Zadanie nr 2 - Wyznaczanie cech oraz badanie trendu

Analiza danych złożonych z detekcją wyjątków

Karol Kazusek - 254189, Sebastian Zych - 254264

21.10.2024

1 Cel zadania

Zadanie polega na wyznaczeniu cech w wybranym strumieniu danych oraz wyznaczeniu trendu. Do analiz należało wybrać tematykę medycyny lub rozpoznawania faz ruchu aktywności człowieka (systemy HAR)

2 Opis zaproponowanych Klasyfikatorów i algorytmów detekcji zmian

2.1 Metoda klasyfikacyjne

2.1.1 K-nn

Metoda **K-Najbliższych Sąsiadów (K-NN)** to jedna z najprostszych technik klasyfikacji. Działa na zasadzie porównywania nowego punktu danych z istniejącymi przykładami treningowymi. Algorytm wybiera K najbliższych sąsiadów (na podstawie odległości euklidesowej lub innych miar), a następnie przypisuje nowy punkt do klasy, która występuje najczęściej wśród tych sąsiadów.

- `n_neighbors`: Liczba sąsiadów (K) do uwzględnienia przy klasyfikacji.
- `weights`: Sposób ważenia sąsiadów. Możliwe wartości to `'uniform'` (wszyscy sąsiedzi mają taką samą wagę) i `'distance'` (bliżsi sąsiedzi mają większy wpływ).
- `metric`: Miara odległości używana do porównania punktów, w naszym przypadku wyznaczamy odległość za pomocą metryki wyznaczoną przez DTW oraz euklidesową do porównania.

2.1.2 DTW

Metryka **Dynamic Time Warping (DTW)** to algorytm służący do mierzenia odległości między dwoma sekwencjami czasowymi, które mogą różnić się czestotliwością, fazą lub długością. Jego celem jest znalezienie optymalnej nieliniowej ścieżki dopasowania między dwiema sekwencjami, minimalizując różnice między nimi. Algorytm wykorzystuje dynamiczne programowanie, aby efektywnie znaleźć najlepsze dopasowanie.

2.1.3 TSF

TimeSeriesForest (TSF) to algorytm klasyfikacji szeregów czasowych zaprojektowany z myślą o analizie i modelowaniu danych sekwencyjnych. Jest on rozwinięciem klasycznych metod klasyfikacji, takich jak lasy losowe, jednak dostosowany do pracy z danymi zmieniającymi się w czasie. TimeSeriesForest wykorzystuje koncepcję lasów losowych, aby połączyć wyniki z wielu drzew decyzyjnych, dzięki czemu model osiąga wyższą dokładność i lepszą generalizację. W TSF każde drzewo jest trenowane na losowych podpróbkach danych czasowych, a jego struktura jest dostosowana do analizy charakterystycznych cech serii czasowych.

- `n_estimators`: Liczba drzew które będą podejmować decyzję o klasyfikacji wzorca.
- `n_windows`: Liczba okien, z których wyodrębniane są cechy.
- `min_window_size`: Minimalna długość okien wykorzystwyana w klasyfikacji.

2.2 Metoda detekcji zmian

2.2.1 ADWIN

(Okno przesuwne) **Adaptive Windowing (ADWIN)** to metoda wykrywania concept drift, czyli zjawiska zmiany rozkładu danych w czasie, co może wpływać na skuteczność modeli uczących się. ADWIN działa poprzez dynamiczne dostosowywanie okna czasowego, które monitoruje dane strumieniowe, starając się zidentyfikować momenty, w których statystyczne właściwości danych ulegają zmianie. Kiedy wykrywana jest istotna różnica między starą a nową częścią okna, starsze dane są odrzucane, a nowe włączane do analizy. Dzięki automatycznej adaptacji wielkości okna ADWIN skutecznie reaguje na różne rodzaje i prędkości driftu, co sprawia, że jest użytecznym narzędziem w środowiskach strumieniowych.

