POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I TELEKOMUNIKACJI

KIERUNEK: CYBERBEZPIECZEŃSTWO

Metody AI w badaniu zagrożeń – Projekt

DoH

AUTORZY:
Dominik Błaszczyk, 249429
Sebastian Słodki, 249307

Spis treści

1. Wstęp

DNS (ang. Domain Name System) [1] jest rozproszonym systemem komputerowym, który przyporządkowuje nazwy domenowe (np. przyklad.pl) do adresów IP (np. 192.0.2.1). Działa jako tłumacz, umożliwiając użytkownikom korzystanie z łatwych do zapamiętania nazw domenowych zamiast pamiętania skomplikowanych adresów IP.

Kiedy użytkownik wpisuje nazwę domenową, aplikacja klienta wysyła zapytanie do serwera DNS. Serwer DNS przeszukuje hierarchiczną strukturę DNS, aby odnaleźć odpowiedni rekord DNS dla danej domeny i zwraca adres IP przypisany do tej nazwy. Dzięki temu, użytkownicy mogą komunikować się z serwerami i urządzeniami w sieci za pomocą znanych im nazw domenowych.

DNS jest istotnym elementem infrastruktury internetowej, umożliwiającym nawigację w Internecie i skuteczne odnajdywanie zasobów sieciowych. Posiada kilka luk bezpieczeństwa, wielokrotnie wykorzystywanych na przestrzeni lat. Nadużywanie DNS zawsze było obszarem dużego zainteresowania badaczy cyberbezpieczeństwa. Jednak zapewnienie bezpieczeństwa i prywatności żądaniom i odpowiedziom DNS jest nadal trudnym zadaniem, ponieważ napastnicy używają wyrafinowanych metodologii ataku, aby wykradać informacje.

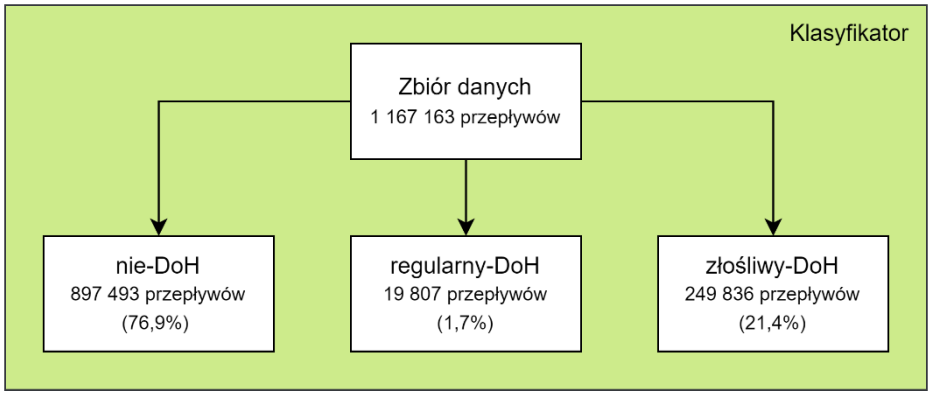
Aby przezwyciężyć niektóre z luk w DNS związanych z prywatnością i manipulacją danymi, IETF wprowadził DNS po HTTPS (DoH) [2] w RFC8484 - protokół, który zwiększa prywatność i zwalcza możliwość podsłuchiwania oraz ataki człowiek-w-środku poprzez szyfrowanie zapytań DNS i wysyłanie ich w zaszyfrowanym kanale.

Podczas gdy przedstawiono wiele przykładów dotyczących aspektów zastosowania i wydajności DoH, badania nad aspektami bezpieczeństwa DoH pozostają ograniczone - tunele DNS były i są szeroko wykorzystywane przez złośliwe oprogramowanie do potajemnego przesyłania danych do i z sieci jednocześnie omijając zapory sieciowe. Wielu badaczy bezpieczeństwa skrytykowało DoH za utrudnianie wykrywania i łagodzenia takich ataków.

Celem projektu jest praktyczne wykorzystanie metod i algorytmów uczenia maszynowego w kontekście danych związanych z użyciem protokołu DoH - ruchu regularnego jak i tego złośliwego, a następnie analiza wybranych algorytmów oraz porównanie ich wyników.

