraz 30 cech opisujących właściwości jądra komórkowego, uzyskanych z cyfrowych obrazów aspiratów cienkoigłowych (FNA) guzów piersi. Przypadki w tym zbiorze zostały sklasyfikowane jako łagodne (B) lub złośliwe (M), co czyni te dane przydatnymi w analizie problemów klasyfikacyjnych związanych z diagnozowaniem raka piersi. Każda z 30 cech opisuje właściwości jądra komórkowego, takie jak promień, tekstura, obwód, powierzchnia, gładkość, zwartość, wklęsłość, punkty wklęsłości, symetria i wymiar fraktalny. Zbiór danych jest kompletny, bez brakujących wartości. Spośród 569 obiektów, 357 zostało sklasyfikowanych jako łagodne (*ang. benign*), a 212 jako złośliwe (*ang. malignant*). W zbiorze znajdują się 4 obiekty odstające, co daje odsetek wynoszący 1,11%. Analiza tych danych często wymaga odwzorowania w przestrzeni trójwymiarowej, której szczegółowy opis można znaleźć w pracy [433].

Analiza wyników eksperymentów z użyciem systemu Trinity SALT

Zgodnie z opisaną wcześniej metodologią, wyniki eksperymentów oraz szczegółowy proces ich realizacji zostały zilustrowane na przykładzie zbioru *„Covid19”* (pełna nazwa: *„Covid19 case surveillance real”*). W dalszej części podrozdziału przedstawiono podsumowanie wyników dla wszystkich czternastu analizowanych zbiorów danych. Przykład procesu eksperymentalnego został przeprowadzony zgodnie z algorytmem 6, a szczegółowy opis jego działania znajduje się również w algorytmie 8. Ostateczne wyniki (dla wszystkich czternastu zbiorów) uwzględniają końcowy etap analizy za pomocą systemu Trinity SALT, który został przedstawiony w algorytmie 7.

Tabela 8.6: Zbiory danych użyte w analizie obejmują zbiory rzeczywiste (R) oraz syntetyczny zbiór *„Citibike”* (S). Dane zostały podzielone na część szkoleniową (Szk.) i testową (Tst.) w proporcji 70% do 30%, z wyjątkiem zbiorów *„Bank mktg”* oraz *„Breast cancer”*, które zostały podzielone niemal po równo (50% na 50%). Wszystkie zbiory danych posiadają cechy kategoryczne, z wyjątkiem czterech: *„Breast cancer”*, *„Credit Card”*, *„Thyroid”* oraz *„Wine quality”*. W kolumnach tabeli „licz.” oznacza liczność zbioru, „outlrs” odnosi się do liczby anomalii, a „%” reprezentuje procent anomalii w zbiorze. Źródło: opracowanie własne.

Zbiory (R)	Cechy	Szk. (licz.; outlrs; %)	Tst. (licz.; outlrs; %)	Cał. (licz.; outlrs; %)
Bank mktg	16	20 072; 111; 0,55	20 072; 111; 0,55	40 144; 222; 0,55
Chess krkopt	6	19 639; 19; 0,10	8 417; 8; 0,10	28 056; 27; 0,10
Covid19	10	23 737; 232; 0,98	10 174; 100; 1,00	33 911; 332; 0,98
Mushroom	22	2 975; 29; 0,97	1 275; 13; 1,02	4 250; 42; 0,99
Breast cancer	30	180; 2; 1,11	181; 2; 1,10	361; 4; 1,11
Credit Card	30	199 364; 344; 0,17	85 433; 148; 0,17	284 807; 492; 0,17
Thyroid	21	4 748; 82; 1,73	2 035; 35; 1,72	6 783; 117; 1,72
Veh. Claims	19	149 095; 1 042; 0,70	63 899; 446; 0,70	212 994; 1 488; 0,70
KDD CUP ’99	41	68 303; 209; 0,31	29 274; 90; 0,31	97 577; 299; 0,31
w7a libsvm	300	34 073; 284; 0,83	14 603; 122; 0,84	48 676; 406; 0,83
Wine quality	11	3 428; 17; 0,50	1 470; 8; 0,54	4 898; 25; 0,51
Adult	14	26 159; 151; 0,58	11 211; 64; 0,57	37 370; 215; 0,58
Car Eval.	6	1 209; 45; 3,72	519; 20; 3,85	1 728; 65; 3,76
Citibike (S)	11	14 000; 140; 1,00	6 000; 60; 1,00	20 000; 200; 1,00

Tabela 8.7 zawiera szczegółowy opis atrybutów zbioru *„Covid19”*, użytego do zilustrowania całego procesu eksperymentalnego. Jak opisano w metodologii, optymalne hiperparametry dla algorytmów SOM, AE i LOF są wybierane poprzez wielokrotne trenowanie modeli na danych treningowych oraz ocenę wyników za pomocą miary F1. Na przykładzie zbioru *„Covid19”*, dane zostały podzielone na część szkoleniową (70%) i testową (30%). Następnie, dla każdego z modeli bazowych (SOM, AE, LOF) zdefiniowano przestrzeń hiperparametrów. Funkcja celu, minimalizując negatywną wartość miary F1 (co odpowiada maksymalizacji samej miary F1), optymalizuje kombinacje hiperparametrów, wykonując 301 iteracji za pomocą funkcji `gp_minimize` [409]. W trakcie tego procesu zapisywane są wyniki miary F1, a na końcu obliczana jest mediana wyników, co pozwala na wybranie najbardziej stabilnych hiperparametrów. Ostatecznie uzyskane hiperparametry są uznawane za optymalne dla każdego modelu, a wyniki na zbiorze testowym dostarczają informacji o ich skuteczności. Proces optymalizacji bayesowskiej, dobierającej hiperpara-

Tabela 8.7: Opis atrybutów zbioru danych *„Covid19”*, wykorzystanego do demonstracji procedur eksperymentalnych. Źródło: opracowanie własne.

Nazwa atrybutu	**Znaczenie i wartości**
cdc_report_dt	Data zgłoszenia do CDC
pos_spec_dt	Data pierwszego pozytywnego pobrania próbki (MM/DD/YY-YY)
onset_dt	Data wystąpienia objawów
current_status	Aktualny status osoby, przyjmuje wartości: Przypadek potwierdzony laboratoryjnie 94%, Przypadek prawdopodobny 6%
sex	Płeć, przyjmuje wartości: Kobieta 52%, Mężczyzna 47%, Inne 1%
age_group	Kategoria wiekowa, przyjmuje wartości: 20−29 lat 19%, 30−39 lat 18%, Inne 64%
Race_ and_ethnicity	(Demografia przypadku), przyjmuje wartości: Nieznane 33%, Biały 31%, Nie-hiszpan/latynos 36%, Inne 36%
hosp_yn	Czy pacjent był hospitalizowany?, przyjmuje wartości: Nie 42%, Brak danych 38%, Inne 20%
icu_yn	Czy pacjent był przyjęty na oddział intensywnej terapii?, przyjmuje wartości: Brak danych 74%, Nieznane 15%, Inne 11%
medcond_yn	Czy miał jakiekolwiek istniejące wcześniej schorzenia i/lub zachowania ryzyka?, przyjmuje wartości: Brak danych 72%, Nieznane 10%, Inne 18%.
death_yn	Czy pacjent zmarł w wyniku tej choroby?, przyjmuje wartości: Nie 44%, Brak danych 41%, Inne 15%
is_outlier	

metry, przeprowadzono w płatnym środowisku Google Colab Pro+, co umożliwiało dostęp do większej pamięci RAM (52 GB). Uzyskane hiperparametry dla zbioru *„Covid19”* przedstawiono w tabelach 8.8, 8.9 oraz 8.10. Analogicznie postąpiono z trzynastoma innymi zbiorami, a wszystkie optymalne hiperparametry dla pozostałych zbiorów zamieszczono w Dodatku C 9.1.

Tabela 8.8: Optymalne hiperparametry i wartości metryk dla modelu SOM na zbiorach szkoleniowym (70%, n = 23 737, 232 anom., 0.98%) oraz testowym (30%, n = 10 174, 100 anom., 1.00%) dla zbioru *Covid19 case surveillance real*.

Zbiór	**x**	**y**	**Współ. ucz.**	**Prom.**	**Fun. Zaniku**
Covid19 case real	28	11	0,004952419078874148	15	declin_decay

Topol.	**Metr.**	**Fun. Sąs.**	**L. Iter.**	**Seed**
hexagon.	cosine	mexican hat	361	20

Tabela 8.9: Optymalne hiperparametry i wartości metryk dla modelu AE na zbiorach szkoleniowym (70%, n = 23 737, 232 anom., 0.98%) oraz testowym (30%, n = 10 174, 100 anom., 1.00%) dla zbioru *Covid19 case surveillance real*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Covid19 case real	246	0,0662	667	L1	0.5216	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
1	relu	sigmoid	MSE	23

Tabela 8.10: Optymalne hiperparametry i wartości metryk dla modelu LOF na zbiorach szkoleniowym (70%, n = 23 737, 232 anom., 0.98%) oraz testowym (30%, n = 10 174, 100 anom., 1.00%) dla zbioru *Covid19 case surveillance real*.

Zbiór	Rozm. bloku	MinPts	Metr.
Covid19 case real	61	33	euclidean

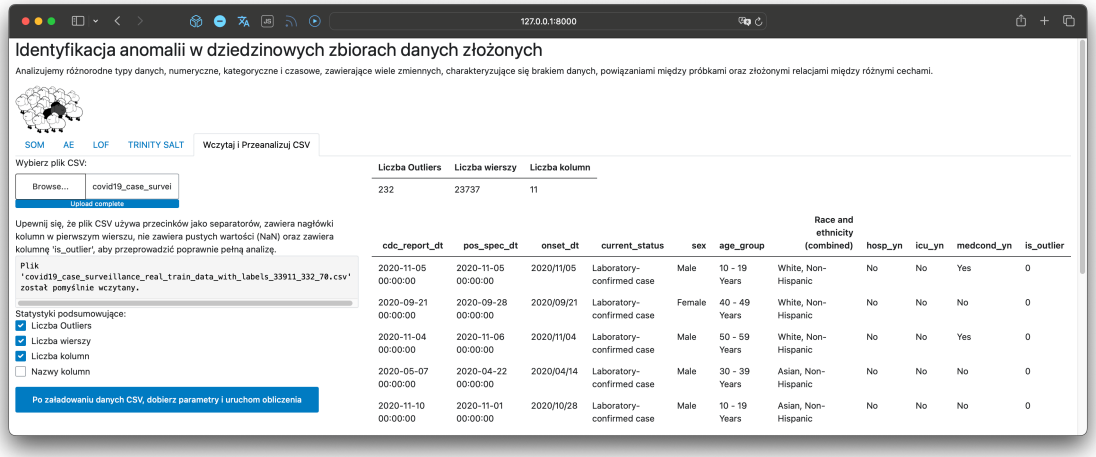
Po ustaleniu optymalnych hiperparametrów algorytmu, przechodzi się do finalnego etapu analizy, opisanego w algorytmie 7, realizowanego za pomocą aplikacji webowej Trinity SALT (szczegóły znajdują się w rozdziale 7). Na początku wczytywany jest badany plik CSV, co pokazano na rysunku 8.17. Na tym etapie ocenia się skuteczność detekcji anomalii w zbiorach szkoleniowych i testowych, zarówno dla pojedynczych algorytmów, jak i zespołu z wykorzystaniem metodologii Maksymalnej Normalizowanej Agregacji MNA oraz uwzględnieniem bonusu za zgodność modeli.

W tabeli 8.11 przedstawiono wyniki dla zbioru danych *„Covid19”*, które ilustrują praktyczne zastosowanie algorytmu. W dalszej części podrozdziału zaprezentowano również wyniki dla zbioru *„Mushroom (R)”* (gdzie „(R)” oznacza dane rzeczywiste) oraz uśrednione metryki dla wszystkich 14 analizowanych zbiorów danych. W podrozdziale omówiono rezultaty dla dwóch wybranych zbiorów oraz uśrednione metryki dla pozostałych. Szczegółowe dane dla wszystkich zbiorów znajdują się w Dodatku B 9.1.

Kontrolowane warunki badawcze, w których znana jest liczba rzeczywistych anomalii (rzeczywiste etykiety), umożliwiają precyzyjną ocenę algorytmów wykrywających anomalie oraz przyjęcie odpowiednich założeń dotyczących ich działania w różnych scenariuszach. W badaniu zastosowano metodę klasyfikacji opartą na rankingu anomalii, zakładając, że elementy z najwyższymi wartościami w rankingu będą klasyfikowane jako anomalie. Liczba rzeczywistych anomalii, zarówno generowanych sztucznie, jak i pochodzących z wcześniej oznaczonych zbiorów danych, została ustalona i odpowiadała liczbie obiektów uznanych za anomalie. Na przykład, gdy anomalie stanowiły 1% rozmiaru zbioru, identyczna liczba elementów była klasyfikowana jako anomalie. W tej sytuacji każdy błąd klasyfikacji skutkował identyczną liczbą fałszywie pozytywnych (FP) i fałszywie negatywnych (FN) wyników. W przypadku błędnej klasyfikacji normalnych obiektów jako anomalie, pomijano taką samą liczbę rzeczywistych anomalii. Równość wyników

czułości i precyzji wynikała z przyjętych założeń oraz metodologii badania. Spójność tego podejścia opierała się na tym, że anomalie były generowane w sposób kontrolowany lub pochodziły z wcześniej zweryfikowanych zbiorów danych, co umożliwiło dokładną analizę skuteczności algorytmów przy znanej i ustalonej liczbie anomalii.

Kolumna *is_outlier* służy wyłącznie do oznaczania obiektów wcześniej zidentyfikowanych jako odstające (ground truth) i jest wykorzystywana do budowy macierzy pomyłek oraz innych wyników, takich jak tabele i wykresy. Przed rozpoczęciem analizy kolumna ta jest usuwana, aby nie wpływała na wyniki. Dane są przetwarzane niezależnie, a po zakończeniu analizy kolumna *is_outlier* jest przywracana, co umożliwia porównanie wyników z wcześniejszymi oznaczeniami. Pozwala to ocenić zgodność nowo wykrytych anomalii z istniejącymi oznaczeniami bez ingerencji kolumny *is_outlier* w proces analizy.



Rysunek 8.17: Zbiór danych *„Covid19”* wczytany do aplikacji webowej. Źródło: opracowanie własne.

Problem, przed którym staje algorytm, jest niezwykle trudny do rozwiązania. Anomalie w analizowanych danych stanowią około jednego procenta całego zbioru, a dokładnie 0,98% w tym przypadku. Podobnie jest w innych analizowanych zestawach danych, co oznacza konieczność radzenia sobie z ogromną nierównowagą klas [434]. Dysproporcja ta powoduje naturalną tendencję algorytmu do przewidywania dominującej klasy, co znacząco utrudnia precyzyjne wykrywanie rzadkich anomalii. Większość współczesnych badań nad nierównowagą klas koncentruje się na przypadkach, gdzie stosunek między klasami wynosi od 1:4 do 1:100. Istnieje jednak ograniczona liczba prac zajmujących się klasyfikacją danych o skrajnie nierównomiernej strukturze [332]. W wielu analizowanych zbiorach odsetek anomalii wynosi mniej niż 1% (często około 0,5%, co odpowiada stosunkowi 1:200), choć niektóre zbiory charakteryzują się znacznie bardziej ekstremalnymi współczynnikami nierównowagi. Przykładowo, zbiór *„Chess krkopt”* zawiera zaledwie

0,1% anomalii, co przekłada się na stosunek 1:999 (anomalie: przypadki normalne). Zbiór *„KDD CUP '99"* ma 0,31% anomalii (stosunek 1:322), a zbiór *„Credit Card"* wykazuje 0,17% anomalii (stosunek 1:588). Takie przypadki stanowią dla algorytmów poważne wyzwania, wymagając odpowiedniego dostosowania zarówno na etapie wstępnego przetwarzania danych, jak i w trakcie klasyfikacji, aby skutecznie poradzić sobie z taką skrajną nierównowagą. Mimo rosnącego zainteresowania problemem nierównowagi klas w mniej ekstremalnych scenariuszach, klasyfikacja zbiorów o tak skrajnej dysproporcji między klasami pozostaje stosunkowo słabo zbadana. Takie przypadki są wyjątkowo trudne do rozwiązania i wymagają zastosowania zaawansowanych technik w algorytmach [332].

Dodatkowo, badane zbiory są najczęściej wielowymiarowe i zawierają również cechy kategoryczne, które po przekształceniu w wektory zero-jedynkowe znacząco komplikują analizę. Ponadto wyzwaniem jest ograniczona liczba przykładów anomalii w danych treningowych - w każdym z analizowanych zbiorów występują one bardzo rzadko. W omawianym przypadku występują zaledwie 232 przypadki anomalii na ponad 23 000 obiektów, co sprawia, że modele dysponują zbyt małą liczbą danych, aby skutecznie nauczyć się wzorców anomalii. Taka sytuacja stwarza ryzyko, że modele nie będą w stanie uogólnić wzorców anomalii lub wręcz zbytnio dopasują się do niewielkiej liczby dostępnych przypadków, co prowadzi do przeuczenia. Duża liczba cech, połączona z ograniczoną liczbą anomalii, powoduje rozproszenie danych, znacząco utrudniając skuteczne uczenie się modeli. Przy tak niskiej liczbie anomalii modele często klasyfikują większość przypadków jako normalne, co skutkuje niską precyzją w wykrywaniu rzeczywistych anomalii. Nawet zaawansowane algorytmy mogą mieć trudności z identyfikacją tak rzadkich zdarzeń.

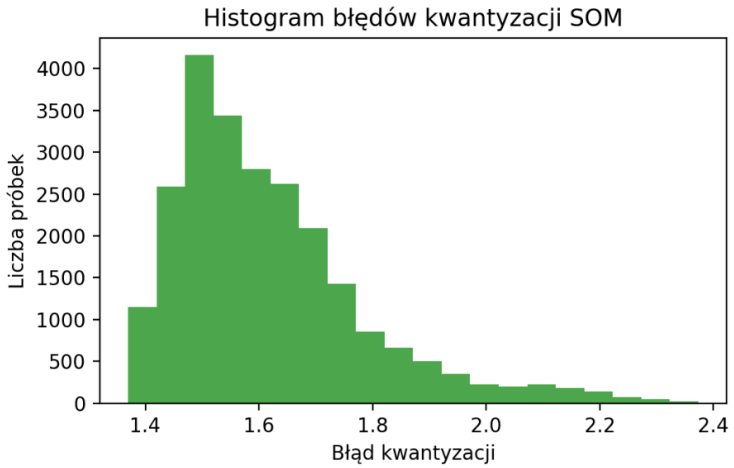
Trinity SALT, jako podejście integrujące wyniki z trzech różnych algorytmów, stara się odpowiedzieć na te wyzwania. Ta metoda ma na celu minimalizowanie ryzyka przeoczenia istotnych anomalii, choć wiąże się również z wyzwaniem związanym ze zwiększoną liczbą fałszywych alarmów. Zamiast optymalizować model pod kątem najlepszych wyników na danych treningowych, Trinity SALT unika nadmiernego dopasowania poprzez integrację wyników z różnych konfiguracji algorytmów i wykorzystanie mediany wyników. Jak pokazano w algorytmie 8, zamiast wybierać konfiguracje, które maksymalizują miarę F1, metoda korzysta z mediany wyników, co pozwala na uzyskanie bardziej ogólnych modeli. Taka strategia minimalizuje ryzyko nadmiernego dopasowania, co pozwala na otrzymanie bardziej stabilnych i realistycznych wyników, zwłaszcza w sytuacjach, gdy w rzeczywistości nie dysponujemy oznaczonymi etykietami. Choć początkowo podejście z medianą może wydawać się mało skuteczne, w rzeczywistości pozwala ono stworzyć model lepiej dostosowany do złożoności świata rzeczywistego. W efekcie model ten wykazuje większą zdolność do efektywnej generalizacji na nowe, nieznane dane, co jest niezbędne w praktycznych zastosowaniach.

Po wczytaniu zbioru danych do aplikacji webowej i ustawieniu optymalnych hiperparametrów, zgodnie z opisem systemu w rozdziale 7, użytkownik może przeglądać wyniki w formie wykresów i tabel. Nie wszystkie dostępne wykresy i tabele są tutaj prezentowane.

Tabela 8.11: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Covid19”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

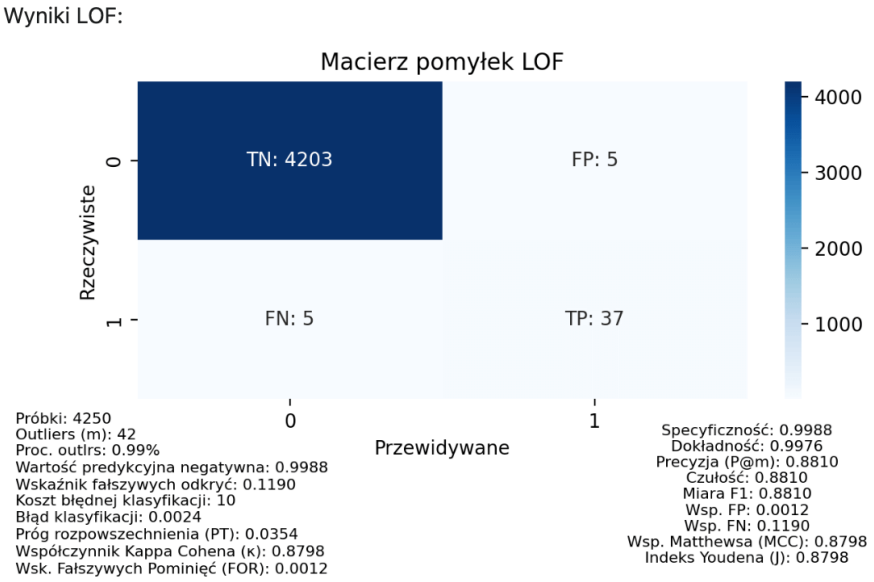
Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Covid19 (R)	Bł. kl.	0,0114	0,0124	0,0122	0,0130	0,0127	0,0130	0,0150	0,0149
Cechy: 10 (w tym kateg.)	Spec.	0,9943	0,9937	0,9938	0,9934	0,9936	0,9934	0,9893	0,9893
Szk.:23737;232;0,98%	Prec.	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,3386	0,3415
	Czuł.	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,5560	0,5600
Tst.:10174;100;1,00%	F1	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,4209	0,4242
	FP	0,0057	0,0063	0,0062	0,0066	0,0064	0,0066	0,0107	0,0107
n:33911;332;0,98%	FN	0,5819	0,6300	0,6250	0,6600	0,6509	0,6600	0,4440	0,4400
	Czas	0,0935	0,0528	4,1264	1,8368	0,1166	0,0499	7,7652	2,6829

Poniżej pokazano dwa przykłady: histogram błędów kwantyzacji dla algorytmu SOM na zbiorze *„Covid19”* (rysunek 8.18) oraz macierz pomyłek z różnymi metrykami dla algorytmu LOF na całym zbiorze *„Mushroom (R)”* (rysunek 8.19). Pozostałe wizualizacje, w tym inne wykresy i tabele, są dostępne w aplikacji.



Rysunek 8.18: Zrzut ekranu aplikacji webowej przedstawiający histogram błędów kwantyzacji dla algorytmu SOM na zbiorze danych *„Covid19”*. Źródło: opracowanie własne.

Analiza wyników przedstawionych w tabeli 8.11 oraz na rysunku 8.20, ograniczonych przez wcześniej omówione wyzwania, pozwala ocenić skuteczność detekcji anomalii, szczególnie pod kątem czułości. Dla algorytmu SOM czułość w zbiorze szkoleniowym wynosi 41,8%. Jest to stosunkowo dobry wynik, biorąc pod uwagę niską częstość występowania anomalii (0,98%) oraz dużą liczbę cech. Algorytmy AE (37,5%) i LOF (34,91%)



Rysunek 8.19: Zrzut ekranu aplikacji webowej przedstawiający macierz pomyłek dla algorytmu LOF na całym zbiorze danych *„Mushroom (R)”*. Źródło: opracowanie własne.

Tabela 8.12: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Mushroom (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Mushroom (R)	Bł. kl.	0,0007	0,0063	0,0000	0,0047	0,0007	0,0047	0,0007	0,0071
Cechy: 22 (w tym kateg.)	Spec.	0,9997	0,9968	1,0000	0,9976	0,9997	0,9976	0,9993	0,9937
Szk.:2975;29;0,97%	Prec.	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	0,9355	0,6000
	Czuł.	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	1,0000	0,9231
Tst.:1275;13;1,02%	F1	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	0,9667	0,7273
	FP	0,0003	0,0032	0,0000	0,0024	0,0003	0,0024	0,0007	0,0063
T:4250;42;0,99%	FN	0,0345	0,3077	0,0000	0,2308	0,0345	0,2308	0,0000	0,0769
	Czas	0,0630	0,0522	11,5105	4,5333	0,6881	0,1224	13,4732	4,4240

uzyskały nieco niższe, ale wciąż akceptowalne wyniki w kontekście tych wyzwań. Trinity SALT został zaprojektowany w celu zwiększenia czułości przy jednoczesnym utrzymaniu precyzji na akceptowalnym poziomie. System zwiększa czułość do 55,6% (podobnie w zbiorze testowym, gdzie wynosi 56%), uwzględniając wszystkie obiekty oznaczone jako anomalie przez co najmniej jeden z detektorów, jednocześnie utrzymując wynik F1 na poziomie około 42% zarówno na zbiorze szkoleniowym, jak i testowym. Co istotne, podobny trend można zaobserwować w tabeli 8.13, która przedstawia uśrednione wyniki

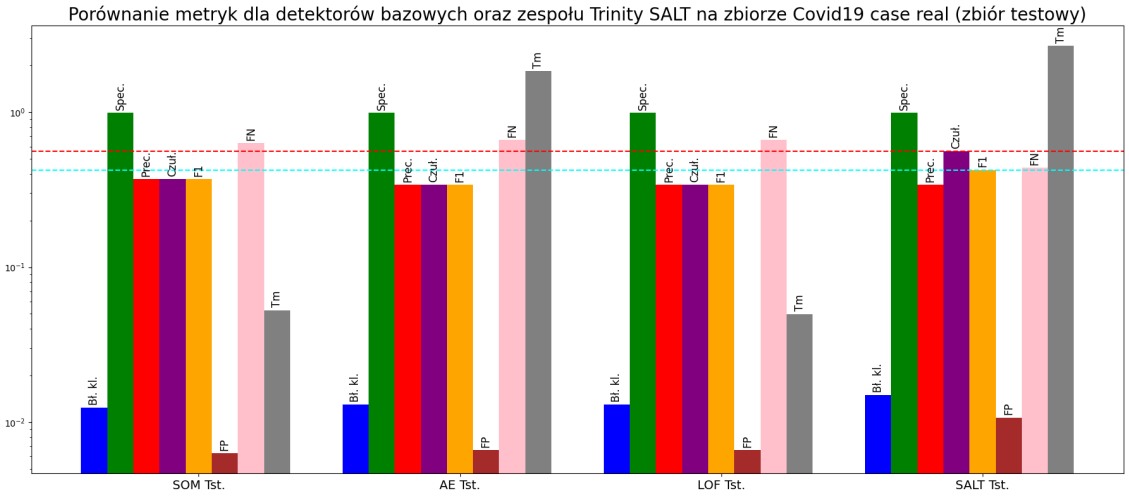
dla 14 różnorodnych i złożonych zbiorów danych, z których wszystkie charakteryzują się dużą nierównowagą, a niektóre mają wręcz ekstremalne współczynniki nierównowagi. Wyniki te zostały zwizualizowane na rysunkach 8.22 oraz 8.23. Nieuchronnym kosztem tego podejścia jest zwiększona liczba fałszywych alarmów. Jednak w scenariuszach, gdzie priorytetem jest czułość, a pominięcie anomalii może prowadzić do poważnych konsekwencji, Trinity SALT staje się szczególnie uzasadnionym wyborem.