- **delta**:
  - Typ: `float`
  - Opis: Kontroluje poziom istotności statystycznej przy wykrywaniu driftu. Niższa wartość oznacza bardziej restrykcyjne wykrywanie zmian, wyższa – częstsze wykrywanie.
- **clock**:
  - Typ: `int`
  - Opis: Określa, co ile próbek algorytm sprawdza możliwość wystąpienia driftu.
- **min_window_length**:
  - Typ: `int`
  - Opis: Minimalna długość okna obserwacyjnego, chroni przed zbyt szybkim usuwaniem starszych danych.

2.2.2 DDM

DDM (Drift Detection Method) to metoda wykrywania zmian koncepcyjnych oparta na modelu PAC (Probably Approximately Correct), zakładająca, że jeśli rozkład danych jest stacjonarny, to wraz ze wzrostem liczby analizowanych próbek wskaźnik błędów klasyfikatora powinien się zmniejszać.

Jeśli algorytm wykryje wzrost wskaźnika błędów, który przekroczy wyznaczony próg, uznaje, że nastąpiła zmiana lub ostrzega użytkownika o możliwości zmiany w najbliższym czasie – tę sytuację nazywa się strefą ostrzeżenia (warning zone).

3 Charakterystyka wybranych do danych

MIT-BIH Arrhythmia Database to szeroko stosowany zestaw danych wykorzystywany w badaniach dotyczących analizy sygnałów elektrokardiograficznych (EKG) oraz automatycznej klasyfikacji arytmii. Zbiór ten zawiera 48 zapisów sygnałów EKG zarejestrowanych u 47 pacjentów, w tym zarówno osób z różnymi typami arytmii, jak i zdrowych. Każdy zapis obejmuje około 30 minut sygnału EKG, który został pobrany w częstotliwości 125 Hz.

Dane zostały ręcznie oznakowane przez kardiologów, co pozwala na identyfikację różnych typów arytmii, takich jak np. skurcze dodatkowe, migotanie przedsionków czy blokady serca. MIT-BIH Arrhythmia Database jest często wykorzystywany w algorytmach sztucznej inteligencji do trenowania systemów do automatycznej diagnozy arytmii.
Kroki używane do ekstrakcji uderzeń z sygnału EKG według autorów DBLP:journals/corr/abs-1805-0079DBLP:journals/corr/abs-1805-007944 były następujące:

1. Podział ciągłego sygnału EKG na okna 10s i wybór jednego okna 10s z sygnału EKG.
2. Normalizacja wartości amplitudy do zakresu od zera do jeden.
3. Znalezienie zbioru wszystkich lokalnych maksimów na podstawie zerokrosów pierwszej pochodnej.
4. Znalezienie zbioru kandydatów na szczyty R EKG poprzez zastosowanie progu 0.9 na znormalizowanej wartości lokalnych maksimów.
5. Znalezienie mediany interwałów czasowych R-R jako nominalnego okresu bicia serca dla danego okna (T).
6. Dla każdego szczytu R wybór części sygnału o długości równej 1.2T.
7. Uzupełnienie każdej wybranej części zerami, aby jej długość była równa zdefiniowanej stałej długości.

Liczba próbek: 21900
Liczba kategorii: 5
Częstotliwość próbkowania: 125Hz

Nazwa klasy	**etykieta**	**liczebność**
Normalne uderzenie	0	1658
Przedwczesne uderzenie nadkomorowe	1	556
Przedwczesny skurcz komorowy	2	1448
Fuzja skurczu komorowego i normalnego uderzenia	3	162
Niezdyscyplinowane uderzenie	4	1608

Dane w celu trenowania klasyfikatorów 1-NN-DTW i 3-NN-DTW zostały okrojeone do liczebności 100 na każdą klasę, przez wysokie wymagania sprzętowe algorytmów.

