POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Technologie informacyjne w systemach automatyki (ART)

PRACA DYPLOMOWA INŻYNIERSKA

Uniwersalny system do wykrywania anomalii w danych wybranego typu

System for general task of anomaly detection in data of a chosen type

AUTOR:
Daniel Jabłoński

PROWADZĄCY PRACĘ:
dr Marek Bazan
Katedra Informatyki Technicznej

WROCŁAW 2021

Spis treści

Rozdział 1

Wstęp

Abstrakt Rozdział wprowadza pojęcie anomalii oraz obserwacji odstająca. Przedstawia rosnące znaczenie detekcji anomalii w zbiorach danych oraz ich praktyczne zastosowanie. Argumentuje przyjęte w pracy podejście do wykrywania anomalii. Prezentuje cel oraz strukturę niniejszej pracy.

1.1 Wprowadzenie

Yuval Noah Harari w swojej książce *„21 lekcji dla XXI wieku"* zwiastuje potencjał *Big data*[1] w XXI wieku jako dobra, której wartość swoim potencjałem przyćmi tradycyjne dobra takie jak posiadanie ziemi, fabryk czy maszyn. „Harvard Business Review" określa pozycję *Data Scientist*[2] jako najbardziej pożądanej w XXI wieku [12].

Znaczenie eksploracji danych zwiększa się każdego dnia. Ważną dziedziną tego procesu jest zadanie wykrywania anomalii. Początki badań nad metodą wykrywania anomalii w danych można znaleźć już w XIX w. [14]. Wraz z rozwojem techniki komputerowej oraz zwiększoną ilością dostępnych danych, skuteczne metody wykrywania anomalii stały się wysoce pożądane. Detekcja anomalii w zbiorach danych przekłada się na uzyskaniu cennych informacji w wielu dziedzinach takich jak:

- W diagnostyce medycznej anomalny obraz z rezonansu magnetycznego może wskazać obecność nowotworu [30]
- W detekcji oszustw finansowych takich jak kradzież karty kredytowej [8]
- W przemyśle do detekcji awarii sensorów [13]
- W zabezpieczeniach sieciowych do wykrywania włamań do sieci [16]

Wykrywanie anomalii odnosi się do problemu znajdowania obserwacji, które odbiegają zachowaniem od normy. Słownik języka polskiego definiuje anomalię jako: „odchylenie od normy" [2]. W literaturze można spotkać się z określeniami takich punktów jako: nieprawidłowość, obserwacja odstająca, niezgodność, dewiacja [4]. Z czego najczęstszym określeniem w danym kontekście jest anomalia oraz obserwacja odstająca, które zazwyczaj używane są zamiennie [11]. Związane jest to z podejściem zakładającym, że problem

[1] Gromadzenie i przetwarzanie dużych zbiorów danych w celu uzyskania wartościowych informacji.
[2] Specjalista analizujący dane w celu uzyskania pożądanych i wartościowych informacji.

wykrywania anomalii jest uczeniem nienadzorowanym. Gdzie do detekcji anomalii bazujemy na metodach statystycznych w celu znalezieniu obserwacji odstających z założeniem, że owe punkty będą anomaliami. Jest to dominujące założenie w literaturze [15]. Drugim podejściem jest modelowanie osobno normalnych i anomalnych punktów. Jednakże to podejście wymaga znajomości procesu powstania obu typów punktów w zbiorze danych lub wystarczającą liczbę oznaczonych danych treningowych. Co w zadaniach detekcji anomalii gdzie operujemy na nieoznaczonych danych, jak również nie znamy procesu generującego anomalne punkty, jest podejściem nieprzydatnym. W związku z tym w niniejszej pracy do detekcji punktów anomalnych wykorzystano podejście bazujące na detekcji obserwacji odstających.

1.2 Cel pracy

Celem pracy jest opracowanie oraz implementacja uniwersalnego systemu internetowego służącego do detekcji anomalii dla dowolnych zbiorów danych obserwacji statystycznych (bez etykiet). System ma pozwalać na przesłanie danych w popularnych formatach CSV oraz JSON. Przetworzeniu uzyskanych danych m.in. oczyszczeniu danych z brakujących wartości oraz wyskalowaniu danych. Stworzeniu raportu z działania systemu oraz zapewnieniu użytkownikowi oczyszczonych danych z anomalii w formacie, w którym te dane zostały przesłane, jak również danych z wartością anomalności obserwacji – jak bardzo dana obserwacja odstaje od reszty zbioru. System powinien być jak najprostszy w obsłudze, szybki w działaniu oraz zapewniać kluczowe informacje pozwalającemu użytkownikowi na dalszą analizę uzyskanych wyników. System ma zapewnić użytkownikowi, nieposiadającemu wiedzy na temat wykrywania anomalii, automatyzację procesu wyboru algorytmu detekcji oraz jego parametrów w celu zwiększenia skuteczności detekcji.

1.3 Struktura pracy

Praca składa się sześciu rozdziałów. Rozdział pierwszy wprowadza czytelnika w tematykę pracy. Rozdział drugi przedstawia teorię detekcje anomalii wraz z różnicami w podejściu i problematyce tematu. Rozdział trzeci prezentuje zastosowane podejście doboru modelu oraz opisuje algorytmy wraz z parametrami (modele) składające się na przestrzeń bazową modeli. Rozdział czwarty dokonuje omówienia stworzonego systemu wykrywania anomalii. Rozdział piąty analizuje skuteczność działania systemu. Ostatni rozdział podsumowuje całość pracy.

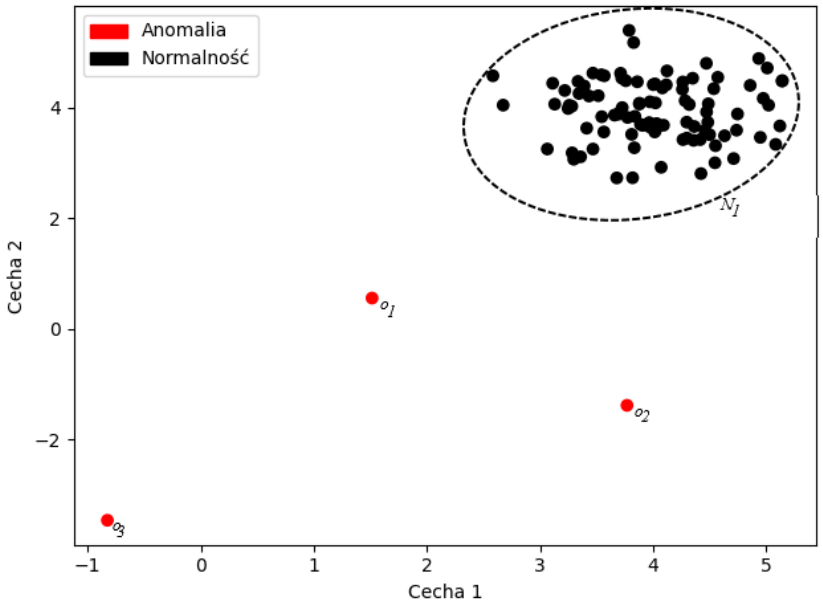
Rozdział 2

Wykrywanie anomalii

Abstrakt W rozdziale przybliżono teorię, problematykę oraz różnice podejść w detekcji anomalii (obserwacji odstających).

2.1 Definicja anomalii (obserwacji odstającej)

Obserwacje odstające to punkty, które nie odpowiadają wzorcowi w zbiorze danych. Barnett i Lewis definiują obserwację odstającą następująco: „Obserwacja odstająca jest to obserwacja, której obecność jest różna od pozostałych obserwacji" [7]. Rysunek 2.1 obrazuje przykład obserwacji odstających (anomalii) dla dwuwymiarowego zbioru danych. Zbiór danych posiada obszar N_1, większość punktów znajduje się wewnątrz tego obszaru. Punkty wystarczająco oddalone od obszaru N_1: o_1, o_2, o_3 – sklasyfikowane są jako obserwacje odstające (anomalie).



Rysunek 2.1 Prosty przykład anomalii w dwuwymiarowym zbiorze danych.
źródło: Opracowanie własne

2.2 Zadanie detekcji anomalii

W pracy rozważamy nienadzorowane zadanie detekcji anomalii na zbiorze N punktów $x_1, ..., x_N$ każdy punkt jest d-wymiarowym wektorem liczb rzeczywistych. Zbiór danych składa się z obserwacji poprawnych i anomalnych, jednakże zbiór nie posiada wektora $y_1, .., y_N$ – klasyfikującego obserwację do obserwacji poprawnych lub anomalnych. Celem zadania detekcji anomalii jest wykrycie punktów anomalnych w danym zbiorze.

2.3 Różnice w podejściu detekcji anomalii

2.3.1 Wykrywanie anomalii lokalnych a globalnych

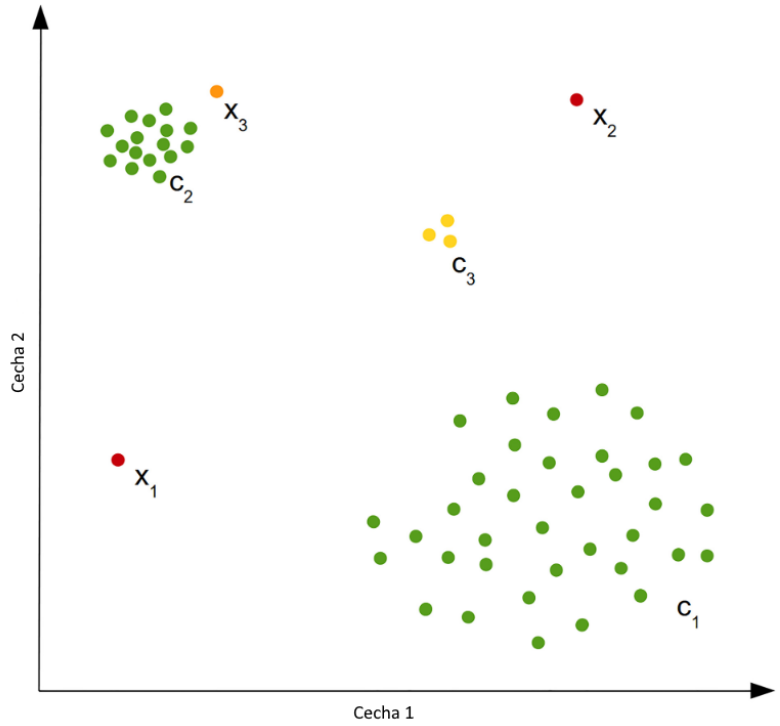
Podejście dotyczy wyboru zakresu zbioru jako zbioru odniesienia dla rozpatrywania odstawania danej obserwacji. Główne podejścia to podejście globalne oraz lokalne. Rysunek 2.2 obrazuje prosty dwuwymiarowy zbiór danych z dwoma klastrami (skupieniami) poprawnych obserwacji: c_1 i c_2. Dla punktów x_1 oraz x_2 klasyfikacja jako anomalie jest możliwa wizualnie. Oba punkty są znacząco oddalone od gęstych obszarów obserwacji. Są to anomalie globalne. Rozpatrując wszystkie obserwacje zbioru punkt x_3 mógłby być uznany za poprawną obserwację należącą do klastra c_2, jednak kiedy rozpatrzymy podzbiór punktów w sąsiedztwie klastra c_2 (zbiór odniesienia), punkt x_3 może być uznany za anomalię. Jest to przykład anomalii lokalnej. Zatem anomalia lokalna jest to obserwacja, której odchylenie od normy (anomalność) rozpatrywane jest w obszarze najbliższego sąsiedztwa (zbioru odniesienia).

- Podejście globalne
  - Zbiór odniesienia obejmuje wszystkie obserwacje.
  - Założenie o istnieniu jeden prawidłowego mechanizmu generującego normalne punkty.
  - Problem: występowanie anomalii lokalnych lub więcej niż jeden prawidłowy mechanizm generujący punkty, może wypaczyć wynik detekcji.
- Podejście lokalne
  - Zbiór odniesienia jest podzbiorem całego zbioru.
  - Brak założenia o liczbie mechanizmów generujących.
  - Problem: wybór odpowiedniego podzbioru jako zbioru odniesienia.