Bazujące na analizie 14 zbiorów danych oraz zbioru *„Covid19”*, ogólne wyniki mogą sugerować umiarkowaną skuteczność Trinity SALT w wykrywaniu anomalii. Jednak bardziej szczegółowa analiza poszczególnych zbiorów (wyniki w Dodatku B 9.1) ukazuje znacznie bardziej imponujące rezultaty. Na przykład, w przypadku zbioru *„Mushroom (R)”*, co pokazuje tabela 8.12 oraz rysunek 8.21 system Trinity SALT osiągnął bardzo wysoką czułość na poziomie 92,31% na zbiorze testowym, co podkreśla jego wyjątkową skuteczność w wykrywaniu anomalii. Wysoki wynik F1 (72,73%) dodatkowo potwierdza efektywne zrównoważenie czułości i precyzji, co jest ważne w tego typu analizach. Podobnie w zbiorze *„Breast cancer (R)”*, algorytm uzyskał czułość wynoszącą 100%, co świadczy o pełnej skuteczności w detekcji anomalii. Wynik F1 na poziomie 57,14% sugeruje, że algorytm efektywnie balansuje czułość i precyzję, utrzymując solidną jakość wyników pomimo niskiej częstości anomalii (1,11% zbioru danych). W przypadku syntetycznego zbioru *„Citibike Synthetic (S)”*, Trinity SALT uzyskał perfekcyjne wyniki zarówno dla czułości, jak i F1, osiągając 100% na obu zbiorach — szkoleniowym i testowym. To pokazuje, że algorytm jest w stanie bezbłędnie wykryć wszystkie anomalie w danych. Podsumowując, mimo umiarkowanych wyników w uśrednionych metrykach, indywidualne przypadki, takie jak *„Mushroom (R)”* i *„Breast cancer (R)”* czy *„Citibike Synthetic (S)”*, pokazują, że algorytm Trinity SALT może osiągać znakomite rezultaty, gdy jest dobrze dostosowany do specyficznych cech danych. Warto podkreślić, że te wyniki zostały osiągnięte mimo zastosowania mediany przy wyborze hiperparametrów, co pozwala modelowi lepiej dostosować się do rzeczywistości, zamiast optymalizować się pod kątem maksymalnych wyników na danych treningowych. Zastosowanie mediany hiperparametrów sprawia, że algorytm unika nadmiernego optymizmu, co mogłoby wystąpić przy wyborze wyników maksymalnych. Takie podejście pozwala uzyskać wyniki bardziej zbliżone do tych, które można by zaobserwować na rzeczywistych zbiorach danych, minimalizując ryzyko rozczarowujących rezultatów podczas wdrażania modelu w realnych aplikacjach.

Analiza wyników na zbiorze testowym *„Covid19”*, na którym algorytm nie był trenowany, wykazuje, że metoda Trinity SALT nie tylko poprawia czułość w porównaniu z innymi detektorami bazowymi, ale także osiąga najwyższe wartości miary F1, co świadczy o równowadze między precyzją a czułością, jak pokazano na rysunku 8.20. Ponadto, wyższa miara F1 dla Trinity SALT w porównaniu z pozostałymi algorytmami (SOM, AE, LOF) na zbiorze testowym sugeruje, że poprawa czułości została osiągnięta bez znaczącej utraty precyzji. Zgodnie z literaturą naukową, wysoką czułość można uznać za istotny

Tabela 8.13: Podsumowanie wyników analizy detekcji anomalii w systemie Trinity SALT dla 14 zbiorów danych. Wartości w kolumnach reprezentują uśrednione metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Podsumowanie dla 14 zbiorów	Bł. kl.	0,0103	0,0121	0,0096	0,0121	0,0119	0,0137	0,0170	0,0196
	Spec.	0,9948	0,9938	0,9951	0,9938	0,9939	0,9930	0,9857	0,9838
	Prec.	0,4609	0,3820	0,4543	0,4131	0,4180	0,3429	0,3766	0,3342
	Czuł.	0,4609	0,3820	0,4543	0,4131	0,4180	0,3429	0,6473	0,6152
	F1	0,4609	0,3820	0,4543	0,4131	0,4180	0,3429	0,4614	0,4209
	FP	0,0052	0,0062	0,0049	0,0062	0,0061	0,0070	0,0143	0,0163
	FN	0,5391	0,6180	0,5457	0,5869	0,6177	0,6571	0,3527	0,3848
	Czas	1,3470	0,6209	172,2044	70,3957	20,6982	8,4522	188,6098	91,6515



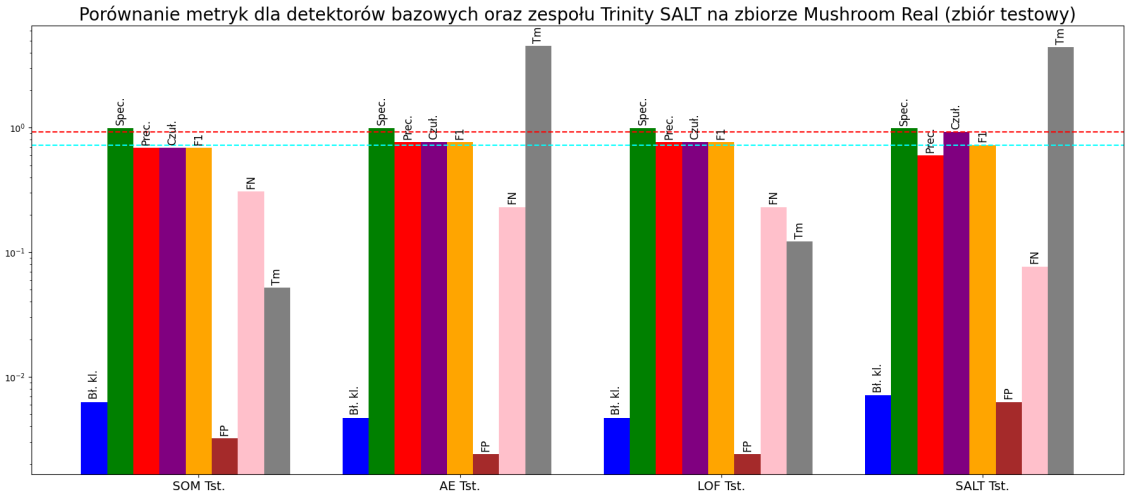
Rysunek 8.20: Porównanie metryk dla detektorów bazowych oraz zespołu Trinity SALT na zbiorze *„Covid19”* (zbiór testowy). Czerwona przerywana linia wskazuje wartość Czułości, a jasnoniebieska przerywana linia oznacza wartość miary F1. Niebieski słupek oznacza Błąd klasyfikacji (Bł. kl.), zielony Specyficzność (Spec.), czerwony Precyzję (Prec.), fioletowy Czułość (Czuł.), pomarańczowy miarę F1, brązowy Fałszywie Pozytywne (FP), różowy Fałszywie Negatywne (FN), a szary Czas (Tm). Źródło: opracowanie własne.

wskaźnik skuteczności modelu, zwłaszcza w przypadkach niezrównoważonych zbiorów danych, gdzie przeoczenie pozytywnych przypadków może mieć poważne konsekwencje. Czułość, jako miara wydajności, bierze pod uwagę rzeczywiste pozytywne przypadki, co jest szczególnie ważne w sytuacjach, w których błędna klasyfikacja przypadków z klasy mniejszościowej (np. pacjentów z chorobą) ma znacznie większe konsekwencje niż

przypadków większościowych (np. zdrowych pacjentów) [435]. Autorzy artykułu [436] podkreślają, że błędna klasyfikacja rzadkiego zdarzenia, takiego jak wykrycie komórek nowotworowych w diagnostyce medycznej, może prowadzić do znacznie poważniejszych problemów niż błędna klasyfikacja bardziej powszechnego zdarzenia. W tym przypadku, błędne zaklasyfikowanie komórek nowotworowych (czyli przypadków z klasy mniejszościowej) niesie znacznie większe ryzyko dla zdrowia, co wskazuje na konieczność priorytetowego traktowania czułości w takich sytuacjach. Autorzy artykułu [352] wskazują na ograniczenia krzywych ROC w niezrównoważonych zbiorach danych i sugerują, że bardziej odpowiednie do oceny klasyfikatorów w takich przypadkach są krzywe precyzji-czułości PRC. Oznacza to, że czułość pozostaje istotną metryką, ale może być lepiej interpretowana w kontekście krzywych PRC niż ROC. Zbiór testowy służy do oceny zdolności modelu do generalizacji, czyli jego skuteczności na nowych, wcześniej niewidzianych danych. Utrzymanie wysokiej skuteczności Trinity SALT na tym zbiorze świadczy o tym, że model jest dobrze przygotowany do pracy w rzeczywistych warunkach, gdzie dane mogą różnić się od tych użytych w procesie treningowym. Ponadto, zgodnie z literaturą naukową, zespoły algorytmów, takie jak Trinity SALT, wykazują większą stabilność i są mniej podatne na fluktuacje wyników w porównaniu z pojedynczymi algorytmami bazowymi. Jest to spowodowane redukcją wariancji oraz błędów losowych dzięki agregacji wyników wielu modeli, co zostało dobrze udokumentowane zarówno teoretycznie, jak i praktycznie w pracach takich jak [310, 204]. W związku z tym, wyniki uzyskane przez zespół Trinity SALT są bardziej stabilne, co wskazuje na lepszą ogólną niezawodność oraz mniejsze ryzyko zmienności wyników w różnych warunkach.

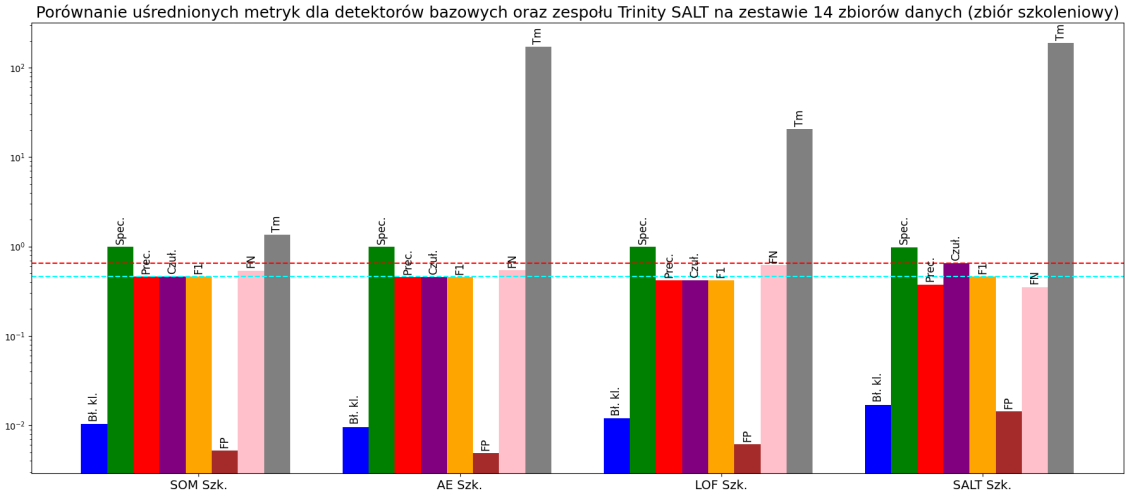
Wyniki uzyskane na zbiorze testowym *„Mushroom (R)”* (przedstawione na rysunku 8.21) również wskazują, że metoda Trinity SALT znacząco poprawia czułość w porównaniu z detektorami bazowymi, takimi jak SOM, AE i LOF. Co prawda miara F1 nie jest wyższa niż w przypadku innych algorytmów, ale pozostaje na zadowalającym poziomie, co sugeruje, że wzrost czułości został osiągnięty bez znaczącej utraty precyzji. Wyniki te pokazują, że Trinity SALT dobrze radzi sobie z detekcją anomalii, szczególnie pod względem czułości. Wysoka czułość Trinity SALT (92,31%) i satysfakcjonująca wartość miary F1 (72,73%) dowodzą, że model ten dobrze radzi sobie z detekcją anomalii również w rzeczywistych warunkach. Optymalizacja modelu pod kątem czułości nie spowodowała znaczącego spadku precyzji, co potwierdza skuteczność Trinity SALT.

Wyniki na zestawie 14 zbiorów danych, z uśrednionymi metrykami dla zbioru szkoleniowego (szczegółowe wyniki znajdują się w Dodatku B 9.1), przedstawione na rysunkach 8.22 oraz 8.24, potwierdzają wcześniejsze obserwacje z pojedynczego zbioru *„Covid19”*. Metoda Trinity SALT nadal wykazuje znaczną przewagę w zakresie czułości w porównaniu z innymi detektorami bazowymi, co sugeruje, że model ten skutecznie identyfikuje rzeczywiste anomalie na różnych typach danych. Co więcej, wskaźnik F1 dla Trinity SALT, który jest nieznacznie wyższy niż dla SOM, AE i LOF, wskazuje na dobre zrównoważenie między czułością a precyzją, co jest potrzebne dla zachowania niskiego poziomu fałszy-



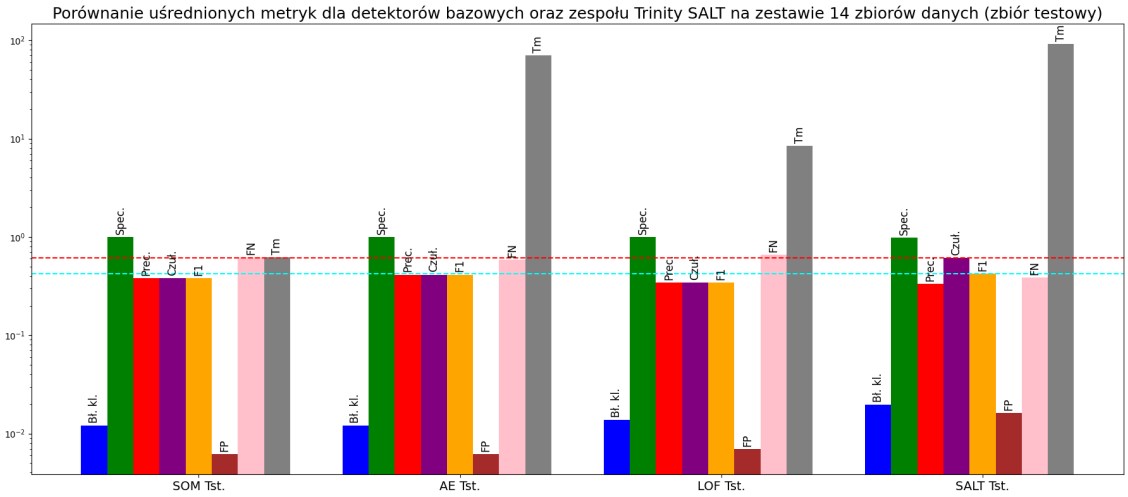
Rysunek 8.21: Porównanie metryk dla detektorów bazowych oraz zespołu Trinity SALT na zbiorze *„Mushroom (R)”* (zbiór testowy). Czerwona przerywana linia wskazuje wartość Czułości, a jasnoniebieska przerywana linia oznacza wartość miary F1. Niebieski słupek oznacza Błąd klasyfikacji (Bł. kl.), zielony Specyficzność (Spec.), czerwony Precyzję (Prec.), fioletowy Czułość (Czuł.), pomarańczowy miarę F1, brązowy Fałszywie Pozytywne (FP), różowy Fałszywie Negatywne (FN), a szary Czas (Tm). Źródło: opracowanie własne.

wych alarmów. Co nowego w przypadku analizy 14 zbiorów danych to potwierdzenie, że Trinity SALT utrzymuje swoją przewagę nawet w bardziej zróżnicowanym środowisku danych. Oznacza to, że metoda ta jest bardziej wszechstronna i skuteczna w szerokim zakresie różnych problemów detekcji anomalii. Zwiększona stabilność wyników w analizie 14 zbiorów danych podkreśla, że agregacja wyników wielu modeli, jaką oferuje Trinity SALT, prowadzi do redukcji wariancji i błędów losowych, co potwierdza jej niezawodność i mniejsze ryzyko fluktuacji wyników w zmiennych warunkach. To badanie pokazuje, że metoda Trinity SALT nie tylko jest skuteczniejsza od pozostałych algorytmów bazowych na pojedynczym zbiorze danych, jakim jest *„Covid19”*, ale także zachowuje swoją efektywność na szerszej gamie zbiorów, co czyni ją uniwersalnym narzędziem w identyfikacji anomalii. Wyniki analizy 14 zbiorów danych na zbiorze testowym, zilustrowane na rysunkach 8.23 oraz 8.24, również potwierdzają wcześniejsze obserwacje z pojedynczego zbioru Covid19 i w szerszym kontekście podkreślają wszechstronność oraz skuteczność metody Trinity SALT. Podobnie jak w przypadku zbioru *„Covid19”* czy *„Mushroom (R)”*, Trinity SALT osiąga znacznie wyższą średnią czułość na zestawie 14 zbiorów danych, przewyższając inne detektory bazowe. To wyraźnie świadczy o zdolności tego algorytmu do efektywnego wykrywania anomalii w różnorodnych zbiorach testowych, które wcześniej nie były widziane przez model.

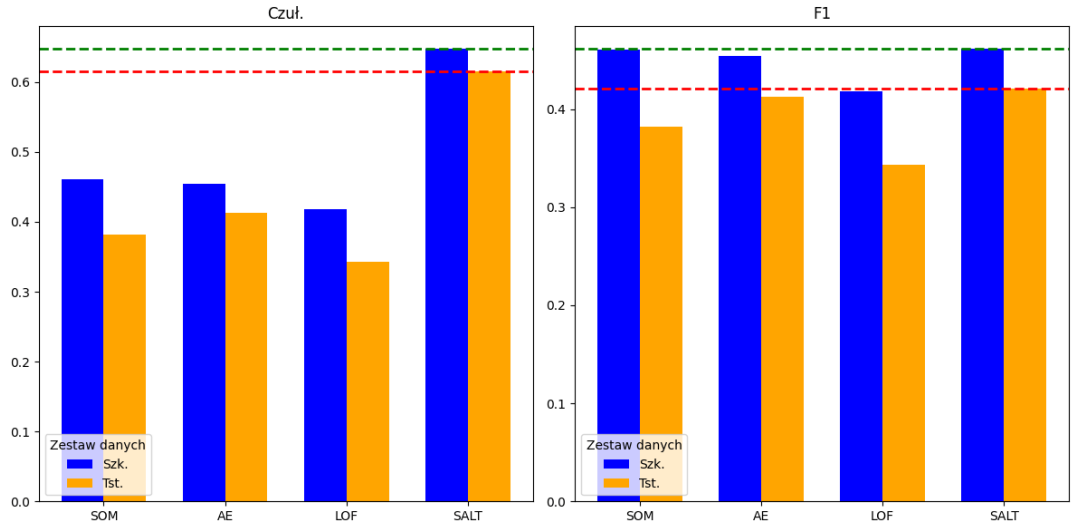


Rysunek 8.22: Porównanie uśrednionych metryk dla detektorów bazowych oraz zespołu Trinity SALT na zestawie 14 zbiorów danych (zbiór szkoleniowy). Czerwona przerywana linia wskazuje wartość Czułości, a jasnoniebieska przerywana linia oznacza wartość miary F1. Niebieski słupek oznacza Błąd klasyfikacji (Bł. kl.), zielony Specyficzność (Spec.), czerwony Precyzję (Prec.), fioletowy Czułość (Czuł.), pomarańczowy miarę F1, brązowy Fałszywie Pozytywne (FP), różowy Fałszywie Negatywne (FN), a szary Czas (Tm). Źródło: opracowanie własne.

Trinity SALT pozostaje jednym z najlepszych detektorów w każdej sytuacji, nawet gdy bazowe wyniki innych algorytmów są słabsze. Ta zdolność do utrzymania wysokiej wydajności na różnych zestawach danych sugeruje, że Trinity SALT może być solidnym wyborem w praktycznych zastosowaniach, gdzie dostępność danych referencyjnych jest ograniczona. Istnieje również możliwość dalszego poprawienia wyników Trinity SALT w przyszłości, na przykład poprzez włączenie dodatkowych metod bazowych. Chociaż Trinity SALT często działa najlepiej, różnica między nim a innymi detektorami w zestawach danych była czasami niewielka. Sugeruje to, że wersje zespołowe poszczególnych detektorów mogą osiągnąć swoje maksymalne możliwości, a łączenie ich z Trinity SALT może przynieść tylko niewielkie korzyści, choć z pewnością zwiększa stabilność wyników na różnych zbiorach danych. W praktyce mogą wystąpić sytuacje, w których jeden lub więcej detektorów może działać gorzej na specyficznym zbiorze danych. W takich przypadkach heterogeniczna kombinacja bazowych detektorów, jak Trinity SALT, zapewnia dodatkową ochronę przed nieprzewidywalnymi wynikami. Interpretowalność wyników oraz zrozumienie, które cechy danych prowadzą do identyfikacji anomalii, mogą w przyszłości odgrywać istotną rolę przy wyborze nowych algorytmów do zespołu Trinity lub przy zastępowaniu obecnych. Dzięki temu zespół algorytmów będzie bardziej elastyczny i lepiej dostosowany do specyficznych potrzeb różnych zbiorów danych. Analiza wyników

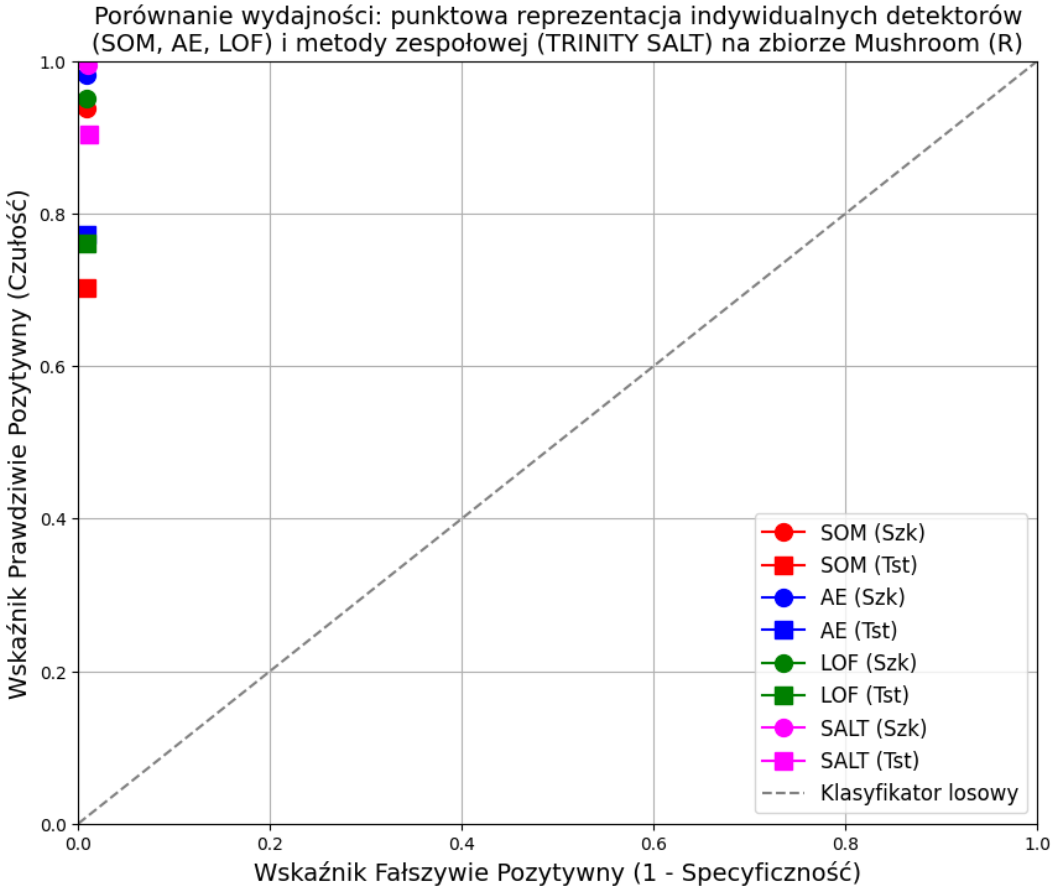


Rysunek 8.23: Porównanie uśrednionych metryk dla detektorów bazowych oraz zespołu Trinity SALT na zestawie 14 zbiorów danych (zbiór testowy). Czerwona przerywana linia wskazuje wartość Czułości, a jasnoniebieska przerywana linia oznacza wartość miary F1. Niebieski słupek oznacza Błąd klasyfikacji (Bł. kl.), zielony Specyficzność (Spec.), czerwony Precyzję (Prec.), fioletowy Czułość (Czuł.), pomarańczowy miarę F1, brązowy Fałszywie Pozytywne (FP), różowy Fałszywie Negatywne (FN), a szary Czas (Tm). Źródło: opracowanie własne.



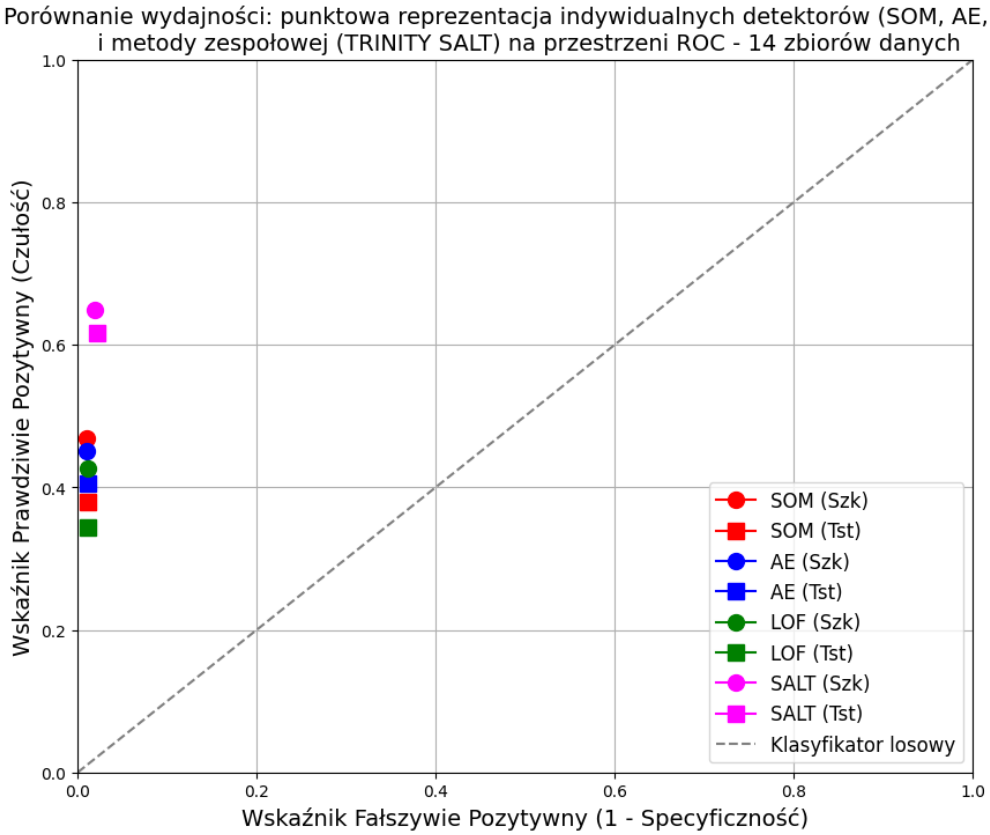
Rysunek 8.24: Porównanie uśrednionych czułości i miary F1 dla 14 zestawów danych z dużą nierównowagą klas na zbiorach szkoleniowych i testowych. Źródło: opracowanie własne.

dla 14 zbiorów danych, z uśrednionymi metrykami, pokazana na rysunku 8.24, koncentruje się na najważniejszych wskaźnikach — czułości i F1, zarówno dla zbiorów treningowych, jak i testowych. Wyniki te wyraźnie pokazują przewagę metody Trinity SALT nad innymi detektorami (SOM, AE, LOF). Na zbiorach treningowych Trinity SALT osiąga najwyższą czułość, co potwierdza jego zdolność do wykrywania rzeczywistych anomalii. Chociaż na zbiorach testowych czułość nieznacznie spada, metoda nadal przewyższa inne techniki, co świadczy o jej zdolności do skutecznej generalizacji na nowe, wcześniej niewidziane dane. Wskaźnik F1, równoważący czułość i precyzję, również jest najwyższy dla Trinity SALT na obu rodzajach zbiorów. Mimo że na danych testowych wynik F1 jest niższy, co zapewne wynika z tego, że są to dane wcześniej niewidziane przez model, Trinity SALT pozostaje najbardziej efektywną i stabilną metodą.



