

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Projekt dyplomowy

*Wykrywanie anomalii w ruchu sieciowym z wykorzystaniem algorytmów populacyjnych*

# Anomaly detection in network traffic using population algorithms

Autor: *Zofia Jankowska*

Kierunek studiów: *Automatyka i Robotyka*

Opiekun pracy: *dr hab. inż. Joanna Kwiecień*

## Kraków, 2024

Spis treści

[Wstęp 2](#_Toc148870250)

[Algorytmy w tej detekcji 2](#_Toc148870251)

[Realizacja aplikacji 4](#_Toc148870252)

[Testy 5](#_Toc148870253)

[Podsumowanie 5](#_Toc148870254)

[Bibliografia 5](#_Toc148870255)

# Wstęp

To jakieś pierdoły

2 strony, cel i założenia projektu, struktura pracy

Celem jest implementacja i porównanie działania algorytmu genetycznego oraz PSO w celu wykrywania anomalii w ruchu sieciowym na podstawie bazy danych KDDCUP’99.

Poszczególne rekordy z bazy będą klasyfikowane jako połączenie normalne lub atak (czy dzielimy na 4 różne główne typy ataków)

Trenowanie algorytmów zostanie wykonane na podstawie zestawu treningowego 10% całej bazy, a następnie przeprowadzona zostanie analiza działania na podstawie całego 100% bazy.

Wykonany zostanie interfejs graficzny przedstawiający przebieg uczenia oraz oraz sprawdzenie efektywności wyznaczonych reguł.

Wykonane zostaną testy statystyczne: test Wilkoxona (rozwinięcie testu t-studenta).

**Opis problemu anomalii**

Ta ładna definicja o tym że anomalie to punkty tak inne że są podejrzenia że jest to punkt normalny generowany przez inny zestaw reguł.

# Algorytmy w tej detekcji

Opis ga i pso i adaptacja do problemu, state of art, jakie jeszcze mogą być anomalie w sieciach – nie tylko włamania

**State of art**

**Algorytm genetyczny**

Ze względu na popularność problemu wykrywania anomalii w ruchu sieciowym, stosowane rozwiązania ciągle ewoluują, a na ich temat powstały niezliczone publikacje.

W artykule A. Goyala oraz Ch. Kumara „GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System” przedstawiono takie rozwiązania:

W celu zakodowania poszczególnych połączeń, wartości w formacie string, zostały przedstawione za pomocą liczb całkowitych zaczynających się od 0, a każdemu nowemu typowi ciągu przypisano zwiększoną liczbę. Genotypy stworzono z przeliczonych wartości, a te pola, które miały format liczbowy zostały włączone w sposób niezmieniony.

Do oceny osobników, skorzystali oni z funkcji ewaluacyjnej, w której wyznacza się różnicę między odsetkiem poprawnie zaklasyfikowanych ataków, a połączeń normalnych. W ten sposób, jeżeli wartość tej funkcji była większa od 0, osobnik poprawnie klasyfikował więcej ataków niż połączeń normalnych, co było działaniem pożądanym. Te osobniki, dla których funkcja ewaluacyjna wynosiła więcej niż 0,95, były wybierane do produkcji kolejnych pokoleń.

Poddawane krzyżowaniu były pokolenia składające się 60 najlepszych osobników, których 40 najlepszych było jeszcze dodatkowo kopiowanych w celu polepszenia jakości populacji.

Autorzy zastosowali krzyżowanie jednopunktowe, gdzie dochodziło do niego tylko na granicy dwóch pól, a nie między nimi. Mutowano natomiast pojedyncze pola, w ich zakresie wyznaczonym na początku eksperymentu, ze współczynnikiem mutacji ustawionym na 1%.

W artykule Wei Li „Using Genetic Algorithm for Network Intrusion Detection” zastosowano odmienne podejście.

Do kodowania pojedynczych osobników wykorzystano 9 atrybutów: źródłowy oraz docelowy adres IP, numer portu źródłowego i docelowego, czas trwania, stan, protokół oraz liczbę wysłanych bajtów przez autora i respondera. Rozdzielono wszystkie wartości liczbowe, tak aby jeden gen był jedną cyfrą.

W celu obliczenia wartości funkcji oceny, ogólny wynik wyznaczony na podstawie tego czy pole połączenia pasuje do wstępnie sklasyfikowanego zestawu danych mnożono przez wagę tego pola.