2.3.2 Wynik metody

Ważnym aspektem dla zadania wykrywania anomalii jest sposób oceny każdej obserwacji. Dane wyjściowe metody mogą przypisywać obserwacji wartość wskaźnika mierzącego anomalność na dwa warianty:

- binarny: przypisanie obserwacji etykiety. Klasyfikacja obserwacji jako anomalii lub poprawnej obserwacji w zbiorze
- ciągły: dla każdej obserwacji obliczana jest wartość anomalności np. prawdopodobieństwo obserwacji jako anomalii



Rysunek 2.2 Przykład anomalii globalnych (x_1, x_2), lokalnej x_3 oraz mikro klastra c_3.
źródło: Opracowanie własne na podstawie [19]

Podejście przypisujące wartość wskaźnika mierzącego anomalność w sposób ciągły (wartość anomalności) jest podejściem wszechstronnym. Wykorzystując podejście osoba analizująca dane może za pomocą np. progu wartości anomalności, otrzymać binarną etykietę obserwacji, dostosowując próg dla danej dziedziny. Wiele podejść opartych na przypisywaniu wartości anomalności skupia się na wyznaczeniu grupy n-obserwacji o najwyższym wyniku (parametr n często podawany jest przez użytkownika np. procent kontaminacji zbioru danych). Podejście przypisujące wartość anomalności przydatne jest w rozważaniu mikro klastrów, czyli małych regularnych klastrów. Rysunek 2.2 obrazuje mikro klaster c_3. Algorytm detekcji anomalii powinien przypisać punktom klastra wartość anomalności wyższą od normalnych obserwacji oraz mniejszą od anomalii globalnych. Przykład pokazuje, że podejście przypisywania ciągłej wartości anomalności punktu jest podejściem bardziej użytecznym od binarnej klasyfikacji zwłaszcza w dalszej analizie danych.

2.3.3 Rodzaje anomalii

W detekcji anomalii wyróżnia się trzy rodzaje anomalii: anomalie punktowe, anomalie zgrupowane oraz anomalie kontekstowe. Większość dostępnych algorytmów detekcji anomalii dla uczenia nienadzorowanego opiera się na wykrywaniu anomalii punktowych. Detekcja anomalii zgrupowanych została przedstawiona na rysunku 2.2 i opisana w sekcji 2.3.2 – mikro klaster c_3. Anomalie kontekstowe są to anomalie najczęściej spotykane w szeregach czasowych. Dotyczy to punktów, które mogą być uznane za poprawne obserwacje, jednak rozpatrywane w kontekście charakteru zbioru danych zostaną uznane za anomalie. Weźmy przykład pomiaru temperatury w skali roku we Wrocławiu. Temperatura 25°C wydaje się poprawnym odczytem temperatury, jednak w kontekście miesiąca np. stycznia. Tak wysoka temperatura jest anomalią.

Szczęśliwie można wykorzystać algorytmy detekcji anomalii punktowych do wykrywa-

nia anomalii zgrupowanych i kontekstowych. Detekcja różnych typów anomalii wymaga obróbki danych m.in. przekształcenia zbioru danych reprezentującego cechy z wykorzystaniem korelacji, agregacji oraz grupowania [18] w celu analizy obserwacji jako anomalii punktowych.

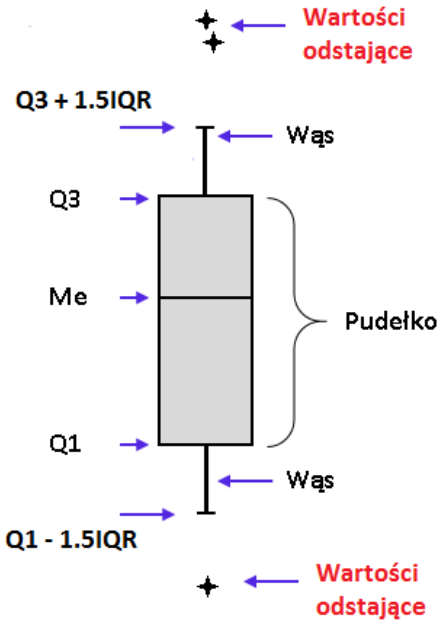
W pracy skupiono się na problemach detekcji anomalii punktowych, wykorzystując bazę zbiorów danych ODDS [28]. Baza zawiera zbiory danych już przygotowanych do zadania detekcji anomalii punktowych (Dodatek A).

2.4 Główne metodyki detekcji anomalii

Dobór metody do detekcji anomalii jest złożonym zadaniem. Detektor anomalii powinien jak najbardziej separować obserwacje odstające od reszty obserwacji. Zdefiniowanie normalnego obszaru zwierającego wszystkie poprawne obserwacje jest ciężkie do osiągnięcia dla rzeczywistych zbiorów danych. Dodatkowo wybrany model powinien zapewnić interpretację wartości anomalności – dlaczego obserwacja uważana jest za anomalię. Co pozwala na dalszą analizę obserwacji w celu uzyskania informacji na temat jej powstania anomalii oraz znaczenia dla danego zbioru danych.

2.4.1 Metody oparte na wiedzy statystycznej

Pierwsze metody detekcji anomalii wywodziły się z analizy statystycznej danych. Popularną metodą jest detekcja anomalii z wykorzystaniem rozkład cechy statystycznej. Analizując rozkład cechy statystycznej, wartości znacząco odstające od średniej wartości (regułą trzech sigm) są wartościami odstającymi (anomaliami). Rysunek 2.4 przedstawia wykres pudełkowy – sposób wizualizacji rozkładu cechy – dla którego wartości poza dolnym i górnym wąsem są wartościami odstającymi.



Rysunek 2.3 Wykres pudełkowy z wartościami odstającymi
źródło: [3]

2.4.2 Metody oparte na sąsiedztwie obserwacji

Metody do wyznaczenia wartości anomalności obserwacji rozpatrują sąsiedztwo (zbiór odniesienia) analizowanej obserwacji. Najbardziej popularne podejścia obliczenia anomalności obserwacji względem sąsiedztwa to:

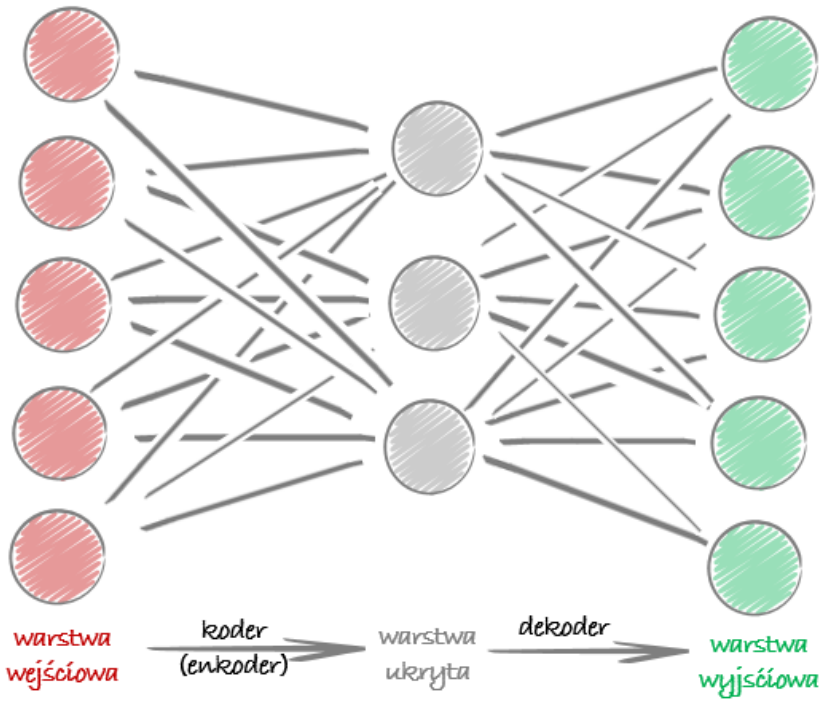
- Wykorzystujące klasteryzację: metoda klasteryzuje dane wejściowe na większe i mniejsze skupienia. Wartość anomalności wyznaczana jest z wykorzystaniem odległości obserwacji do najbliższego dużego skupienia oraz rozmiar klastra, do którego przypisano obserwację.
- Wykorzystujące metrykę: wartość anomalności obliczana jest jako długość między obserwacją a k najbliższymi sąsiadami. (Przykład opisany w podsekcji 3.3.2)
- Wykorzystujące gęstość: wartość anomalności obliczana jest jako stosunek lokalnej gęstości obserwacji do średniej lokalnej gęstości k sąsiadów. (Przykład opisany w podsekcji 3.3.3)
- Wykorzystujące miarę kąta: wartość anomalności obliczana jest na podstawie miary kąta między obserwacją a k sąsiadami. Dla wielowymiarowych zbiorów danych miara kąta jest stabilniejszą metryką. (Przykład opisany w podsekcji 3.3.5)

2.4.3 Metody oparte na łączeniu klasyfikatorów (zespół klasyfikatorów)

Wykorzystują podejście budowania silnego klasyfikatora składającego się ze słabszych klasyfikatorów. Główne algorytmy korzystające z podejścia to: *Isolation Forest* oraz *Lightweight Online Detector of Anomalies* (LODA). *Isolation Forest* wykorzystuje drzewa decyzyjne do estymacji anomalności obserwacji, natomiast LODA wykorzystuje do tego celu histogramy (algorytmu opisane zostały w podsekcji 3.3.8 oraz 3.3.7)

2.4.4 Metody oparte na sieciach neuronowych

Podejście wykorzystuje sieci neuronowe do detekcji anomalii. Popularnym rodzajem wykorzystywanych sieci neuronowych są autoenkodery. Jest to jednokierunkowa sieć neuronowe zdolna do rekonstrukcji sygnału wejściowego. Autoenkodery składają się z dwóch elementów: kodera oraz dekodera. Koder przekształca dane wejściowe do niższej wymiarowości, usuwając zbędne informacje, następnie dekoder rekonstruuje dane do wejściowej wymiarowości. Funkcja straty autoenkodera jest różnicą między danymi wejściowymi a rekonstrukcją danych. Im wyższa funkcja straty tym wyższa anomalność obserwacji.



Rysunek 2.4 Budowa autoenkodera
źródło: [24]

Rozdział 3

Proponowana metoda detekcji anomalii

Abstrakt W rozdziale przedstawiono proponowane rozwiązanie doboru modelu detekcji anomalii pozwalające na automatyzację tego procesu. Rozdział zawiera również opis algorytmów wraz z parametrami, składających się na przestrzeń bazową modeli.

3.1 Rozpatrywane podejścia

Mając na uwadze problematykę detekcji anomalii poruszoną w rozdziale 2 na fazie projekcyjnej rozpatrywano początkowo wybór jednego algorytmu o jak najlepszej skuteczności detekcji dla jak największej ilości zróżnicowanych zbiorów danych.

Początkowy wyborem był algorytm *Isolation Forest* [23]. Algorytm pozytywnie oceniany był w badaniach porównawczych [15, 6]. Jednakże obecnie żaden algorytm konsekwentnie przewyższa – skutecznością detekcji anomalii – inne porównywane algorytmy [6]. Dla przykładu w bardzo skupionych anomaliach algorytm *Angle-based Outlier Detection* (ABOD) [22] oraz *Local Outlier Factero* (LOF) [9] sprawowały się lepiej [15].

Następnie zamiast wyboru jednego algorytmu, rozpatrywano podejście zakładające wybór jak najlepszego algorytmu z listy algorytmów oraz dobór jego parametrów w celu usprawnienia detekcji anomalii dla zróżnicowanych właściwości zbiorów danych. W celu automatyzacji doboru optymalnego algorytmu oraz jego parametrów zdecydowano na wykorzystanie biblioteki *Automating Outlier Detection via Meta-Learning* (MetaOD) [35].