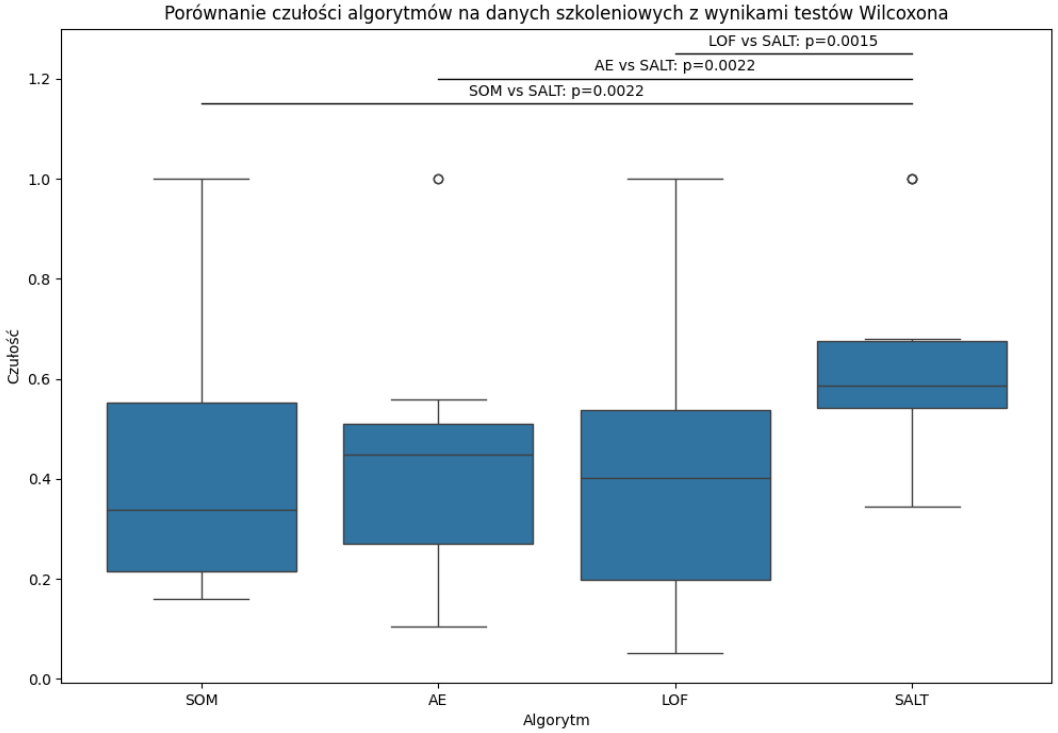
Rysunek 8.25: Porównanie wydajności: punktowa reprezentacja indywidualnych detektorów (SOM, AE, LOF) i metody zespołowej (Trinity SALT) na zbiorze *„Mushroom (R)”*. Źródło: opracowanie własne.

Analiza wyników przedstawionych na wykresie porównującym wydajność różnych algorytmów na zbiorze danych *„Mushroom (R)”* w przestrzeni ROC, co ilustruje rysunek 8.25 wskazuje na istotne różnice pomiędzy metodą zespołową Trinity SALT a innymi pojedynczymi detektorami (SOM, AE, LOF). Trinity SALT wyróżnia się wysoką czułością na zbiorze testowym, co wskazuje na jego zdolność do skuteczniejszej detekcji anomalii w porównaniu do SOM, AE i LOF. Choć na zbiorze treningowym AE osiąga taką samą czułość jak SALT, na zbiorze testowym SALT znacząco przewyższa inne algorytmy. To sugeruje, że Trinity SALT lepiej radzi sobie z uogólnieniem wyników na nieznanych danych. Wzrost czułości w Trinity SALT nie odbywa się kosztem specyficzności. Specyficzność pozostaje na poziomie zbliżonym do innych metod, co oznacza, że nie doszło do istotnego wzrostu liczby fałszywie pozytywnych wyników. To sprawia, że jest skuteczniejsza od pojedynczych detektorów w identyfikacji anomalii.



Rysunek 8.26: Porównanie wydajności: punktowa reprezentacja indywidualnych detektorów (SOM, AE, LOF) i metody zespołowej (Trinity SALT) w przestrzeni ROC - 14 zbiorów danych. Źródło: opracowanie własne.

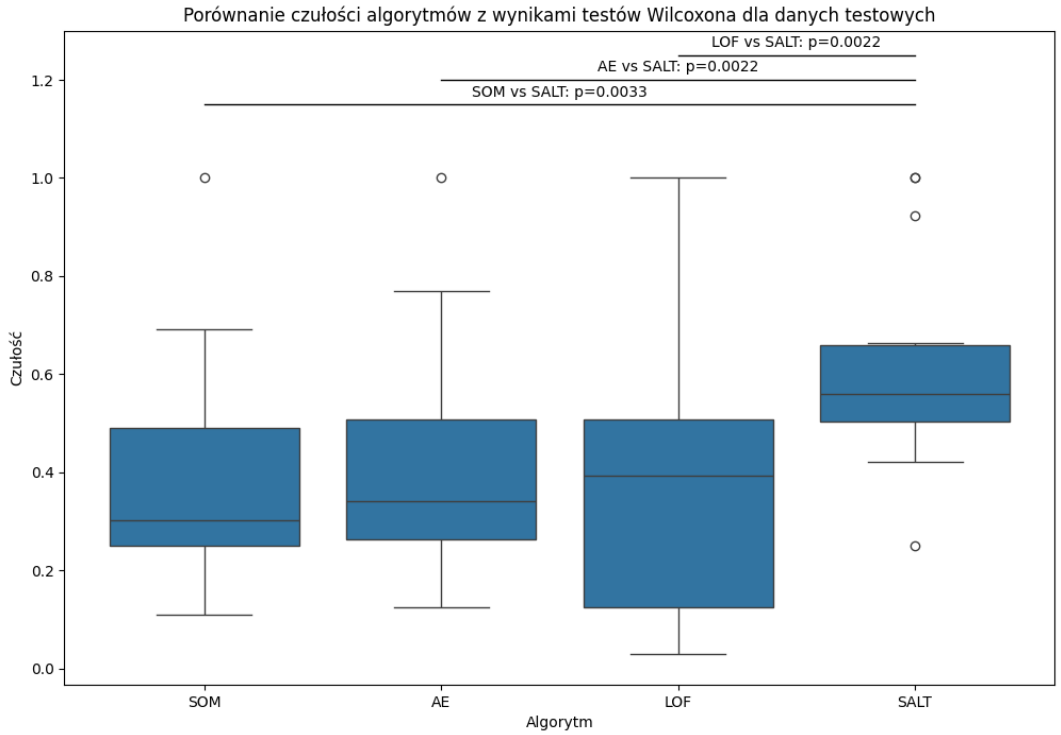
Podobne wyniki można zaobserwować w analizie 14 zbiorów danych, przedstawionej na wykresie ROC, co przedstawia rysunek 8.26. Uśrednione metryki wyraźnie pokazują przewagę Trinity SALT nad detektorami SOM, AE i LOF. Trinity SALT osiąga wyższą czułość na zbiorach szkoleniowym i testowym, co sugeruje jego większą skuteczność w wykrywaniu anomalii, choć kosztem nieco niższej specyficzności. W porównaniu do pozostałych algorytmów Trinity SALT lepiej generalizuje, co widać w stabilnych wynikach na zbiorze testowym. Algorytmy SOM, AE i LOF wykazują większe różnice w czułości między zbiorami szkoleniowymi a testowymi, co może wskazywać na ich mniejsze zdolności do pracy z różnorodnymi danymi.



Rysunek 8.27: Porównanie czułości algorytmów na danych szkoleniowych z wynikami testów Wilcoxona. Źródło: opracowanie własne.

Wyniki przedstawione na rysunku 8.27 pokazują, że metoda Trinity SALT znacząco przewyższa inne algorytmy, takie jak SOM, AE i LOF, pod względem czułości. Test Wilcoxona potwierdza istotność statystyczną tych różnic, co wskazuje na wyraźną przewagę SALT w detekcji anomalii. Zastosowanie tej metody pozwala na osiągnięcie wyższej czułości, co jest ważne w kontekście rzadkich i trudnych do wykrycia przypadków anomalii. Mimo że wiąże się to z pewnym zwiększeniem liczby fałszywych pozytywów, w sytuacjach, gdzie kluczowe jest wykrycie każdej anomalii, Trinity SALT oferuje bardziej zrównoważone i niezawodne podejście. Test Wilcoxona jest szczególnie odpowiedni

w tym kontekście, ponieważ uwzględnia zależność między wynikami różnych algorytmów testowanych na tych samych zbiorach danych. Dzięki temu możemy precyzyjnie ocenić, czy różnice w czułości między metodą Trinity SALT a innymi algorytmami są statystycznie istotne. Co więcej, test Wilcoxona nie zakłada normalności rozkładów, co czyni go bardziej elastycznym i wiarygodnym w przypadku danych o nieznanym lub niezgodnym z normalnością rozkładzie, co często ma miejsce w analizie anomalii. Tym samym wybór tego testu pozwala na uzyskanie bardziej rzetelnych wyników i podkreśla wyższość zespołu Trinity SALT w specyficznych warunkach badawczych.



Rysunek 8.28: Porównanie czułości algorytmów z wynikami testów Wilcoxona dla danych testowych. Źródło: opracowanie własne.

Na rysunku 8.28 przedstawiono porównanie czułości czterech algorytmów: SOM, AE, LOF oraz Trinity SALT, zastosowanych na danych testowych. Wyraźnie widoczna jest przewaga metody Trinity SALT, która osiąga znacznie wyższą czułość w porównaniu do pozostałych metod. Wykres pudełkowy przedstawia medianę wyników dla każdego algorytmu oraz zakresy wyników, co umożliwia głębszą analizę ich stabilności i zdolności do wykrywania anomalii. Algorytm Trinity SALT cechuje się wysoką czułością, co jest istotne przy wykrywaniu rzadkich i trudnych do identyfikacji przypadków. Zastosowanie testu Wilcoxona pokazuje, że różnice między Trinity SALT a pozostałymi algorytmami są statystycznie istotne. P-wartości są znacznie niższe niż typowy poziom istotności 0,05, co

wyraźnie wskazuje na przewagę Trinity SALT. Zespół Trinity SALT nie tylko przewyższa inne algorytmy pod względem czułości, ale również wykazuje większą spójność wyników, co sugeruje, że jest bardziej niezawodny w różnych scenariuszach testowych. W praktyce oznacza to, że Trinity SALT jest lepiej przygotowany do pracy w warunkach rzeczywistych, gdzie wykrywanie anomalii jest fundamentalne dla bezpieczeństwa i stabilności systemów. Wysoka czułość osiągnięta przez Trinity SALT wynika z jego unikalnej konstrukcji, która integruje wyniki z różnych konfiguracji algorytmów, minimalizując ryzyko przeoczenia anomalii.

Przechodząc do szczegółowej analizy bazowych algorytmów wchodzących w skład zespołu Trinity SALT (szczegółowe wyniki znajdują się w Dodatku B 9.1), dokonano oceny ich skuteczności na podstawie uśrednionych wyników uzyskanych z 14 różnych zbiorów danych. Należy jednak zaznaczyć, że prezentowane wartości procentowe stanowią uśrednienie, będące wypadkową wyników z poszczególnych zbiorów – niektóre algorytmy osiągnęły bardzo wysoką skuteczność i wydajność, podczas gdy inne wykazywały znacznie słabsze rezultaty, w zależności od specyfiki danych. W związku z tym, średnie wartości należy traktować jako uogólnienie, które nie w pełni oddaje zróżnicowanie wyników indywidualnych zbiorów, gdyż specyficzne właściwości danych mogą znacząco wpływać na efektywność algorytmów w konkretnych scenariuszach. Niemniej jednak, uśrednione wyniki dostarczają ogólnego obrazu i trendów w skuteczności badanych metod. Uśrednione wyniki przedstawione w tabeli 8.13 oraz na rysunkach 8.22 i 8.23 wskazują, że algorytm SOM uzyskuje nieco lepsze wyniki pod względem czułości na zbiorze szkoleniowym, jednak różnice w porównaniu do AE są niewielkie. SOM osiąga czułość na poziomie 46,09%, podczas gdy AE uzyskuje 45,43%. Na zbiorze testowym AE wykazuje przewagę nad SOM w zakresie czułości (41,31% dla AE w porównaniu do 38,20% dla SOM), co sugeruje, że AE lepiej radzi sobie z wykrywaniem anomalii na nowych, wcześniej nieznanych danych. Pod względem czułości, wyniki uzyskane przez SOM i AE są zbliżone zarówno na zbiorze szkoleniowym (46,09% dla SOM kontra 45,43% dla AE), jak i testowym (38,20% dla SOM kontra 41,31% dla AE). Różnice w czułości są minimalne, co wskazuje na porównywalną efektywność obu algorytmów. Natomiast LOF uzyskuje najniższe wyniki w czułości w porównaniu z SOM i AE, co czyni go mniej skutecznym w kontekście analizowanych 14 zbiorów danych. Warto jednak zauważyć, że LOF, pomimo niższych wyników uśrednionych, w niektórych przypadkach osiąga wyniki zbliżone lub nawet lepsze od SOM i AE, co podkreśla jego potencjalną wartość w specyficznych scenariuszach. Czas trenowania jest również istotnym czynnikiem wpływającym na praktyczność algorytmów. AE wymaga znacznie dłuższego czasu trenowania (172,20 sekundy na zbiorze szkoleniowym w porównaniu do 1,35 sekundy dla SOM), co może stanowić ograniczenie przy analizie dużych zbiorów danych. Algorytmy SOM i LOF działają znacznie szybciej, co może stanowić istotną zaletę w zastosowaniach wymagających szybkich obliczeń. Szczególnie LOF wyróżnia się pod względem wydajności czasowej dzięki zastosowaniu nietypowego dla tego algorytmu rozwiązania podziału na bloki (który w swojej standardowej wersji ma

złożoność obliczeniową $O(n^2)$), co umożliwia równoległe przetwarzanie danych i znacząco skraca czas obliczeń, zwłaszcza w przypadku dużych zbiorów danych. Podsumowując, choć SOM wykazuje nieco wyższą skuteczność na zbiorze szkoleniowym, AE oferuje lepszą czułość i generalizację na zbiorze testowym. Oba algorytmy są wartościowe, jednak AE może okazać się potencjalnie lepszy, zwłaszcza gdy badanie dotyczy nowych, nieznanych danych. LOF, mimo najsłabszej skuteczności pod względem uśrednionej czułości, pozostaje atrakcyjny ze względu na krótki czas wykonania, co może stanowić decydujący czynnik w zastosowaniach wymagających szybkiego działania.

Algorytm Trinity SALT może znaleźć szerokie zastosowanie w różnych dziedzinach, takich jak medycyna, gdzie może służyć do wykrywania pacjentów z potencjalnie niebezpiecznymi zaburzeniami zdrowotnymi, w finansach do identyfikacji podejrzanych transakcji mogących wskazywać na oszustwa, oraz w inżynierii do monitorowania stanu maszyn i wykrywania awarii. Aby skutecznie zastosować algorytm Trinity SALT, konieczne jest przyjęcie pewnych założeń dotyczących oczekiwanej liczby anomalii w sytuacjach, gdy brakuje pełnej wiedzy na temat danych referencyjnych (ground truth), co jest często spotykane w rzeczywistym świecie. W praktyce, gdy brak jest precyzyjnych informacji o rzeczywistej liczbie anomalii, można przyjąć, że określony procent próbek w zbiorze danych stanowi anomalie. Na przykład, w przypadku założenia, że 5% populacji pacjentów może być dotkniętych chorobą, Trinity SALT należy skalibrować tak, aby identyfikował co najmniej 5% najbardziej podejrzanych przypadków. Alternatywnie, możliwe jest zastosowanie reguły trzech sigm 3σ, zakładając, że wykrywane będą najbardziej ekstremalne przypadki — czyli te, które znajdują się poza 99,7% populacji, co odpowiada około 0,3% danych. W niniejszym badaniu skupiono się na wykrywaniu najbardziej ekstremalnych przypadków, co odpowiadało około 1% analizowanego zbioru danych. Początkowe założenia dotyczące liczby wykrywalnych odchyleń mogą opierać się na wiedzy eksperckiej, wcześniejszych analizach lub danych historycznych. W miarę postępu badań, możliwe jest iteracyjne dostosowanie tych założeń, aby uzyskać bardziej realistyczne i przydatne wyniki. Takie podejście pozwala algorytmowi Trinity SALT skutecznie identyfikować najbardziej narażonych pacjentów, najbardziej podejrzane transakcje finansowe oraz krytyczne anomalie w systemach inżynieryjnych, zapewniając jednocześnie wysoką czułość przy zachowaniu kontrolowanej precyzji.

8.3 Podsumowanie i wnioski z eksperymentów

Eksperymenty przeprowadzone w ramach rozprawy dostarczają silnych dowodów na skuteczność optymalizacji bayesowskiej w kontekście modelowania niepewności oraz precyzyjnego dostosowywania hiperparametrów algorytmów, takich jak SOM, AE i LOF. Zaimplementowany system Trinity SALT, łączący sieć samouczącą się SOM, autoenkoder AE oraz algorytm gęstościowy LOF, wykazuje znaczące korzyści wynikające z zastosowania metod zespołowych, które poprawiają wydajność, stabilność i odporność na szumy

w porównaniu do pojedynczych modeli, a jednocześnie stanowi nowatorskie podejście w dziedzinie wykrywania anomalii. Takie podejście znacząco redukuje nadmierne dopasowanie i zwiększa zdolność generalizacji modelu. Chociaż metody zespołowe są szeroko badane w grupowaniu, rekomendacji i klasyfikacji, ich zastosowanie w detekcji anomalii jest stosunkowo nowe i rozwija się intensywnie dopiero od kilku lat, co czyni je mniej zbadanymi w porównaniu do innych obszarów eksploracji danych.

Jak powszechnie wiadomo, tradycyjna metoda LOF, wymagająca przeszukiwania całego zbioru danych, charakteryzuje się wysoką złożonością obliczeniową, co może prowadzić do wydłużonych czasów przetwarzania, zwłaszcza w przypadku dużych zbiorów danych. W odpowiedzi na te wyzwania zaproponowano innowacyjne podejście optymalizacji rozmiaru bloku w algorytmie LOF, co nie tylko skraca czas obliczeń, ale także zachowuje wysoką skuteczność wykrywania anomalii. Heterogeniczny zespół detektorów anomalii Trinity SALT, dzięki odpowiednio dobranym silnym algorytmom bazowym i zastosowaniu optymalizacji bayesowskiej, umożliwił dynamiczne dostosowanie hiperparametrów do specyfiki analizowanych danych i dostępnych zasobów sprzętowych, co pozwala na skuteczne i wydajne wykrywanie anomalii. Ponadto, w przypadku algorytmu LOF wykorzystano zaproponowany podział na bloki, co znacząco wpłynęło na ogólną wydajność przetwarzania. System okazał się również skuteczny w konfrontacji z ekstremalnymi problemami nierównowagi klas, gdzie anomalie stanowią zaledwie ułamek procenta danych, co dodatkowo podkreśla jego zdolność do radzenia sobie z wyjątkowo trudnymi scenariuszami. Trinity SALT wykazał się wysoką czułością, co jest niezbędne w wykrywaniu rzadkich anomalii, a jednocześnie utrzymał miarę F1 na zadowalającym poziomie, co wskazuje na dobrą równowagę między czułością a precyzją. To potwierdza, że model skutecznie identyfikuje rzeczywiste przypadki pozytywne bez znaczącej utraty precyzji, co jest szczególnie ważne w kontekście rzeczywistych zastosowań, takich jak diagnostyka medyczna, gdzie ważne jest minimalizowanie błędnych alarmów i przeoczeń. Najważniejsze wyniki i wnioski z przeprowadzonych eksperymentów — w tym dotyczące optymalizacji rozmiaru bloku oraz oceny skuteczności zespołu Trinity SALT — zostały zestawione w tabeli 8.14. Szczegółowy opis badań dodatkowych znajduje się w załączniku 9.1, zatytułowanym „Wykaz badań dodatkowych". Opisy eksperymentów poniżej pogłębiają zrozumienie i ilustrują ważne aspekty optymalizacji LOF i systemu Trinity SALT.

W eksperymencie nr 1 szczególną uwagę zwrócono na wpływ optymalizacji rozmiaru bloku na skuteczność algorytmu w wykrywaniu anomalii. Okazało się, że podział danych na mniejsze bloki nie tylko skrócił czas przetwarzania, ale także często poprawił zdolność algorytmu do identyfikacji anomalii. W przypadkach, gdzie ograniczenia sprzętowe, takie jak dostępna pamięć RAM, wymusiły redukcję rozmiaru analizowanych danych, optymalizacja ta pozwoliła na efektywne wykorzystanie zasobów obliczeniowych, co ma kluczowe znaczenie przy analizie dużych zbiorów danych. Wskazuje to na konieczność precyzyjnego dostosowania parametrów algorytmu do specyfiki zbioru danych i dostępnych zasobów.

Tabela 8.14: Podsumowanie eksperymentów dotyczących rozmiaru bloku w LOF, optymalizacji bayesowskiej oraz zespołu Trinity SALT

Nr eksperymentu	Cel eksperymentu	Wyniki i wnioski
1	Optymalizacja rozmiaru bloku w LOF	Zmniejszenie rozmiaru bloku w algorytmie LOF poprawiło zarówno czas przetwarzania, jak i skuteczność wykrywania anomalii. Dostosowanie rozmiaru bloku minimalizuje odległości między obiektami, co usprawnia lokalną analizę danych. Pomimo ograniczeń sprzętowych, optymalizacja umożliwiła efektywne wykorzystanie dostępnych zasobów.
2	Adaptacyjna optymalizacja rozmiaru bloku	Wprowadzenie dynamicznej optymalizacji rozmiaru bloku oraz dostosowanie hiperparametrów algorytmu LOF do specyfiki danych i dostępnych zasobów zwiększyło jego efektywność. Pozwoliło to na skrócenie czasu obliczeń i zwiększenie skuteczności detekcji anomalii poprzez efektywne zarządzanie zasobami obliczeniowymi.
3	Ocena zespołu Trinity SALT	Metoda Trinity SALT wykazała wysoką skuteczność i stabilność w identyfikacji anomalii na różnych zestawach danych, przewyższając inne detektory. Zespół osiągnął wyższą czułość w wykrywaniu anomalii, utrzymując dobry balans między czułością a precyzją i wysokie wartości miary F1. Optymalizacja bayesowska precyzyjnie dostosowała hiperparametry, poprawiając zarządzanie zasobami i skuteczność modelu na danych testowych i szkoleniowych.

Eksperyment nr 2 potwierdził, że dynamiczne dostosowywanie nie tylko rozmiaru bloku, ale także innych hiperparametrów, takich jak liczba najbliższych sąsiadów czy wybór metryki odległości, umożliwia jeszcze efektywniejsze zarządzanie zasobami sprzętowymi. Badania wykazały również, że algorytm LOF, pierwotnie zaprojektowany do analizy danych numerycznych, może być z powodzeniem adaptowany do przetwarzania danych kategorycznych poprzez technikę one-hot encoding, co znacząco rozszerza jego potencjalne zastosowania. Praktyczne zastosowanie tej optymalizacji można zaobserwować na przykładzie dużych zbiorów, takich jak *„Credit Card”* i *„Vehicle Claims”*, gdzie dzięki efektywnemu podziałowi na mniejsze bloki czas przetwarzania pozostaje stosunkowo krótki. W przeciwieństwie do tego, zbiór *„Chess KRKOPT”*, mimo mniejszego

rozmiaru, wymaga znacznie dłuższego czasu przetwarzania z powodu braku podziału na bloki, co zmusza algorytm do analizy całego zbioru jako jednego bloku. To podkreśla, jak istotne jest odpowiednie dostosowanie rozmiaru bloku w kontekście optymalizacji czasu przetwarzania. Mniejsze zbiory, takie jak *„Wine Quality”* i *„Mushroom Real”*, mimo zastosowania większych bloków, są przetwarzane szybko dzięki swoim niewielkim rozmiarom, co pokazuje, że czas przetwarzania zależy zarówno od rozmiaru bloku, jak i wielkości zbioru. Jednak nawet małe zbiory mogą wymagać znacznego czasu na przetwarzanie, gdy przypisane im zostaną duże bloki lub gdy są analizowane w całości. Te wyniki pokazują, że odpowiednie dostosowanie parametrów algorytmu, takich jak rozmiar bloku, jest bardzo ważne dla efektywnego przetwarzania i zarządzania zasobami sprzętowymi w różnorodnych kontekstach analizy danych.

Druga część badań, określona jako Eksperyment 3, koncentrowała się na ocenie zespołu detektorów Trinity SALT, wykorzystując doświadczenia z poprzednich eksperymentów. Zastosowanie podejścia polegającego na podziale danych na mniejsze bloki dla algorytmu LOF zoptymalizowało zarządzanie zasobami obliczeniowymi i poprawiło wydajność analizy dużych zbiorów danych. Dzięki optymalizacji bayesowskiej precyzyjnie dostosowano hiperparametry algorytmów bazowych, co przyczyniło się do zwiększenia efektywności zespołu. System Trinity SALT wyróżnia się również dzięki autorskiej technice maksymalnej znormalizowanej agregacji MNA, która premiuje konsensus modeli, co dodatkowo wzmacnia jego skuteczność w porównaniu z tradycyjnymi metodami.

Wyniki wskazują, że Trinity SALT osiągnął wysoką czułość zarówno na danych szkoleniowych, jak i testowych, co potwierdza jego skuteczność w zróżnicowanych warunkach. Na przykład w zbiorze *„Mushroom (R)”*, Trinity SALT osiągnął czułość na poziomie 92,31% i miarę F1 wynoszącą 72,73%, co potwierdza jego zdolność do skutecznego wykrywania anomalii bez znaczącej utraty precyzji. W zbiorze *„Breast cancer (R)”*, algorytm uzyskał czułość na poziomie 100% oraz miarę F1 wynoszącą 57,14%, co wskazuje na jego zdolność do skutecznego wykrywania anomalii, nawet w obliczu rzadkości występowania tych przypadków. W przypadku syntetycznego zbioru *„Citibike Synthetic (S)”*, Trinity SALT osiągnął doskonałe wyniki, uzyskując 100% czułości i miarę F1 na obu zbiorach, szkoleniowym i testowym, co dowodzi jego zdolności do bezbłędnego wykrywania anomalii w danych. Chociaż algorytm nie zawsze osiągał najlepsze wyniki na wszystkich zbiorach, w uśrednionych wynikach z 14 zestawów danych model wykazywał stosunkowo wysoką czułość i zrównoważony stosunek precyzji do czułości, co przekłada się na wysokie wartości miary F1. Jest to istotne w krytycznych zastosowaniach, takich jak diagnostyka medyczna, cyberbezpieczeństwo i detekcja oszustw, gdzie priorytetem jest szybkie wykrywanie i minimalizowanie ryzyka przeoczenia istotnych przypadków.