2. Zbiór danych

W zbiorze danych [3] zastosowano dwuwarstwowe podejście do regularnego i złośliwego ruchu DoH wraz z ruchem nie-DoH. W celu wygenerowania reprezentatywnego zbioru danych, ruch HTTPS (DoH i nie-DoH) był generowany poprzez odwiedzenie 10 tysięcy najpopularniejszych stron internetowych użytkowników Alexy Amazona, przy użyciu przeglądarek i narzędzi tunelowania DNS, które obsługują odpowiednio protokół DoH. W pierwszej warstwie, przechwycony ruch jest klasyfikowany jako DoH i non-DoH. W drugiej warstwie, ruch DoH jest charakteryzowany jako regularny DoH i złośliwy DoH.



Rys. 1 Zbiór danych

Zmienne zależne:

- 0 - **nie-DoH**: Ruch generowany przez dostęp do strony, która używa protokołu HTTPS jest przechwytywany i oznaczany jako ruch nie-DoH.

- 1 - **Regularny DoH** to nie-złośliwy ruch DoH generowany przy użyciu tej samej techniki co wspomniany nie-DoH.

- 2 - **złośliwy-DoH**: Narzędzia tunelowania DNS takie jak dns2tcp, DNSCat2 i Iodine były użyte do generowania złośliwego ruchu DoH. Narzędzia te potrafią wysyłać ruch TCP zakodowany w zapytaniach DNS przy pomocy zaszyfrowanych danych DoH.

Wśród zbioru danych można zdefiniować niektóre z cech:

- Współczynnik zmienności - statystyczna miara względnego rozproszenia punktów danych w serii danych wokół średniej. Matematycznie, $CV = \sigma/\mu$, gdzie σ i μ oznaczają odpowiednio odchylenie standardowe i średnią.

- Odchylenie od mediany - mierzy asymetrię rozkładu prawdopodobieństwa zmiennej losowej od rzeczywistej wartości wokół jej mediany.

- Odchylenie od mody - mierzy asymetrię rozkładu prawdopodobieństwa zmiennej losowej od rzeczywistej wartości wokół jej mody.

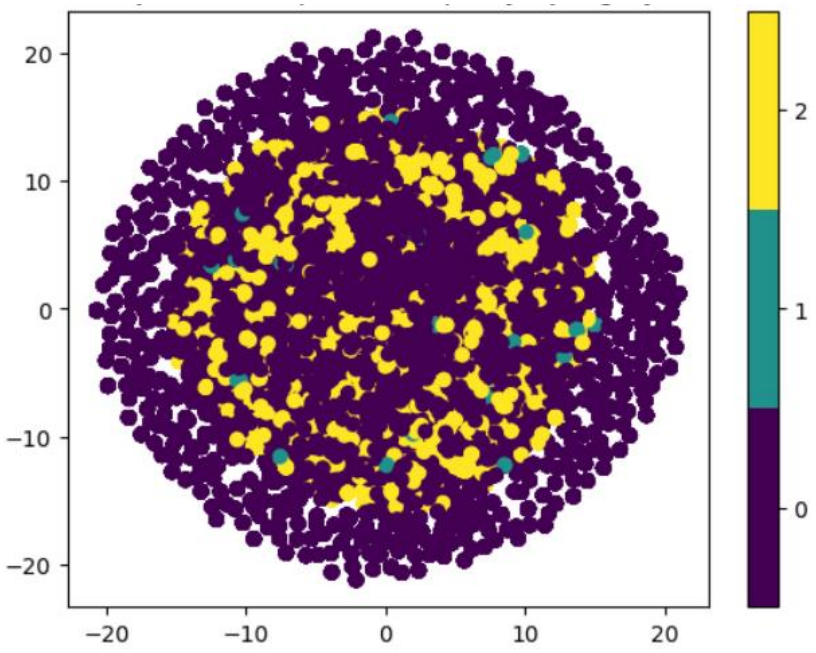
W oparciu o powyższe definicje, te trzy cechy są obliczane dla długości pakietu, czasu pakietu oraz różnicy czasu żądania/odpowiedzi:

- Przesłane bajty przepływu
- Tempo wysyłania przepływu
- Odebrane bajty przepływu
- Tempo odbierania przepływu
- Wariancja długości pakietu
- Odchylenie standardowe długości pakietu
- Średnia długość pakietu
- Mediana długości pakietu
- Modalna długość pakietu