Wagi zostały podzielone na kategorie związane z poszczególnymi atrybutami. Przykładowo – wszystkie geny reprezentujące pole źródłowego adresu IP mają taką samą wagę. Wagi te przypisano zgodnie z taką kolejnością: docelowy adres IP, źródłowy adres IP, numer portu docelowego, czas trwania, liczba bajtów wysłanych przez autora, przez respondera, stan, protokół, oraz numer portu źródłowego.

Bezwzględna różnica miedzy wynikiem, a rzeczywistym poziomem podejrzenia (próg, wskazujący w jakim stopniu dwa połączenia sieciowe są „pasujące” jest obliczana i używana do wyznaczenia wartości kary.

Gdy wystąpi niezgodność, wartość kary jest obliczana na podstawie bezwzględnej różnicy. Jest on mnożony razy ranking (wskazuje czy włamanie jest łatwe do zidentyfikowania) i dzielone przez 100. Końcowa wartość funkcji ewaluacyjnej chromosomu jest obliczona jako 1- kara.

Ze względu na to, że jedna reguła nie jest w stanie jednoznacznie określić czy dane połączenie jest atakiem czy nie, poszukiwany jest zestaw reguł. Autorzy dążyli do znalezienia wielu lokalnych maksimów (wystarczająco dobrych rozwiązań). Wykorzystano do tego metody *nichingu.* Metodami nichingu są *crowding* i *sharing. Crowding* wykorzystuje najbardziej podobnego członka do zastąpienia, aby spowolnić populację do zbiegnięcia się do jednego punktu w kolejnych pokoleniach. Polega to na parowaniu każdego osobnika poza populacją z podobnym osobnikiem w bieżącej populacji (*faza parowania)* i decydowaniu, który z nich pozostanie w populacji *(faza zastępowania)* .[9]  *Sharing* zmniejsza fitness osobników, które mają bardzo podobnych członków i nagradza te, które przeszukują przestrzeń w mniej zaludnionych przestrzeniach. Dzięki temu istnieje możliwość wyznaczania większej ilości lokalnych maksimów. [10]

Mutacja wykonana została w taki sam sposób jak w artykule powyżej.

W artykule “An effective genetic algorithm-based feature selection method for intrusion detection systems”, w celu zakodowania chromosomów dokonano normalizacji poszczególnych genów. Tutaj również zastosowanie metodę 1-z-n w celu konwersji wartości tekstowych na liczby całkowite. Każdy chromosom w populacji reprezentuje jedno rozwiązanie do problemu selekcji. (nie rozumiem do końca jak to działa).

Funkcja ewaluacyjna chromosomów bazuje na korelacji między wybranymi cechami. Dzięki temu znajdowane są NIE ROZUMIEM AAAAAAA

Jako metodę selekcji zaproponowano ruletkę z wagami – im lepszy wynik funkcji ewaluacyjnej uzyskał dany osobnik, tym więcej razy pojawia się on na kole ruletki.

Wykorzystano krzyżowanie jednopunktowe, z uściśleniem jednak, że z dwóch rodziców powstawał tylko jeden osobnik.

Jako współczynnik mutacji testowano prawdopodobieństwa 0,25, 0,5 oraz 0,75 i najlepsze wyniki osiągnięto dla 0,5. Zastosowano podejście, że mutowany jest rodzic, a nie potomek. Ilość cech, która jest mutowana oraz wartości im przypisywane są losowe.

Autorzy, do klasyfikacji poszczególnych chromosomów wykorzystują algorytm k-NN.

Praca „Particle Swarm Optimalisation for Outlier Detection” mówi o podejściu do wykrywania anomalii w zbiorach z wykorzystaniem algorytmu optymalizacji rojem cząstek. Podejście autorów opiera się na mierze wykorzystującej odległość.

Aby zwizualizować swój pomysł nanieśli oni przykładowe punkty na wykres oraz nakreślili okręgi o promieniu *r* na punkty, które mają potencjał na bycie anomaliami. Poszukiwane są punkty mające *k* punktów w obrębie okręguprzy jednoczesnej próbie minimalizowania współczynnika *k/r* . DANO TU FAJNĄ DEFINICJĘ PUNKTÓW ODSTAJĄCYCH KTÓRĄ MOŻNA POTENCJALNIE WYKORZYSTAĆ.

# Realizacja aplikacji

**Opis bazy danych i preprocessingu**

Opis bazy danych

W ramach preprocessingu – odpowiednia klasyfikacja rekordów :

- Zmiana danych tekstowych na 1 z n.

- Wyznaczenie wartości minimalnych i maksymalnych w każdej kolumnie

- należy jeszcze sprawdzić czy każda kolumna z tych 41 jest rzeczywiście potrzebna (w genetic\_network zastosowano tylko 9, genetic\_intrusion 8)

- zakodowanie odpowiednie adresów IP – rozbicie na 4 wartości?