3.2 Proponowane rozwiązanie: MetaOD

MetaOD wykorzystuje meta-uczenie do systematycznego i kryterialnego podejścia automatyzacji wyboru optymalnego algorytmu wraz z jego parametrami (wybór modelu) w celu detekcji anomalii. Meta-uczenie, czyli uczenie jak się uczyć, jest techniką, która na podstawie wcześniejszych doświadczeń ułatwia efektywne uczenie dla nowego zadania (zbioru danych). Podejście jest skuteczne dla automatyzacji uczenia maszynowego [32]. MetaOD rozpatruje problem wyboru modelu (algorytm wraz z parametrami) z przestrzeni bazowej modeli (Tabela 3.1), która składa się z następujących algorytmów (opisanych w sekcji 3.3):

- *Angle-based Outlier Detection* (ABOD)
- *Connectivity based Outlier Factor* (COF)

- *Histogram-based Outlier Score* (HBOS)
- *Isolation Forest* (IForest)
- *k-Nearest Neighbors* (kNN)
- *Lightweight Online Detector of Anomalies* (LODA)
- *Local Outlier Factor* (LOF)
- *One-Class SVM* (OCSVM)

MetaOD rozważa zadanie wyboru modelu w celu detekcji anomalii analogicznie do problemu generacji rekomendacji (Collaborative Filtering) dla nowego użytkownika – brak ewaluacji (problem "zimnego rozruchu"). Uwzględniając różnice między problemami m.in. brak sprzężenia zwrotnego do poprawy rekomendacji oraz wykorzystanie tylko jednego najlepszego modelu dla problemu detekcji anomalii zamiast Top-N rekomendacji.

MetaOD generuje na podstawie zbioru danych meta-dane (200 meta-cech), które reprezentują charakterystykę oraz właściwości zbioru danych m.in. miarę rozkładu. Generacja meta-cech przeprowadzany jest offline, natomiast wybór modelu z przestrzeni bazowej dokonywany jest online. MetaOD zwraca uporządkowaną listę modeli, od najlepszego do najgorszego, na podstawie zgromadzonych w bazie wyników skuteczności detekcji anomalii na zbiorach danych uznanych – na podstawie meta-danych – za podobne, z wykorzystaniem przestrzeni bazowej modeli.

W analizie skuteczności przeprowadzonych przez autorów: „MetaOD znacząco poprawił skuteczność detekcji w porównaniu z popularnymi metodami."[35].

Algorytm	Parametr 1	Parametr 2
ABOD	liczba sąsiadów	nie dotyczy
COF	liczba sąsiadów	nie dotyczy
HBOS	liczba prostokątów histogramu	tolerancja
IForest	liczba estymatorów	procent rozpatrywanych cech
kNN	liczba sąsiadów	metoda
LODA	liczba prostokątów histogramu	liczba losowych cięć
LOF	liczba sąsiadów	metryka
OCSVM	współczynnik ν	funkcja jądra

Tabela 3.1 Przestrzeń bazowa modeli (algorytm i parametry) MetaOD
źródło: Opracowanie własne na podstawie [35]

3.3 Wykorzystywane algorytmy

W sekcji zostanie przybliżone działanie algorytmów (Rysunek 3.1) z uwzględnieniem znaczenia parametrów dobranych przez MetaOD. Wszystkie algorytmy implementowane są z wykorzystaniem biblioteki *Python Outlier Detection* (PyOD) [34]

Rysunek 3.1 Wykorzystywane algorytmy z podziałem na metodologię.
źródło: Opracowanie własne

3.3.1 One-Class SVM

One-Class SVM [Jednoklasowa maszyna wektorów nośnych] (OCSVM) [26]. OCSVM jest binarną metodą klasyfikacji odwzorowującą dane wejściowe z wykorzystaniem funkcji jądrowej na wielowymiarową w poszukiwaniu hiperpłaszczyzny separującej poprawne obserwacje od anomalii. Funkcje jądrowe:

$$Liniowa : x^T x_i$$
$$Wielomianowa : (\gamma x^T x_i + c)^n$$
$$RBF : exp(-\gamma||x - x_i||^2)$$
$$Sigmoidalna : tgh(\gamma x^T x_i + c)$$

Parametr "nu" jest górną granicą akceptowalnego błędu klasyfikacji oraz dolną granicą liczby wektorów nośnych do liczby analizowanych obserwacji. Wykorzystany w celu uniknięcia przeuczenia klasyfikatora.

3.3.2 k-Nearest Neighbors

k-Nearest Neighbors[Algorytm k najbliższych sąsiadów] (kNN) [27] opiera się na wnioskowaniu, że poprawne obserwacje będą w sąsiedztwie innych poprawnych obserwacji. Natomiast anomalie będą znacząco oddalone od skupień normalnych obserwacji. Dla każdego elementu obliczamy metrykę do k = 1,...,N sąsiada. Wynik anomalności elementu zależy od metody (wybrana przez MetaOD):

- *largest* - anomalność punktu jest odległością do k-sąsiada (najdalej oddalonego)
- *mean* - anomalność punktu jest średnią ze wszystkich odległości do sąsiadów
- *median* - anomalność punktu jest medianą wszystkich odległości do sąsiadów

3.3.3 Local Outlier Factor

Local Outlier Factor (LOF) [9] algorytm w odróżnieniu od algorytmu kNN rozpatruje lokalną gęstość względem sąsiadów. Podejście odnosi się do wad metod opartych na metryce w przypadkach zbiorów danych o różnych gęstościach klastrów poprawnych obserwacji (Rysunek 2.2). W celu obliczenia anomalności punktu należy dokonać 3 kroków:

1. Dla rozpatrywanego punktu x, należy znaleźć k najbliższych sąsiadów
2. Rozpatrując k najbliższych sąsiadów N_k, wyznaczyć zagęszczenie w k-sąsiedztwie punktu x obliczając *local reachability density*(LRD):

$$LRD_k(x) = 1/\left(\frac{\sum_{o \in N_k(x)} d_k(x,o)}{|N_k(x)|}\right) \tag{3.1}$$

   gdzie $d_k(\cdot)$ oznacza *reachability distance*
3. Ostatecznie obliczamy wartość LOF, porównując LRD punktu x z LRD k-sąsiada:

$$LOF(x) = \frac{\sum_{o \in N_k(x)} \frac{LRD_k(o)}{LRD_k(x)}}{|N_k(x)|} \tag{3.2}$$

Wynik LOF jest stosunkiem LRD punku x do średniej LRD dla k sąsiadów. Jeżeli wartość $LOF(x) > 1$ oznacza to małą lokalną gęstość (anomalia lokalna). $LOF(x) \approx 1$ oznacza, że lokalna gęstość punktu x jest zbliżona do sąsiadów.

3.3.4 Connectivity based Outlier Factor

Connectivity based Outlier Factor(COF) [31] algorytm COF jest zbliżony działaniem do LOF, główną różnicą wybór k najbliższych sąsiadów (Rysunek 3.3). LOF wybiera k najbliższych sąsiadów, wykorzystując metrykę np. odległość euklidesową, tworząc wokół analizowanego punktu sferę. COF tworzy zbiór k-sąsiadów dla punktu x, wybierając najbliższego sąsiada, którego dodaje do zbioru. Następnie wybiera punkt najbliższy do dowolnego elementu istniejącego zbioru k-sąsiadów i dodaje go do zbioru. Proces przebiega do momentu, aż zbiór zawierać będzie k elementów. Wynikiem czego powstaje minimalne drzewo rozpinające. Odpowiednikiem *reachability distance* jest *chaining distance* czyli długość krawędzi w minimalnym drzewie rozpinającym.. Obliczanie anomalności punktu x jest analogiczne do LOF – stosunek *chaining distance* dla punktu x do średniej z *chaining distance* dla k-sąsiadów :

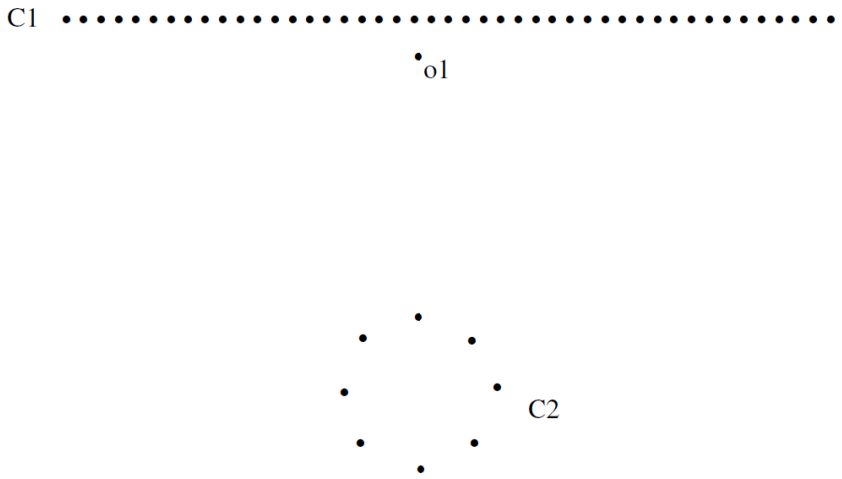
$$COF_k(x) = \frac{|N_k(x)| \cdot ac - dist_{N_K(x)}(x)}{\sum_{o \in N_k(x)} ac - dist_{N_k(o))}(o)} \tag{3.3}$$

gdzie $ac_{dist}(x)$ oznacza średnią długość krawędzi w minimalnym drzewie rozpinającym punktu x

3.3.5 Angle-based Outlier Detection

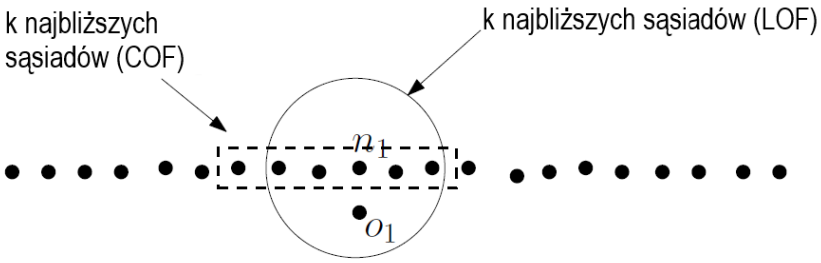
Angle-based Outlier Detection(ABOD) [22] zamiast metryki między punktami w sąsiedztwie obserwacji, porównuje kierunek wektorów skierowanych wychodzących z rozpatrywanego punktu x. W wielowymiarowych przestrzeniach kąty są stabilniejsze niż metryka, spowodowane jest to „przekleństwem wymiarowości". Rezultatem badań na temat zagadnienia „przekleństwa wymiarowości"[5] jest wniosek, że wraz ze wzrostem wymiarów, różnica największej odległości między punktami a najmniejszej dla dowolnego punktu, zbiega do 0 :

$$\lim_{d \to \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \longrightarrow 0 \tag{3.4}$$



Rysunek 3.2 Zbiór, dla którego skuteczna detekcja anomalii algorytmem LOF nie powiedzie się.