- Odchylenie długości pakietu względem mediany
- Odchylenie długości pakietu względem modalnej
- Współczynnik zmienności długości pakietu
- Wariancja czasu pakietu
- Odchylenie standardowe czasu pakietu
- Średni czas pakietu
- Mediana czasu pakietu
- Modalny czas pakietu
- Odchylenie czasu pakietu względem mediany
- Odchylenie czasu pakietu względem modalnej
- Współczynnik zmienności czasu pakietu
- Wariancja czasu odpowiedzi
- Odchylenie standardowe czasu odpowiedzi
- Średni czas odpowiedzi
- Mediana czasu odpowiedzi
- Modalny czas odpowiedzi
- Odchylenie czasu odpowiedzi względem mediany
- Odchylenie czasu odpowiedzi względem modalnej
- Współczynnik zmienności czasu odpowiedzi

\

Technika redukcji wymiarów, która umożliwia użytkownikom tworzenie wizualizacji zbioru danych. Wykresy mogą być tworzone w dwóch oraz trzech wymiarach. W projekcie wykorzystano technikę UMAP do wizualizacji cech.



Rys. 2 Wizualizacja z wykorzystaniem techniki UMAP

3. Przygotowanie danych

Zbiór danych odpowiednio oznaczono zmiennymi zależnymi oraz usunięto z niego podstawowe informacje, nieistotne dla analizy (źródłowe i docelowe adresy ip wraz z numerami portów, znaczniki czasowe).

```
import pandas as pd
df1 = pd.read_csv("drive/MyDrive/l2-malicious.csv")
df1 = df1.dropna()
df1.columns = df1.columns.str.replace('Label','FlowType')
df1 = df1.replace("Malicious", 2)

df2 = pd.read_csv("drive/MyDrive/l1-nondoh.csv")
df2 = df2.dropna()
df2.columns = df2.columns.str.replace('Label','FlowType')
df2 = df2.replace("NonDoH", 0)

df3 = pd.read_csv("drive/MyDrive/l2-benign.csv")
df3 = df3.dropna()
df3.columns = df3.columns.str.replace('Label','FlowType')
df3 = df3.replace("Benign", 1)

frames = [df1, df2, df3]
df = pd.concat(frames)
df.shape

X = df.drop(df.columns[[0,1,2,3,4,5,34]], axis='columns')
Y = df['FlowType']
Y = Y.astype('int')
X.head()
```

4. Algorytmy

Rozdział opisuje wykorzystane algorytmy do uczenia maszynowego. W skład algorytmów wchodzi: Algorytm Drzewa Decyzyjnego [4], Algorytm Lasu Losowego [5], Algorytm Perceptron Wielowarstwowy [6] oraz Algorytm K-Najbliższych Sąsiadów [7], Wzmocnienie Adaptacyjne[8].