**Algorytm genetyczny**

- kodowanie jako tablica

- osobniki będą miały zastosowane uogólnienie w postaci dwóch ostatnich wartości oznaczonych xxx.xxx (w genetic\_network jako -1)

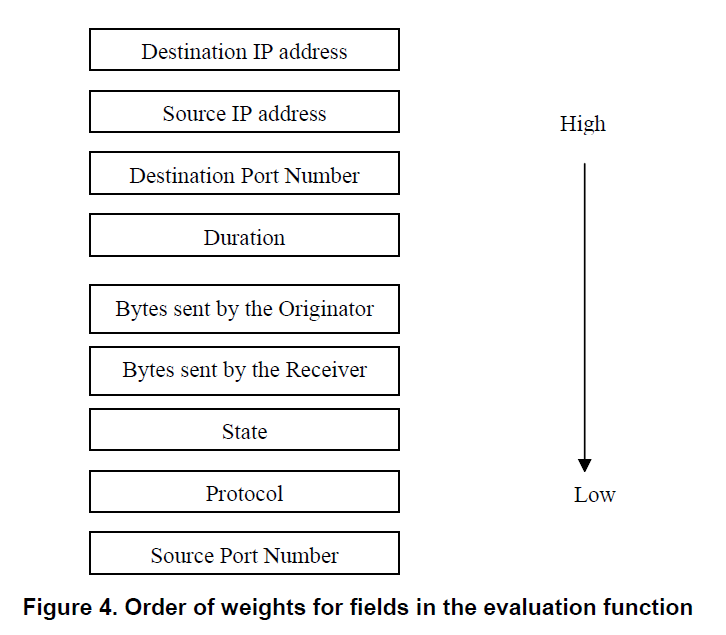
- generowanie losowej populacji początkowej (100-200 osobników)

- krzyżowanie jednopunktowe – po prostu tniemy i zamieniamy (możliwe że w 2 miejscach)

- mutacja – jedno z pól zmienia swoją wartość na wylosowaną z wyznaczonego w trakcie preprocessingu dla zakresu wartości

- funkcja ewaluacyjna – ile dany potomek poprawnie zaklasyfikuje anomalii w zbiorze testowym, ale z wykorzystaniem funkcji F = a/A – b/B, gdzie a i b to poprawnie zaklasyfikowane ataki i nie, a A i B to ilość ataków i nie w zbiorze treningowym. W ten sposób jeśli F > 0 to znaczy że poprawnie sklasyfikowano więcej ataków niż normalsów.

- funkcja ewaluacyjna może też wykorzystywać wagi dla mismatchów pojedynczych pól zgodnie z tabelą wag – genetic\_network



- no i dalej schody, bo podejrzewam że jedna reguła nie przyniesie bardzo widowiskowych efektów, dlatego nie wiem czy nie użyć kilku na raz

- selekcja - w artykule genetic\_intrusion wybierano 60 najlepszych potomków i 40 najlepszych jeszcze z nich jest kopiowane. W artykule effective… metoda selekcji to ruletka z wagami. Im lepsza funkcja ewaluacyjna tym więcej pól na ruletce zajmuje.

- kryterium stopu – funkcja fitness nie rośnie po n iteracjach

**Algorytm PSO**

# Testy

# Podsumowanie

# Bibliografia

<https://www.kaggle.com/code/abhaymudgal/intrusion-detection-system>

1. Goyal, A., Kumar, Ch., GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System

<https://www.kaggle.com/code/limamateus/kdd1999-preprocessing>

1. Tavallaee, M., Bagheri, E., Lu, Wei, Ghorbani, A., 2009, A detailed analysis of the KDD CUP 99 data set
2. Wei, Li, Using Genetic Algorithm for Network Intrusion Detection
3. Mohemmed, A., Zhang, M., Browne, W., 2010, Particle Swarm Optimisation for Outlier Detection
4. Harshit, S., Vineet, R., 2014, Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain
5. <https://pandas.pydata.org/>
6. <https://numpy.org/doc/stable/>
7. <https://docs.python.org/3/library/tkinter.html>
8. Galan, S.F., Mengshoel, O.J., 2010, Generalized Crowding for Genetic Algorithms
9. Miller, B.L., Shaw, M.J., 2002, Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimalization
10. Crawford, K.D., Wainwright, R.L., 1995, Applying Genetic Algorithms to Outlier Detection
11. Halim, Z., Yousaf, M.N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., Ahmad, I., Hanif, M., 2021, An effective genetic algorithm-based feature selection method for intrusion detection systems