źródło: [31]



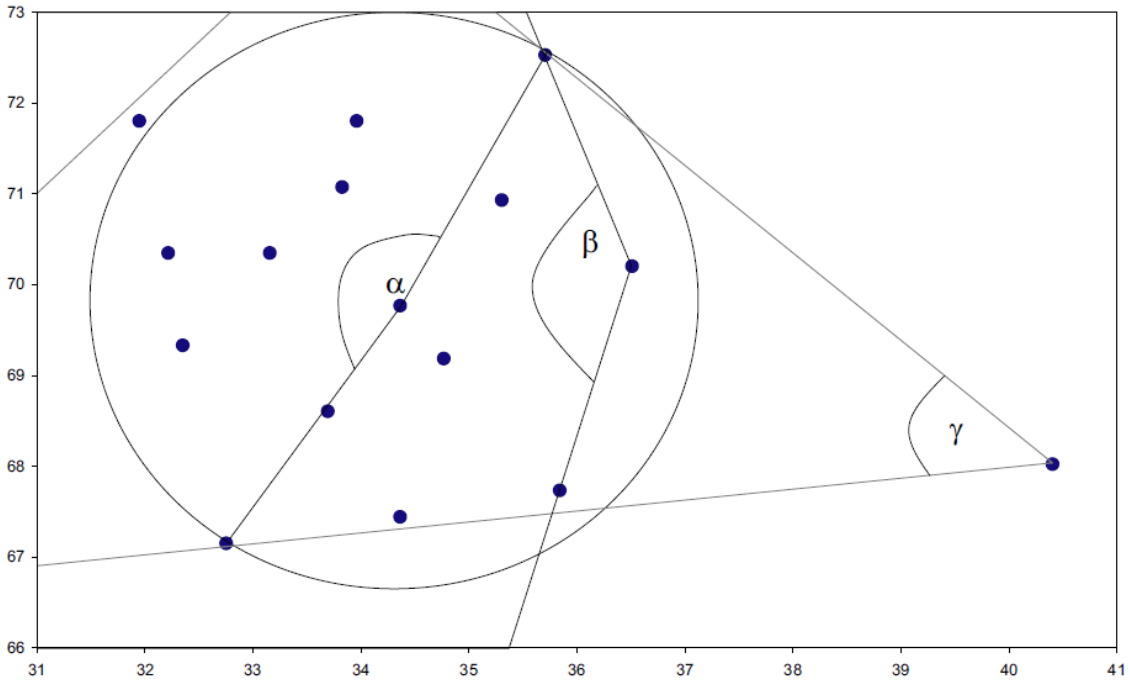
Rysunek 3.3 Różnica w wyborze k najbliższych sąsiadów dla COF i LOF (k=5).
źródło: Opracowanie własne na podstawie [11]

Z tego powodu np. w grupowaniu dokumentów tekstowych jako miary podobieństwa wykorzystuje się miarę kosinusową [21]. Algorytm ABOD rozpatruje wszystkie punkty w zbiorze w odniesieniu do analizowanego punktu, jest to podejście o wysokiej złożoności obliczeniowej $O(n^3)$. W celu zmniejszenia złożoności obliczeniowej w pracy wykorzystano FastABOD, który rozpatruje k-sąsiadów.

Rysunek 3.4 obrazuje intuicję algorytmu ABOD, który do oceny anomalności punktu stosuje współczynnik *Angle-based outlier factor*(ABOF):

$$ABOF(x) = Var\frac{\langle \overrightarrow{xy}, \overrightarrow{xz} \rangle}{\left\|\overrightarrow{xy}\right\|\left\|\overrightarrow{xz}\right\|}, y, z \in B \tag{3.5}$$

gdzie B jest odpowiednio dobranym zbiorem, jako iż w pracy wykorzystano algorytm FastABOD, jest to zbiór k-sąsiadujących punktów. Im niższy ABOF tym większe podejrzenie anomalii



Rysunek 3.4 Intuicja kierująca algorytmem ABOD.
źródło: [22]

3.3.6 Histogram-based Outlier Score

Histogram-based Outlier Score (HBOS) [17] zakłada niezależność cech. Określa anomalność punktu, budując histogramy o n prostokątach dla każdej cechy. Częstotliwość obserwacji przypadających do każdego prostokąta histogramu tożsama jest z zagęszczeniem cechy. Następnie każdy histogram przechodzi normalizacje tak, aby maksymalna wysokość dla każdego histogramu wynosiła 1. Zapewnia to równą wagę każdego histogramu w procesie obliczania anomalności punktu (HBOS). Anomalność punktu x wyznacza się według wzoru:

$$HBOS(x) = \sum_{i=0}^{d} log(\frac{1}{hist_i(x)}) \tag{3.6}$$

3.3.7 Lightweight Online Detector of Anomalies

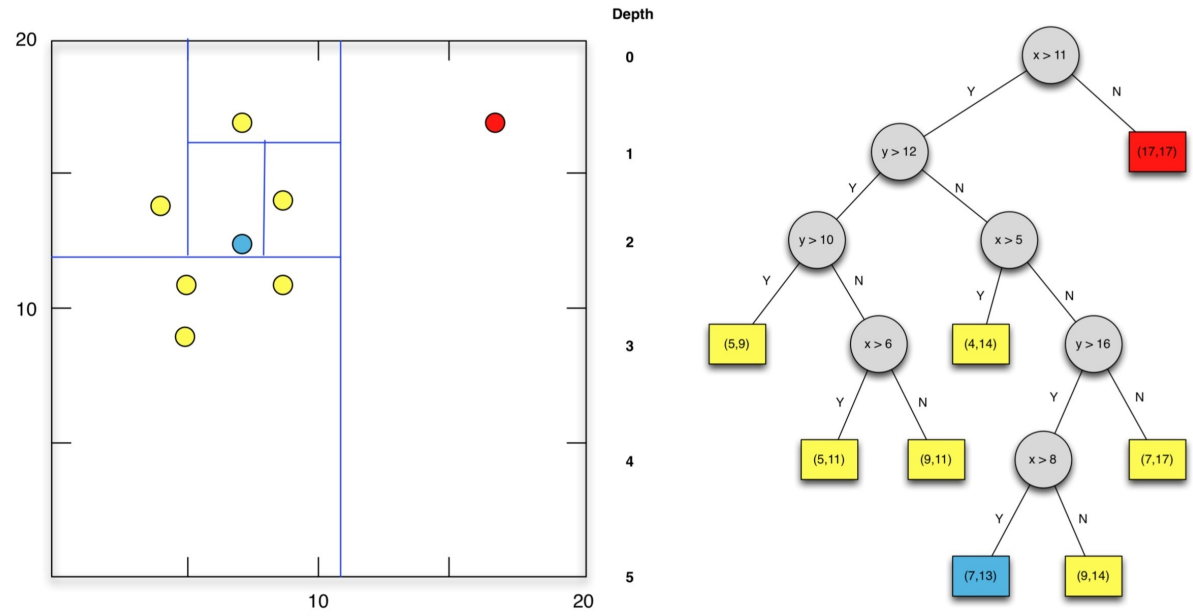
Lightweight Online Detector of Anomalies(LODA) [25] wykorzystuje podejście budowania silnego klasyfikatora składającego się ze słabszych klasyfikatorów (zespół klasyfikatorów). Stosuje technikę losowych projekcji (*Random projection*) w celu redukcji wymiaru, $\mathbb{R}^d \rightarrow \mathbb{R}^k$ [29], gdzie k dobrane przez MetaOD – liczba losowych cięć. LODA składa się z k jednowymiarowych histogramów $\{h_i\}_{i=1}^{k}$ o liczbie prostokątów dobranym przez MetaOD oraz wektorów projekcji $\{w_i \in \mathbb{R}^d\}_{i=1}^{k}$. Każdy z histogramów przybliża gęstość zmiennej losowej (obserwacji) rzutowanej w kierunku wektora w_i. Wynik anomalności dla punktu x wyznacza się ze wzoru:

$$f(x) = -\frac{1}{k}\sum_{i=1}^{k} log\hat{p}_i(x^T w_i) \tag{3.7}$$

gdzie $\hat{p}$ oznacza gęstość zmiennej losowej przybliżonej przez i-ty histogram.

3.3.8 Isolation Forest

Isolation Forest(IForest) [23] wykorzystuje teorię drzew i lasów losowych. Izoluje obserwacje, wybierając losowo cechę i rozdziela ją na podstawie losowej wartości progowej (Rysunek 3.5). W odróżnieniu od drzew decyzyjnych i losowych lasów, zamiast problemu klasyfikacji obliczana jest odległość między korzeniem a liściem (odizolowaną obserwacją). Krótsza odległość między korzeniem a liściem tym łatwiej odizolować obserwację na podstawie losowo wybranych cech. Uśredniona odległość pomiędzy wszystkie drzewa decyzyjne jest wartością anomalności obserwacji. Liczba estymatorów określa liczbę drzew decyzyjnych. Procent rozpatrywanych cech jest odsetkiem losowo wybranych cech wykorzystanych dla każdego estymatora.



Rysunek 3.5 Drzewo decyzyjne w *Isolation Forest*.
źródło: [1]

Rozdział 4

System wykrywania anomalii

Abstrakt Rozdział prezentuje powstały system wykrywania anomalii. Wymienia wymagania funkcjonalne oraz niefunkcjonalne postawione przed systemem. Przedstawia technologie wykorzystane w tworzeniu systemu oraz opisuje implementację funkcjonalności

4.1 Wytyczne projektowe

Uniwersalny system wykrywania anomalii ma za zadanie ułatwić korzystającemu detekcję anomalii dla dowolnego zbioru danych statystycznych. W tym celu najważniejszą funkcją systemu jest automatyzacja wyboru optymalnego modelu detekcji anomalii. System po procesie analizy danych ma wizualizować wynik detekcji w sposób przejrzysty i zrozumiały dla korzystającego.

4.1.1 Wymagania funkcjonalne

- Przesłanie pliku zawierającego zbiór danych w formatach: JSON, CSV
- Oczyszczenie danych z brakujących wartości oraz wyskalowanie
- Wybór optymalnego modelu (algorytm i parametry)
- Detekcja anomalii w zbiorze danych z wykorzystaniem wybranego przez system modelu
- Utworzenie oczyszczonego zbioru danych z anomalii (wartość anomalności w 99. per centylu)
- Utworzenie zbioru danych z wartością anomalności dla każdej obserwacji
- Stworzenie raportu z przebiegu i wyniku detekcji anomalii

4.1.2 Wymagania niefunkcjonalne

- Zbiory danych do pobrania (oczyszczone i zawierające wartość anomalności) powinny być w formacie przesłanych danych
- System powinien być jak najbardziej intuicyjny i prosty w obsłudze

- Raport powinien zawierać niezbędne informacje potrzebne do dalszej analizy przez użytkownika
- System oraz raport powinny być przystępne dla użytkownika bez wiedzy na temat detekcji anomalii

4.2 Analiza technologiczna

4.2.1 Wykorzystane technologie oraz biblioteki

System powstał z wykorzystaniem języka programistycznego Python 3.7. Jest to prosty do nauczenia, przejrzysty, a zarazem wszechstronny język programistyczny o rosnącej popularności zwłaszcza w środowisku uczenia maszynowego. Do stworzenia aplikacji webowej wykorzystano mikro-framework Flask 1.1.2. Do projektowania struktury i interfejsu graficznego strony internetowej wykorzystano: HTML, JavaScript oraz biblioteki CSS – Bootstrap. Do wyboru optymalnego modelu detekcji anomalii wykorzystano bibliotekę MetaOD, którą opisano w sekcji 3.2. Do detekcji anomalii w zbiorze danych, wykorzystując algorytmy z przestrzeni bazowej modeli, skorzystano z gotowych implementacji algorytmów zawartych w bibliotece PyOD. Do analizy oraz przechowywania danych w strukturach danych użyto pakietu Pandas (tabela danych) oraz Numpy(tablice). Do skalowania zbioru danych wykorzystano bibliotekę Scikit-learn. Matplotlib został wykorzystany jako kreator wykresów pudełkowych oraz histogramów.

4.2.2 Wykorzystane narzędzia programistyczne

Zintegrowanym środowiskiem programistycznym wykorzystanym przy tworzeniu aplikacji był PyCharm Professional, czeskiej firmy JetBrains. To zaawansowane i wszechstronne środowisko ułatwia proces pisania kodu źródłowego, testowania oraz rozwijania oprogramowania. Posiada graficzny debugger ułatwiający identyfikację błędów. Wspiera pisanie aplikacji webowych. Posiada integrację z systemem kontroli wersji Git, wykorzystanego przy produkcji systemu do tworzenia kolejnych wersji rozwijanego projektu. Dzięki czemu implementacja nowych funkcjonalności odbywa się bez obawy utraty działającej wersji systemu.