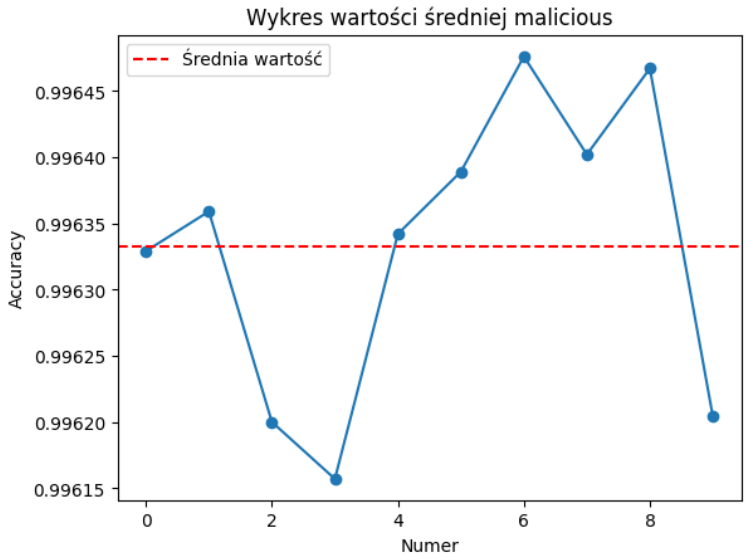
4.1 Algorytm Drzewa Decyzyjnego

Algorytm Drzewa Decyzyjnego [4] to algorytm uczenia maszynowego, który wykorzystuje model podobny do drzewa decyzyjnego lub przewidywań na podstawie cech wejściowych. Rekursywnie dzieli dane na podstawie różnych wartości cech, tworząc gałęzie i węzły w drzewie. Każdy węzeł reprezentuje cechę, a każda gałąź reprezentuje możliwy wynik. Algorytm uczy się na podstawie oznaczonych danych treningowych, aby określić najlepsze podziały i decyzje w każdym węźle, optymalizując dokładność lub inne zdefiniowane metryki. Podczas tworzenia prognoz, algorytm przemierza drzewo od korzenia do węzła liścia, podążając za odpowiednimi gałęziami w oparciu o wartości cech wejściowych i dostarcza odpowiednie wyjście lub przewidywania związane z tym węzłem liścia. Drzewa decyzyjne są

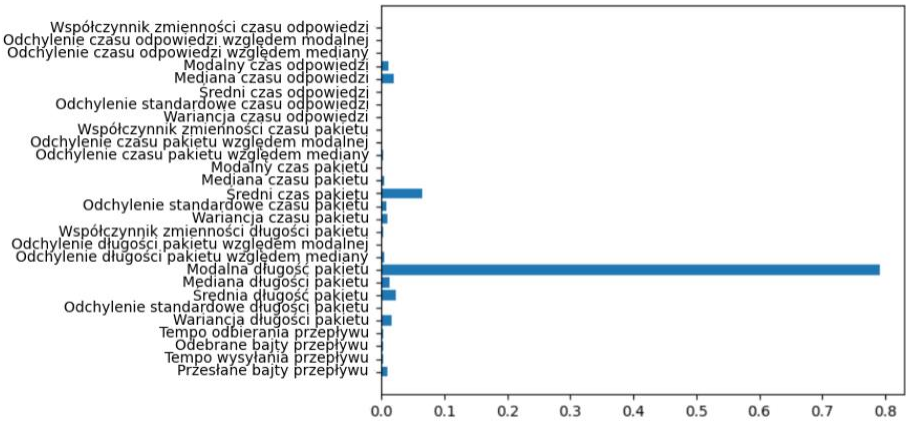
powszechnie stosowane do zadań klasyfikacji i regresji w różnych dziedzinach, zapewniając interpretację i zdolność do obsługi zarówno danych kategorycznych, jak i liczbowych.

Tabela 1. Wyniki dla algorytmu Drzewa Decyzyjnego.

Numer próby	Precyzja	Czas [s]
1	0,996329	82,6184
2	0,996359	82,8867
3	0,996200	79,2304
4	0,996157	82,1111
5	0,996342	89,7878
6	0,996389	83,8418
7	0,996476	81,8864
8	0,996402	81,3428
9	0,996467	84,2928
10	0,996204	84,7265



Rys. 3 Wykresy wartości średniej dla Algorytmu Drzewa Decyzyjnego.



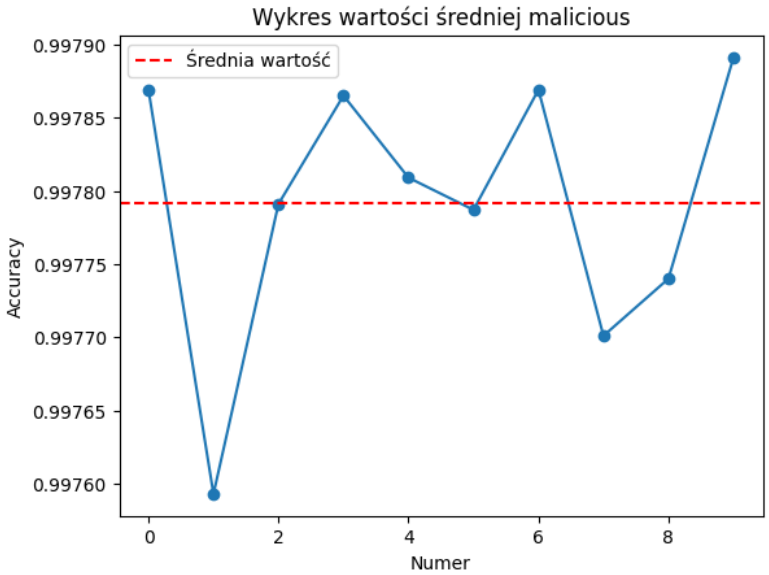
Rys. 4 Wykres słupkowy z ważnością cech DoH dla Drzewa Decyzyjnego.