Eksperymenty na ekstremalnie niezrównoważonych zbiorach danych (o stosunku klas nawet 1:999) potwierdzają, że Trinity SALT skutecznie radzi sobie z dużymi dysproporcjami między klasami, gdzie tradycyjne metody zawodzą. Badania nad tak ekstremalnymi przypadkami są rzadko prowadzone i stanowią istotną lukę w literaturze, ponieważ więk-

szość prac skupia się na mniej skrajnych scenariuszach nierównowagi klas. Złożoność problemu, jaką stwarzają takie ekstremalne proporcje, wymaga zaawansowanych podejść, które są wciąż niedostatecznie eksplorowane. Dodatkowym wyzwaniem są zbiory danych zawierające kategoryczne i mieszane cechy, które mogą komplikować proces klasyfikacji i detekcji anomalii. Wysoka wymiarowość takich danych dodatkowo utrudnia identyfikację anomalii, zwłaszcza gdy algorytmy muszą radzić sobie zarówno z cechami liczbowymi, jak i jakościowymi. Wykrywanie wartości odstających często koncentruje się na danych ciągłych i przestrzeni numerycznej, pomijając dane jakościowe, co może prowadzić do wskazania zupełnie innych obiektów jako potencjalnie nietypowych. To sprawia, że klasyczne algorytmy mogą nie radzić sobie z analizą kategorycznych danych, co podkreśla konieczność stosowania zaawansowanych technik, takich jak Trinity SALT, które są przystosowane do pracy z różnorodnymi typami danych, skutecznie radząc sobie z wyzwaniami związanymi z wysoką wymiarowością i złożonością analizowanych zbiorów. Dzięki swojej elastyczności, innowacyjnym technikom i precyzyjnemu dostosowaniu parametrów, Trinity SALT stanowi potężne narzędzie do identyfikacji anomalii w różnych krytycznych dziedzinach, od zdrowia publicznego po cyberbezpieczeństwo, przyczyniając się do poprawy bezpieczeństwa i efektywności systemów.

Wyniki eksperymentów podkreślają ogromne znaczenie zaawansowanych technik optymalizacji, takich jak optymalizacja bayesowska, w procesie analizy danych. Zastosowanie tych technik, w tym optymalizacji rozmiaru bloku i dostosowania hiperparametrów, zwiększa wydajność algorytmów i umożliwia oszczędne wykorzystanie zasobów obliczeniowych. Te rezultaty wskazują na potrzebę dalszych badań nad alternatywnymi metodami optymalizacji, aby jeszcze bardziej usprawnić proces wykrywania anomalii w złożonych zbiorach danych. Metoda Trinity SALT okazała się skuteczna w identyfikacji prawdziwie pozytywnych przypadków, co jest najważniejsze w zastosowaniach wymagających wysokiej czułości. Trinity SALT reprezentuje nowatorskie podejście, które aktywnie przyczynia się do rozwoju analizy zespołowej w wykrywaniu anomalii, wprowadzając innowacyjne techniki, które zwiększają skuteczność tych systemów. Wprowadzenie tej metody może zainspirować dalsze badania nad rozwojem zaawansowanych, zespołowych technik wykrywania odchyleń. Wyniki otwierają nowe perspektywy dla dalszych badań nad rozwijaniem efektywnych i dokładnych metod detekcji anomalii, sugerując kierunki, które mogą przyczynić się do poprawy analizy danych w różnych krytycznych obszarach. Podsumowując omawiane kwestie, wprowadzenie innowacyjnych metod i technik analizy zespołowej w identyfikacji anomalii otwiera nowe horyzonty, jednocześnie stawiając wyzwania związane z brakiem formalizacji i danych referencyjnych. Nowe podejścia wymagają głębszych badań i kreatywnych rozwiązań, aby w pełni wykorzystać potencjał analizy zespołowej. Rozwój tych metod w praktyce może zwiększyć niezawodność i efektywność systemów wykrywania anomalii, co jest ważne w kontekście powstających coraz większych, bardziej złożonych i dynamicznych zbiorów danych.

Rozdział 9

Podsumowanie

W niniejszym rozdziale podsumowano najważniejsze wnioski wynikające z przeprowadzonych badań, omówiono znaczenie uzyskanych rezultatów oraz przedstawiono rekomendacje i potencjalne kierunki przyszłych badań w zakresie detekcji anomalii w złożonych zbiorach danych. Wyniki zaprezentowane w rozprawie potwierdzają skuteczność proponowanych metod i wskazują na ich potencjał w realnych zastosowaniach, jednocześnie udowadniając postawioną tezę, że zaawansowane techniki zespołowe mogą znacząco zwiększyć czułość i wydajność detekcji anomalii.

Obszerność pracy jest uzasadniona kompleksowym podejściem do problematyki detekcji anomalii w złożonych zbiorach danych. Struktura pracy została starannie zaprojektowana, aby dogłębnie przeanalizować wszystkie kluczowe aspekty tego zagadnienia, od teoretycznych podstaw po szczegółowe opisy eksperymentów i wdrożenia praktyczne. Praca omawia trzy algorytmy: sieci samouczące SOM, autoenkodery AE oraz algorytm gęstościowy LOF, z których każdy mógłby stanowić temat osobnej pracy naukowej. Ich połączenie w zespół Trinity SALT tworzy dodatkową wartość, wymagającą szczegółowej analizy i optymalizacji. Choć metody zespołowe są szeroko badane w kontekście grupowania, rekomendacji i klasyfikacji, ich zastosowanie w detekcji anomalii jest stosunkowo nowe i wciąż nie w pełni zbadane, co uzasadnia podjęcie tego tematu. Rozprawa stanowi tym samym ciekawy wkład w rozwój tego podejścia, poszerzając wiedzę na temat ich efektywności i możliwości zastosowania w wykrywaniu anomalii. Rozdziały pracy szczegółowo przedstawiają różnorodne metody i podejścia, co jest niezbędne, aby w pełni zrozumieć złożoność tematu oraz przedstawić nowatorskie rozwiązania i ich skuteczność w praktyce.

9.1 Znaczenie wyników i przyszłe badania

Wyniki uzyskane w tej rozprawie mają znaczenie zarówno dla rozwoju teoretycznego metod wykrywania anomalii, jak i dla ich praktycznego zastosowania w różnych dziedzinach, gdzie analiza dużych zbiorów danych odgrywa ważną rolę. Przeprowadzone eksperymenty dostarczyły istotnych dowodów na skuteczność zaawansowanych technik wykrywania anomalii, szczególnie w kontekście trudnych w analizie, złożonych zbiorów danych kategorycznych, mieszanych i wielowymiarowych z nierównomierną reprezentacją klas. W niektórych przypadkach zbiory danych w rozprawie zbliżają się do ekstremalnych przypadków nierównowagi, co jest istotne, gdyż takie scenariusze pozostają słabo zbadane w literaturze. Rozprawa przyczynia się do lepszego zrozumienia tych wyzwań, pokazując, że klasyczne metody często zawodzą przy tak skrajnych dysproporcjach, co podkreśla znaczenie rozwijania nowych podejść, które mogą skutecznie radzić sobie z ekstremalną nierównowagą klas.

System Trinity SALT, który integruje różnorodne podejścia, takie jak sieci samouczące SOM, autoenkodery AE oraz algorytm gęstościowy LOF, wykazał skuteczność w identyfikacji anomalii, co potwierdzają wyniki osiągnięte na rzeczywistych zbiorach danych. Zastosowanie autorskiej techniki maksymalnej znormalizowanej agregacji MNA wraz z premiowaniem konsensusu modeli przy integracji wyników oraz optymalizacją bayesowską umożliwiło precyzyjne dostosowanie hiperparametrów do specyfiki danych, co przekłada się na wyższą skuteczność w porównaniu do tradycyjnych metod. Potwierdza to, że zaproponowane podejście spełnia założenia tezy pracy, udowadniając, że zaawansowane zespołowe techniki analizy mogą skutecznie radzić sobie z problemami związanymi z nierównowagą klas i złożonością danych.

Ważnym elementem rozprawy jest zaproponowana autorska definicja anomalii, która podkreśla unikalność i złożoność tego zjawiska. Według tej definicji, anomalia to unikalny wzorzec lub zestaw wzorców w danych, który znacząco odbiega od przewidywanego zachowania i jest identyfikowany dzięki zaawansowanym technikom uczenia maszynowego i analizy wielowymiarowej. Taka definicja wskazuje na potencjalne znaczenie odkrycia anomalii w kontekście ujawniania nowych, wcześniej nieznanych mechanizmów lub zjawisk, co może prowadzić do głębszego zrozumienia analizowanych systemów. Wprowadzenie tej definicji w pracy nie tylko pomaga lepiej zrozumieć naturę anomalii, ale także uzasadnia konieczność stosowania zaawansowanych metod detekcji, które zostały zaprezentowane i przetestowane w badaniach.

Szczególnie istotne jest, że w skrajnych warunkach nierównowagi klas Trinity SALT utrzymał stosunkowo wysoką czułość, biorąc pod uwagę wyzwania, przed którymi stanął, przy jednoczesnym zachowaniu satysfakcjonującego poziomu miary F1. To czyni go efektywnym narzędziem w krytycznych obszarach, takich jak medycyna i cyberbezpieczeństwo. Wysoka czułość jest nieodzowna, ponieważ umożliwia wykrycie jak największej liczby istotnych przypadków, minimalizując ryzyko przeoczenia rzadkich, ale potencjalnie

groźnych anomalii. Ma to wyjątkowe znaczenie w diagnostyce chorób nowotworowych oraz w zarządzaniu kryzysami zdrowotnymi, takimi jak pandemia COVID-19. W przypadku COVID-19 wartości odstające mogą być stosunkowo rzadkie - na przykład spośród ponad 11 800 zgonów zgłoszonych w Nowym Jorku do końca kwietnia 2020 roku tylko 10 było w wieku poniżej 44 lat i nie miało żadnej choroby podstawowej [1]. Nawet jeśli wysoka czułość skutkuje większą liczbą fałszywych alarmów, jest to akceptowalny kompromis w kontekście medycyny, gdzie przeoczenie kluczowej anomalii może mieć poważne konsekwencje dla pacjenta. Przykłady te jasno pokazują, że szybkie i precyzyjne wykrywanie nietypowych wzorców w danych medycznych bezpośrednio przekłada się na lepsze wyniki kliniczne, skuteczniejsze leczenie oraz sprawniejsze zarządzanie zdrowiem publicznym. Dzięki skutecznej detekcji anomalii można w porę zidentyfikować istotne zmiany chorobowe, co minimalizuje ryzyko przeoczenia oraz pozwala na szybką reakcję, co jest ważne zarówno dla indywidualnych pacjentów, jak i dla społeczeństwa w sytuacjach kryzysowych, takich jak epidemie.

Wyniki eksperymentów na różnych zestawach danych wskazują, że dynamiczne dostosowywanie rozmiaru bloku, unikalnie zastosowane w algorytmie LOF oraz optymalizacja innych hiperparametrów, umożliwia optymalne zarządzanie zasobami sprzętowymi, nawet przy ograniczeniach pamięciowych. To nowatorskie rozwiązanie zaproponowane w rozprawie nie tylko usprawnia działanie samego algorytmu LOF, ale również znacząco poprawia wydajność całego zespołu Trinity SALT. Optymalizuje przetwarzanie danych i utrzymuje wysoką skuteczność identyfikacji anomalii, a w niektórych przypadkach nawet ją zwiększa, co jest niezbędne w aplikacjach wymagających szybkiej reakcji.

Najważniejsze wnioski z badań obejmują skuteczność metod zespołowych, takich jak Trinity SALT, który, łącząc różne techniki, znacząco poprawia czułość i stabilność wyników w detekcji anomalii, przewyższając pojedyncze algorytmy. Innowacyjna metoda MNA dodatkowo wzmacnia skuteczność systemu, premiując konsensus modeli, co zostało potwierdzone w eksperymentach. Istotne znaczenie ma również optymalizacja rozmiaru bloku w algorytmie LOF, która skraca czas przetwarzania i utrzymuje zdolność do identyfikacji anomalii na podobnym poziomie, a w niektórych przypadkach nawet ją zwiększa, co jest szczególnie ważne przy pracy z dużymi zbiorami danych. Dynamiczne dostosowywanie hiperparametrów w algorytmie LOF do specyfiki danych, w tym liczby sąsiadów i wyboru metryki, nie tylko utrzymuje wysoką skuteczność wykrywania anomalii, ale także zawsze zwiększa wydajność, co zostało udowodnione zarówno w kontekście wielowymiarowych danych numerycznych, jak i kategorycznych. Wykrywanie anomalii zazwyczaj koncentruje się na danych ciągłych z atrybutami o wartościach numerycznych, podczas gdy dane kategoryczne często pozostają niewykorzystane. Tradycyjne metody mają tendencję do ignorowania danych jakościowych, skupiając się wyłącznie na analizie przestrzeni liczbowej. W rezultacie, zupełnie różne obiekty mogą być identyfikowane jako

[1] https://www.nyc.gov/assets/doh/downloads/pdf/imm/covid-19-daily-data-summary-deaths-04282020-1.pdf

potencjalne anomalie. Pomijanie kategorycznych atrybutów powoduje utratę wielu istotnych informacji, które mogłyby znacząco poprawić jakość detekcji. System Trinity SALT uwzględnia ten typ danych, integrując zarówno numeryczne, jak i kategoryczne atrybuty, co pozwala na bardziej precyzyjną identyfikację anomalii i zwiększa ogólną skuteczność zespołu. System Trinity SALT wykazał się także wyjątkową zdolnością do radzenia sobie z ekstremalną nierównowagą klas, co jest poważnym wyzwaniem dla tradycyjnych metod detekcji. Badania wykazały również, że klasyczne algorytmy mają ograniczoną skuteczność w analizie danych kategorycznych i wielowymiarowych, co podkreśla potrzebę stosowania zaawansowanych metod, takich jak Trinity SALT, które skutecznie radzą sobie z wyzwaniami wynikającymi z wysokiej wymiarowości i złożoności danych.

Zespół detektorów Trinity SALT, dzięki swojej elastyczności i zdolności do efektywnej detekcji anomalii, ma potencjalne zastosowania w wielu kluczowych obszarach. W diagnostyce medycznej Trinity SALT może skutecznie wykrywać anomalie w danych pacjentów, co przyczynia się do wczesnej identyfikacji chorób i zapobiegania potencjalnym komplikacjom zdrowotnym. W obszarze cyberbezpieczeństwa system umożliwia szybką identyfikację nieprawidłowości w sieciach komputerowych, co pozwala na natychmiastową reakcję na potencjalne zagrożenia i ochronę systemów przed atakami. W detekcji oszustw finansowych Trinity SALT znacząco poprawia wykrywanie nieuczciwych transakcji, redukując straty spowodowane przez oszustwa i wzmacniając bezpieczeństwo finansowe.

Wyniki wskazują także na potrzebę dalszego rozwoju zespołowych metod detekcji anomalii, które integrują różne techniki, w tym nowe algorytmy, co może prowadzić do stworzenia bardziej skutecznych i odpornych systemów wykrywania anomalii. Automatyczna optymalizacja hiperparametrów, zwłaszcza przez rozwój zaawansowanych metod dostosowywania hiperparametrów algorytmów, mogłaby znacząco zwiększyć ich adaptacyjność do nowych typów danych. Ważnym kierunkiem jest także adaptacja systemu Trinity SALT do detekcji w strumieniach danych, co odpowiada na wyzwania związane z dynamicznie generowanymi danymi. Dodatkowo, zaawansowana analiza danych kategorycznych i wielowymiarowych wymaga rozwijania technik specjalnie dedykowanych do tych skomplikowanych typów danych, które tradycyjne algorytmy często pomijają. Przyszłe wdrożenia powinny także obejmować testy Trinity SALT w systemach czasu rzeczywistego, takich jak monitoring infrastruktury, gdzie kluczowe jest natychmiastowe wykrywanie anomalii.

Podsumowując, przeprowadzone badania dowodzą, że zaawansowane techniki wykrywania anomalii, takie jak Trinity SALT, mają znaczący potencjał do poprawy skuteczności detekcji w złożonych zbiorach danych. Rozprawa wykazuje, że innowacyjne podejścia, łączące metody głębokiego uczenia z tradycyjnymi algorytmami, mogą skutecznie stawiać czoła wyzwania związane z nierównowagą klas, wysoką wymiarowością i złożonością danych. Te wyniki nie tylko potwierdzają postawioną tezę, ale także sugerują dalsze kierunki badań, które mogą prowadzić do jeszcze bardziej zaawansowanych rozwiązań, mających szerokie zastosowanie w wielu krytycznych obszarach współczesnej technologii i nauki.

Bibliografia

[1] B. Liu, J. M. Conroy, C. D. Morrison, A. O. Odunsi, M. Qin and L. Wei, D. L. Trump, C. S. Johnson, S. Liu, J. Wang. Structural variation discovery in the cancer genome using next generation sequencing: computational solutions and perspectives. *Oncotarget*, 6(8):5477–5489, Mar. 2015. doi: 10.18632/oncotarget.3491. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4467381/.

[2] O. Balogun, A. Kupolusi, A. Akomolafe. Credit card fraud detection using machine learning algorithms. *British Journal of Computer, Networking and Information Technology*, 7:1–35, 2024. doi: 10.52589/BJCNIT-YDIJNXG2. URL https://doi.org/10.52589/BJCNIT-YDIJNXG2.

[3] S. Ermito, A. Dinatale, S. Carrara, A. Cavaliere, L. Imbruglia, S. Recupero. Prenatal diagnosis of limb abnormalities: role of fetal ultrasonography. *J Prenat Med*, 3(2): 18–22, Apr. 2009. URL https://pubmed.ncbi.nlm.nih.gov/22439035/.

[4] E. Najjar, S. Al-augby. Sentiment analysis combination in terrorist detection on twitter: A brief survey of approaches and techniques. In *Research in Intelligent and Computing in Engineering*, volume 1254 of *Advances in Intelligent Systems and Computing*, pages 279–290. Springer, Singapore, 2021. doi: 10.1007/978-981-15-7527-3_23. URL https://doi.org/10.1007/978-981-15-7527-3_23.

[5] W. Wang. Machine learning in financial time-series data. *Advances in Economics, Management and Political Sciences*, 92(1):293–299, 2024. doi: 10.54254/2754-1169/92/20231279. URL https://doi.org/10.54254/2754-1169/92/20231279.

[6] Z. L. Liew. Time series clustering and anomaly detection of covid-19 global cases and deaths. *Final Year Project (FYP), Nanyang Technological University*, 2022. URL https://hdl.handle.net/10356/156387.

[7] Z. Luo, L. Zhang, N. Liu, Y. Wu. Time series clustering of covid-19 pandemic--related data. *Data Science and Management*, 6(2):79–87, 2023. ISSN 2666--7649. doi: 10.1016/j.dsm.2023.03.003. URL https://www.sciencedirect.com/science/article/pii/S2666764923000115.

[8] E. S. Pearson, C. C. Sekar. The efficiency of statistical tools and a criterion for the rejection of outlying observations. *Biometrika*, 28, 1936.

[9] D. Bernoulli. The most probable choice between several discrepant observations and the formation therefrom of the most likely induction. *Biometrika*, 48(1-2):3–18, 1961. Tłumaczenie C.G. Allen, Oryginalna praca opublikowana po łacinie w Acta Acad. Petrop., 1777.

[10] F. Y. Edgeworth. On discordant observations. *Phil. Mag. J. Sci.*, 23(5):364–375, 1887. doi: 10.1080/14786448708628471. URL https://doi.org/10.1080/14786448708628471.

[11] P. R. Rider. *Criteria for Rejection of Observations*. Washington University Studies, New Series, Science and Technology. Washington University, St. Louis, 1933. Number 8.

[12] F. E. Grubbs. Sample criteria for testing outlying observations. *Annals of Mathematical Statistics*, 21:27–58, 1950.

[13] F. J. Anscombe. Rejection of outliers. *Technometrics*, 2:123–147, 1960.

[14] T. S. Ferguson. On the rejection of outliers. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1:253–287, 1961.

[15] A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 34(3):350–363, 1972. doi: 10.1111/j.2517-6161.1972.tb00912.x. URL https://doi.org/10.1111/j.2517-6161.1972.tb00912.x.

[16] R. S. Tsay. Time series model specification in the presence of outliers. *Journal of the American Statistical Association*, 81(393):132–141, 1986. doi: 10.1080/01621459.1986.10478250. URL https://doi.org/10.1080/01621459.1986.10478250.

[17] D. M. Hawkins. *Identification of Outliers*. Monographs on Statistics and Applied Probability. Springer Dordrecht, Dordrecht, 1980. ISBN 978-94-015--3996-8. doi: 10.1007/978-94-015-3994-4. URL https://doi.org/10.1007/978-94-015-3994-4. Published: 21 April 2014.

[18] V. Barnett, T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 3rd edition, 1994. ISBN 978-0-471-93094-5. URL https://www.wiley.com/en-us/

Outliers+in+Statistical+Data%2C+3rd+Edition-p-9780471930945. May 1994.

[19] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969. doi: 10.1080/00401706.1969.10490657. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1969.10490657.

[20] P. J. Rousseeuw, A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, 1987. ISBN 978-0471852339. doi: 10.1002/0471725382. URL https://doi.org/10.1002/0471725382.

[21] V. Chandola, A. Banerjee, V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):1–58, 2009. doi: 10.1145/1541880.1541882. URL https://dl.acm.org/doi/abs/10.1145/1541880.1541882.

[22] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, May 2021. doi: 10.1109/JPROC.2021.3052449. URL https://ieeexplore.ieee.org/document/9347460/authors#authors.

[23] M. Markou, M. Singh. Novelty detection: A review—part 1: Statistical approaches. *Signal Processing*, 83:2481–2497, Dec. 2003. doi: 10.1016/j.sigpro.2003.07.018. URL https://dl.acm.org/doi/10.1016/j.sigpro.2003.07.018.

[24] M. Markou, S. Singh. Novelty detection: A review - part 2: Neural network based approaches. *Signal Processing*, 83:2499–2521, Dec. 2003. doi: 10.1016/j.sigpro.2003.07.019. URL https://dl.acm.org/doi/10.1016/j.sigpro.2003.07.019.

[25] R. Saunders, J. Gero. The importance of being emergent. In *Proceedings of Artificial Intelligence in Design*, page 4. Citeseer, Jan. 2000. URL https://www.academia.edu/1042731/The_importance_of_being_emergent. 88 Views, 1 File.

[26] R. Foorthuis. On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics*, 12(4):297–331, Aug. 2021. doi: 10.1007/s41060-021-00265-1. URL https://link.springer.com/article/10.1007/s41060-021-00265-1.

[27] C. C. Aggarwal. *Outlier Analysis*. Springer International Publishing, Cham, Switzerland, second edition edition, 2017. ISBN 978-3-319-47577-6. doi: 10.1007/978-3-319-47578-3. URL https://link.springer.com/book/10.1007/978-3-319-47578-3.

[28] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data--Centric Systems and Applications. Springer Berlin, Heidelberg, 2 edition, 2011. ISBN 978-3-642-19459-7. doi: 10.1007/978-3-642-19460-3. URL https://doi.org/10.1007/978-3-642-19460-3.

[29] M. Herman, S. Rivera, S. Mills, J. Sullivan, P. Guerra, A. Cosmas, D. Farris, E. Kohlwey, P. Yacci, B. Keller, A. Kherlopian, M. Kim. *The Field Guide to Data Science*. Booz Allen Hamilton, November 2013. URL http://www.boozallen.com/s/insight/publication/field-guide-to-data-science.html.

[30] R. E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, New Jersey, 1961. doi: 10.1515/9781400874668. URL https://doi.org/10.1515/9781400874668.

[31] N. N. R. R. Suri, N. M. M., G. Athithan. *Outlier Detection: Techniques and Applications: A Data Mining Perspective*. Intelligent Systems Reference Library. Springer International Publishing, 2019. ISBN 9783030051273. URL https://link.springer.com/book/10.1007/978-3-030-05127-3.

[32] L. Ertöz, M. Steinbach, V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the SIAM International Conference on Data Mining (ICDM)*, May 2003. doi: 10.1137/1.9781611972733.5. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972733.5.

[33] C. C. Aggarwal, P. S. Yu. Outlier detection in high dimensional data. In *Proceedings of the ACM SIGMOD Conference*, New York, NY, USA, 2001. ACM. doi: 10.1145/376284.375668. URL https://doi.org/10.1145/376284.375668.

[34] D. Cozzolino, A. Power, J. Chapman. Interpreting and reporting principal component analysis in food science analysis and beyond. *Food Analytical Methods*, 12: 2469–2473, 2019. doi: 10.1007/s12161-019-01605-5. URL https://doi.org/10.1007/s12161-019-01605-5.

[35] L. R. Ackoff. *Decyzje optymalne w badaniach stosowanych*. Państwowe Wydawnictwo Naukowe, Warszawa, 1969. URL https://katalogi.bn.org.pl/.

[36] M.A. Kooyman. *Dummy Variables in Econometrics*. Tilburg Studies in Economics. Springer Dordrecht, 1 edition, 1976. ISBN 978-94-011-7744-3. URL https://link.springer.com/book/9789401177443.

[37] J. O. Rawlings, S. G. Pantula, D. A. Dickey. *Applied Regression Analysis: A Research Tool*. Springer Texts in Statistics. Springer-Verlag, New York, NY, USA, 2 edition, 1998. ISBN 0-387-98454-2.

[38] M. K. Dahouda, I. Joe. A deep-learned embedding technique for categorical features encoding. *IEEE Access*, 9:114381–114391, 2021. doi: 10.1109/ACCESS.2021.3104357. URL https://ieeexplore.ieee.org/document/9512057.

[39] A. Y. Rodríguez-González, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, J. Ruiz-Shulcloper. Rp-miner: A relaxed prune algorithm for frequent similar pattern mining. *Knowledge and Information Systems*, 27(3):451–471, June 2011. doi: 10.1007/s10115-010-0309-9. URL https://link.springer.com/article/10.1007/s10115-010-0309-9.

[40] S. Moro, P. Rita, P. Cortez. Bank marketing. UCI Machine Learning Repository, 2012. URL https://doi.org/10.24432/C5K306.

[41] M. Walesiak, E. Gatnar. *Statystyczna analiza danych z wykorzystaniem programu R*. Wydawnictwo Naukowe PWN, 2009. ISBN 978-83-01-15661-9.

[42] S. S. Stevens. Measurement and man. *Science*, 127(3295):383–389, 1958. doi: 10.1126/science.127.3295.383. URL https://www.science.org/doi/abs/10.1126/science.127.3295.383.

[43] E. Gatnar, M. Walesiak. *Analiza danych jakościowych i symbolicznych z wykorzystaniem programu R*. Wydawnictwo CH Beck, Warszawa, 2011. ISBN 978-83-255-2636-8. URL https://wir.ue.wroc.pl/info/book/WUT687c02ae6e7e4ae8abb2281a3a6dce37/.

[44] B. Pawełek. *Metody normalizacji zmiennych w badaniach porównawczych złożonych zjawisk ekonomicznych*. Wydawnictwo Uniwersytetu Ekonomicznego w Krakowie, 2008. URL https://katalog.uek.krakow.pl/cgi-bin/koha/opac-detail.pl?biblionumber=33642&query_desc=se%2Cphr%3A%22Zeszyty%22%20and%20location%3A10006.

[45] M. Walesiak. Przegląd formuł normalizacji wartości zmiennych oraz ich własności w statystycznej analizie wielowymiarowej. *Przegląd Statystyczny*, 61(4):363–372, 2014. URL https://open.icm.edu.pl/items/3e38df04-245b-40df-8c32-3ca64f2efe30.

[46] G.E. Shilov. *Linear Algebra*. Dover books on advanced mathematics. Dover Publications, 1977. ISBN 9780486635187. URL https://books.google.pl/books?id=K-dQAAAAMAAJ.

[47] K. B. Petersen, M. S. Pedersen. The matrix cookbook, November 2012. URL http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf. Version 20121115.