4.3 Implementacja funkcjonalności

4.3.1 Uzyskanie danych

System rozpoczyna działanie od strony startowej, na której klient może wybrać plik z danymi, które chce poddać zadaniu detekcji anomalii. Plik musi mieć rozszerzenie CSV lub JSON. Po naciśnięciu przycisku „Analizuj” następuje przesłanie pliku do systemu. System sprawdza zgodność rozszerzenia pliku, jeżeli rozszerzenie jest obsługiwane, następuje zapisanie pliku na serwerze.

4.3.2 Przygotowanie danych

Po zapisaniu pliku na serwerze następuje wczytanie danych z wykorzystaniem biblioteki Pandas. Wczytane dane są weryfikowane pod kątem brakujących wartości, jeśli zbiór danych zawiera brakujące wartość, system w miejsce brakującej wartości wstawia średnia z

Rysunek 4.1 Strona startowa systemu wykrywania anomalii
źródło: Opracowanie własne

cechy (kolumny). System również dokonuje standaryzacji zbioru danych, wykorzystując funkcje biblioteki Scikit-learn – *MinMaxScaler*, która skaluje obserwację według wzoru:

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

gdzie min, max są to minimalne i maksymalne wartości obserwacji. *MinMaxScaler* został wybrany ze względu na największą skuteczność dla zestawu testowych zbiorów danych (podsekcja 5.3.1).

Listing 4.1 Funkcja wczytująca dane z formatu JSON oraz sprawdzająca brakujące wartości

```
def access_data_json(filename):
    data = pd.read_json(os.path.join(app.config['UPLOAD_FOLDER'], filename))
    check_NAN = data.isnull().values.any()
    if check_NAN == True:
        imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
        imputer = imputer.fit(data)
        data = imputer.transform(data)
    data = data.to_numpy()
    return data
```

4.3.3 Wybór modelu

Wybór modelu dokonywany jest z wykorzystaniem MetaOD (opisanej w Sekcji 3.2).

Listing 4.2 Funkcja skalująca zbiór danych oraz dokonująca wyboru modelu z przestrzeni bazowej modeli

```
def choose_Model(data):
    scaler = MinMaxScaler().fit(data)
    Data_RobustScaled = scaler.transform(data)
    prepare_trained_model()
```

```
selected_models = select_model(Data_RobustScaled, n_selection=1)
for foo, model in enumerate(selected_models):
    model = model.item(0)
    best_clf = name.split("␣")
    clf = best_clf[0]
    param = best_clf[1:]
    parameters = {0:0}
    if clf == "ABOD":
        n_neighbour = param[0]
        n_neighbour = int(n_neighbour)
        model = ABOD(contamination=0.01,
                     n_neighbors=n_neighbour,
                     method='fast')
        parameters = {'Liczba␣sasiadow': n_neighbour}
    if clf == "COF":
        n_neighbours = param[0]
        model = COF(contamination=0.01,
                    n_neighbours=int(n_neighbours))
        parameters = {'Liczba␣sasiado': n_neighbours}
    if clf == "HBOS":
        n_histograms, tolerance = get_param(param)
        model = HBOS(contamination=0.01,
                     n_bins=int(n_histograms),
                     tol=float(tolerance))
        parameters = {'Liczba␣prostokatow␣histogramu': n_histograms,
                      'Tolerancja': tolerance}
    if clf == "Iforest":
        n_estimators, max_features = get_param(param)
        model = IForest(contamination=0.01,
                        n_estimators=int(n_estimators),
                        max_features=float(max_features))
        parameters = {'Liczba␣estymatorow': n_estimators,
                      'Procent␣rozpatrywanych␣cech': max_features}
    if clf == "kNN":
        n_neighbours, method = get_param(param)
        method = method[1:-1]
        model = KNN(contamination=0.01,
                    n_neighbors=int(n_neighbours),
                    method=method)
        parameters = {'Liczba␣sasiadow': n_neighbours,
                      'Metoda': method}
    if clf == "LODA":
        n_bins, n_random_cuts = get_param(param)
        model = LODA(contamination=0.01,
                     n_bins=int(n_bins),
                     n_random_cuts=int(n_random_cuts))
        parameters = {'Liczba␣slupkow␣histogramu': n_bins,
                      'Liczba␣losowych␣ciec': n_random_cuts}
    if clf == "LOF":
        n_neighbours, method = get_param(param)
        method = method[1:-1]
        model = LOF(contamination=0.01,
                    n_neighbors=int(n_neighbours),
                    metric=method)
        parameters = {'Liczba␣sasiadow': n_neighbours,
                      'Metryka': method}
    if clf == "OCSVM":
        nu, kernel = get_param(param)
        kernel = kernel[1:-1]
```

```
            model = OCSVM(contamination=0.01,kernel=str(kernel), nu=float(nu))
            parameters = {'nu': nu,
                          'Funkcja jadra': kernel}
        dump(model, "static/model/clf.joblib")
        return clf, parameters
```

4.3.4 Wykrywanie anomalii

Po uzyskaniu optymalnego modelu, wykorzystując bibliotekę PyOD, dokonujemy zadania detekcji anomalii. Wykorzystując W tym celu w jeden z algorytmów opisanych w sekcji 3.3. Uzyskane wartości anomalności są skalowane do przedziału [0,1] za pomocą funkcji MinMaxScaler

Listing 4.3 Funkcja dokonująca detekcji anomalii oraz przypisująca wartość anomalności dla każdej obserwacji

```
def output_MetaOD(data):
    clf = load('static/model/clf.joblib')
    clf_name = str(clf).split("(")[0]
    transformer = MinMaxScaler().fit(data)
    data_standarize = transformer.transform(data)
    clf.fit(data_standarize)
    labels, decision_score = clf.labels_, clf.decision_scores_
    labels_T = labels.reshape((-1, 1))
    decision_score_T = decision_score.reshape(-1, 1)
    scaler = MinMaxScaler().fit(decision_score_T)
    decision_score_T  = scaler.transform(decision_score_T)
    labels_T = np.where(labels_T == 1, "Anomalia", "Prawid Ćowo Ż ")
    output = np.append(labels_T,decision_score_T,axis=1)
    data_with_labels = np.append(output,data, axis=1)
    dataframe = pd.DataFrame(data_with_labels)
    return dataframe
```

4.3.5 Oczyszczanie danych z anomalii

Ze zbioru danych w celu oczyszczenia usuwamy obserwacje, których wartość anomalności mieści się w 99. per centylu (zaklasyfikowane jako anomalia). Dokonano tego, ustawiają próg zanieczyszczenia danych podczas tworzenia modelu (contamination = 0.01).

4.3.6 Raport działania systemu

System po przeprowadzeniu analizy zbioru danych wyświetla użytkownikowi stronę zawierającą wykres pudełkowy oraz histogram wartości anomalności obserwacji. Jak również zbiór danych posegregowanych według wartości anomalności.

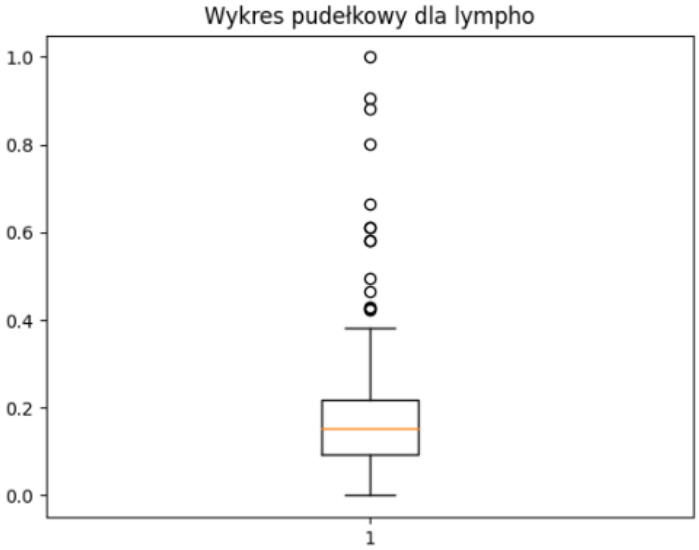
Raport

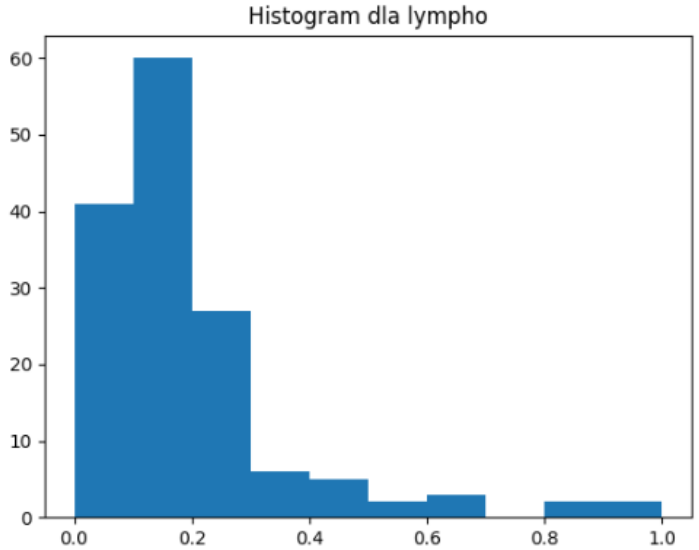
Dane dla: lympho.json

Wykorzystany algorytm: Iforest

Parametry: Liczba estymatorów: 30, Procent rozpatrywanych cech: 0.2,

Pobierz dane | Pobierz oczyszczone dane





Pokaż/ukryj dane

	Klasyfikacja	Anomalność	Cecha 1	Cecha 2	Cecha 3	Cecha 4	Cecha 5	Cecha 6	Cecha 7	Cecha 8	Cecha 9	Cecha 10	Cecha 11	Cecha 12	Cecha 13	Cecha 14	Cecha 15	Cecha 16	Cecha 17	Cecha 18
Obserwacja 2	Anomalia	1.000000	3	2	2	2	2	2	2	2	3	1	1	2	2	8	1	2	2	4
Obserwacja 0	Anomalia	0.923488	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Obserwacja 5	Prawidłowość	0.873471	3	1	1	1	2	2	2	1	3	1	1	2	1	5	3	1	1	7
Obserwacja 4	Prawidłowość	0.813627	3	2	2	2	2	2	2	1	2	2	2	4	2	4	3	2	2	7

Rysunek 4.2 Strona systemu po skończeniu zadania detekcji anomalii
źródło: Opracowanie własne

Rozdział 5

Ewaluacja

Abstrakt Rozdział przedstawia wyniki działania systemu oraz porównuje je z istniejącymi metodami wykrywania anomalii. Argumentuje wykorzystanie MinMaxScaler. Prezentuje możliwość rozwoju systemu.

5.1 Miary ewaluacji

W celu oceny skuteczności systemu detekcji wykorzystano następujących miary:

- precyzja dla top n obserwacji (P@N)
- pole powierzchni pod krzywą Receiver Operating Characteristic (ROC)

Wykorzystane metryki wybrane były jako informatywny sposób oceny skuteczności detekcji anomalii [10]. Jak również wykorzystane były w benchmarku algorytmów zawartego w dokumentacji biblioteki PyOD.