4.2 Algorytm Lasu Losowego

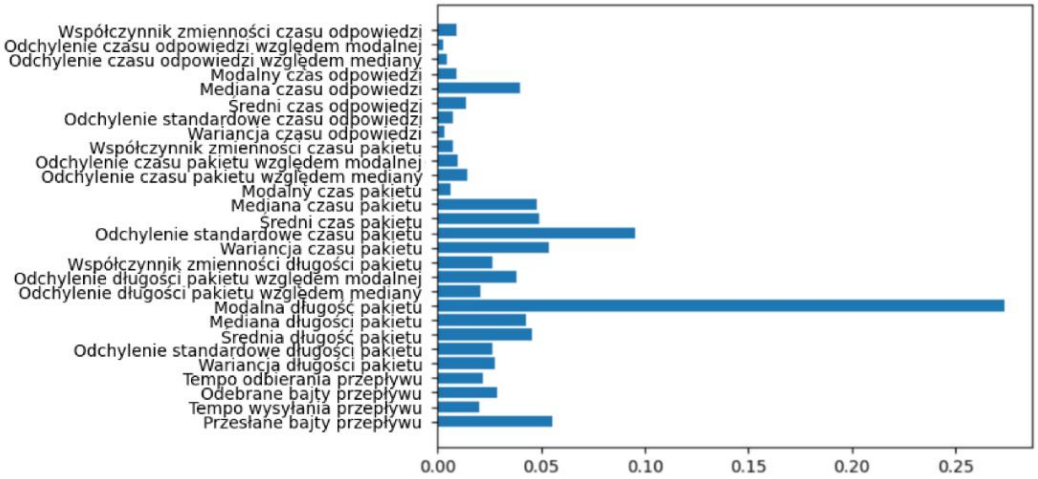
Algorytm Lasu Losowego [5] jest algorytmem uczenia maszynowego, który łączy wiele drzew decyzyjnych, aby dokonać przewidywań lub decyzji. Tworzy on zbiór drzew decyzyjnych, gdzie każde drzewo jest trenowane na losowym podzbiorze danych treningowych i losowych podzbiorach cech wejściowych. Podczas szkolenia, każde drzewo decyzyjne w lesie uczy się niezależnie od innych.
Podczas tworzenia predykcji, każde drzewo w lesie losowym niezależnie generuje predykcję, a ostateczna predykcja jest określana przez agregację predykcji wszystkich drzew. Ta agregacja może być wykonana przez wzięcie większości głosów w problemach klasyfikacji lub uśrednienie przewidywań w problemach regresji.

Tabela 2. Wyniki dla algorytmu Lasu Losowego.

Numer próby	Precyzja	Czas [s]
1	0,997869	819,607
2	0,997593	822,039
3	0,997791	864,931
4	0,997865	828,118
5	0,997809	825,111
6	0,997787	842,579
7	0,997869	854,054
8	0,997701	839,037
9	0,997740	872,946
10	0,997891	848,998



Rys. 5 Wykresy wartości średniej dla Algorytmu Lasu Losowego.



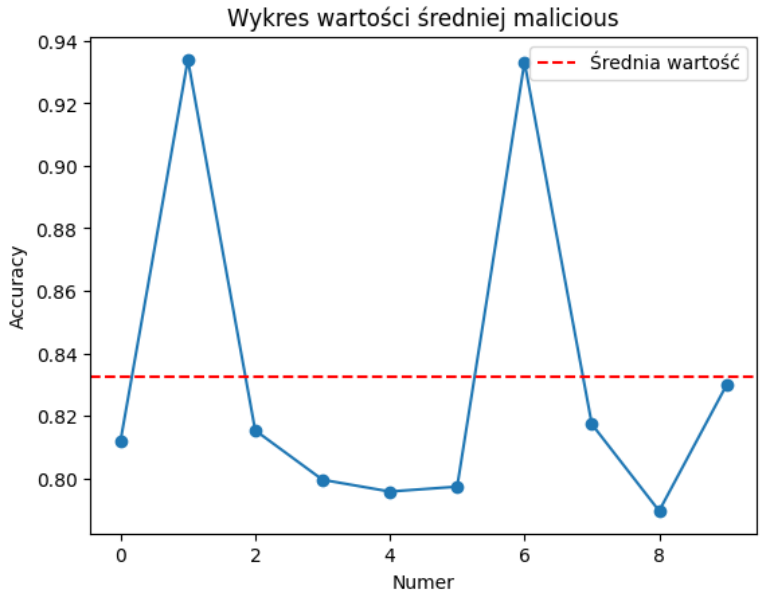
Rys. 6 Wykres słupkowy z ważnością cech DoH dla lasu losowego.