3.1 Zbiór danych wykorzystywany w Concept drift

Jest to zbiór opisujący wygenerowany ruch siciowy, wykorzystywany jest jako zbiór testowy wykrywania anomali. Zawiera w sobie informację o:

1. Źródle i celu transmisji.
2. Porcie źródłowym oraz celu.
3. Protokół.
4. Rozmiar transmisji i ilość pakietów.
5. Czas trwania transmisji.

4 Eksperymenty i wyniki

4.1 Eksperyment nr 1

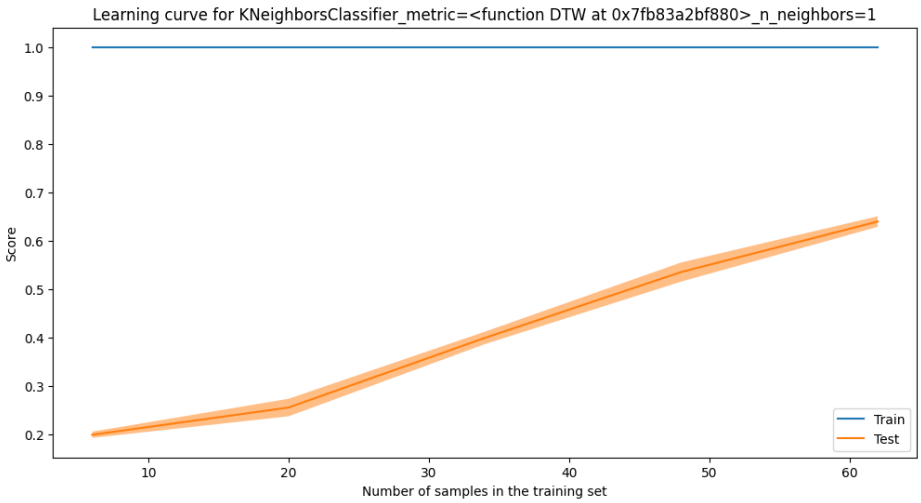
4.1.1 Założenia

Wykonanie klasyfikatora K-nn z metryką DTW dla różnych liczby sąsiadów.

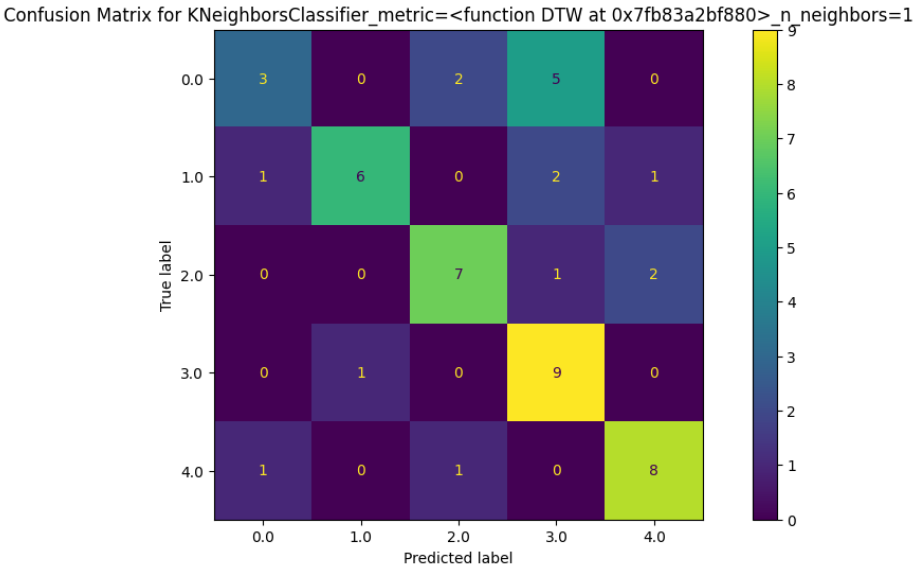
4.1.2 Rezultat 1-NN-DTW i 3-NN-DTW

Classifier	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**
KNeighborsClassifier_metric=DTW_n_neighbors=1	0.660	0.682765	0.660	0.915
KNeighborsClassifier_metric=DTW_n_neighbors=3	0.620	0.622787	0.620	0.905