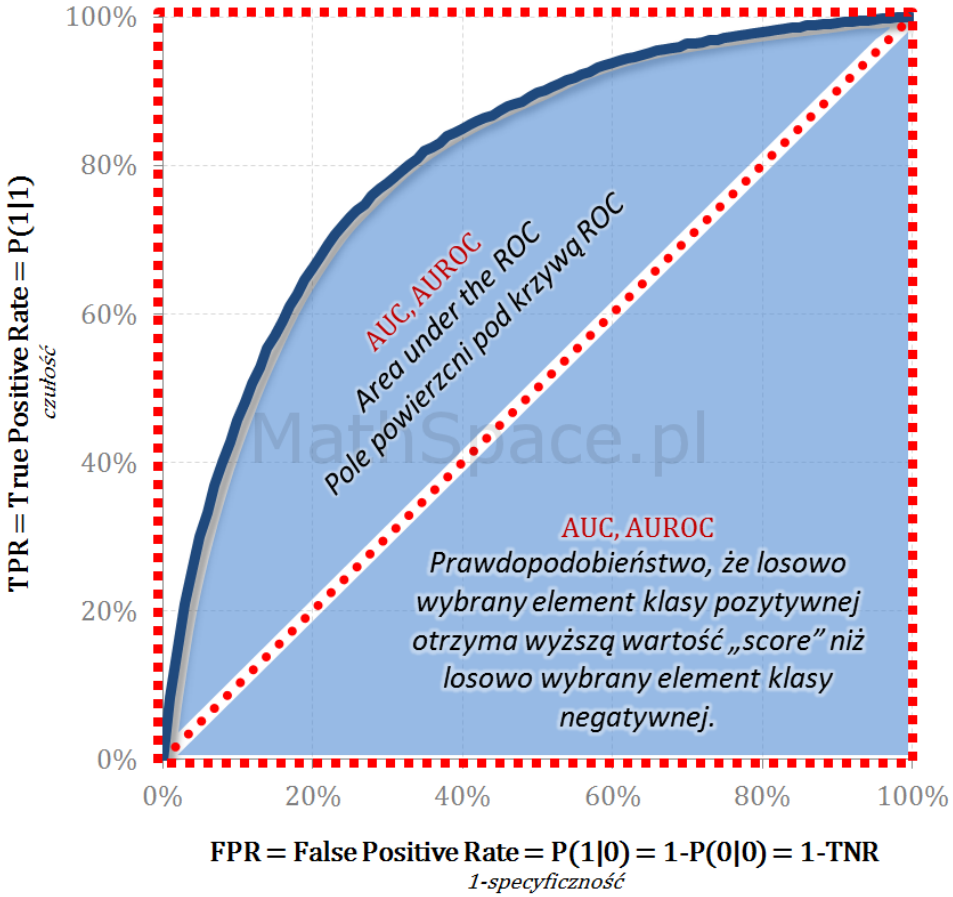
5.1.1 Precyzja dla top n obserwacji (P@N)

Miara oblicza precyzję dla top n obserwacji, gdzie n oznaczać będzie sumę obserwacji odstających w zbiorze danych. N obserwacji, którym system nadał najwyższą wartość wskaźnika anomalności, zaklasyfikowane zostały jako anomalie. P@N określa stosunek klasyfikacji prawdziwie pozytywnych (TP) do sumy klasyfikacji prawdziwie pozytywnych (TP) i fałszywie pozytywnych (FP) dla N obserwacji zaklasyfikowanych jako anomalie:

$$P@N = \frac{TP}{TP + FP} \tag{5.1}$$

5.1.2 Pole powierzchni pod krzywą Receiver Operating Characteristic (ROC)

Miara określa prawdopodobieństwo, że losowy element klasy pozytywnej(anomalii) otrzyma wyższą wartość anomalności niż losowo wybrany element klasy negatywnej (poprawna obserwacja). Krzywa ROC:



Rysunek 5.1 Interpretacja pola pod krzywą ROC
źródło: [20]

5.2 Wykorzystane zbiory danych

Wykorzystano bazę zbiorów danych ODDS [28]. W celu ewaluacji wykorzystano zbiory danych, dla których w dokumentacji biblioteki PyOD przeprowadzono analizę skuteczności detekcji anomalii dla wybranych zbiorów danych. Informacje opisujące zbiory danych zawarto w Dodatku A. Informacje o charakterze zbioru danych oraz znaczeniu wykrycia anomalii w zbiorze opisano w Tabeli A.1. Informacje o zawartości zbiorów: liczba cech i obserwacji, procent anomalii w zbiorze, typ danych przedstawiono w Tabeli A.2.

5.3 Porównanie skuteczności systemu detekcji

5.3.1 Rozpatrywane metody standaryzacji danych

Rozpatrzono skuteczność modelu wybranego przez MetaOD dla czterech metody standaryzacji danych oraz danych nieprzekształconych. Rozpatrywane metody skalowania:

- StandardScaler: standaryzacja Z wykorzystująca średnią oraz odchylenie standardowe cechy:

$$z = \frac{x - \mu}{\sigma}. \tag{5.2}$$

- RobustScaler: standaryzacja odporna na obserwacje odstające, wykorzystuje media-

nę oraz rozstęp ćwiartkowy cechy:

$$X_{std} = \frac{X - mediana(X)}{IQR} \tag{5.3}$$

- MinMaxScaler: standaryzacja danych do przedziału [0,1], gdzie minimalna i maksymalna wartość cechy wyznaczają granice przedziału

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.4}$$

- PowerTransformer (transformacja Yeo–Johnson [33]): transformacja symetryzująca rozkład zmiennej losowej. Dzięki czemu rozkład zmiennej losowej (cecha zbioru danych) przypomina rozkład normalny.

$$y_i^{(\lambda)} = \begin{cases} ((y_i+1)^{\lambda} - 1)/\lambda & \text{jeśli } \lambda \neq 0, y \geqslant 0 \\ \log(y_i + 1) & \text{jeśli } \lambda = 0, y \geqslant 0 \\ -[(-y_i+1)^{(2-\lambda)} - 1]/(2-\lambda) & \text{jeśli } \lambda \neq 2, y < 0 \\ -\log(-y_i + 1) & \text{jeśli } \lambda = 2, y < 0 \end{cases} \tag{5.5}$$

5.3.2 Wyniki

Wyniki zostały zamieszczone w Dodatku B.

- Porównanie metod skalowania na podstawie pola pod krzywą ROC – Tabela B.1
- Porównanie metod skalowania na podstawie P@N – Tabela B.2
- Porównanie wybranego modelu przez MetaOD do modeli PyOD na podstawie pola pod krzywą ROC – Tabela B.3
- Mapa ciepła pola pod krzywą ROC – Rysunek B.1
- Porównanie wybranego modelu przez MetaOD do modeli PyOD na podstawie P@N – Tabela B.4
- Mapa ciepła P@N – Rysunek B.2

5.4 Analiza wyników

Na podstawie wyników z badania skuteczności metody standaryzacji danych (Tabela B.1 oraz B.2) zdecydowano na standaryzację danych, wykorzystując MinMaxScaler. Standaryzacja danych z wykorzystaniem MinMaxScaler zwiększyła średnią skuteczność wybranego modelu w porównaniu do modelu wybranego na podstawie danych niepoddanych standaryzacji o:

- średnie pole pod krzywą ROC: wzrosło o 5,11%
- średnie P@N: wzrosło o 26%

W wyniku standaryzacji danych wykorzystując MinMaxScaler, MetaOD skutecznością zajęło – w porównaniu do algorytmów PyOD – odpowiednio:

- Na podstawie średniego pola pod krzywą ROC: 2 miejsce – lepsza skuteczność: *Isolation Forest*
- Na podstawie średniego P@N: 1 miejsce

Mapy ciepła dla P@N i pola pod krzywą ROC (Rysunek B.1 i B.2) pokazują, że wybrany model przez MetaOD dla zbiorów danych, dla których inne modele były nieskuteczne, również jest nieskuteczny (zbiór danych: vertebral). Jednakże porównując średnią wartość P@N oraz pola pod krzywą ROC model wybrany przez MetaOD wyróżnia się skutecznością detekcji na tle innych modeli, wraz z algorytmem *Isolation Forest*. Analizując wyniki, dochodzimy do konkluzji, że początkowe rozwiązanie wykorzystujące algorytm *Isolation Forest* byłoby porównywalnie skuteczne co wybrana metoda wykorzystująca meta-uczenie.

Jednak podczas analizy porównawczej skuteczności modeli wybranych przez MetaOD w zależności od metody standaryzacji danych można zaobserwować, iż wybór skutecznej metody standaryzacji dla konkretnego zbioru danych może znacząco podnieść skuteczność detekcji anomalii, np. model wybrany po standaryzacji danych wykorzystując RobustScaler dla zbioru *vertebral*. Otwiera to możliwość badań nie tylko nad wyborem modelu, ale również metody standaryzacji oraz transformacji danych.

Rozdział 6

Podsumowanie

Celem niniejszej pracy inżynierskiej było zaprojektowanie oraz stworzenie systemu, który zapewniłby użytkownikowi na dokonanie analizy zbioru danych pod kątem występowania anomalii. Ideą kierującą projektowanie systemu było stworzenie swojego rodzaju „czarnej skrzynki", która automatyzowałaby proces detekcji anomalii, dzięki czemu system byłby przystępny dla użytkownika bez wiedzy z zakresu detekcji anomalii. Problematyką zadania detekcji anomalii jest mnogość podejść w zależności od zastosowania, różnorodność danych. Brak możliwości ewaluacji na zbiorach danych bez etykiet znacząco utrudnia wybór jednego skutecznego modelu, który skutecznie wykrywałby anomalii w zbiorze danych.

W tradycyjnym podejściu detekcji anomalii wybór modelu kierowany był wstępną wiedzą na temat zbioru danych. Posiadanych cechy obserwacji, metodę generującą obserwacje. Prezentowany w niniejszej pracy system skutecznie wykrywa anomalie dla zróżnicowanych zbiorów danych bez wcześniejszej wiedzy na temat samego zbioru. System wykorzystuje w procesie doboru modelu detektora anomalii meta-uczenie. Dzięki czemu system na podstawie meta-cech uzyskanych z analizy zbioru danych rekomenduje modelu, zapewniając skuteczne zastosowanie systemu dla zróżnicowanych zbiorów danych.

Niniejsza praca oprócz samego stworzenia systemu jest też zwięzłym wprowadzeniem do tematu detekcji anomalii wraz z opisem wykorzystywanych algorytmów, dzięki czemu może być wykorzystana jako wstęp polskiego czytelnika w metody oraz postępów w zakresie detekcji anomalii. Tematu niszowego w polskiej literaturze naukowej. Jak również prezentuje innowacyjne podejście automatyzacji doboru detektora anomalii z wykorzystaniem biblioteki MetaOD.

Jednakże obecny stan techniki detekcji anomalii powoduje, że stworzenie uniwersalnie idealnego systemu jest niemożliwe. Wyniki anomalności wyznaczony przez system należy traktować jako wskaźnik, nad którymi obserwacjami należy przeprowadzić dalszą analizę w celu uzyskania wartościowej informacji. Przenosząc na użytkownika decyzję czy dana obserwacja jest, czy też nie jest anomalią, jednakże zapewniając informacje ułatwiające tę decyzję.

Stworzony system nie jest definitywnym rozwiązaniem problemu detekcji anomalii, charakter pracy skupiony jest na demonstracji meta-uczenia w zadaniu detekcji anomalii implementując go w stworzonym systemie w celu automatyzacji wyboru modelu detektora. Co w teorii zapewnia wszechstronność i uniwersalność systemu przez dostosowanie rozwiązania do problemu. Dodatkowo wraz z rozwojem biblioteki MetaOD, przez zwiększenie bazy historycznych wyników detekcji anomalii dla podobnych zbiorów, skuteczność systemu będzie rosła.