4.3 Algorytm Perceptron Wielowarstwowy

Algorytm MLP [6], znany również jako Perceptron wielowarstwowy, jest rodzajem sieci neuronowej używanej do zadań klasyfikacji w uczeniu maszynowym. Jest on oparty na koncepcji sieci neuronowej, składającej się z wielu warstw połączonych ze sobą sztucznych neuronów lub węzłów.
Algorytm uczy się na podstawie oznaczonych danych treningowych, aby dostosować wagi i uprzedzenia sieci neuronowej, umożliwiając jej dokonywanie przewidywań na niewidzianych danych. Wykorzystuje on proces zwany propagacją wsteczną, w którym błędy w przewidywaniach są propagowane wstecz przez sieć, a wagi są odpowiednio aktualizowane, aby zminimalizować błąd.

Tabela 3. Wyniki dla algorytmu: Perceptron wielowarstwowy.

Numer próby	Precyzja	Czas [s]
1	0,812028	1203,12
2	0,933928	3334,42
3	0,815307	922,272
4	0,799506	936,637
5	0,795731	1465,51
6	0,797297	3437,44
7	0,933134	1743,88
8	0,817291	1029,64
9	0,789541	5287,13
10	0,829762	866,049



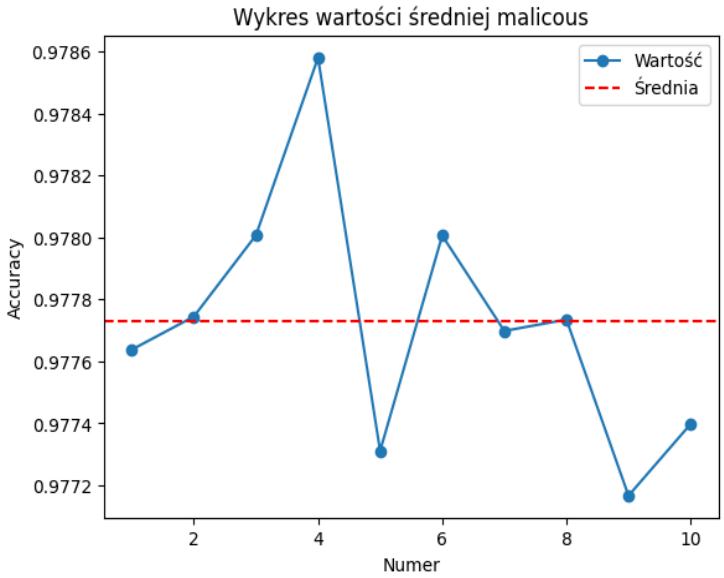
Rys. 7 Wykresy wartości średniej oraz odchylenia standardowego dla Algorytmu Perceptron Wielowarstwowy.

4.4 Algorytm K-Najbliższych Sąsiadów

Algorytm KNN - K-nearest neighbors algorithm [7] (K-Najbliższych Sąsiadów) uczenia maszynowego, stosowany zarówno w zadaniach klasyfikacji, jak i regresji. Jego podstawą jest założenie, że podobne dane powinny znajdować się blisko siebie w przestrzeni cech. Jest to tak zwany algorytm oparty na technice uczenia nadzorowanego. KNN to algorytm nieparametryczny, oznacza to, że nie przyjmuje żadnych założeń dotyczących danych.

Tabela 4. Wyniki dla algorytmu K-Najbliższych Sąsiadów.