[48] E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, 2003. ISBN 978-0-521-59271-0. URL http://www.cambridge.org/9780521592710.

[49] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988. ISBN 978-0-934613--73-6. URL https://www.sciencedirect.com/book/9780080514895/probabilistic-reasoning-in-intelligent-systems.

[50] I. Goodfellow, Y. Bengio, A. Courville. *Systemy uczące się*. Wydawnictwo Naukowe PWN, Warszawa, 2018. ISBN 978-83-01-19583--0. URL https://ksiegarnia.pwn.pl/Deep-Learning,731182149,p.html, https://www.deeplearningbook.org/.

[51] F. P. Ramsey. Truth and probability. In *Foundations of Mathematics and other Essays*, pages 156–198. Kegan Paul, Trench, Trubner, & Co., London, 1926.

[52] T. M. Cover, J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2nd edition, 2006. ISBN 978-0-471-24195-9.

[53] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, version 7.2 (fourth printing) edition, 2003. URL http://www.inference.phy.cam.ac.uk/mackay/itila/.

[54] A. Cauchy. Méthode générale pour la résolution de systèmes d'équations simultanées. *Comptes Rendus Hebd. Séances Acad. Sci.*, 25:536–538, 1847.

[55] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio. Theano: a cpu and gpu math expression compiler. In *Proc. of the 9th Python in Science Conf. (SciPy 2010)*, 2010. doi: 10.25080/Majora-92bf1922-003. URL https://conference.scipy.org/proceedings/scipy2010/bergstra.html.

[56] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, Y. Bengio. Theano: new features and speed improvements. In *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*, 2012. doi: 10.48550/arXiv.1211.5590. URL https://doi.org/10.48550/arXiv.1211.5590.

[57] S. J. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 2003. ISBN 0-13-790395-2. URL https://aima.cs.berkeley.edu/.

[58] J. Nocedal, S. Wright. *Numerical Optimization*. Springer, New York, 2006. URL https://link.springer.com/book/10.1007/978-0-387-40065-5.

[59] S. Boyd, L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paolo, Delhi, 2004. ISBN 978-0-521-83378-3. URL https://stanford.edu/~boyd/cvxbook/.

[60] R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, USA, 1997. ISBN 978-0-691-01586-6. doi: 10.1515/9781400873173. URL https://doi.org/10.1515/9781400873173.

[61] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960. doi: 10.1137/0108011. URL https://doi.org/10.1137/0108011.

[62] W. Karush. Minima of functions of several variables with inequalities as side constraints. Master's thesis, Dept. of Mathematics, Univ. of Chicago, 1939. URL https://catalog.lib.uchicago.edu/vufind/Record/4111654.

[63] H. W. Kuhn, A. W. Tucker. Nonlinear programming. In G. Giorgi and T. Kjeldsen, editors, *Traces and Emergence of Nonlinear Programming*, pages 393–414. Birkhäuser, Basel, 2014. doi: 10.1007/978-3-0348-0439-4_11. URL https://doi.org/10.1007/978-3-0348-0439-4_11.

[64] H. W. Kuhn, A. W. Tucker. Nonlinear programming. In *Proc. Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, Calif., 1951. University of California Press.

[65] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV)*, pages 2146–2153. IEEE, 2009. doi: 10.1109/ICCV.2009.5459469. URL https://doi.org/10.1109/ICCV.2009.5459469.

[66] V. Nair, G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814. Omnipress, 2010. doi: 10.5555/3104322.3104425. URL https://dl.acm.org/doi/10.5555/3104322.3104425.

[67] X. Glorot, A. Bordes, Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011. URL https://proceedings.mlr.press/v15/glorot11a.html.

[68] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau. Incorporating second-order functional knowledge for better option pricing. In V. Tresp T. Leen, T. Dietterich, editor,

Advances in Neural Information Processing Systems 13 (NIPS'00), volume 13. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/44968aece94f667e4095002d140b5896-Paper.pdf.

[69] R. Collobert. *Large Scale Machine Learning*. Ph.d. thesis, Université de Paris VI, LIP6, 2004. URL https://publications.idiap.ch/publications/show/485.

[70] J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems*, pages 211–217. Morgan Kaufmann, 1990.

[71] P.-N. Tan, M. Steinbach, V. Kumar. *Introduction to Data Mining*. Pearson Education Limited, Edinburgh Gate, Harlow, Essex CM20 2JE, England, 2014. ISBN 978--1-292-02615-2. URL https://www-users.cse.umn.edu/~kumar001/dmbook/index.php.

[72] J. Han, M. Kamber, J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, third edition edition, 2011. URL https://www.bibsonomy.org/bibtex/2beb274b9aeaebb87f5423781b6839f54/hotho.

[73] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, NY, 2 edition, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7. URL https://doi.org/10.1007/978-0-387-84858-7.

[74] P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936. doi: 10.1007/s13171-019-00164-5. URL https://link.springer.com/article/10.1007/s13171-019-00164-5.

[75] M. Hubert, M. Debruyne. Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):36–43, 2010. doi: 10.1002/wics.61. URL https://doi.org/10.1002/wics.61.

[76] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29:147–160, 1950. URL https://archive.org/details/bstj29-2-147.

[77] M. Kłopotek, S. Wierzchoń. *Algorytmy analizy skupień*. WNT, Warszawa, 2017. ISBN 978-83-01-19178-8.

[78] D. R. Wilson, T. R. Martinez. Value difference metrics for continuously valued attributes. In *Proceedings of the International Conference on Artificial Intelligence, Expert Systems and Neural Networks (AIE'96)*, pages 11–14, Provo, UT, 1996. Brigham Young University. URL https://axon.cs.byu.edu/papers/wilson.aie96.ivdm.pdf.

[79] K. Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. doi: 10.1098/rspl.1895.0041. URL https://www.jstor.org/stable/115794.

[80] P. Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37(140): 241–272, 1901. doi: 10.5169/seals-266440. URL https://www.e-periodica.ch/digbib/view?pid=bsv-002:1901:37::745#251.

[81] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 44(163):223–270, 1908. doi: 10.5169/seals-268384. URL https://www.e-periodica.ch/digbib/view?pid=bsv-002:1908:44::485#248.

[82] A. Nowak-Brzezińska, C. Horyń. Wartości odstające i ich wpływ na jakość analizy skupień–pakiet soacraport. *Studia Informatica*, 41(1):31–46, 2020. ISSN 1642-0489. doi: 10.5281/zenodo.7238199. URL https://zenodo.org/records/7238199.

[83] J. C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971. doi: 10.2307/2528823. URL https://doi.org/10.2307/2528823.

[84] C. Horyń. Analiza wpływu odchyleń na jakość grupowania. Praca dyplomowa magisterska, Uniwersytet Śląski w Katowicach, Wydział Nauk Ścisłych i Technicznych, Instytut Informatyk, Sosnowiec, 2020.

[85] A. Nowak-Brzezińska, C. Horyń. Outliers in rules - the comparision of lof, cof and kmeans algorithms. *Procedia Computer Science*, 176:1420–1429, 2020. ISSN 1877-0509. doi: 10.1016/j.procs.2020.09.152. URL https://doi.org/10.1016/j.procs.2020.09.152.

[86] T. Morzy. *Eksploracja danych. Metody i algorytmy*. Kultowe podręczniki IT. Wydawnictwo Naukowe PWN, Warszawa, 1 edition, 2013. ISBN 978-83-01-17175-9.

[87] R. Brachman, T. Anand. *The Process of Knowledge Discovery in Databases: A First Sketch*, pages 1–11. AAAI/MIT Press, 07 1994. URL https://dl.acm.org/doi/abs/10.5555/3000850.3000852.

[88] E. Fix, J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989. URL https://doi.org/10.2307/1403797.

[89] J. Zhang, A. Yin, G. Chen, Y. Li, Z. Lu, B. Wang. Research on the intelligent design of office chair patterns. *Applied Sciences*, 12:2124, 02 2022. doi: 10.3390/app12042124. URL https://www.mdpi.com/2076-3417/12/4/2124.

[90] A. Skowron, A. Wojna. K nearest neighbor classification with local induction of the simple value difference metric. In *Proceedings of the 5th International Conference on Intelligent Information Processing and Web Mining*, pages 157–164, Warsaw, Poland, 2003. Springer. doi: 10.1007/978-3-540-25929-9_27. URL https://doi.org/10.1007/978-3-540-25929-9_27.

[91] S. Byers, A. E. Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93 (442):577–584, 1998. doi: 10.1080/01621459.1998.10473711. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1998.10473711.

[92] S. E. Guttormsson, R. J. Marks, M. A. El-Sharkawi, I. Kerszenbaum. Elliptical novelty grouping for on-line short-turn detection of excited running rotors. *IEEE Transactions on Energy Conversion*, 14(1):16–22, 1999. doi: 10.1109/60.749142. URL https://ieeexplore.ieee.org/document/749142.

[93] R. K. Pearson. *Mining Imperfect Data: Dealing with Contamination and Incomplete Records*. Society for Industrial and Applied Mathematics, 2005. ISBN 978-0-89871--582-8. doi: 10.1137/1.9780898717884. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898717884.

[94] S. Seo. A review and comparison of methods for detecting outliers in univariate data sets. Master's thesis, University of Pittsburgh, Aug. 2006. URL http://d-scholarship.pitt.edu/id/eprint/7948.

[95] S. Cherednichenko. Outlier detection in clustering. Master's thesis, University of Joensuu, Department of Computer Science, Jan. 2005. URL https://cs.uef.fi/pub/Theses/2005_MSc_Cherednichenko_Svetlana.pdf.

[96] A. Nowak-Brzezińska. Eksploracja odchyleń w regułowych bazach wiedzy. *Studia Informatica*, 33(2A (105)):479–492, 2012. URL https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-a6ddf4a5-bb7e-4202-a9f8-1b4366c608fe.

[97] M. Ester, H.-P. Kriegel, J. Sander, X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd*

International Conference on Knowledge Discovery and Data Mining (KDD), pages 226–231. AAAI Press, 1996. URL https://dl.acm.org/doi/10.5555/3001460.3001507.

[98] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132174. doi: 10.1145/342009.335388. URL https://doi.org/10.1145/342009.335388.

[99] H. Saeedi Emadi, S.M. Mazinani. A novel anomaly detection algorithm using dbscan and svm in wireless sensor networks. *Wireless Personal Communications*, 98(3): 2025–2035, 2018. doi: 10.1007/s11277-017-4961-1. URL https://doi.org/10.1007/s11277-017-4961-1.

[100] J. Tang, Z. Chen, A. Fu, D. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In P. S. Yu M.-S. Chen and B. Liu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 535–548, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-47887-4. doi: 10.1007/3-540-47887-6_53. URL https://doi.org/10.1007/3-540-47887-6_53.

[101] A. Nowak-Brzezińska, C. Horyń. Exploration of outliers in if-then rule-based knowledge bases. *Entropy*, 22(10):1096, 2020. doi: 10.3390/e22101096. URL https://doi.org/10.3390/e22101096.

[102] C. Horyń. Grupowanie danych zawierających odchylenia w środowisku r. Praca dyplomowa inżynierska, Uniwersytet Śląski w Katowicach, Wydział Nauk Ścisłych i Technicznych, Instytut Informatyki, Sosnowiec, 2019.

[103] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Proceedings of the 19th International Conference on Data Engineering (ICDE)*, pages 315–326. IEEE, 2003. doi: 10.1109/ICDE.2003.1260802. URL https://doi.org/10.1109/ICDE.2003.1260802.

[104] A. F. Hassan, S. Barakat, A. Rezk. Towards a deep learning-based outlier detection approach in the context of streaming data. *Journal of Big Data*, 9(120), 2022. doi: 10.1186/s40537-022-00670-8. URL https://doi.org/10.1186/s40537-022-00670-8.

[105] G. Pang, C. Shen, L. Cao, A. van den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys*, 54:1–38, 2021. doi: 10.1145/3439950. URL https://doi.org/10.1145/3439950.

[106] S. Hawkins, H. He, G. Williams, R. Baxter. Outlier detection using replicator neural networks. In M. Arikawa Y. Kambayashi, W. Winiwarter, editor, *Data Warehousing and Knowledge Discovery, DaWaK 2002, Lecture Notes in Computer Science*, volume 2454, pages 170–180. Springer, Berlin, Heidelberg, 2002. ISBN 978-3-540-44123-6. doi: 10.1007/3-540-46145-0_17. URL https://doi.org/10.1007/3-540-46145-0_17.

[107] M. A. Atoui, A. Cohen, S. Verron, A. Kobi. A single bayesian network classifier for monitoring with unknown classes. *Engineering Applications of Artificial Intelligence*, 85:681–690, 2019. ISSN 0952-1976. doi: 10.1016/j.engappai.2019.07.016. URL https://www.sciencedirect.com/science/article/pii/S0952197619301800.

[108] B. Cai, Y. Liu, M. Xie. A dynamic-bayesian-network-based fault diagnosis methodology considering transient and intermittent faults. *IEEE Transactions on Automation Science and Engineering*, 14(1):276–285, 2017. doi: 10.1109/TASE.2016.2574875. URL https://ieeexplore.ieee.org/document/7495018.

[109] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer, New York, NY, 2 edition, 2000. ISBN 978-0-387-98780-4. doi: 10.1007/978-1-4757-3264-1. URL https://link.springer.com/book/10.1007/978-1-4757-3264-1.

[110] Vladimir N. Vapnik. *Statistical Learning Theory*. Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control. Wiley & Sons Ltd, New York, 1st edition, October 1998. ISBN 978-0-471--03003-4. URL https://www.wiley.com/en-us/Statistical+Learning+Theory-p-9780471030034.

[111] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, 1 edition, August 2006. ISBN 978-0-387--31073-2. doi: 10.1007/978-0-387-45528-0. URL https://link.springer.com/book/10.1007/978-0-387-45528-0.

[112] K. Michalik. *Systemy ekspertowe we wspomaganiu procesów zarządzania wiedzą w organizacji*. Wydawnictwo Uniwersytetu Ekonomicznego, Katowice, January 2014. ISBN 978-83-7875-175-5. URL http://bazekon.icm.edu.pl/bazekon/element/bwmeta1.element.ekon-element-000171283865.

[113] S. Guha, R. Rastogi, K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000. ISSN 0306-4379. doi: 10.1016/S0306-4379(00)00022-3. URL https://www.sciencedirect.com/science/article/pii/S0306437900000223.

[114] S. Guha, R. Rastogi, K. Shim. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering (ICDE)*, pages 512–521, 1999. doi: 10.1109/ICDE.1999.754967. URL https://ieeexplore.ieee.org/document/754967.

[115] L. Duan, L. Xu, F. Guo, J. Lee, B. Yan. A local-density based spatial clustering algorithm with noise. *Information Systems*, 32(7):978–986, 2007. ISSN 0306--4379. doi: 10.1016/j.is.2006.10.006. URL https://www.sciencedirect.com/science/article/pii/S0306437906000871.

[116] J. Sander, M. Ester, H.-P. Kriegel, X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998. doi: 10.1023/A:1009745219419. URL https://doi.org/10.1023/A:1009745219419.

[117] A. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31: 651–666, 2010. doi: 10.1016/j.patrec.2009.09.011. URL https://doi.org/10.1016/j.patrec.2009.09.011.

[118] H. Steinhaus. Sur la division des corps matériels en parties. *Bulletin de l'Académie Polonaise des Sciences*, 4(12):801–804, 1956. URL http://www.laurent-duval.eu/Documents/Steinhaus_H_1956_j-bull-acad-polon-sci_division_cmp-k-means.pdf.

[119] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi: 10.1109/TIT.1982.1056489. URL https://ieeexplore.ieee.org/document/1056489. Algorytm przedstawiono w raporcie pochodzącym z 31 lipca 1957 roku, a drukiem praca ukazała się dopiero w 1982 roku.

[120] J.A. Cuesta-Albertos, A. Gordaliza, C. Matrán. Trimmed k-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997. doi: 10.1214/aos/1031833664. URL http://dx.doi.org/10.1214/aos/1031833664.

[121] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. URL http://www.jstor.org/stable/2984875.

[122] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA and London, England, 2012. ISBN 978-0-262-01802-9. URL https://dl.acm.org/doi/10.5555/2380985.

[123] Z. He, X. Xu, S. Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003. ISSN 0167-8655. doi: https://doi.org/10.1016/

S0167-8655(03)00003-5. URL https://www.sciencedirect.com/science/article/pii/S0167865503000035.

[124] M. Amer, M. Goldstein. Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. In S. Fischer, editor, *Proceedings of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012)*, pages 1–12, Aachen, 2012. Shaker Verlag GmbH. URL https://www.goldiges.de/publications/Anomaly_Detection_Algorithms_for_RapidMiner.pdf.

[125] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Publishing Company, 1977.

[126] V. Brys, M. Hubert, A. Struyf. A robust measure of skewness. *Journal of Computational and Graphical Statistics*, 13(4):996–1017, 2004. doi: 10.1198/106186004X12632. URL https://www.tandfonline.com/doi/abs/10.1198/106186004X12632.

[127] F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974. doi: 10.1080/01621459.1974.10482962. URL https://doi.org/10.1080/01621459.1974.10482962.

[128] P. J. Rousseeuw, C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993. doi: 10.1080/01621459.1993.10476408. URL https://doi.org/10.1080/01621459.1993.10476408.

[129] J. Laurikkala, M. Juhola, E. Kentala. Informal identification of outliers in medical data. In *Proceedings of the Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, July 2000. URL https://api.semanticscholar.org/CorpusID:15341032.

[130] W. Marek, Z. Pawlak. Information storage and retrieval system – mathematical foundations. *Theoretical Computer Science*, 1:331–354, April 1976. URL https://www.sciencedirect.com/science/article/pii/0304397576900773/pdf?md5=bc0b25345469fc5511dc52ffc079ad47&pid=1-s2.0-0304397576900773-main.pdf&_valck=1.

[131] R. D. Cook. Detection of influential observations in linear regression. *Technometrics*, 19(1):15–18, 1977. doi: 10.2307/1268249. URL https://doi.org/10.2307/1268249.

[132] D. A. Belsley, E. Kuh, R. E. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley Series in Probability and Statistics. John

Wiley & Sons, Inc., 1980. ISBN 978-0471058564. doi: 10.1002/0471725153. URL https://doi.org/10.1002/0471725153.

[133] F. J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. doi: 10.1080/01621459.1951.10500769. URL https://doi.org/10.1080/01621459.1951.10500769.

[134] J. D. Gibbons, S. Chakraborti. *Nonparametric Statistical Inference*. Chapman and Hall/CRC, New York, 5th edition edition, 2010. ISBN 9781439856500. doi: 10.1201/9781439896129. URL https://doi.org/10.1201/9781439896129.

[135] H. Mann, D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947. doi: 10.1214/aoms/1177730491. URL https://doi.org/10.1214/aoms/1177730491.

[136] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman and Hall/CRC, New York, 5th edition edition, 2011. ISBN 9781584888147. doi: 10.1201/9780429186196. URL https://doi.org/10.1201/9780429186196.

[137] W. H. Kruskal, W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.10483441. URL https://doi.org/10.1080/01621459.1952.10483441.

[138] M. Hollander, D. A. Wolfe, E. Chicken. *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2015. ISBN 9780470387375. doi: 10.1002/9781119196037. URL https://doi.org/10.1002/9781119196037.

[139] D. Ruppert, M. P. Wand, R. J. Carroll. *Semiparametric Regression*. Cambridge University Press, New York, NY, 2003. ISBN 9780521785169. doi: 10.1017/CBO9780511755453. URL https://doi.org/10.1017/CBO9780511755453.

[140] J. Koronacki, J. Ćwik. *Statystyczne Systemy Uczące Się*. WNT, Warszawa, 2005.

[141] M. Manguoglu. A highly efficient parallel algorithm for computing the fiedler vector. *arXiv*, 2013. URL https://doi.org/10.48550/arXiv.1003.3689.

[142] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970. doi: 10.1287/mnsc.17.3.219. URL https://doi.org/10.1287/mnsc.17.3.219.

[143] A. Pothen, H. D. Simon, K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990. doi: 10.1137/0611030. URL https://doi.org/10.1137/0611030.

[144] J. Shi, J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. URL https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf.

[145] M. Filippone, F. Camastra, F. Masulli, S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1):176–190, January 2008. ISSN 0031-3203. doi: 10.1016/j.patcog.2007.05.018. URL https://doi.org/10.1016/j.patcog.2007.05.018.

[146] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, December 2007. ISSN 0960-3174. doi: 10.1007/s11222-007-9033-z. URL https://link.springer.com/article/10.1007/s11222-007-9033-z.

[147] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, August 2007. ISSN 1574-0137. doi: 10.1016/j.cosrev.2007.05.001. URL https://doi.org/10.1016/j.cosrev.2007.05.001.

[148] A. Agovic, A. Banerjee, A. Ganguly, V. Protopopescu. Anomaly detection using manifold embedding and its applications in transportation corridors. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 435–455, 2007. doi: 10.1145/1281192.1281234. URL https://doi.org/10.1145/1281192.1281234.

[149] Z. Li, M. van Leeuwen. Explainable contextual anomaly detection using quantile regression forests. *Data Mining and Knowledge Discovery*, 37:2517–2563, 2023. doi: 10.1007/s10618-023-00967-z. URL https://doi.org/10.1007/s10618-023-00967-z.

[150] L. Bontemps, V. L. Cao, J. McDermott, N.-A. Le-Khac. Collective anomaly detection based on long short term memory recurrent neural network. *arXiv preprint arXiv:1703.09752*, 2017. URL https://doi.org/10.48550/arXiv.1703.09752.

[151] C. C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. Springer Cham, 1 edition, 2018. ISBN 978-3-030-06856-1. doi: 10.1007/978-3-319-94463-0. URL https://doi.org/10.1007/978-3-319-94463-0.

[152] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, New York, 1st edition, 1984. ISBN

9781315139470. doi: 10.1201/9781315139470. URL https://doi.org/10.1201/9781315139470.

[153] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[154] Y. LeCun. *PhD thesis: Modèles connexionnistes de l'apprentissage (connectionist learning models)*. PhD thesis, Université P. et M. Curie (Paris 6), 1987. URL https://nyuscholars.nyu.edu/en/publications/phd-thesis-modeles-connexionnistes-de-lapprentissage-connectionis.

[155] H. Bourlard, Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59:291–294, 1988. doi: 10.1007/BF00332918. URL https://doi.org/10.1007/BF00332918.

[156] G. E. Hinton, R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 3–10, Denver, Colorado, 1993. Morgan Kaufmann Publishers Inc. URL https://dl.acm.org/doi/abs/10.5555/2987189.2987190.

[157] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. doi: 10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

[158] S. Hochreiter, J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[159] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. doi: 10.48550/arXiv.1406.2661. URL https://doi.org/10.48550/arXiv.1406.2661.

[160] F. Arslan, A. Javaid, M. D. Z. Awan, E. ur-Rehman. *Anomaly Detection in Time Series: Current Focus and Future Challenges*. IntechOpen, 2024. ISBN 978-1-83769-027-5. doi: 10.5772/intechopen.111886. URL https://www.intechopen.com/chapters/87583.

[161] S. H. Rafique, A. Abdallah, N. S. Musa, T. Murugan. Machine learning and deep learning techniques for internet of things network anomaly detection—current research trends. *Sensors*, 24:1968, 2024. doi: 10.3390/s24061968. URL https://doi.org/10.3390/s24061968.

[162] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, C. Faloutsos. Loci: Fast outlier detection using the local correlation integral. Technical Report CMU-CS-02-188, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, November 2002. URL https://ieeexplore.ieee.org/document/1260802.

[163] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. doi: 10.1214/aoms/1177704472. URL https://projecteuclid.org/euclid.aoms/1177704472.

[164] A. Hamache, S. Destercke, T. Denoeux, F. Cuzzolin, A. Martin. Uncertainty-aware parzen-rosenblatt classifier for multiattribute data. In F. Cuzzolin S. Destercke, T. Denoeux and A. Martin, editors, *Belief Functions: Theory and Applications. BELIEF 2018*, volume 11069 of *Lecture Notes in Computer Science*, pages 180–189. Springer, Cham, 2018. ISBN 978-3-319-99382-9. doi: 10.1007/978-3-319-99383-6_14. URL https://doi.org/10.1007/978-3-319-99383-6_14.

[165] S. Guha, R. Rastogi, K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25:345–366, 02 2001. doi: 10.1016/S0306-4379(00)00022-3. URL https://doi.org/10.1016/S0306-4379(00)00022-3.

[166] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304, 1998. ISSN 1384-5810. doi: 10.1023/A:1009769707641. URL https://link.springer.com/article/10.1023/A:1009769707641.

[167] Z. He, X. Xu, S. Deng. Squeezer: An efficient algorithm for clustering categorical data. *Journal of Computer Science and Technology*, 17:611–624, 09 2002. doi: 10.1007/BF02948829. URL https://doi.org/10.1007/BF02948829.

[168] Z. He, X. Xu, S. Deng. k-anmi: A mutual information based clustering algorithm for categorical data. *Information Fusion*, 9(2):223–233, 2008. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2006.05.006. URL https://www.sciencedirect.com/science/article/pii/S1566253506000637.

[169] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. URL https://onlinelibrary.wiley.com/doi/10.1002/j.1538-7305.1948.tb01338.x.

[170] J. Kreer. A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208, September 1957. doi: 10.1109/TIT.1957.1057418. URL https://ieeexplore.ieee.org/document/1057418.

[171] N. N. R. Ranga Suri, M. Narasimha Murty, G. Athithan. A ranking-based algorithm for detection of outliers in categorical data. *International Journal of Hybrid Intelligent Systems*, 11(1):1–11, 2014. doi: 10.3233/HIS-130179. URL https://doi.org/10.3233/HIS-130179.

[172] A. Koufakou, E. G. Ortiz, M. Georgiopoulos, G. C. Anagnostopoulos, K. M. Reynolds. A scalable and efficient outlier detection strategy for categorical data. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, pages 210–217, 2007. doi: 10.1109/ICTAI.2007.125. URL https://doi.org/10.1109/ICTAI.2007.125.

[173] Z. He and X. Xu, S. Deng. A fast greedy algorithm for outlier mining. In *Advances in Knowledge Discovery and Data Mining, 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006, Proceedings*, volume 3918 of *Lecture Notes in Computer Science*, pages 567–576. Springer, 2006. doi: 10.1007/11731139_67. URL https://link.springer.com/chapter/10.1007/11731139_67.

[174] Z. He, X. Xu, J. Huang, S. Deng. Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems*, 2(1):103–118, 2005. doi: 10.2298/CSIS0501103H. URL https://doiserbia.nb.rs/Article.aspx?ID=1820-02140501103H.

[175] I.T. Jolliffe. Principal component analysis. In M. Lovric, editor, *International Encyclopedia of Statistical Science*. Springer, Berlin, Heidelberg, 2011. ISBN 978--3-642-04897-5. doi: 10.1007/978-3-642-04898-2_455. URL https://doi.org/10.1007/978-3-642-04898-2_455.

[176] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang. A novel anomaly detection scheme based on principal component classifier. In *IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003. URL https://www.researchgate.net/publication/228709094_A_Novel_Anomaly_Detection_Scheme_Based_on_Principal_Component_Classifier.

[177] S. Boriah, V. Chandola, V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 8th SIAM International Conference on Data Mining*, pages 243–254. SIAM, 2008. doi: 10.1137/1.9781611972788.22. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972788.22.

[178] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. doi: 10.1108/00220410410560573. URL https://doi.org/10.1108/00220410410560573.

[179] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection. In D. Barbará and S. Jajodia, editors, *Applications of Data Mining in Computer Security*, volume 6 of *Advances in Information Security*, pages 78–100. Springer, Boston, MA, 2002. doi: 10.1007/978-1-4615-0953-0_4. URL https://link.springer.com/chapter/10.1007/978-1-4615-0953-0_4.