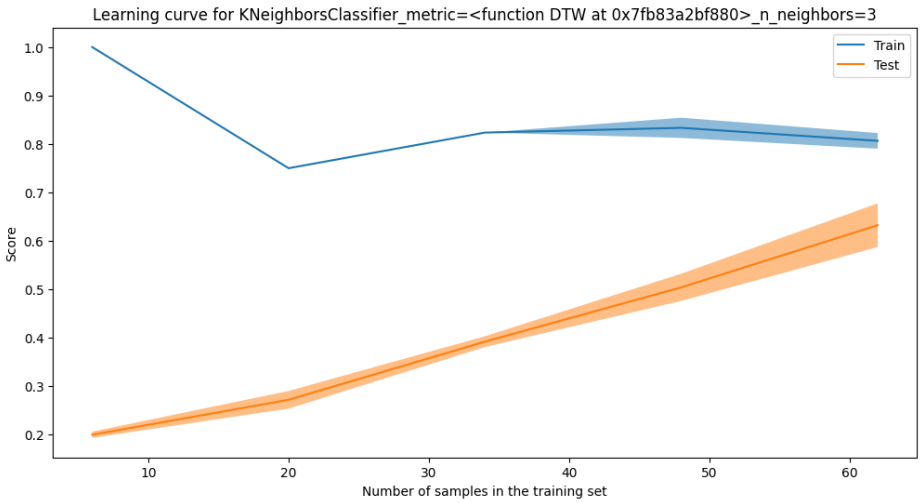
Tabela 1: Statystki dla 1-NN-DTW i 3-NN-DTW