Numer próby	Precyzja	Czas [s]
1	0,977637473	717,515
2	0,977744273	392,510
3	0,978006641	259,800
4	0,978579469	366,561
5	0,977310839	241,835
6	0,978005651	295,464
7	0,977697931	242,172
8	0,977734731	239,552
9	0,977166329	244,798
10	0,977397507	244,641



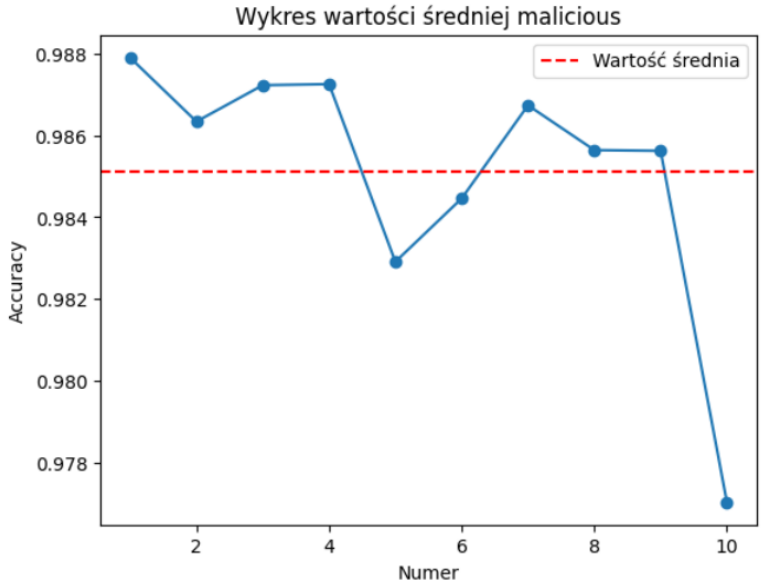
Rys. 8 Wykresy wartości średniej oraz odchylenia standardowego dla Algorytmu K-Najbliższych Sąsiadów.

4.5 Wzmocnienie Adaptacyjne

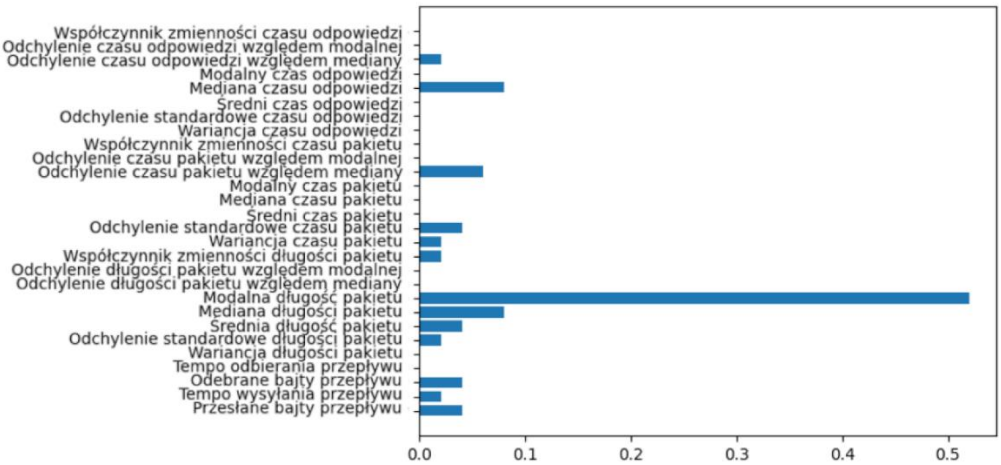
Algorytm AdaBoost [8] łączący ze sobą słabe klasyfikatory do stworzenia silnego klasyfikatora. Działa on w sposób iteracyjny, uczy on słabe klasyfikatory na różnych podzbiorach ważonych danych treningowych.

Tabela 5. Wyniki dla algorytmu Wzmocnienie Adaptacyjne

Numer próby	Precyzja	Czas [s]
1	0,987896	368,391
2	0,986334	383,078
3	0,987223	375,891
4	0,987249	392,688
5	0,982905	393,375
6	0,984454	397,328
7	0,986731	397,141
8	0,985636	378,703
9	0,985618	394,609
10	0,977021	368,078



Rys. 9 Wykresy wartości średniej oraz odchylenia standardowego dla Algorytmu Wzmocnienie Adaptacyjne.



Rys. 10 Wykresy wartości średniej oraz odchylenia standardowego dla Algorytmu Wzmocnienia Adaptacyjnego.