[180] K. Sparck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972. URL https://www.bibsonomy.org/bibtex/25a4ce4a29250ab2ddf0c00299373c2d4/lepsky.

[181] D. W. Goodall. A new similarity index based on probability. *Biometrics*, 22 (4):882–907, 1966. doi: 10.2307/2528080. URL https://doi.org/10.2307/2528080.

[182] M. R. Anderberg. *Cluster Analysis for Applications*. Probability and Mathematical Statistics: A Series of Monographs and Textbooks. Academic Press, 1973. ISBN 978-0-12-057650-0. doi: 10.1016/C2013-0-06161-0. URL https://doi.org/10.1016/C2013-0-06161-0.

[183] T. P. Burnaby. On a method for character weighting a similarity coefficient: Employing the concept of information. *Mathematical Geology*, 2(1):25–38, 1970. doi: 10.1007/BF02332078. URL https://link.springer.com/article/10.1007/BF02332078.

[184] P. Gambaryan. A mathematical model of taxonomy. *Izvest. Akad. Nauk Armen, SSR*, 17(12):47–53, 1964.

[185] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568. URL https://dl.acm.org/doi/10.5555/645527.657297.

[186] E. S. Smirnov. On exact methods in systematics. *Systematic Zoology*, 17(1):1–13, 1968. doi: 10.2307/2412390. URL https://doi.org/10.2307/2412390.

[187] N. Goodman. Seven strictures on similarity. *Problems and Projects*, pages 437–447, 1972. URL https://www.bibsonomy.org/bibtex/164406d561cb3414fd708e38b0f8ec706/quesada.

[188] P. Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société vaudoise des sciences naturelles*, 37:547–579, 1901. doi: 10.5169/SEALS-266450. URL https://doi.org/10.5169/SEALS-266450.

[189] S. Salvatore, K. D. Rand, I. Grytten, E. Ferkingstad, D. Domanska, L. Holden, M. Gheorghe, A. Mathelier, I. Glad, G. K. Sandve. Beware the jaccard: the choice of similarity measure is important and non-trivial in genomic colocalisation analysis. *Briefings in Bioinformatics*, 21(5):1523–1530, Sep 2020. doi: 10.1093/bib/bbz083. URL https://doi.org/10.1093/bib/bbz083.

[190] G. Das, H. Mannila. Context-based similarity measures for categorical databases. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2000)*, pages 201–210. Springer-Verlag, 2000. doi: 10.1007/3-540-45372-5_20. URL https://link.springer.com/chapter/10.1007/3-540-45372-5_20.

[191] A. Gordon. *Classification*. Chapman and Hall/CRC, New York, 2nd edition, 1999. ISBN 9780367805302. doi: 10.1201/9780367805302. URL https://doi.org/10.1201/9780367805302.

[192] R. Gnanadesikan, J. R. Kettenring, S. L. Tsao. Weighting and selection of variables for cluster analysis. *Journal of Classification*, 12(1):113–136, March 1995. doi: 10.1007/BF01202271. URL https://doi.org/10.1007/BF01202271.

[193] G. W. Milligan. A validation study of a variable weighting algorithm for cluster analysis. *Journal of Classification*, 6:53–71, 1989. doi: 10.1007/BF01908588. URL https://doi.org/10.1007/BF01908588.

[194] M. Walesiak. *Problemy selekcji i ważenia zmiennych w zagadnieniu klasyfikacji*, volume 1076 of *Taksonomia 12, Prace Naukowe AE we Wrocławiu*, page 185. Wydawnictwo Akademii Ekonomicznej we Wrocławiu, 2005. URL https://www.researchgate.net/publication/309399343_Problemy_selekcji_i_wazenia_zmiennych_w_zagadnieniu_klasyfikacji.

[195] V. Makarenkov, P. Legendre. Optimal variable weighting for ultrametric and additive trees and k-means partitioning: Methods and software. *Journal of Classification*, 18:245–271, 2001. doi: 10.1007/s00357-001-0018-x. URL https://doi.org/10.1007/s00357-001-0018-x.

[196] Z. Zhao, R. Anand, M. Wang. Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform. *arXiv preprint arXiv:1908.05376*, 2019. doi: 10.48550/arXiv.1908.05376. URL https://doi.org/10.48550/arXiv.1908.05376.

[197] K. Sun, P. Tian, H. Qi, F. Ma, G. Yang. An improved normalized mutual information variable selection algorithm for neural network-based soft sensors. *Sensors*, 19 (24):5368, 2019. doi: 10.3390/s19245368. URL https://doi.org/10.3390/s19245368.

[198] M. Afzal, S. M. Arif Ashraf. Genetic algorithm for outlier detection. *International Journal of Computer Science and Information Technologies*, 7(2):833–835, 2016. URL https://www.ijcsit.com/ijcsit-v7issue2.php.

[199] A. O. Efunboade, O. R. Oyeniran, O. A.Odeniyi. Performance evaluation of genetic algorithm selection methods in outlier detection: Further analysis. *International Journal of Medical and Clinical Research (IJMCR)*, 10(6):2701–2704, 2022. doi: 10.47191/ijmcr/v10i6.01. URL https://doi.org/10.47191/ijmcr/v10i6.01.

[200] J. Zhang, Q. Gao, H. Wang. A novel method for detecting outlying subspaces in high-dimensional databases using genetic algorithm. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006)*, pages 711–715, Hong Kong, China, December 2006. doi: 10.1109/ICDM.2006.6. URL https://doi.org/10.1109/ICDM.2006.6.

[201] J. Zhang, M. Lou, T. W. Ling, H. Wang. Hos-miner: A system for detecting outlying subspaces of high-dimensional data. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB)*, pages 31–September 3, Toronto, Canada, August 2004. doi: 10.1016/B978-012088469-8/50123-6. URL https://doi.org/10.1016/B978-012088469-8.50123-6.

[202] J. Zhang, H. Wang. Detecting outlying subspaces for high-dimensional data: The new task, algorithms, and performance. *Knowledge and Information Systems*, 10 (3):333–355, 2006. doi: 10.1007/s10115-006-0020-z. URL https://doi.org/10.1007/s10115-006-0020-z.

[203] A. Lazarevic, V. Kumar. Feature bagging for outlier detection. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 157–166, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081891. URL https://doi.org/10.1145/1081870.1081891.

[204] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, Berlin, Heidelberg, 2000. Springer-Verlag. ISBN 3540677046. doi: 10.1007/3-540-45014-9_1. URL https://dl.acm.org/doi/10.5555/648054.743935.

[205] I. Assent, R. Krieger, E. Müller, T. Seidl. Inscy: Indexing subspace clusters with in-process-removal of redundancy. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 719–724, USA, 2008. IEEE Computer Society. doi: 10.1109/ICDM.2008.65. URL https://ieeexplore.ieee.org/document/4781168.

[206] G. Moise, J. Sander. Finding non-redundant, statistically significant regions in high dimensional data: A novel approach to projected and subspace clustering. In

Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), pages 533–541, USA, 2008. ACM. doi: 10.1145/1401890.1401956. URL https://dl.acm.org/doi/10.1145/1401890.1401956.

[207] E. Müller, I. Assent, S. Günnemann, R. Krieger, T. Seidl. Relevant subspace clustering: Mining the most interesting non-redundant concepts in high dimensional data. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 377–386, USA, 2009. IEEE Computer Society. doi: 10.1109/ICDM.2009.10. URL https://ieeexplore.ieee.org/document/5360263.

[208] K. Sim, V. Gopalkrishnan, A. Zimek, G. Cong. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery (DMKD)*, 26(2):332–397, 2012. doi: 10.1007/s10618-012-0258-x. URL https://link.springer.com/article/10.1007/s10618-012-0258-x.

[209] E. Müller, S. Günnemann, I. Assent, T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proc. VLDB Endow.*, 2(1):1270–1281, 2009. doi: 10.14778/1687627.1687770. URL https://dl.acm.org/doi/10.14778/1687627.1687770.

[210] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM SIGMOD Int. Conf. Management of Data (SIGMOD)*, pages 94–105, USA, 1998. ACM. doi: 10.1145/276304.276314. URL https://dl.acm.org/doi/10.1145/276304.276314.

[211] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, J. S. Park. Fast algorithms for projected clustering. In *Proc. ACM SIGMOD Int. Conf. Management of Data (SIGMOD)*, pages 61–72, USA, 1999. ACM. doi: 10.1145/304182.304188. URL https://dl.acm.org/doi/10.1145/304182.304188.

[212] M. L. Yiu, N. Mamoulis. Frequent-pattern based iterative projected clustering. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 689–692, USA, 2003. IEEE Computer Society. doi: 10.1109/ICDM.2003.1251009. URL https://ieeexplore.ieee.org/document/1251009.

[213] K. Sequeira, M. Zaki. Schism: A new approach for interesting subspace mining. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 186–193, USA, 2004. IEEE Computer Society. doi: 10.1109/ICDM.2004.10099. URL https://ieeexplore.ieee.org/document/1410283.

[214] H.-P. Kriegel, P. Kröger, M. Renz, S. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *Proc. IEEE Int. Conf. Data Mining*

(ICDM), pages 250–257, USA, 2005. IEEE Computer Society. doi: 10.1109/ICDM.2005.5. URL https://ieeexplore.ieee.org/document/1565686.

[215] H.-P. Kriegel, E. Schubert, A. Zimek, P. Kroger. Outlier detection in axis-parallel subspaces of high dimensional data. In *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD)*, pages 831–838, Berlin, Heidelberg, 2009. Springer. doi: 10.1007/978-3-642-01307-2_86. URL https://link.springer.com/chapter/10.1007/978-3-642-01307-2_86.

[216] E. Muller, M. Schiffer, T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *Proc. IEEE Int. Conf. Data Engineering (ICDE)*, pages 434–445, USA, 2011. IEEE Computer Society. doi: 10.1109/ICDE.2011.5767916. URL https://ieeexplore.ieee.org/document/5767916.

[217] E. Muller, I. Assent, P. Iglesias, Y. Mulle, K. Bohm. Outlier analysis via subspace analysis in multiple views of the data. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, ICDM '12, pages 529–538, USA, 2012. IEEE Computer Society. ISBN 9780769549057. doi: 10.1109/ICDM.2012.112. URL https://dl.acm.org/doi/10.1109/ICDM.2012.112.

[218] F. T. Liu, K. M. Ting, Z.-H. Zhou. Isolation forest. In *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 413–422, USA, 2008. IEEE Computer Society. doi: 10.1109/ICDM.2008.17. URL https://ieeexplore.ieee.org/document/4781136.

[219] M. Fernandez-Delgado, E. Cernadas, S. Barro, D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, 15(1):3133–3181, 2014. doi: 10.5555/2627435.2697065. URL https://dl.acm.org/doi/10.5555/2627435.2697065.

[220] C. C. Aggarwal. Outlier ensembles: position paper. *SIGKDD Explor. Newsl.*, 14(2):49–58, Apr. 2013. doi: 10.1145/2481244.2481252. URL https://doi.org/10.1145/2481244.2481252.

[221] S. Sathe, C. C. Aggarwal. Subspace outlier detection in linear time with randomized hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 459–468, Barcelona, Spain, 2016. doi: 10.1109/ICDM.2016.0057. URL https://doi.org/10.1109/ICDM.2016.0057.

[222] F. Keller, E. Müller, K. Böhm. Hics: High contrast subspaces for density-based outlier ranking. *2012 IEEE 28th International Conference on Data Engineering*, pages 1037–1048, 2012. doi: 10.1109/ICDE.2012.88. URL https://ieeexplore.ieee.org/document/6228154.

[223] W. S. McCulloch, W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. URL https://link.springer.com/article/10.1007/BF02478259.

[224] W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz, editor. *Inżynieria biomedyczna: Podstawy i zastosowania. Tom 9: Sieci neuronowe w inżynierii biomedycznej*. Akademicka Oficyna Wydawnicza EXIT, 2013. ISBN 9788378370246. URL http://www.exit.pl/tn9.htm.

[225] D. B. Lenat, R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 978-0201517521. URL https://dl.acm.org/doi/book/10.5555/575523.

[226] R. Neches. Building large knowledge-based systems: Representation and inference in the cyc project: D.b. lenat and r.v. guha. *Artificial Intelligence*, 61(1):65–79, 1993. ISSN 0004-3702. doi: 10.1016/0004-3702(93)90094-R. URL https://www.sciencedirect.com/science/article/pii/000437029390094R.

[227] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th International Conference on Machine Learning*, pages 609–616, 2009. doi: 10.1145/1553374.1553453. URL https://dl.acm.org/doi/10.1145/1553374.1553453.

[228] B. M. Lake and R. Salakhutdinov, J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. doi: 10.1126/science.aab3050. URL https://doi.org/10.1126/science.aab3050.

[229] S. Dehaene. *Reading in the Brain: The New Science of How We Read*. Penguin, New York, 2009. ISBN 978-0-14-311805-3. doi: 10.1111/ijal.12055. URL https://doi.org/10.1111/ijal.12055.

[230] G. Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018. URL https://doi.org/10.48550/arXiv.1801.00631.

[231] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, F. A. Wichmann. Generalisation in humans and deep neural networks. *arXiv preprint arXiv:1808.08750*, 2018. URL https://doi.org/10.48550/arXiv.1808.08750.

[232] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, Y. Zhang.

Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. URL https://doi.org/10.48550/arXiv.2303.12712.

[233] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. doi: 10.1037/h0042519. URL https://doi.org/10.1037/h0042519.

[234] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. doi: 10.1093/oso/9780198538493.001.0001. URL https://doi.org/10.1093/oso/9780198538493.001.0001.

[235] J. Chen, S. Sathe, C. C. Aggarwal, S. D. Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 90–98, Houston, TX, USA, 2017. SIAM. doi: 10.1137/1.9781611974973.11. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611974973.11.

[236] C. C. Aggarwal, S. Sathe. *Outlier Ensembles: An Introduction*. Springer International Publishing, Cham, Switzerland, 2017. ISBN 978-3-319-54764-0. doi: 10.1007/978-3-319-54765-7. URL https://doi.org/10.1007/978-3-319-54765-7.

[237] R. Hecht-Nielsen. Replicator neural networks for universal optimal source coding. *Science*, 269(5232):1860–1863, 1995. doi: 10.1126/science.269.5232.1860. URL https://www.science.org/doi/10.1126/science.269.5232.1860.

[238] G. Hinton, R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647. URL https://www.science.org/doi/10.1126/science.1127647.

[239] P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL https://proceedings.mlr.press/v27/baldi12a.html.

[240] T. Mitchell. *Machine Learning*. McGraw-Hill New York, 1997. ISBN 0070428077.

[241] H. A. Simon. Why should machines learn? In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning*, Symbolic Computation, pages 25–37. Springer, Berlin, Heidelberg, 1983. ISBN 978-3-662-12407--9. doi: 10.1007/978-3-662-12405-5_2. URL https://doi.org/10.1007/978-3-662-12405-5_2.

[242] V. N. Vapnik, A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In A. Gammerman V. Vovk, H. Papadopoulos, editor, *Measures of Complexity*, pages 11–30. Springer, Cham, 2015. ISBN 978-3-319-21851-9. doi: 10.1007/978-3-319-21852-6_3. URL https://doi.org/10.1007/978-3-319-21852-6_3.

[243] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Information Science and Statistics. Springer-Verlag, New York, NY, 1 edition, 2006. ISBN 978-0-387-30865-4. doi: 10.1007/0-387-34239-7. URL https://doi.org/10.1007/0-387-34239-7.

[244] D. M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, Dec. 2004. doi: 10.1021/ci0342472. URL https://doi.org/10.1021/ci0342472.

[245] R. Roelofs, S. Fridovich-Keil, J. Miller, V. Shankar, M. Hardt, B. Recht, L. Schmidt. A meta-analysis of overfitting in machine learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. doi: 10.5555/3454287.3455110. URL https://dl.acm.org/doi/10.5555/3454287.3455110.

[246] S. Watanabe, H. Yamana. Overfitting measurement of convolutional neural networks using trained network weights. *International Journal of Data Science and Analytics*, 14(3):261–278, 2022. doi: 10.1007/s41060-022-00332-1. URL https://doi.org/10.1007/s41060-022-00332-1.

[247] D. Bashir, G.D. Montañez, S. Sehra, P.S. Segura, J. Lauw. *An Information-Theoretic Perspective on Overfitting and Underfitting*, volume 12576 of *Lecture Notes in Computer Science*, pages 364–379. Springer, Cham, 2020. doi: 10.1007/978-3-030-64984-5_27. URL https://doi.org/10.1007/978-3-030-64984-5_27.

[248] P. Cunningham, S.J. Delany. *Underestimation Bias and Underfitting in Machine Learning*, volume 12641 of *Lecture Notes in Computer Science*, pages 15–30. Springer, Cham, 2021. doi: 10.1007/978-3-030-73959-1_2. URL https://doi.org/10.1007/978-3-030-73959-1_2.

[249] G. Bonaccorso. *Mastering Machine Learning Algorithms: Expert techniques to implement popular machine learning algorithms and fine-tune your models*. Packt Publishing, Birmingham, UK, 2018. ISBN 978-1-78862-111-3. Polish edition: Algorytmy uczenia maszynowego Zaawansowane techniki implementacji. Helion SA, 2019.

[250] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[251] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108. URL https://doi.org/10.1162/neco.1995.7.1.108.

[252] L. Bottou. Online algorithms and stochastic approximations. In D. Saad, editor, *Online Learning in Neural Networks*. Cambridge University Press, Cambridge, UK, 1998. URL http://leon.bottou.org/papers/bottou-98x.

[253] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi: 10.48550/arXiv.1412.6980. URL https://doi.org/10.48550/arXiv.1412.6980.

[254] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

[255] G. Hinton. Neural networks for machine learning. Coursera, video lectures, 2012. URL https://www.youtube.com/playlist?list=PLoRl3Ht4JOcdU872GhiYWf6jwrk_SNhz9.

[256] T. Schaul, I. Antonoglou, D. Silver. Unit tests for stochastic optimization. In *International Conference on Learning Representations*, 2014. doi: 10.48550/arXiv.1312.6055. URL https://arxiv.org/abs/1312.6055.

[257] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(1):6765–6816, 2017. doi: 10.48550/arXiv.1603.06560. URL https://jmlr.org/papers/v18/16-558.html.

[258] M.V. Narkhede and P.P. Bartakke, M.S. Sutaone. A review on weight initialization strategies for neural networks. *Artificial Intelligence Review*, 55:291–322, 2022. doi: 10.1007/s10462-021-10033-z. URL https://doi.org/10.1007/s10462-021-10033-z.

[259] N. Lynch, C. Musco, M. Parter. Winner-take-all computation in spiking neural networks. *arXiv preprint arXiv:1904.12591*, 2019. URL https://doi.org/10.48550/arXiv.1904.12591.

[260] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, C. A. Mead. Winner-take-all networks of o(n) complexity. Technical report, California Institute of Technology, Pasadena, CA, 1989. URL https://dl.acm.org/doi/10.5555/89851.89944.

[261] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982. doi: 10.1007/BF00337288. URL https://doi.org/10.1007/BF00337288.

[262] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer Berlin, Heidelberg, Berlin, Heidelberg, 3 edition, 2001. ISBN 978-3-540-67921-9. doi: 10.1007/978-3-642-56927-2. URL https://doi.org/10.1007/978-3-642-56927-2. Published: 16 November 2000, eBook Published: 06 December 2012.

[263] K. Migdał-Najman, K. Najman. *Samouczące się sztuczne sieci neuronowe w grupowaniu i klasyfikacji danych. Teoria i zastosowania w ekonomii*. Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk, 2013. ISBN 978-83-7865-111-6. URL https://kiw.arch.ug.edu.pl/pl/30-ekonomia?id_category=30&n=43#.

[264] A. Nowak-Brzezińska, C. Horyń. Self-organizing map algorithm as a tool for outlier detection. *Procedia Computer Science*, 207:2162–2171, 2022. ISSN 1877-0509. doi: 10.1016/j.procs.2022.09.276. URL https://doi.org/10.1016/j.procs.2022.09.276.

[265] T. Kohonen. *Speech recognition based on topology conserving feature maps*, pages 26–40. MIT Press, 1989. doi: 10.7551/mitpress/4926.003.0004. URL https://doi.org/10.7551/mitpress/4926.003.0004.

[266] E. Erwin, K. Obermayer, K. Schulten. Self-organizing maps: stationary states, metastability, and convergence rate. *Biological Cybernetics*, 67:35–45, 1992. doi: 10.1007/BF00201800. URL https://doi.org/10.1007/BF00201800.

[267] L. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. Prentice Hall, Florida Institute of Technology, 1994. ISBN 978-0-13--334186-7. URL https://dl.acm.org/doi/10.5555/197023.

[268] D. Niebur. An example of unsupervised networks: Kohonen's self-organizing feature map. In *IEEE Power Engineering Society Tutorial Course on Applications of Artificial Neural Networks to Power Systems*, pages 28–38. IEEE Power Engineering Society Tutorial Course, 1996.

[269] C. von der Malsburg. Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik*, 14(2):85–100, 1973. doi: 10.1007/BF00288907. URL https://link.springer.com/article/10.1007/BF00288907.

[270] D. J. Willshaw, C. von der Malsburg. How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London. Series*

B. Biological Sciences, 194(1117):431–445, 1976. doi: 10.1098/rspb.1976.0087. URL https://doi.org/10.1098/rspb.1976.0087.

[271] C. von der Malsburg, D. J. Willshaw. A marker induction mechanism for the establishment of ordered neural mappings: Its application to the retinotectal problem. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 287 (1021):203–243, 1979. doi: 10.1098/rstb.1979.0056. URL https://doi.org/10.1098/rstb.1979.0056.

[272] S. Amari. Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42(3):339–364, 1980. doi: 10.1007/BF02460791. URL https://doi.org/10.1007/BF02460791.

[273] H. Chen and C. Schuffels, R. Orwig. Internet categorization and search: A machine learning approach. *Journal of Visual Communications and Image Representation*, 7 (1):88–102, March 1996. doi: 10.1006/jvci.1996.0008. URL https://doi.org/10.1006/jvci.1996.0008.

[274] J. Vesanto. Neural network tool for data mining: Som toolbox. In *Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000)*, pages 184–196, Oulu, Finland, 2000. Oulun yliopistopaino. URL http://www.cis.hut.fi/projects/somtoolbox/.

[275] D. G. Roussinov, H. Chen. A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. *Communication and Cognition in Artificial Intelligence*, 15(1-2):81–111, 1998.

[276] K. Kiviluoto. Topology preservation in self-organizing maps. In *Proceedings of the International Conference on Neural Networks (ICNN'96)*, volume 1, pages 294–299, 1996. doi: 10.1109/ICNN.1996.548907. URL https://doi.org/10.1109/ICNN.1996.548907.

[277] H. U. Bauer, K. R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, 3(4): 570–579, 1992. doi: 10.1109/72.143371. URL https://ieeexplore.ieee.org/document/143371.

[278] A. Angelopoulou, J. G. Rodríguez, A. Psarrou. Learning 2d hand shapes using the topology preservation model gng. In *Proceedings of the 9th IEEE European Conference on Computer Vision (ECCV 2006)*, volume 3951 of *Lecture Notes in Computer Science*, pages 313–324, Hamburg, Germany, 2006. Springer. doi: 10.1007/11744023_25. URL https://doi.org/10.1007/11744023_25.

[279] J. Lampinen, E. Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2:261–272, 1992. doi: 10.1007/BF00118594. URL https://doi.org/10.1007/BF00118594.

[280] J. Vesanto, M. Sulkava, J. Hollmén. On the decomposition of the self-organizing map distortion measure. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pages 11–16, Kitakyushu, Japan, September 2003. Helsinki University of Technology, Laboratory of Computer and Information Science.

[281] S. Kaski, K. Lagus. Comparing self-organizing maps. In *Artificial Neural Networks — ICANN 96*, volume 1112 of *Lecture Notes in Computer Science*, pages 809–814, Berlin, Heidelberg, 1996. Springer. doi: 10.1007/3-540-61510-5_136. URL https://doi.org/10.1007/3-540-61510-5_136.

[282] J. E. Nash, J. V. Sutcliffe. River flow forecasting through conceptual models part i – a discussion of principles. *Journal of Hydrology*, 10(3):282–290, 1970. doi: 10.1016/0022-1694(70)90255-6. URL https://www.sciencedirect.com/science/article/pii/0022169470902556.

[283] C. J. Willmott. On the validation of models. *Physical Geography*, 2:184–194, 1981. doi: 10.1080/02723646.1981.10642213. URL https://doi.org/10.1080/02723646.1981.10642213.

[284] C. J. Willmott. Some comments on the evaluation of model performance. *Bulletin of the American Meteorological Society*, 63(11):1309–1313, 1982. doi: 10.1175/1520-0477(1982)063<1309:SCOTEO>2.0.CO;2. URL http://dx.doi.org/10.1175/1520-0477(1982)063<1309:SCOTEO>2.0.CO;2.

[285] U. Asan, S. Ercan. An introduction to self-organizing maps. In C. Kahraman, editor, *Computational Intelligence Systems in Industrial Engineering*, volume 6. Atlantis Press, Paris, 2012. doi: 10.2991/978-94-91216-77-0_14. URL https://doi.org/10.2991/978-94-91216-77-0_14.

[286] C. Horyń, A. Nowak-Brzezińska. Detecting outliers in rule-based knowledge bases using self-organizing map and local outlier factor algorithms. *Procedia Computer Science*, 225:2116–2125, 2023. ISSN 1877-0509. doi: 10.1016/j.procs.2023.10.202. URL https://doi.org/10.1016/j.procs.2023.10.202.

[287] M. Refinetti, S. Goldt. The dynamics of representation learning in shallow, non--linear autoencoders. *Proceedings of the 39th International Conference on Machine Learning, PMLR*, 162:18499–18519, 2022. doi: 10.48550/arXiv.2201.02115. URL https://doi.org/10.48550/arXiv.2201.02115.

[288] Y. Liu, C. Ponce, S. L. Brunton, J. N. Kutz. Multiresolution convolutional autoencoders. *Journal of Computational Physics*, 474:111801, 2023. ISSN 0021-9991. doi: 10.1016/j.jcp.2022.111801. URL https://www.sciencedirect.com/science/article/pii/S0021999122008646.

[289] D. Bank, N. Koenigstein, R. Giryes. Autoencoders. *arXiv preprint arXiv:2003.05991*, 2020. doi: 10.48550/arXiv.2003.05991. URL https://doi.org/10.48550/arXiv.2003.05991,https://www.researchgate.net/publication/339945889_Autoencoders.

[290] G. Pang, C. Shen, L. Cao, A. V. D. Hengel. Deep learning for anomaly detection: a review. *ACM Comput. Surv.*, 54(2):1–38, 2021. doi: 10.1145/3439950. URL https://doi.org/10.1145/3439950.

[291] M. Ma, C. Sun, X. Chen. Deep coupling autoencoder for fault diagnosis with multimodal sensory data. *IEEE Trans. Ind. Inf.*, 14(3):1137–1145, 2018. doi: 10.1109/TII.2018.2793246. URL https://doi.org/10.1109/TII.2018.2793246.

[292] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, Y. Xu. Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review*, 57:28, 2024. doi: 10.1007/s10462-023-10662-6. URL https://doi.org/10.1007/s10462-023-10662-6.

[293] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning internal representations by error propagation. Technical report, Institute for Cognitive Science, University of California, San Diego, 1985. URL https://stanford.edu/~jlmcc/papers/PDP/Volume%201/Chap8_PDP86.pdf.

[294] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, April 1982. doi: 10.1073/pnas.79.8.2554. URL https://doi.org/10.1073/pnas.79.8.2554.

[295] H. Bourlard, S. H. Kabil. Autoencoders reloaded. *Biological Cybernetics*, 116:389–406, 2022. doi: 10.1007/s00422-022-00937-6. URL https://doi.org/10.1007/s00422-022-00937-6.