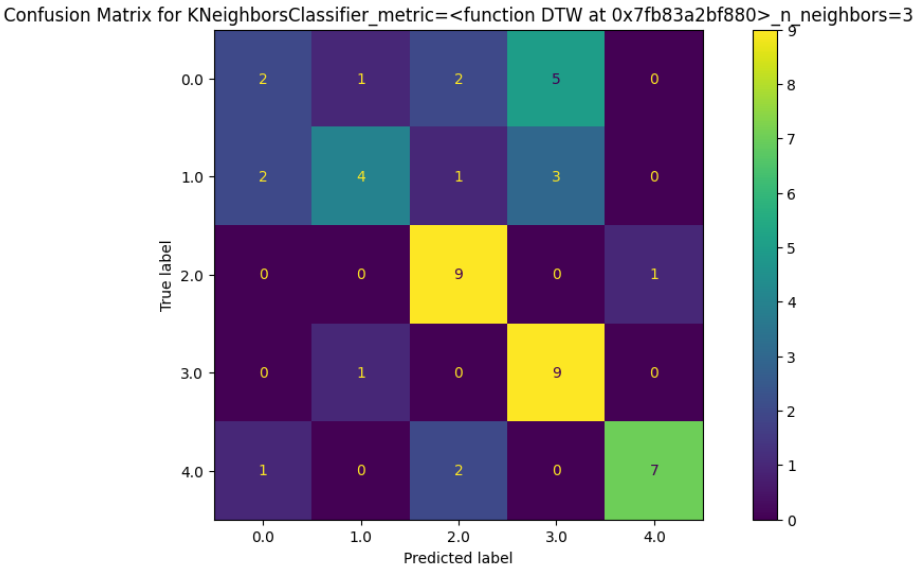
Rysunek 1: Krzywa uczenia dla 1-NN-DTW



Rysunek 2: Macierz pomyłek dla 1-NN-DTW



Rysunek 3: Krzywa uczenia dla 3-NN-DTW

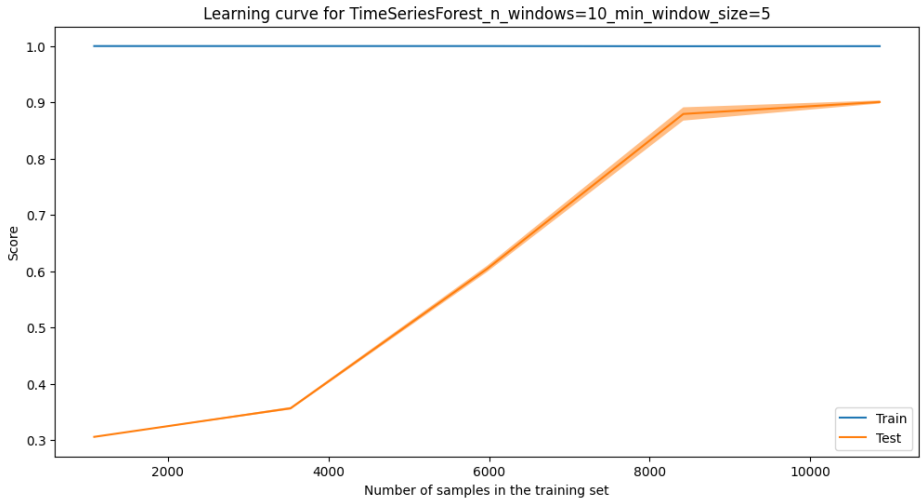


Rysunek 4: Macierz pomyłek dla 3-NN-DTW

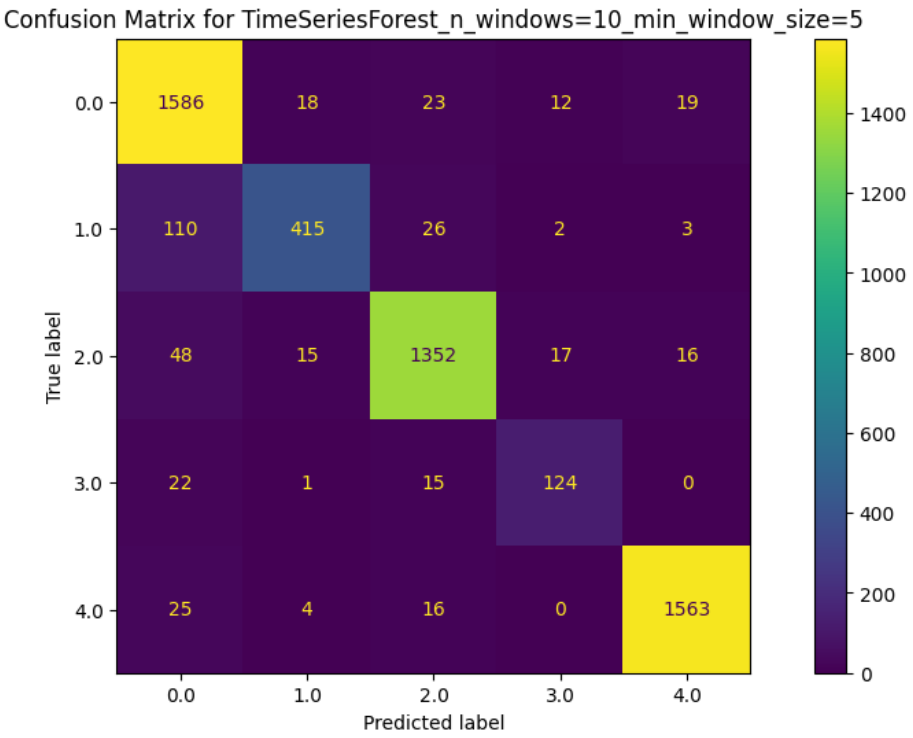
4.1.3 Rezultat TSF

Classifier	Accuracy	Precision	Sensitivity	Specificity
TimeSeriesForest_n_w=1_min_window_size=1	0.712997	0.655248	0.601963	0.921563
TimeSeriesForest_n_w=1_min_window_size=5	0.746870	0.732415	0.698350	0.930439
TimeSeriesForest_n_w=1_min_window_size=10	0.736009	0.712384	0.674085	0.927971
TimeSeriesForest_n_w=5_min_window_size=1	0.870214	0.845206	0.805423	0.964475
TimeSeriesForest_n_w=5_min_window_size=5	0.832474	0.810761	0.775638	0.954112
TimeSeriesForest_n_w=5_min_window_size=10	0.878314	0.856982	0.823460	0.966818
TimeSeriesForest_n_w=10_min_window_size=1	0.856222	0.816359	0.792334	0.960982
TimeSeriesForest_n_w=10_min_window_size=5	0.927835	0.904411	0.874825	0.980398
TimeSeriesForest_n_w=10_min_window_size=10	0.924521	0.908856	0.870622	0.979297

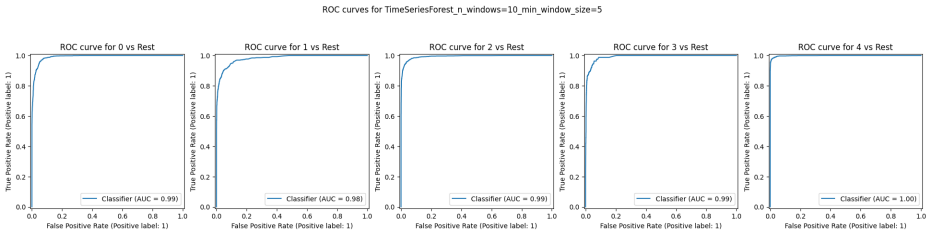
Tabela 2: Statystki dla TimeSeriesForest z różnymi rozmiarami okien



Rysunek 5: Krzywa uczenia dla TSF



Rysunek 6: Macierz pomyłek dla TSF



Rysunek 7: Wykres AUROC dla TSF

4.2 Eksperyment nr 2

Wykorzystania algorytmów detekcji zmian w celu poznania ich działania oraz lepszego poznania wybranych zbioru danych.

4.2.1 Rezultat ADWIN

Numer indeksu wykrycia zmiany	**Na jaką klasę**
Change detected at index 1695	1
Change detected at index 2239	2
Change detected at index 3711	3