Dodatki

Dodatek A

Wykorzystane zbiór danych

Zbiór	Charakter zbioru danych	Znaczenie anomalii
arrhythmia	Dane pacjenta wraz z informacjami badania EKG	Wykrycie arytmi serca
cardio	Wyniki badania Kardiotokografii	Wykrycie patologii pracy serca płodu
glass	Skład chemiczny różnych rodzai szkła	Wykrycie próbek szkła, należących do klasy zdecydowanie mniejszościowej
ionosphere	Sygnały odbite radaru (jonosondy)	Sygnał jonosondy przechodzi przez jonosferę – brak wykrycia struktury w jonosferze
letter	Wybrane 3 litery tworzą klasę normalnych obserwacji	Litery spoza normalnej klasy
lympho	Wyniki badania limfografii	Wykrycie metastazy lub zwłóknienia
mnist	Obserwacje cyfry 0 uznane za normalne obserwacje	700 zdjęć cyfry 6 uznane za anomalie
musk	Zbiór opisuje budowę oraz strukturę różnych związków chemicznych	Wykrycie związku uznanego za piżmo
optdigits	Zbiór składa się z cyfr. Zbiór zawiera ponad 97% obserwacji cyfr od 1 do 9	Wykrycie cyfry 0 (2.86%) – anomalia
pendigits	Zbiór zbliżony charakterem do „optdigits”	Wykrycie cyfry 0 – anomalia
pima	Zbiór danych o kobietach z plemienia Pima takich jak poziom glukozy, BMI, ciśnienie tętnicze	Wykrycie cukrzycy
satellite	Zdjęcia satelitarne programu Landsat. Zbiór danych oryginalnie służył do klasyfikacji gleby (7 klas)	Najmniej liczne klasy (2,4,5) – anomalie
satimage-2	Zbiór danych „satellite”, dla którego ilość obserwacji klasy 2 została zmniejszona do 71	Wykrycie gleby należącej do klasy 2 (pole bawełny)
shuttle	Informacje pokładowe wahadłowca kosmicznego. Ponad 92% obserwacji należy do klasy 1	Obserwacje nie należące do klasy 1 – anomalie
vertebral	Cechy biometryczne opisujące miednicę oraz odcinek lędzwiowy kręgosłupa	Obserwacje normalnej biomechaniki zmniejszono do 30 obserwacji – anomalie
vowels	Dane zawierają zakodowane wypowiedzenia samogłoski /ae/, przez 4 mężczyzn. Wypowiedzenia samogłoski przez pierwszego mężczyzny zmniejszono do 50	Wykrycie wypowiedzenia samogłoski /ae/ przez 1 mężczyznę – anomalia
wbc	Wyniki biopsji aspiracyjnej cienkoigłowej guzów piersi	Złośliwy rak piersi – anomalia

Tabela A.1 Informacje o charakterze zbioru danych i obecnych w zbiorze anomalii
źródło: Opracowanie własne na podstawie [28]

Zbiór	Liczba Obserwacji	Liczba Cech	Procent Anomalii	Brakujące wartości	Cechy jakościowe	Cechy ilościowe
arrhythmia	452	274	14.6018	Tak	Tak	Tak
cardio	1831	21	9.6122	Nie	Nie	Tak
glass	214	9	4.2056	Nie	Nie	Tak
ionosphere	351	33	35.8974	Nie	Nie	Tak
letter	1600	32	6.2500	Nie	Nie	Tak
lympho	148	18	4.0541	Nie	Tak	Nie
mnist	7603	100	9.2069	Nie	Nie	Tak
musk	3062	166	3.1679	Nie	Nie	Tak
optdigits	5216	64	2.8758	Nie	Nie	Tak
pendigits	6870	16	2.2707	Nie	Nie	Tak
pima	768	8	34.8958	Nie	Nie	Tak
satellite	6435	36	31.6395	Nie	Nie	Tak
satimage-2	5803	36	1.2235	Nie	Nie	Tak
shuttle	49097	9	7.1511	Nie	Nie	Tak
vertebral	240	6	12.5000	Nie	Nie	Tak
vowels	1456	12	3.4341	Nie	Nie	Tak
wbc	378	30	5.5556	Nie	Nie	Nie

Tabela A.2 Informacje o właściwościach zbiorów danych
źródło: Opracowanie własne na podstawie [28]

Dodatek B

Wyniki porównawcze

Zbiór	Dane oryginalne	Robust Scaler	Standard Scaler	MinMax Scaler	Power Transformer
arrhythmia	0.7480	0.7527	0.7685	0.8093	0.7619
cardio	0.5474	0.6150	0.8604	0.9109	0.3830
glass	0.6249	0.4016	0.7122	0.7518	0.6076
ionosphere	0.8336	0.8308	0.8065	0.8678	0.8481
letter	0.8790	0.8096	0.8976	0.8411	0.9062
lympho	0.9977	0.9930	0.9953	0.9941	1.0000
musk	0.9460	0.9994	0.7267	0.9923	0.1024
optdigits	0.7431	0.4615	0.5380	0.4757	0.4856
pendigits	0.6975	0.7786	0.6792	0.7218	0.5707
pima	0.6996	0.7025	0.5650	0.6751	0.4289
satellite	0.5549	0.5394	0.6123	0.6380	0.3594
satimage-2	0.9795	0.9715	0.9688	0.9808	0.9469
vertebral	0.3771	0.7484	0.3465	0.2254	0.2835
vowels	0.9551	0.9727	0.9551	0.9543	0.9781
wbc	0.8980	0.8844	0.9689	0.8814	0.6293
mnist	0.7737	0.8051	0.6330	0.8130	0.6960
shuttle	0.6166	0.9860	0.6167	0.9973	0.6105
Średnia	0.7572	0.7795	0.7442	0.7959	0.6234

Tabela B.1 Porównanie skuteczności modelu wybranego przez MetaOD w zależności od metody standaryzacji danych – pole pod krzywą ROC

źródło: Opracowanie własne

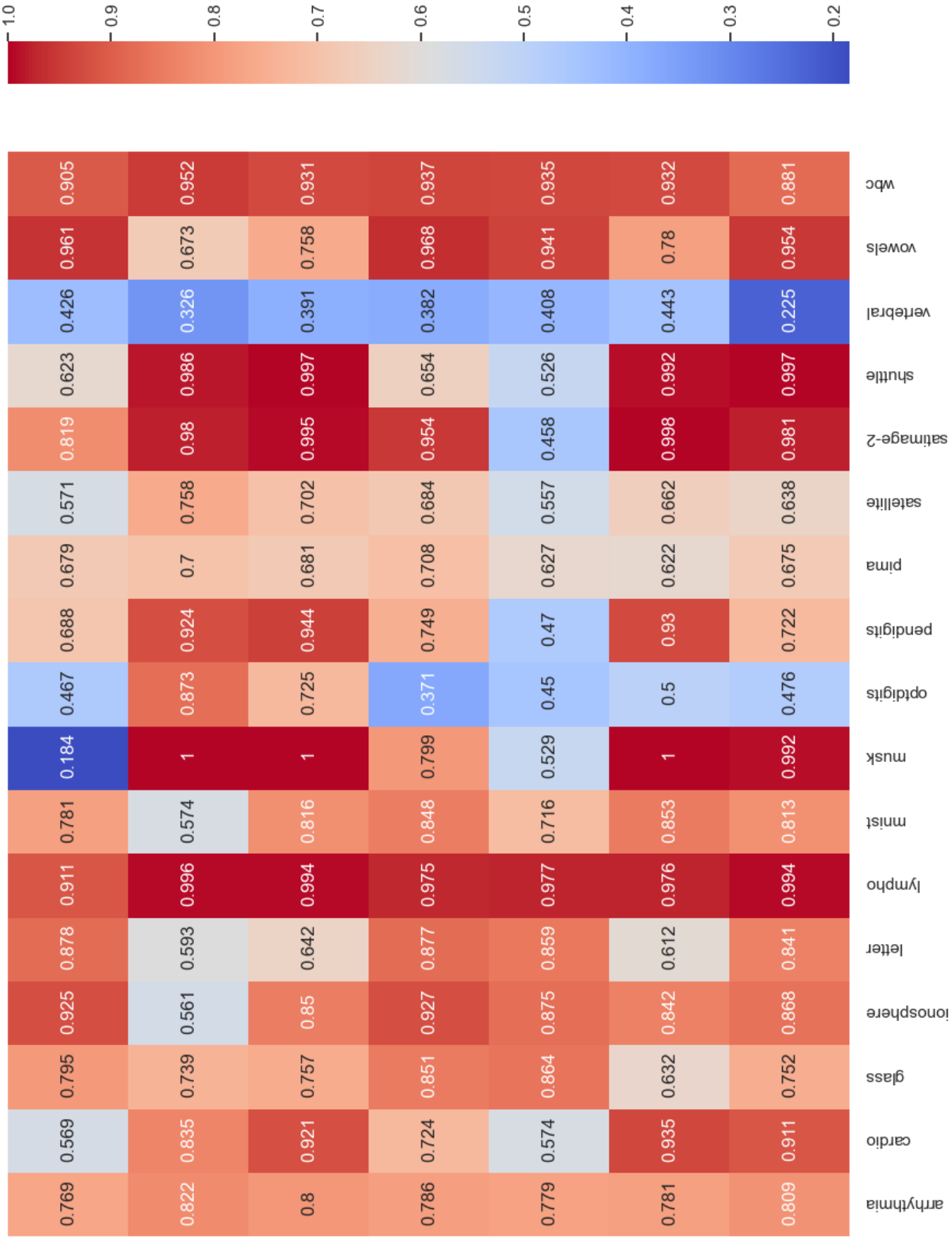
Zbiór	Dane oryginalne	Robust Scaler	Standard Scaler	MinMax Scaler	Power Transformer
arrhythmia	0.3788	0.4545	0.3788	0.4697	0.4242
cardio	0.2273	0.2159	0.4489	0.5114	0.1307
glass	0.1111	0.2222	0.1111	0.1111	0.0000
ionosphere	0.6349	0.6429	0.6032	0.6984	0.6349
letter	0.3800	0.2600	0.5400	0.3100	0.4800
lympho	0.8333	0.6667	0.8333	0.8333	1.0000
musk	0.6495	0.9381	0.0309	0.7320	0.0515
optdigits	0.0067	0.0067	0.0733	0.0200	0.0400
pendigits	0.0769	0.0769	0.0705	0.0641	0.0513
pima	0.5485	0.5373	0.3881	0.5187	0.2985
satellite	0.3723	0.3900	0.5231	0.5373	0.1876
satimage-2	0.8451	0.8310	0.8732	0.8592	0.6056
vertebral	0.0667	0.2667	0.0000	0.0000	0.0000
vowels	0.5400	0.7000	0.5400	0.6000	0.7200
wbc	0.2857	0.5714	0.6190	0.5714	0.1429
mnist	0.3500	0.3600	0.2314	0.4100	0.2557
shuttle	0.1925	0.8762	0.4000	0.9425	0.1837
Średnia	0.3823	0.4716	0.392	0.4817	0.3063

Tabela B.2 Porównanie skuteczności modelu wybranego przez MetaOD w zależności od metody standaryzacji danych – P@N

źródło: Opracowanie własne

Zbiór	#Obserwacji	# Cech	% Anomalii	ABOD	HBOS	IForest	KNN	LOF	OCSVM	MetaOD
arrhythmia	452	274	14.6018	0.7688	0.8219	0.8005	0.7861	0.7787	0.7812	0.8093
cardio	1831	21	9.6122	0.5692	0.8351	0.9213	0.7236	0.5736	0.9348	0.9109
glass	214	9	4.2056	0.7951	0.7389	0.7569	0.8508	0.8644	0.6324	0.7518
ionosphere	351	33	35.8974	0.9248	0.5614	0.8499	0.9267	0.8753	0.8419	0.8678
letter	1600	32	6.2500	0.8783	0.5927	0.6420	0.8766	0.8594	0.6118	0.8411
lympho	148	18	4.0541	0.9110	0.9957	0.9941	0.9745	0.9771	0.9759	0.9941
mnist	7603	100	9.2069	0.7815	0.5742	0.8159	0.8481	0.7161	0.8529	0.8130
musk	3062	166	3.1679	0.1844	1.0000	0.9999	0.7986	0.5287	1.0000	0.9923
optdigits	5216	64	2.8758	0.4667	0.8732	0.7253	0.3708	0.4500	0.4997	0.4757
pendigits	6870	16	2.2707	0.6878	0.9238	0.9435	0.7486	0.4698	0.9303	0.7218
pima	768	8	34.8958	0.6794	0.7000	0.6806	0.7078	0.6271	0.6215	0.6751
satellite	6435	36	31.6395	0.5714	0.7581	0.7022	0.6836	0.5573	0.6622	0.6380
satimage-2	5803	36	1.2235	0.8190	0.9804	0.9947	0.9536	0.4577	0.9978	0.9808
shuttle	49097	9	7.1511	0.6234	0.9855	0.9971	0.6537	0.5264	0.9917	0.9973
vertebral	240	6	12.5000	0.4262	0.3263	0.3905	0.3817	0.4081	0.4431	0.2254
vowels	1456	12	3.4341	0.9606	0.6727	0.7585	0.9680	0.9410	0.7802	0.9543
wbc	378	30	5.5556	0.9047	0.9516	0.9310	0.9366	0.9349	0.9319	0.8814
Średnia				0.7031	0.7819	0.8179	0.7758	0.6792	0.7935	0.7959