[296] S. Chen, W. Guo. Auto-encoders in deep learning—a review with new perspectives. *Mathematics*, 11(1777), 2023. doi: 10.3390/math11081777. URL https://doi.org/10.3390/math11081777.

[297] NumPy Developers. Numpy: The fundamental package for scientific computing with python, 2024. URL https://numpy.org/doc/stable/. dostęp: 2024-08-07.

[298] J. Tang, Z. Chen, A. W. C. Fu, D. W. Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining, PAKDD 2002*, volume 2336 of *Lecture Notes in Computer Science*, pages 535–548. Springer, Berlin, Heidelberg, 2002. doi: 10.1007/3-540-47887-6_53. URL https://doi.org/10.1007/3-540-47887-6_53.

[299] S. Upadhyaya, K. Singh. Nearest neighbour based outlier detection techniques. *International Journal of Computer Trends and Technology*, 3(2), 2012. URL https://ijcttjournal.org/archives/ijctt-v3i2p119.

[300] O. Alghushairy, R. Alsini, T. Soule, X. Ma. A review of local outlier factor algorithms for outlier detection in big data streams. *Big Data and Cognitive Computing*, 5(1):1, 2021. doi: 10.3390/bdcc5010001. URL https://doi.org/10.3390/bdcc5010001.

[301] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek. Interpreting and unifying outlier scores. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 13–24, 2011. ISBN 978-0-89871-992-5. doi: 10.1137/1.9781611972818.2. URL https://doi.org/10.1137/1.9781611972818.2.

[302] A. Li, W. Xu, Z. Liu, Y. Shi. Improved incremental local outlier detection for data streams based on the landmark window model. *Knowledge and Information Systems*, 63:2129–2155, 2021. doi: 10.1007/s10115-021-01585-1. URL https://doi.org/10.1007/s10115-021-01585-1.

[303] H.-P. Kriegel, P. Kröger, E. Schubert, A. Zimek. Loop: Local outlier probabilities. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pages 1649–1652, 2009. doi: 10.1145/1645953.1646195. URL https://doi.org/10.1145/1645953.1646195.

[304] A. Nowak-Brzezińska, C. Horyń. Outliers in covid 19 data based on rule representation - the analysis of lof algorithm. *Procedia Computer Science*, 192: 3010–3019, 2021. ISSN 1877-0509. doi: 10.1016/j.procs.2021.09.073. URL https://doi.org/10.1016/j.procs.2021.09.073.

[305] C. Horyń, A. Nowak-Brzezińska. Improving detection efficiency: Optimizing block size in the local outlier factor (lof) algorithm. In *Rough Sets*, pages 627–641, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-50959-9. doi: 10.1007/978-3-031-50959-9_43. URL https://doi.org/10.1007/978-3-031-50959-9_43,https://dl.acm.org/doi/10.1007/978-3-031-50959-9_43.

[306] G. Seni, J. Elder. *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*, volume 2 of *Synthesis Lectures on Data Mining and Knowledge Discovery*. Springer Cham, Jan 2010. doi: 10.2200/S00240ED1V01Y200912DMK002. URL https://doi.org/10.2200/S00240ED1V01Y200912DMK002.

[307] T. K. Ho, J. J. Hull, S. N. Srihari. Combination of structural classifiers. In *Pre-Processings, International Association for Pattern Recognition Workshop on Syntactic & Structural Pattern Recognition*, pages 123–136, 1990. URL https://www.researchgate.net/publication/2398607_Combination_of_Structural_Classifiers.

[308] L. K. Hansen, P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, Oct 1990. doi: 10.1109/34.58871. URL https://doi.org/10.1109/34.58871.

[309] E. Kleinberg. Stochastic discrimination. *Annals of Mathematics and Artificial Intelligence*, 1:207–239, 1990. doi: 10.1007/BF01531079. URL https://doi.org/10.1007/BF01531079.

[310] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996. doi: 10.1007/BF00058655. URL https://doi.org/10.1007/BF00058655.

[311] Y. Freund, R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)*, ICML'96, pages 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. ISBN 1558604197. URL https://dl.acm.org/doi/10.5555/3091696.3091715.

[312] C. C. Aggarwal. *Data Classification: Algorithms and Applications*. Chapman and Hall/CRC, New York, 1st edition, 2014. ISBN 9780429102639. doi: 10.1201/b17320. URL https://doi.org/10.1201/b17320.

[313] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, second edition, 2014. ISBN 978-1-118-31523--1. URL https://www.wiley.com/en-us/Combining+Pattern+Classifiers:+Methods+and+Algorithms,+2nd+Edition-p-9781118315231.

[314] L. Rokach. *Pattern Classification Using Ensemble Methods*. World Scientific Publishing Company, 2010. ISBN 978-981-12-0195-0. doi: 10.1142/11325. URL https://doi.org/10.1142/11325.

[315] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*, volume 14. CRC Press, Jun 2012. ISBN 9781439830031. doi: 10.1201/b12207. URL https://doi.org/10.1201/b12207.

[316] A. Elmahalwy, H. Mousa, K. Amin. New hybrid ensemble method for anomaly detection in data science. *International Journal of Electrical and Computer Engineering (IJECE)*, 13:3498, 06 2023. doi: 10.11591/ijece.v13i3.pp3498-3508. URL https://doi.org/10.11591/ijece.v13i3.pp3498-3508.

[317] E. A. Boateng, B. J. W. Unsupervised ensemble methods for anomaly detection in plc-based process control. *arXiv preprint arXiv:2302.02097*, Feb. 2023. URL https://doi.org/10.48550/arXiv.2302.02097.

[318] N. Jeffrey, Q. Tan, J. R. Villar. Using ensemble learning for anomaly detection in cyber–physical systems. *Electronics*, 13(7):1391, 2024. doi: 10.3390/electronics13071391. URL https://doi.org/10.3390/electronics13071391.

[319] E. M. Ferrouhi, I. Bouabdallaoui. A comparative study of ensemble learning algorithms for high-frequency trading. *Scientific African*, 24:e02161, 2024. ISSN 2468-2276. doi: https://doi.org/10.1016/j.sciaf.2024.e02161. URL https://www.sciencedirect.com/science/article/pii/S2468227624001066.

[320] K. Noto, C. Brodley, D. Slonim. Anomaly detection using an ensemble of feature models. In *Proc. IEEE Int. Conf. Data Min.*, pages 953–958, Dec. 2010. doi: 10.1109/ICDM.2010.140. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3197694/.

[321] S. Guha, N. Mishra, G. Roy, O. Schrijvers. Robust random cut forest based anomaly detection on streams. In *Proc. 33rd Int. Conf. Machine Learning*, volume 48 of *Proc. Machine Learning Research*, pages 2712–2721, New York, NY, USA, Jun. 2016. PMLR. doi: 10.1145/2882903.2915222. URL https://proceedings.mlr.press/v48/guha16.html.

[322] H. V. Nguyen, H. H. Ang, V. Gopalkrishnan. Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *Database Systems for Advanced Applications. DASFAA 2010. Lecture Notes in Computer Science*, volume 5981, pages 368–383. Springer, Berlin, Heidelberg, 2010. ISBN 978-3-642-12025--1. doi: 10.1007/978-3-642-12026-8_29. URL https://doi.org/10.1007/978-3-642-12026-8_29.

[323] S. Rayana, L. Akoglu. Less is more: Building selective anomaly ensembles with application to event detection in temporal graphs. *arXiv preprint arXiv:1501.01924*,

Jan 2015. doi: 10.48550/arXiv.1501.01924. URL https://doi.org/10.48550/arXiv.1501.01924.

[324] S. Rayana, W. Zhong, L. Akoglu. Sequential ensemble learning for outlier detection: A bias-variance perspective. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 1167–1172, Dec 2016. doi: 10.1109/ICDM.2016.0154. URL https://doi.org/10.1109/ICDM.2016.0154.

[325] C. C. Aggarwal, S. Sathe. Theoretical foundations and algorithms for outlier ensembles. *SIGKDD Explor. Newsl.*, 17(1):24–47, Jun. 2015. doi: 10.1145/2830544.2830549. URL https://doi.org/10.1145/2830544.2830549.

[326] D. M. J. Tax, R. P. W. Duin. Combining one-class classifiers. In *Lecture Notes in Computer Science*, volume 2096, pages 299–308. Springer, 2001. doi: 10.1007/3-540-48219-9_30. URL https://doi.org/10.1007/3-540-48219-9_30.

[327] Y. Freund, R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Comput. Learn. Theory*, volume 55, pages 119–139, 1995. URL http://link.springer.com/chapter/10.1007/3-540-59119-2_166.

[328] A. F. Emmott, S. Das, T. Dietterich, A. Fern, W. Wong. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ODD '13, pages 16–21, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450323352. doi: 10.1145/2500853.2500858. URL https://doi.org/10.1145/2500853.2500858.

[329] A. Emmott, S. Das, T. G. Dietterich, A. Fern, W.-K. Wong. A meta-analysis of the anomaly detection problem. *arXiv:1503.01158*, 2015. URL https://doi.org/10.48550/arXiv.1503.01158.

[330] J. Gao, P. -N. Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 212–221. IEEE, 2006. doi: 10.1109/ICDM.2006.43. URL https://doi.org/10.1109/ICDM.2006.43.

[331] S. Bhanja, A. Das. Impact of data normalization on deep neural network for time series forecasting. *arXiv preprint arXiv:1812.05519*, Dec 2018. URL https://doi.org/10.48550/arXiv.1812.05519.

[332] B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, Nov 2016. doi: 10.1007/s13748-016-0094-0. URL https://doi.org/10.1007/s13748-016-0094-0.

[333] L. Dube, T. Verster. Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. *Data Science in Finance and Economics*, 3(4):354–379, October 2023. ISSN 2769-2140. doi: 10.3934/DSFE.2023021. URL https://www.aimspress.com/article/doi/10.3934/DSFE.2023021.

[334] J. M. Johnson, T. M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0192-5. URL https://doi.org/10.1186/s40537-019-0192-5.

[335] M. Buda, A. Maki, M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. ISSN 0893-6080. doi: 10.1016/j.neunet.2018.07.011. URL https://www.sciencedirect.com/science/article/pii/S0893608018302107.

[336] A. Fraser, D. Marcu. Squibs and discussions: Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303, 2007. doi: 10.1162/coli.2007.33.3.293. URL https://aclanthology.org/J07-3002.

[337] D. M. W. Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Mach. Learn. Technol.*, 2, 2008. URL https://doi.org/10.48550/arXiv.2010.16061.

[338] W. C. Bown. Sensitivity and specificity versus precision and recall, and related dilemmas. *Journal of Classification*, 41(3):402–426, 2024. doi: 10.1007/s00357-024-09478-y. URL https://doi.org/10.1007/s00357-024-09478-y.

[339] D. M. W. Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *Journal of Machine Learning Research*, 45(1): 1–28, 2008. doi: 10.48550/arXiv.2010.16061. URL https://arxiv.org/abs/2010.16061.

[340] M. Buckland, F. Gey. The relationship between recall and precision. *Journal of the American Society for Information Science*, 45(1):12–19, 1994. doi: 10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L. URL https://doi.org/10.1002/(SICI)1097-4571(199401)45:1<12::AID-ASI2>3.0.CO;2-L.

[341] Y. Sasaki. The truth of the f-measure. *Teach Tutor Mater*, 1(5): 1–5, 2007. URL https://nicolasshu.com/assets/pdf/Sasaki_2007_The%20Truth%20of%20the%20F-measure.pdf.

[342] D. Chicco, G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21

(1):6, January 2020. ISSN 1471-2164. doi: 10.1186/s12864-019-6413-7. URL https://doi.org/10.1186/s12864-019-6413-7.

[343] J. Swets, R. Dawes, J. Monahan. Better decisions through science. *Scientific American*, 283:82–87, 11 2000. doi: 10.1038/scientificamerican1000-82. URL https://www.scientificamerican.com/article/better-decisions-through-science/.

[344] T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8): 861–874, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X.

[345] E. Şengönül, R. Samet, Q. Abu Al-Haija, A. Alqahtani, B. Alturki, A.A. Alsulami. An analysis of artificial intelligence techniques in surveillance video anomaly detection: A comprehensive survey. *Applied Sciences*, 13(4956), 2023. doi: 10.3390/app13084956. URL https://doi.org/10.3390/app13084956.

[346] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 0031-3203. doi: 10.1016/S0031-3203(96)00142-2. URL https://www.sciencedirect.com/science/article/pii/S0031320396001422.

[347] J. Koronacki, J. Ćwik. *Statystyczne Systemy Uczące Się*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2 edition, 2015. ISBN 978-83-60434-56-7. URL http://www.exit.pl/kor.htm.

[348] A. Petrie, C. Sabin. *Statystyka Medyczna w Zarysie*. Wydawnictwo Lekarskie PZWL, Warszawa, 2006. ISBN 83-200-3312-8. URL https://pzwl.pl/autor/Aviva-Petrie,a,2305576.

[349] Z. Omiotek. *Wybrane problemy modelowania predykcyjnego w diagnostyce technicznej i medycznej*. Monografie – Politechnika Lubelska. Wydawnictwo Politechniki Lubelskiej, Lublin, 2021. ISBN 978-83-7947-455-4. URL https://bc.pollub.pl/dlibra/publication/13915/edition/13580.

[350] J. A. Hanley, B. Mcneil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143:29–36, 1982. doi: 10.1148/radiology.143.1.7063747. URL https://doi.org/10.1148/radiology.143.1.7063747.

[351] J. Miao, W. Zhu. Precision-recall curve (prc) classification trees. *arXiv preprint arXiv:2011.07640*, November 2020. doi: 10.48550/arXiv.2011.07640. URL https://arxiv.org/abs/2011.07640v1.

[352] T. Saito, M. Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3): e0118432, 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

[353] M. Goadrich, L. Oliphant, J. Shavlik. Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction. In *Proceedings of the 14th International Conference on Inductive Logic Programming, Springer*, volume 3194, pages 98–115, 2004. ISBN 978-3-540-22941-4. doi: 10.1007/978-3-540-30109-7_11. URL https://link.springer.com/chapter/10.1007/978-3-540-30109-7_11.

[354] J. Davis, M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning, ACM*, pages 233–240, 2006. doi: 10.1145/1143844.1143874. URL https://doi.org/10.1145/1143844.1143874.

[355] M. Szreder. Istotność statystyczna w czasach big data. *Wiadomości Statystyczne*, 64(11):42–57, 2019. URL https://bazekon.uek.krakow.pl/en/rekord/171579812.

[356] V. Amrhein, S. Greenland, B. McShane. Retire statistical significance. *Nature*, 567: 305–307, 2019. doi: 10.1038/d41586-019-00857-9. URL https://doi.org/10.1038/d41586-019-00857-9.

[357] V. Amrhein, D. Trafimow, S. Greenland. Inferential statistics as descriptive statistics: There is no replication crisis if we don't expect replication. *The American Statistician*, 73(sup 1):262–270, 2019. doi: 10.1080/00031305.2018.1543137. URL https://doi.org/10.1080/00031305.2018.1543137.

[358] J. Koronacki, J. Mielniczuk. *Statystyka dla studentów kierunków technicznych i przyrodniczych*. WNT, Warszawa, 2001. ISBN 83-204-2684-7.

[359] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1: 80–83, 1945. doi: 10.2307/3001968. URL https://doi.org/10.2307/3001968.

[360] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1988. doi: 10.4324/9780203771587. URL https://doi.org/10.4324/9780203771587.

[361] R. A. Fisher. *Statistical Methods for Research Workers*, pages 66–70. Springer Series in Statistics. Springer, New York, NY, 1992. ISBN 978-0-387-94039--7. doi: 10.1007/978-1-4612-4380-9_6. URL https://doi.org/10.1007/978-1-4612-4380-9_6.

[362] D. G. Altman. *Practical Statistics for Medical Research*. Chapman and Hall/CRC, London, 1990. doi: 10.1201/9780429258589. URL https://doi.org/10.1201/9780429258589.

[363] R. A. Johnson, D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Inc. Division of Simon and Schuster, One Lake Street, Upper Saddle River, NJ, United States, 1988. ISBN 978-0-13-041146-4. URL https://dl.acm.org/doi/book/10.5555/59551.

[364] G. James, D. Witten, T. Hastie, R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer, New York, NY, 2nd edition, 2021. ISBN 978-1-0716-1417-4. doi: 10.1007/978-1-0716-1418-1. URL https://doi.org/10.1007/978-1-0716-1418-1.

[365] W. S. Gosset (pseud. Student). The probable error of a mean. *Biometrika*, 6(1): 1–25, 1908. doi: 10.2307/2331554. URL https://doi.org/10.2307/2331554.

[366] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996. URL https://doi.org/10.1007/BF02295996.

[367] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley and Sons, New York, 3rd edition, 1999.

[368] W. H. Kruskal. A nonparametric test for the several sample problem. *Annals of Mathematical Statistics*, 23:525–540, 1952. doi: 10.1214/aoms/1177729332. URL https://doi.org/10.1214/aoms/1177729332.

[369] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200): 675–701, 1937. doi: 10.1080/01621459.1937.10503522. URL https://doi.org/10.1080/01621459.1937.10503522.

[370] G. E. Hinton, R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647. URL https://doi.org/10.1126/science.1127647.

[371] J. Zhao, Y. Kim, K. Zhang, A. M. Rush, Y. LeCun. Adversarially regularized autoencoders. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5902–5911. PMLR, Jul 2018. URL https://proceedings.mlr.press/v80/zhao18b.html.

[372] Y. Bengio, L. Yao, G. Alain, P. Vincent. Generalized denoising auto-encoders as generative models. *arXiv preprint arXiv:1305.6663*, 2013. doi: 10.48550/arXiv.1305.6663. URL https://arxiv.org/abs/1305.6663.

[373] Y. Bengio, L. Yao, G. Alain, P. Vincent. What regularized auto-encoders learn from the data generating distribution. *arXiv preprint arXiv:1211.4246*, 2012. doi: 10.48550/arXiv.1211.4246. URL https://arxiv.org/abs/1211.4246.

[374] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010. URL https://jmlr.csail.mit.edu/papers/v11/vincent10a.html.

[375] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1): 1–39, 2010. doi: 10.1007/s10462-009-9124-7. URL https://doi.org/10.1007/s10462-009-9124-7.

[376] M. Topolski. *Metody ekstrakcji cech w uczeniu maszynowym. Nowe trendy inżynierii cech*. Exit, 2023. ISBN 9788378371397. URL http://exit.pl/top.htm.

[377] P. Bühlmann. Bagging, boosting and ensemble methods. In *Handbook of Computational Statistics*, Springer Handbooks of Computational Statistics, pages 985–1022. Springer, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-21551-3_33. URL https://doi.org/10.1007/978-3-642-21551-3_33.

[378] K. Zhang, M. Hutter, H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Advances in Knowledge Discovery and Data Mining. PAKDD 2009*, volume 5476 of *Lecture Notes in Computer Science*, pages 813–822. Springer, Berlin, Heidelberg, 2009. doi: 10.1007/978-3-642-01307-2_84. URL https://doi.org/10.1007/978-3-642-01307-2_84.

[379] A. L. M. Chiu, A. W.-C. Fu. Enhancements on local outlier detection. In *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.*, pages 298–307. IEEE, 2003. doi: 10.1109/IDEAS.2003.1214939. URL https://doi.org/10.1109/IDEAS.2003.1214939.

[380] S.-Y. Jiang, Q.-H. Li, K.-L. Li, H. Wang, Z.-L. Meng. Glof: A new approach for mining local outlier. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, volume 1, pages 157–162. IEEE, 2003.

[381] K. Cao, L. Shi, G. Wang, D. Han, M. Bai. Density-based local outlier detection on uncertain data. In *Web-Age Information Management. WAIM 2014. Lecture Notes in Computer Science*, volume 8485. Springer, Cham,

2014. doi: 10.1007/978-3-319-08010-9_9. URL https://doi.org/10.1007/978-3-319-08010-9_9.

[382] E. Lozano, E. Acuna. Parallel algorithms for distance-based and density-based outliers. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, page 4 pp. IEEE, 2005. doi: 10.1109/ICDM.2005.116. URL https://doi.org/10.1109/ICDM.2005.116.

[383] M. Salehi, C. Leckie, J. C. Bezdek, T. Vaithianathan, X. L. Zhang. Fast memory efficient local outlier detection in data streams. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3246–3260, Dec 2016. doi: 10.1109/TKDE.2016.2597833. URL https://doi.org/10.1109/TKDE.2016.2597833.

[384] A. Guttman. R-trees: a dynamic index structure for spatial searching. *ACM SIGMOD Record*, 14(2):47–57, 1984. doi: 10.1145/971697.602266. URL https://doi.org/10.1145/971697.602266.

[385] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. doi: 10.1145/361002.361007. URL https://doi.org/10.1145/361002.361007.

[386] A. Beygelzimer, S. Kakade, J. Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 97–104, 2006. doi: 10.1145/1143844.1143857. URL https://doi.org/10.1145/1143844.1143857.

[387] S. Su, L. Xiao, L. Ruan, F. Gu, S. Li, Z. Wang, R. Xu. An efficient density-based local outlier detection approach for scattered data. *IEEE Access*, 7:1006–1020, 2019. doi: 10.1109/ACCESS.2018.2886197. URL https://doi.org/10.1109/ACCESS.2018.2886197.

[388] R. Bell, Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, Dec 2007. doi: 10.1145/1345448.1345465. URL https://doi.org/10.1145/1345448.1345465.

[389] Python Software Foundation. time — time access and conversions, 2024. URL https://docs.python.org/3/library/time.html. dostęp: 2024-08-07.

[390] SciPy Community. distance.cdist, 2024. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html. dostęp: 2024-08-07.

[391] SciPy Community. Scipy: Open source scientific tools for python, 2024. URL https://docs.scipy.org/doc/scipy/reference/. dostęp: 2024-08-07.

[392] M. Bohanec. Car evaluation. UCI Machine Learning Repository, 1997. URL https://doi.org/10.24432/C5JP48. dostęp 1 sierpnia 2024.

[393] UCI Machine Learning Repository. Mushroom classification, 1987. URL https://doi.org/10.24432/C5959T,https://www.kaggle.com/datasets/uciml/mushroom-classification. Dostępny również na Kaggle: Mushroom Classification Dataset, dostęp: luty 2024.

[394] S. Moro, P. Rita, P. Cortez. Bank marketing. UCI Machine Learning Repository, 2012. URL https://doi.org/10.24432/C5K306. dostęp 1 sierpnia 2024.

[395] citibike. Citibike, 2013. URL https://www.citibikenyc.com/system-data, https://www.kaggle.com/datasets/sujan97/citibike-system-data. Badany zbiór pierwszych 20,000 rekordów, dostęp: 1 sierpnia 2024.

[396] B. Becker, R. Kohavi. Adult. UCI Machine Learning Repository, 1996. URL https://doi.org/10.24432/C5XW20. dostęp 1 sierpnia 2024.

[397] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, G. Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10):4915–4928, 2014. URL https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud. dostęp: 1 sierpnia 2024.

[398] S. A. Danziger, R. Baronio, L. Ho, L. Hall, K. Salmon, G. W. Hatfield, P. Kaiser, R. H. Lathrop. Predicting positive p53 cancer rescue regions using most informative positive (mip) active learning. *PLoS Comput. Biol.*, 5(9):e1000498, Sep 2009. doi: 10.1371/journal.pcbi.1000498. URL https://doi.org/10.1371/journal.pcbi.1000498.

[399] Google. Colab pro+. https://colab.research.google.com/signup, 2024. dostęp: 2024-08-07.

[400] W. Jin, A. K. H. Tung, J. Han, W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Advances in Knowledge Discovery and Data Mining. PAKDD 2006*, volume 3918 of *Lecture Notes in Computer Science*, pages 577–593. Springer, Berlin, Heidelberg, 2006. doi: 10.1007/11731139_68. URL https://doi.org/10.1007/11731139_68.

[401] D. Pokrajac, A. Lazarevic, L. J. Latecki. Incremental local outlier detection for data streams. In *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 504–515, 2007. doi: 10.1109/CIDM.2007.368917. URL https://doi.org/10.1109/CIDM.2007.368917.

[402] F. Angiulli, C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Principles of Data Mining and Knowledge Discovery. PKDD 2002*, volume 2431 of *Lecture Notes in Computer Science*, pages 15–27. Springer, Berlin, Heidelberg, 2002. doi: 10.1007/3-540-45681-3_2. URL https://doi.org/10.1007/3-540-45681-3_2.

[403] J. Snoek, H. Larochelle, R. P. Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, June 2012. URL https://arxiv.org/abs/1206.2944.

[404] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. ISBN 026218253X. URL http://www.GaussianProcess.org/gpml.

[405] E. Brochu, V. M. Cora, N. de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, December 2010. URL https://doi.org/10.48550/arXiv.1012.2599. Submitted on 12 Dec 2010.

[406] F. Hutter, H. H. Hoos, K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *LION'05: Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011. doi: 10.1007/978-3-642-25566-3_40. URL https://doi.org/10.1007/978-3-642-25566-3_40.

[407] Scikit-Optimize Developers. Scikit-optimize: Sequential model-based optimization with python, 2024. URL https://scikit-optimize.github.io/stable/. dostęp: 2024-08-07.

[408] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, S.-H. Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019. ISSN 1674-862X. doi: 10.11989/JEST.1674-862X.80904120. URL https://www.sciencedirect.com/science/article/pii/S1674862X19300047.

[409] Scikit-Optimize Developers. Scikit-optimize: Sequential model-based optimization with python, 2024. URL https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html. dostęp: 2024-08-07.

[410] D. Dua, C. Graff. Uci machine learning repository, 2017. URL https://archive.ics.uci.edu/ml/index.php. dostęp: 2024-08-09.

[411] Kaggle Inc. Kaggle: Your machine learning and data science community, 2024. URL https://www.kaggle.com/. dostęp: 2024-08-09.

[412] scikit-learn developers. sklearn.preprocessing.labelencoder — scikit-learn 0.24.2 documentation, 2021. URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html. dostęp: 2024-08--09.

[413] scikit-learn developers. Minmaxscaler — scikit-learn 0.24.2 documentation, 2021. URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. dostęp: 2024-08-09.

[414] Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine quality white dataset, 2009. URL https://doi.org/10.24432/C56S3T. dostęp: luty 2024.

[415] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47 (4):547–553, 2009. doi: 10.1016/j.dss.2009.05.016. URL https://doi.org/10.1016/j.dss.2009.05.016. dostęp: luty 2024.

[416] R. Lathrop. p53 mutants. UCI Machine Learning Repository, 2010. URL https://doi.org/10.24432/C5T89H. https://doi.org/10.1371/journal.pcbi.1000498, Artykuł wprowadzający: Danziger S.A., Baronio R., Ho L., Hall L., Salmon K., Hatfield G.W., Kaiser P., Lathrop R., Predicting Positive p53 Cancer Rescue Regions Using Most Informative Positive (MIP) Active Learning, PLoS Comput. Biol., 2009, dostęp 1 sierpnia 2024.

[417] A. Chawda, S. Grimm, M. Kloft. Vehicle claims labeled dataset for unsupervised anomaly detection. Prezentowane na warsztatach NeurIPS 2022, 2022. URL https://github.com/ajaychawda58/uadad. https://github.com/ajaychawda58/UADAD/tree/main/data/vehicle_claims, dostęp: luty 2024.

[418] A. Chawda, S. Grimm, M. Kloft. Unsupervised anomaly detection for auditing data and impact of categorical encodings. *arXiv preprint arXiv:2210.14056*, 2022. doi: 10.48550/arXiv.2210.14056. URL https://arxiv.org/abs/2210.14056.

[419] M. Bain, A. Hoff. Chess endgame database for white king and rook against black king (krkopt), 1994. URL https://doi.org/10.24432/C57W2S. dostęp: luty 2024.

[420] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, P. Chan. Kdd cup 1999 data. Dataset available at UCI Machine Learning Repository and Kaggle, 1999. URL https://doi.org/10.24432/C51C7N. https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data, dostęp: luty 2024.