Change detected at index 3871	4

Tabela 3: Miejsce i klasa zmiany trendu

4.2.2 Rezultat DDM

Należy pamietać że algorytmy takie jak **DDM** i **EDDM** wykrywają tylko **binarną zmianę**, w tym przypadku musieliśmy wykorzystać drugi zbiór danych.

Numer indeksu wykrycia zmiany
Change detected at index 138
Change detected at index 408
Change detected at index 510
Change detected at index 722

Tabela 4: Miejsce zmiany trendu

5 Wnioski

Wnioski z przeprowadzonych eksperymentów dowodzą, że

- Efektywność klasyfikatorów wykozystających z miary DTW jest niższa w przypadkach większych zbiorów danych posiadających wysoką ilość cech w zbiorze.
- Utrudnione jest trenowanie klasyfikatorów wykorzystujących miarę DTW, ponieważ liczenie tej miary jest wymagającym zadaniem dla procesora przez swoją złożoną implementacje wykorzystuje programowanie dynamiczne.
- Efektywność klasyfikatora TimeSeriesForest jest porównywalny do poprzednie badanych klasyfikatorów, co czyni go skutecznym narzędziem w analizie i klasyfikacji danych sekwencyjnych, pomimo dużej ilości cech w zbiorze. Należy jednak pamiętać o wybraniu w takim przypadku **większego okna kontekstu**.
- Metody detekcji dryfu są w stanie z pewną pewnością określić **concept drift**, w przypadku **ADWIN** wykorzystywane jest okno przesuwne, aby określić ten moment.
- DDM jest skuteczną metodą wykrywania zmiany rozkładu danych.
- Metody detekcji concept drift umożliwają lepiej poznać zbiór danych strumieniowych oraz mogą być wykorzystywane jako triggery poszczególnych operacji w przypadku zmiany trendu.