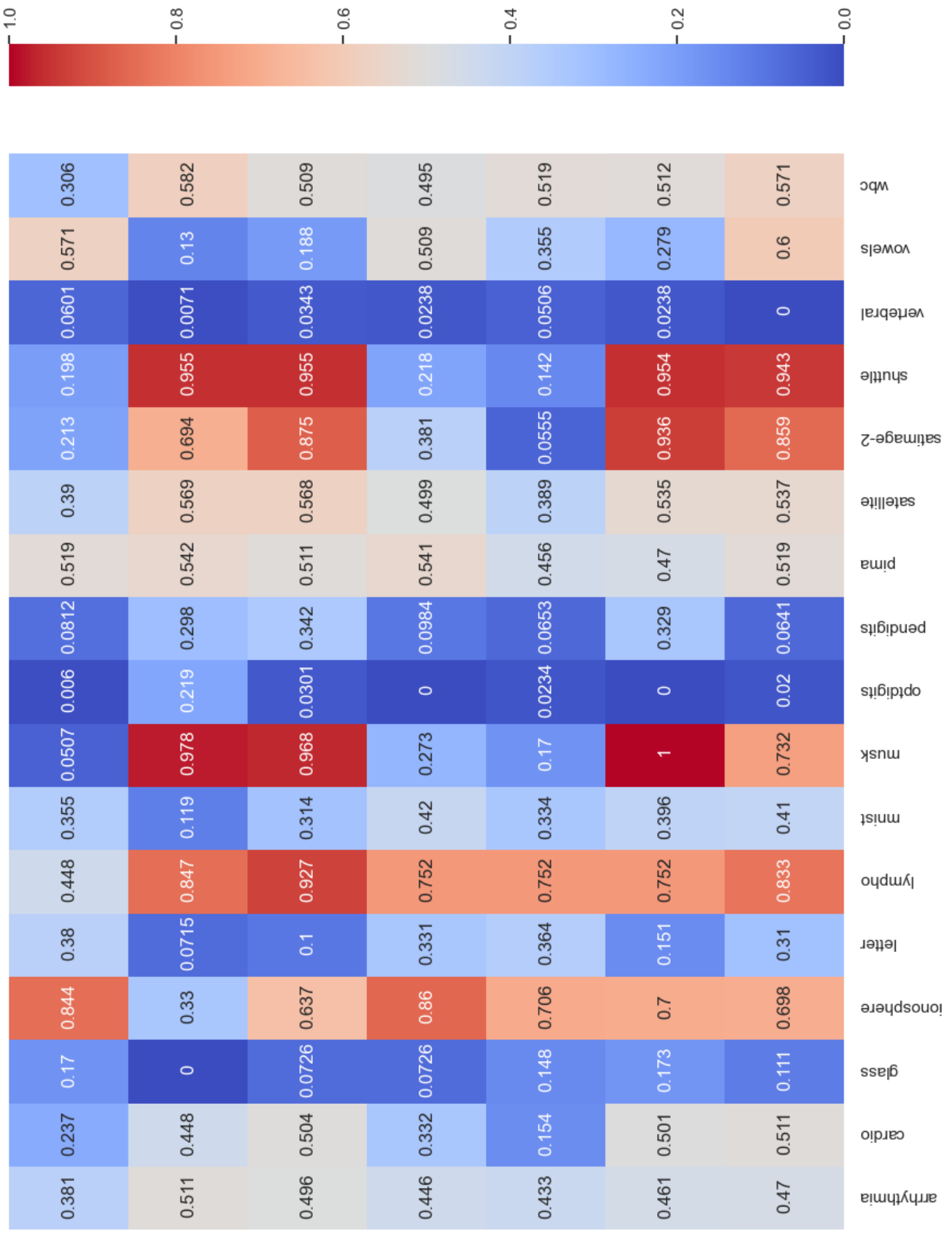
Tabela B.3 Porównanie pola pod krzywą ROC wybranego modelu przez MetaOD do modeli PyOD
źródło: Opracowanie własne na podstawie [34]



Rysunek B.1 Mapa ciepła pola pod krzywą ROC
źródło: Opracowanie własne

Zbiór	#Obserwacji	# Cech	% Anomalii	ABOD	HBOS	IForest	KNN	LOF	OCSVM	MetaOD
arrhythmia	452	274	14.6018	0.3808	0.5111	0.4961	0.4464	0.4334	0.4614	0.4697
cardio	1831	21	9.6122	0.2374	0.4476	0.5041	0.3323	0.1541	0.5011	0.5114
glass	214	9	4.2056	0.1702	0.0000	0.0726	0.0726	0.1476	0.1726	0.1111
ionosphere	351	33	35.8974	0.8442	0.3295	0.6369	0.8602	0.7063	0.7000	0.6984
letter	1600	32	6.2500	0.3801	0.0715	0.1003	0.3312	0.3641	0.1510	0.3100
lympho	148	18	4.0541	0.4483	0.8467	0.9267	0.7517	0.7517	0.7517	0.8333
mnist	7603	100	9.2069	0.3555	0.1188	0.3135	0.4204	0.3343	0.3962	0.4100
musk	3062	166	3.1679	0.0507	0.9783	0.9680	0.2733	0.1695	1.0000	0.7320
optdigits	5216	64	2.8758	0.0060	0.2194	0.0301	0.0000	0.0234	0.0000	0.0200
pendigits	6870	16	2.2707	0.0812	0.2979	0.3422	0.0984	0.0653	0.3287	0.0641
pima	768	8	34.8958	0.5193	0.5424	0.5111	0.5413	0.4555	0.4704	0.5187
satellite	6435	36	31.6395	0.3902	0.5690	0.5676	0.4994	0.3893	0.5346	0.5373
satimage-2	5803	36	1.2235	0.2130	0.6939	0.8754	0.3809	0.0555	0.9356	0.8592
shuttle	49097	9	7.1511	0.1977	0.9551	0.9546	0.2184	0.1424	0.9542	0.9425
vertebral	240	6	12.5000	0.0601	0.0071	0.0343	0.0238	0.0506	0.0238	0.0000
vowels	1456	12	3.4341	0.5710	0.1297	0.1875	0.5093	0.3551	0.2791	0.6000
wbc	378	30	5.5556	0.3060	0.5817	0.5088	0.4952	0.5188	0.5125	0.5714
Średnia				0.3066	0.4294	0.4723	0.3679	0.301	0.4808	0.4817

Tabela B.4 Porównanie P@N wybranego modelu przez MetaOD do modeli PyOD
źródło: Opracowanie własne na podstawie [34]



Rysunek B.2 Mapa ciepła P@N
źródło: Opracowanie własne

Literatura

[1] *Isolation forest.* `https://zhuanlan.zhihu.com/p/32841893`, Dostęp: 2021-01-12.

[2] *Słownik języka polskiego PWN.* `https://sjp.pwn.pl/sjp/anomalia;2550157.html`, Dostęp: 2020-11-23.

[3] *Wykres pudełkowy.* https://www.statystyka-zadania.pl/wykres-pudelkowy. Dostęp: 2021-01-12.

[4] C. C. Aggarwal, *Outlier Analysis*, Springer, 2017.

[5] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space*, in Database Theory — ICDT 2001, J. Van den Bussche and V. Vianu, eds., Springer Berlin Heidelberg, 2001.

[6] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*, Springer, 2017.

[7] V. Barnett and T. Lewis, *Outliers in statistical data*, Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, (1984).

[8] R. J. Bolton, D. J. Hand, et al., *Unsupervised profiling methods for fraud detection*, Credit scoring and credit control VII, (2001), pp. 235–255.

[9] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, *Lof: identifying density-based local outliers*, in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[10] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, *On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study*, Data mining and knowledge discovery, 30 (2016), pp. 891–927.

[11] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR), 41 (2009), pp. 1–58.

[12] T. H. Davenport and D. Patil, *Data scientist: The sexiest job of the 21st century*, Harvard business review, 90 (2012), pp. 70–76.

[13] E. W. Dereszynski and T. G. Dietterich, *Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns*, ACM Transactions on Sensor Networks (TOSN), 8 (2011), pp. 1–36.

[14] F. Y. Edgeworth, *On discordant observations*, Philisoph. Mag, 23 (1887), pp. 364–375.

[15] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, *A meta-analysis of the anomaly detection problem*, arXiv preprint arXiv:1503.01158, (2015).

[16] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, *Anomaly-based network intrusion detection: Techniques, systems and challenges*, computers & security, 28 (2009), pp. 18–28.

[17] M. Goldstein and A. Dengel, *Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm*, KI-2012: Poster and Demo Track, (2012), pp. 59–63.

[18] M. Goldstein and S. Uchida, *Behavior analysis using unsupervised anomaly detection*, in The 10th Joint Workshop on Machine Perception and Robotics (MPR 2014). Online, 2014.

[19] ——, *A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data*, PloS one, 11 (2016), p. e0152173.

[20] M. Gromada, *Receiver operating characteristic – krzywa roc – czyli ocena jakości klasyfikacji (część 7).* `https://mathspace.pl/matematyka/receiver-operating-characteristic-krzywa-roc-czyli-ocena-jakosci-klasyfikacji-czesc-7/`, Dostęp: 2021-01-14.

[21] A. Huang et al., *Similarity measures for text document clustering*, in Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, vol. 4, 2008, pp. 9–56.

[22] H.-P. Kriegel, M. Schubert, and A. Zimek, *Angle-based outlier detection in high-dimensional data*, in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 444–452.

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, *Isolation forest*, in 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422.

[24] M. Mamczur, *Czym są autoenkodery (autokodery) i jakie maja zastosowanie?* https://miroslawmamczur.pl/czym-sa-autoenkodery-autokodery-i-jakie-maja-zastosowanie/. Dostęp: 2021-10-12.

[25] T. Pevnỳ, *Loda: Lightweight on-line detector of anomalies*, Machine Learning, 102 (2016), pp. 275–304.

[26] J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, et al., *Estimating the support of a high-dimensional distribution*, Technical Report MSR-T R-99–87, Microsoft Research (MSR), (1999).

[27] S. Ramaswamy, R. Rastogi, and K. Shim, *Efficient algorithms for mining outliers from large data sets*, SIGMOD Rec., 29 (2000), p. 427–438.

[28] S. Rayana, *ODDS library.*

[29] E. Skubalska-Rafajłowicz, *Losowe projekcje*, 2017.

[30] C. Spence, L. Parra, and P. Sajda, *Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model*, in Proceedings IEEE workshop on mathematical methods in biomedical image analysis (MMBIA 2001), IEEE, 2001, pp. 3–10.

[31] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, *Enhancing effectiveness of outlier detections for low density patterns*, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2002, pp. 535–548.

[32] J. Vanschoren, *Meta-learning: A survey*, arXiv preprint arXiv:1810.03548, (2018).

[33] I.-K. Yeo and R. A. Johnson, *A new family of power transformations to improve normality or symmetry*, Biometrika, 87 (2000), pp. 954–959.

[34] Y. Zhao, Z. Nasrullah, and Z. Li, *Pyod: A python toolbox for scalable outlier detection*, Journal of Machine Learning Research, 20 (2019), pp. 1–7.

[35] Y. Zhao, R. Rossi, and L. Akoglu, *Automating outlier detection via meta-learning*, arXiv preprint arXiv:2009.10606, (2020).

Spis rysunków

Spis tabel