[421] kdd cup. kdd cup 1999. https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Introduction, 1999. dostęp: 10 sierpnia 2024.

[422] ACM SIGKDD. KDD Conference Proceedings. https://kdd.hosting.acm.org/proceedings, 1999. dostęp: 10 sierpnia 2024.

[423] J. C. Platt and G. Pang. w7a dataset for binary classification tasks, 1998. URL https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html. https://github.com/GuansongPang/ADRepository-Anomaly-detection-datasets/blob/main/categorical%20data/w7a-libsvm-nonsparse.tar.xz, https://www.openml.org/search?type=data&sort=runs&id=1587&status=active, dostęp: luty 2024.

[424] J. C. Platt. Fast training of support vector machines using sequential minimal optimization, 1999. URL https://www.researchgate.net/publication/234786663_Fast_Training_of_Support_Vector_Machines_Using_Sequential_Minimal_Optimization. dostęp: luty 2024.

[425] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In *Advances in Kernel Methods-Support Vector Learning*, pages 208–219. MIT Press, 1998.

[426] Centers for Disease Control and Prevention (CDC). Covid 19 case surveillance public use dataset, 2020. URL https://www.kaggle.com/arashnic/covid19-case-surveillance-public-use-dataset. https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/vbim-akqf/about_data, dostęp: luty 2024.

[427] RapidMiner, Inc. *RapidMiner Studio*, 2024. dostępny na: https://rapidminer.com/.

[428] scikit-learn developers. *LabelEncoder documentation*, 2024. URL https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html.

[429] Scikit-learn developers. *OneHotEncoder*, 2024. dostępny na: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[430] Python Software Foundation. time — time access and conversions, 2023. URL https://docs.python.org/3/library/time.html. Python 3.11.0 Documentation.

[431] UCI Machine Learning Repository. Thyroid disease data set, 1987. URL https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease. dostęp: sierpień 2024.

[432] UCI Machine Learning Repository. Breast cancer wisconsin (diagnostic) data set, 1995. URL https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). dostęp: sierpień 2024.

[433] K. P. Bennett, O. L. Mangasarian. Robust Linear Programming Discrimination of Two Linearly Inseparable Sets. *Optimization Methods and Software*, 1(1):23–34, 1992.

[434] H. He, E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sep 2009. doi: 10.1109/TKDE.2008.239. URL https://doi.org/10.1109/TKDE.2008.239.

[435] Y. Tang, Y. Zhang, N. V. Chawla, S. Krasser. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009. doi: 10.1109/TSMCB.2008.2002909. URL https://doi.org/10.1109/TSMCB.2008.2002909.

[436] A. Mahani, A. R. Baba Ali. *Classification Problem in Imbalanced Datasets*. IntechOpen, May 2020. doi: 10.5772/intechopen.89603. URL https://doi.org/10.5772/intechopen.89603.

Spis rysunków

Spis tabel

Spis Algorytmów

Wykaz badań dodatkowych

W ramach przygotowań do rozprawy przeprowadzono dodatkowe eksperymenty i badania, które nie zostały szczegółowo omówione w jej treści lub są w trakcie recenzji:

- Zidentyfikowano brak narzędzi do wykrywania odchyleń przed grupowaniem. W odpowiedzi na ten problem stworzono pakiet SOaCRaport, który umożliwia wykrywanie odchyleń w zbiorach danych przy użyciu algorytmów LOF i COF oraz ocenę jakości utworzonych grup [82],
- Dokonano analizy czterech algorytmów do wykrywania odchyleń w bazach wiedzy opartych na regułach: Local Outlier Factor (LOF), Connectivity-based Outlier Factor (COF), K-means oraz Small Clusters, który został opracowany jako autorski algorytm. LOF i COF okazały się najskuteczniejsze w poprawie jakości klastrów [101],
- Przeprowadzono badania nad algorytmami LOF, COF i K-means w bazach wiedzy opartych na regułach, analizując wpływ usunięcia odchyleń na jakość klastrów. Najlepsze wyniki uzyskano dla COF [85],
- Zastosowano dwufazową procedurę, w której optymalizowano strukturę klastrów reguł w bazie wiedzy Covid-19, a następnie użyto algorytmu LOF do eliminacji nietypowych reguł, co poprawiło jakość klastrów oraz samej bazy wiedzy [304],
- Skoncentrowano się na wykrywaniu nietypowych reguł w systemach wspomagania decyzji opartych na bazach wiedzy, stosując algorytmy LOF i SOM. Oba algorytmy skutecznie identyfikowały odchylenia, co przyczyniło się do poprawy kompletności bazy wiedzy [286],
- Przeprowadzono badania porównawcze SOM i LOF, które wykazały, że SOM jest szybszy w wykrywaniu odchyleń, a jego skuteczność zależy od danych i hiperparametrów uczenia, osiągając dobrą dokładność na danych jakościowych [264].

- Wykrywanie odchyleń w danych kategorycznych i mieszanych zbadano przy użyciu algorytmów LOF, SOM oraz Autoenkodera. Badania porównawcze wykazały szczególnie wysoką skuteczność Autoenkodera w identyfikacji anomalii, a także poprawę efektywności algorytmu LOF dzięki zastosowaniu techniki optymalizacji wielkości bloków (*block size*). Artykuł *„Anomaly Detection Techniques: A Comparative Study”* jest w trakcie recenzji w momencie składania rozprawy.
- Na zaproszenie czasopisma kontynuowano badania nad optymalizacją wielkości bloków w procesie wykrywania anomalii z wykorzystaniem algorytmu LOF. Wyniki potwierdzają skuteczność proponowanego podejścia, które przyspiesza proces wykrywania anomalii przy jednoczesnym zachowaniu wysokiej skuteczności. Artykuł *„Automatic Block Size Optimization in the LOF Algorithm for Efficient Anomaly Detection”* przeszedł recenzję i jest na etapie drobnych poprawek, z dużą szansą na publikację w prestiżowym czasopiśmie *Applied Soft Computing* (CiteScore 15,8; IF 7,2) w momencie składania rozprawy.

Dodatek A. Spis zawartości dołączonej płyty DVD

Na płycie DVD załączono plik *Struktura_danych_na_płycie_DVD.pdf*, który zawiera szczegółowy opis wszystkich plików i materiałów zapisanych na płycie, w tym:

1. Rozprawa doktorska – niniejszy dokument,
2. Kod źródłowy oprogramowania,
3. Zbiory danych wykorzystane w eksperymentach.

Dodatek B. Wyniki Trinity SALT dla wszystkich zbiorów danych

Tabela 1: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Bank mktg (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Bank mktg (R)	Bł. kl.	0,0091	0,0093	0,0068	0,0081	0,0089	0,0101	0,0128	0,0152
Cechy: 16 (w tym kateg.)	Spec.	0,9954	0,9953	0,9966	0,9959	0,9955	0,9949	0,9894	0,9878
Szk.:20072;111;0,55%	Prec.	0,1802	0,1622	0,3874	0,2703	0,1982	0,0901	0,2355	0,1678
	Czuł.	0,1802	0,1622	0,3874	0,2703	0,1982	0,0901	0,5856	0,4414
Tst.:20072;111;0,55%	F1	0,1802	0,1622	0,3874	0,2703	0,1982	0,0901	0,3359	0,2432
	FP	0,0046	0,0047	0,0034	0,0041	0,0045	0,0051	0,0106	0,0122
T:40144;222;0,55%	FN	0,8198	0,8378	0,6126	0,7297	0,8018	0,9099	0,4144	0,5586
	Czas	0,2518	0,2647	1,2018	1,0636	0,3312	0,3201	4,0305	3,9189

Tabela 2: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Chess krkopt (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Chess krkopt (R)	Bł. kl.	0,0013	0,0014	0,0017	0,0014	0,0018	0,0017	0,0029	0,0026
Cechy: 6 (w tym kateg.)	Spec.	0,9993	0,9993	0,9991	0,9993	0,9991	0,9992	0,9976	0,9977
Szk.:19639;19;0,10%	Prec.	0,3158	0,2500	0,1053	0,2500	0,0526	0,1250	0,1607	0,2083
	Czuł.	0,3158	0,2500	0,1053	0,2500	0,0526	0,1250	0,4737	0,6250
Tst.:8417;8;0,10%	F1	0,3158	0,2500	0,1053	0,2500	0,0526	0,1250	0,2400	0,3125
	FP	0,0007	0,0007	0,0009	0,0007	0,0009	0,0008	0,0024	0,0023
T:28056;27;0,10%	FN	0,6842	0,7500	0,8947	0,7500	0,9474	0,8750	0,5263	0,3750
	Czas	0,1171	0,0724	0,3183	0,1712	15,0379	5,3279	16,6642	6,6245

Tabela 3: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Covid19 (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Covid19 (R)	Bł. kl.	0,0114	0,0124	0,0122	0,0130	0,0127	0,0130	0,0150	0,0149
Cechy: 10 (w tym kateg.)	Spec.	0,9943	0,9937	0,9938	0,9934	0,9936	0,9934	0,9893	0,9893
Szk.:23737;232;0,98%	Prec.	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,3386	0,3415
	Czuł.	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,5560	0,5600
Tst.:10174;100;1,00%	F1	0,4181	0,3700	0,3750	0,3400	0,3491	0,3400	0,4209	0,4242
	FP	0,0057	0,0063	0,0062	0,0066	0,0064	0,0066	0,0107	0,0107
T:33911;332;0,98%	FN	0,5819	0,6300	0,6250	0,6600	0,6509	0,6600	0,4440	0,4400
	Czas	0,0935	0,0528	4,1264	1,8368	0,1166	0,0499	7,7652	2,6829

Tabela 4: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Mushroom (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Mushroom (R)	Bł. kl.	0,0007	0,0063	0,0000	0,0047	0,0007	0,0047	0,0007	0,0071
Cechy: 22 (w tym kateg.)	Spec.	0,9997	0,9968	1,0000	0,9976	0,9997	0,9976	0,9993	0,9937
Szk.:2975;29;0,97%	Prec.	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	0,9355	0,6000
	Czuł.	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	1,0000	0,9231
Tst.:1275;13;1,02%	F1	0,9655	0,6923	1,0000	0,7692	0,9655	0,7692	0,9667	0,7273
	FP	0,0003	0,0032	0,0000	0,0024	0,0003	0,0024	0,0007	0,0063
T:4250;42;0,99%	FN	0,0345	0,3077	0,0000	0,2308	0,0345	0,2308	0,0000	0,0769
	Czas	0,0630	0,0522	11,5105	4,5333	0,6881	0,1224	13,4732	4,4240

Tabela 5: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Breast cancer (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Breast cancer (R)	Bł. kl.	0,0000	0,0110	0,0111	0,0110	0,0111	0,0221	0,0111	0,0166
Cechy: 30 (numeryczne)	Spec.	1,0000	0,9944	0,9944	0,9944	0,9944	0,9888	0,9888	0,9832
Szk.:180;2;1,11%	Prec.	1,0000	0,5000	0,5000	0,5000	0,5000	0,0000	0,5000	0,4000
	Czuł.	1,0000	0,5000	0,5000	0,5000	0,5000	0,0000	1,0000	1,0000
Tst.:181;2;1,10%	F1	1,0000	0,5000	0,5000	0,5000	0,5000	0,0000	0,6667	0,5714
	FP	0,0000	0,0056	0,0056	0,0056	0,0056	0,0112	0,0112	0,0168
T:361;4;1,11%	FN	0,0000	0,5000	0,5000	0,5000	1,0000	1,0000	0,0000	0,0000
	Czas	0,0578	0,0519	0,0952	0,1246	0,0021	0,0011	0,2258	0,2260

Tabela 6: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Credit Card (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Credit Card (R)	Bł. kl.	0,0016	0,0019	0,0017	0,0019	0,0019	0,0021	0,0024	0,0021
Cechy: 30 (numeryczne)	Spec.	0,9992	0,9991	0,9992	0,9992	0,9991	0,9990	0,9983	0,9986
Szk.:199364;344;0,17%	Prec.	0,5262	0,4595	0,5145	0,5135	0,4622	0,4459	0,3713	0,4129
	Czuł.	0,5262	0,4595	0,5145	0,5135	0,4622	0,4459	0,5872	0,5608
Tst.:85433;148;0,17%	F1	0,5262	0,4595	0,5145	0,5135	0,4622	0,4459	0,4550	0,4756
	FP	0,0008	0,0009	0,0008	0,0008	0,0009	0,0010	0,0017	0,0014
T:284807;492;0,17%	FN	0,4738	0,5405	0,4855	0,4865	0,5378	0,5541	0,4128	0,4392
	Czas	0,1261	0,0584	130,2744	45,9107	72,5731	31,0610	199,2640	89,9778

Tabela 7: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Thyroid disease (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Thyroid disease (R)	Bł. kl.	0,0274	0,0256	0,0194	0,0226	0,0303	0,0334	0,0423	0,0462
Cechy: 21 (numeryczne)	Spec.	0,9861	0,9870	0,9901	0,9885	0,9846	0,9830	0,9651	0,9615
Szk.:4748;82;1,73%	Prec.	0,2073	0,2571	0,4390	0,3429	0,1220	0,0286	0,2126	0,1895
	Czuł.	0,2073	0,2571	0,4390	0,3429	0,1220	0,0286	0,5366	0,5143
Tst.:2035;35;1,72%	F1	0,2073	0,2571	0,4390	0,3429	0,1220	0,0286	0,3045	0,2769
	FP	0,0139	0,0130	0,0099	0,0115	0,0154	0,0170	0,0349	0,0385
T:6783;117;1,72%	FN	0,7927	0,7429	0,5610	0,6571	0,8780	0,9714	0,4634	0,4857
	Czas	0,0573	0,0449	11,1231	7,4078	1,7464	0,2890	14,3847	7,1774

Tabela 8: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Vehicle Claims (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Vehicle Claims (R)	Bł. kl.	0,0092	0,0091	0,0076	0,0081	0,0076	0,0075	0,0125	0,0126
Cechy: 19 (w tym kateg.)	Spec.	0,9954	0,9954	0,9962	0,9959	0,9962	0,9962	0,9898	0,9899
Szk.:149095;1042;0,70%	Prec.	0,3445	0,3475	0,4559	0,4215	0,4559	0,4619	0,3108	0,3079
	Czuł.	0,3445	0,3475	0,4559	0,4215	0,4559	0,4619	0,6516	0,6413
Tst.:63899;446;0,70%	F1	0,3445	0,3475	0,4559	0,4215	0,4559	0,4619	0,4208	0,4160
	FP	0,0046	0,0046	0,0038	0,0041	0,0038	0,0038	0,0102	0,0101
T:212994;1488;0,70%	FN	0,6555	0,6525	0,5441	0,5785	0,5441	0,5381	0,3484	0,3587
	Czas	15,5888	6,3271	872,181	299,286	161,098	65,4304	1034,10	507,76

Tabela 9: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„KDD CUP 1999 (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
KDD CUP 1999 (R)	Bł. kl.	0,0027	0,0030	0,0027	0,0030	0,0028	0,0030	0,0045	0,0055
Cechy: 41 (w tym kateg.)	Spec.	0,9986	0,9985	0,9986	0,9985	0,9986	0,9985	0,9968	0,9960
Szk.:68303;209;0,31%	Prec.	0,5598	0,5111	0,5598	0,5111	0,5502	0,5111	0,3563	0,2822
	Czuł.	0,5598	0,5111	0,5598	0,5111	0,5502	0,5111	0,5694	0,5111
Tst.:29274;90;0,31%	F1	0,5598	0,5111	0,5598	0,5111	0,5502	0,5111	0,4383	0,3636
	FP	0,0014	0,0015	0,0014	0,0015	0,0014	0,0015	0,0032	0,0040
T:97577;299;0,31%	FN	0,4402	0,4889	0,4402	0,4889	0,4498	0,4889	0,4306	0,4889
	Czas	0,1473	0,0766	180,164	92,4985	0,5658	0,1950	233,751	88,7911

Tabela 10: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„w7a libsvm nonspare (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
w7a libsvm nonspare (R)	Bł. kl.	0,0132	0,0136	0,0147	0,0123	0,0058	0,0059	0,0181	0,0164
Cechy: 300 (w tym kateg.)	Spec.	0,9933	0,9932	0,9926	0,9938	0,9971	0,9970	0,9845	0,9863
Szk.:34073;284;0,83%	Prec.	0,2077	0,1885	0,1162	0,2623	0,6514	0,6475	0,2688	0,2893
	Czuł.	0,2077	0,1885	0,1162	0,2623	0,6514	0,6475	0,6796	0,6639
Tst.:14603;122;0,84%	F1	0,2077	0,1885	0,1162	0,2623	0,6514	0,6475	0,3852	0,4030
	FP	0,0067	0,0068	0,0074	0,0062	0,0029	0,0030	0,0155	0,0137
T:48676;406;0,83%	FN	0,7923	0,8115	0,8838	0,7377	0,3486	0,3525	0,3204	0,3361
	Czas	0,3405	0,1697	313,41	139,77	32,183	13,3827	300,40	154,20

Tabela 11: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Wine quality white (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Wine quality white (R)	Bł. kl.	0,0076	0,0082	0,0076	0,0095	0,0076	0,0095	0,0114	0,0143
Cechy: 11 (numeryczne)	Spec.	0,9962	0,9959	0,9962	0,9952	0,9962	0,9952	0,9915	0,9897
Szk.:3428;17;0,50%	Prec.	0,2353	0,2500	0,2353	0,1250	0,2353	0,1250	0,1944	0,1176
	Czuł.	0,2353	0,2500	0,2353	0,1250	0,2353	0,1250	0,4118	0,2500
Tst.:1470;8;0,54%	F1	0,2353	0,2500	0,2353	0,1250	0,2353	0,1250	0,2642	0,1600
	FP	0,0038	0,0041	0,0038	0,0048	0,0038	0,0048	0,0085	0,0103
T:4898;25;0,51%	FN	0,7647	0,7500	0,7647	0,8750	0,7647	0,8750	0,5882	0,7500
	Czas	1,1555	1,0465	22,2972	9,5659	0,4058	0,1487	22,7037	18,7584

Tabela 12: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Adult (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Adult (R)	Bł. kl.	0,0097	0,0102	0,0092	0,0077	0,0093	0,0096	0,0157	0,0143
Cechy: 14 (w tym kateg.)	Spec.	0,9951	0,9949	0,9954	0,9961	0,9953	0,9952	0,9880	0,9890
Szk.:26159;151;0,58%	Prec.	0,1589	0,1094	0,2053	0,3281	0,1987	0,1562	0,1425	0,1800
	Czuł.	0,1589	0,1094	0,2053	0,3281	0,1987	0,1562	0,3444	0,4219
Tst.:11211;64;0,57%	F1	0,1589	0,1094	0,2053	0,3281	0,1987	0,1562	0,2016	0,2523
	FP	0,0049	0,0051	0,0046	0,0039	0,0047	0,0048	0,0120	0,0110
T:37370;215;0,58%	FN	0,8411	0,8906	0,7947	0,6719	0,8013	0,8438	0,6556	0,5781
	Czas	0,6312	0,3663	668,2978	280,8968	1,0244	0,4225	586,4106	304,0338

Tabela 13: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Car Evaluation (R)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Car Evaluation (R)	Bł. kl.	0,0496	0,0578	0,0397	0,0655	0,0662	0,0694	0,0885	0,1060
Cechy: 6 (w tym kateg.)	Spec.	0,9742	0,9699	0,9794	0,9659	0,9656	0,9639	0,9210	0,9098
Szk.:1209;45;3,72%	Prec.	0,3333	0,2500	0,4667	0,1500	0,1111	0,1000	0,2459	0,1818
	Czuł.	0,3333	0,2500	0,4667	0,1500	0,1111	0,1000	0,6667	0,5000
Tst.:519;20;3,85%	F1	0,3333	0,2500	0,4667	0,1500	0,1111	0,1000	0,3593	0,2667
	FP	0,0258	0,0301	0,0206	0,0341	0,0344	0,0361	0,0790	0,0902
T:1728;65;3,76%	FN	0,6667	0,7500	0,5333	0,8500	0,8889	0,9000	0,3333	0,5000
	Czas	0,0063	0,0084	1,4988	0,1090	0,0647	0,0045	1,3496	0,2056

Tabela 14: Wyniki analizy detekcji anomalii w systemie Trinity SALT dla *„Citibike Synthetic (S)”*. Wartości w kolumnach reprezentują metryki dla części szkoleniowej (Szk.) i testowej (Tst.). Źródło: opracowanie własne.

Opis	Metr.	SOM		AE		LOF		SALT	
		Szk.	Tst.	Szk.	Tst.	Szk.	Tst.	Szk.	Tst.
Citibike Synthetic (S)	Bł. kl.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Cechy: 11 (w tym kateg.)	Spec.	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
Szk.:14000;140;1,00%	Prec.	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Czuł.	1,0000	1,0000	1,0000	1,0000	1,0000	1,000	1,0000	1,0000
Tst.:6000;60;1,00%	F1	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	FP	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
T:20000;200;1,00%	FN	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Czas	0,2216	0,1008	194,36	102,36	3,9371	1,5750	206,01	94,3397

Dodatek C. Optymalne hiperparametry dla wszystkich zbiorów danych

Tabela 15: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Bank mktg (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Bank mktg real	20	30	0,5597	5	hyper_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	hamming	gaussian	600	18

Tabela 16: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Bank mktg (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Bank mktg real	50	0,0001	2048	L2	0,9831	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
1	relu	sigmoid	MSE	39

Tabela 17: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Bank mktg (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Bank mktg real	262	42	hamming

Tabela 18: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Chess krkopt (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Chess krkopt zerovsall	30	17	0,000186	5	exponen_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	gaussian	536	0

Tabela 19: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Chess krkopt (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Chess krkopt zerovsall	74	0,09689	3839	brak	0,07976	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
2	relu	sigmoid	MSE	6

Tabela 20: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Chess krkopt (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Chess krkopt zerovsall	9628	4	cosine

Tabela 21: Optymalne hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym: *„Covid19 (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Covid19 case real	28	11	0,004952	15	declin_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	mexican hat	361	20

Tabela 22: Optymalne hiperparametry i wartości metryk dla modelu AE na zbiorach szkoleniowym i testowym: *„Covid19 (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Covid19 case real	246	0,0662	667	L1	0,5216	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
1	relu	sigmoid	MSE	23

Tabela 23: Optymalne hiperparametry i wartości metryk dla modelu LOF na zbiorach szkoleniowym i testowym: *„Covid19 (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Covid19 case real	61	33	euclidean

Tabela 24: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Mushroom (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Mushroom real	12	18	0,4502	2	exponen_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	cosine	bubble	482	23

Tabela 25: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Mushroom (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Mushroom real	300	0,01107	2219	brak	0,6676	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
35	relu	sigmoid	MSE	0

Tabela 26: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Mushroom (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Mushroom real	2975	29	hamming

Tabela 27: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Breast cancer (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Breast cancer real	39	45	0,7149	3	exponen_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	cut gaussian	266	28

Tabela 28: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Breast cancer (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Breast cancer real	504	0,0137	2945	brak	0,0883	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
41	relu	sigmoid	MSE	72

Tabela 29: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Breast cancer (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Breast cancer real	76	2	euclidean

Tabela 30: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Credit card (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Credit card real	2	19	1,4656	10	inverse_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	euclidean	gaussian	335	33

Tabela 31: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Credit card (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Credit card real	279	0,0651	1746	L1	0,5913	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
46	relu	sigmoid	MSE	45

Tabela 32: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Credit card (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Credit card real	4400	17	euclidean

Tabela 33: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Thyroid disease (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Thyroid disease real	27	23	0,0077	5	hyper_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	gaussian	456	42

Tabela 34: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Thyroid disease (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Thyroid disease real	489	0,0036	2373	L2	0,1713	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
100	relu	sigmoid	MSE	77

Tabela 35: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Thyroid disease (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Thyroid disease real	4748	4	cosine

Tabela 36: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Vehicle claims (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
Vehicle claims real	40	31	0,0288	8	declin_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	bubble	700	120

Tabela 37: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Vehicle claims (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
Vehicle claims real	954	0,0155	2505	L2	0,1725	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
60	relu	sigmoid	MSE	38

Tabela 38: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Vehicle claims (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
Vehicle claims real	5725	31	euclidean

Tabela 39: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„KDD CUP 1999 (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
KDD CUP 1999 real	2	37	0,3366	3	power_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	hamming	triangle	370	18

Tabela 40: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„KDD CUP 1999 (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
KDD CUP 1999 real	892	0,0873	1440	L2	0,0334	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
34	relu	sigmoid	MSE	65

Tabela 41: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„KDD CUP 1999 (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
KDD CUP 1999 real	57	54	hamming

Tabela 42: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„w7a libsvm nonspare (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
w7a libsvm nonspare real	1	62	1,3522	6	invers_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	mexican hat	408	101

Tabela 43: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„w7a libsvm nonspare (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
w7a libsvm nonspare real	808	0,0001	1486	L1	0,01	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed	Konfig.
100	relu	sigmoid	MSE	46	konfig. 1

Tabela 44: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„w7a libsvm nonspare (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
w7a libsvm nonspare real	3865	68	cosine

Tabela 45: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Wine quality white (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
wine quality white real	70	70	1,1000	3	exponen_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
hexagon.	cosine	mexican hat	3008	67

Tabela 46: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Wine quality white (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
wine quality white real	1644	0,0232	713	L1	0,21	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
75	relu	sigmoid	MSE	0

Tabela 47: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Wine quality white (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
wine quality white real	1768	45	cosine

Tabela 48: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Adult (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
adult real	2	199	0,5278	7	hyperbol._decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	hamming	gaussian	1167	43

Tabela 49: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Adult (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
adult real	3294	0,0847	306	L1	0,1604	ADAM

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
112	relu	sigmoid	MSE	81

Tabela 50: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Adult (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
adult real	486	43	hamming

Tabela 51: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Car evaluation (R)”*.

Zbiór	x	y	Współ. ucz.	Prom.	Fun. Zaniku
car evaluation real	1	8	0,2070	9	inverse_decay

Topol.	Metr.	Fun. Sąs.	L. Iter.	Seed
rectang.	hamming	bubble	256	64

Tabela 52: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Car evaluation (R)”*.

Zbiór	Epok.	Współ. ucz.	Roz. partii	Regul.	Lambda	Optym.
car evaluation real	386	0,9859	3414	L2	0,1120	SGD

Neurony (w. ukryte)	f. aktyw. wej.	f. aktyw. wyj.	f. straty	seed
16	relu	sigmoid	MSE	82

Tabela 53: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Car evaluation (R)”*.

Zbiór	Rozm. bloku	MinPts	Metr.
car evaluation real	21	3	euclidean

Tabela 54: Hiperparametry i wartości metryk dla modelu SOM na zbiorach: szkoleniowym i testowym *„Citibike synthetic (S)”*.

Zbiór	**x**	**y**	**Współ. ucz.**	**Prom.**	**Fun. Zaniku**
citibike synthetic	1	8	0,0497	6	declining_decay

Topol.	**Metr.**	**Fun. Sąs.**	**L. Iter.**	**Seed**
rectang.	euclidean	bubble	261	82

Tabela 55: Hiperparametry i wartości metryk dla modelu AE na zbiorach: szkoleniowym i testowym *„Citibike synthetic (S)”*.

Zbiór	**Epok.**	**Współ. ucz.**	**Roz. partii**	**Regul.**	**Lambda**	**Optym.**
citibike synthetic	300	0,001	808	brak	0,1591	SGD

Neurony (w. ukryte)	**f. aktyw. wej.**	**f. aktyw. wyj.**	**f. straty**	**seed**
50	relu	sigmoid	MSE	100

Tabela 56: Hiperparametry i wartości metryk dla modelu LOF na zbiorach: szkoleniowym i testowym *„Citibike synthetic (S)”*.

Zbiór	**Rozm. bloku**	**MinPts**	**Metr.**
citibike synthetic	281	29	euclidean