Podsumowanie

Podczas trwania projektu „Metody AI w badaniu zagrożeń” do analizy danych ruchu DNS over HTTPS wykorzystano pięć następujących algorytmów: Drzewo Decyzyjne, Las Losowy, Perceptron Wielowarstwowy, K-Najbliższych Sąsiadów oraz Wzmocnienie Adaptacyjne. Analizując wyniki precyzji algorytmów zebrane w tabeli 6, 7 oraz na podstawie powyższych wykresów można stwierdzić, że algorytmy Drzewo Decyzyjne oraz Las Losowy posiadają najwyższy poziom precyzji, oba uzyskały accuracy na poziomie 0.99. K-Najbliższych Sąsiadów, Wzmocnienie Adaptacyjne uzyskały wyniki między 0.97 – 0.98. Najgorzej wypadł Perceptron wielowarstwowy z wynikiem 0,83. Analiza wykresów wartości średniej pokazuje, że każdy z algorytmów podczas wykonywania prób posiada wyniki zbliżone do ich najlepszych wyników. Odchylenie standardowe w przypadku wszystkich algorytmów jest poniżej jednej tysięcznej wartości. Sugeruje to, że każdy z nich działa w stabilny i spójny sposób. Analiza najważniejszych cech wykazuje, że najbardziej dominującą z nich potrzebną jest „modalna długość pakietu” i to przy jej pomocy najlepiej algorytmy dokonują detekcji złośliwego DoH. Ostatnim ważnym parametrem jest czas nauki, w przypadku wykorzystanych w projekcie algorytmów najkorzystniej prezentuje się Drzewo Decyzyjne z wynikiem średnim 83 sekund. Na podstawie zebranych danych o pięciu testowanych algorytmach można wywnioskować, że najkorzystniej w przypadku analizy danych DoH wypada Drzewo Decyzyjne. Dzięki niemu wykrycie zagrożeń ma wysoki poziom powodzenia. W przypadku Decision Tree istotne jest też, że uczy się najszybciej wśród badanych algorytmów co może pozwolić na zaoszczędzenie zasobów.

Tabela 6. Zbiorcza tabela wyników dla wszystkich testowanych algorytmów z wartością średnią każdego z nich.

Numer próby	Drzewo Decyzyjne	Las Losowy	Perceptron Wielowarstwowy	K-Najbliższych Sąsiadów	Wzmocnienie Adaptacyjne
1	0,996329	0,997869	0,812028	0,97763747	0,987896
2	0,996359	0,997593	0,933928	0,97774427	0,986334
3	0,996200	0,997791	0,815307	0,97800664	0,987223
4	0,996157	0,997865	0,799506	0,97857947	0,987249
5	0,996342	0,997809	0,795731	0,97731084	0,982905
6	0,996389	0,997787	0,797297	0,97800565	0,984454
7	0,996476	0,997869	0,933134	0,97769793	0,986731
8	0,996402	0,997701	0,817291	0,97773473	0,985636
9	0,996467	0,997740	0,789541	0,97716633	0,985618
10	0,996204	0,997891	0,829762	0,97739751	0,977021
średnia	**0,9963325**	**0,9977915**	**0,832352**	**0,97772808**	**0,9851067**

Tabela 7. Zbiorcza tabela wyników dla wszystkich testowanych algorytmów z wartością średnią precyzji i czasu nauki każdego z nich.

	Czas nauki [s] (średnia)	Precyzja (średnia)
Drzewo Decyzyjne	83,2725	0,996333
Las Losowy	841,742	0,997791
Perceptron wielowarstwowy	2022,61	0,832352
K-Najbliższych sąsiadów	324,484	0,977728
Wzmocnienie Adaptacyjne	384,928	0,985106

Bibliografia

[1] “Domain Name System” https://pl.wikipedia.org/wiki/Domain_Name_System
[2] “DNS over HTTPS” https://en.wikipedia.org/wiki/DNS_over_HTTPS
[3] „CIRA-CIC-DoHBrw-2020” https://www.unb.ca/cic/datasets/dohbrw-2020.html
[4] “Decision Tree Classifier” https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
[5] “Random Forest Classifier” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[6] “MLP Classifier” https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
[7] „K-Nearest Neighbor(KNN) Algorithm for Machine Learning”%20https:/www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning
[8] "Master the AdaBoost Algorithm: Guide to Implementing & Understanding AdaBoost” https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/
[9] Analiza ruchu DoH – Google Collab (kod) https://colab.research.google.com/drive/1f3RP--3sFmZyGGkQyJiTlj6aLswoGWIg?usp=sharing