# Projekt zespołowy badawczy

Ćwiczenie numer: 3

Temat: **Dokumentacja realizacji projektu**

Politechnika Białostocka

Wykonujący ćwiczenie:

**- Jakub Kowalewski**

Studia dzienne

Kierunek: Informatyka, II stopnia

Semestr: II

Grupa zajęciowa: PS 3

Prowadzący ćwiczenie: dr inż. Tomasz Grześ

Data wykonania ćwiczenia:
29.01.2025 r.

Min. 5 stron przedstawiających proces realizacji projektu: metodyka badawcza, zbiory danych, narzędzia, algorytmy, wyniki badań w postaci zarówno tabel, jak i wykresów, wnioski częściowe i ogólne.

Proces realizacji projektu „Analiza skuteczności algorytmów sztucznej inteligencji w rozpoznawaniu sposobów szyfrowania plików”

## 1. Metodyka badawcza

Celem projektu jest analiza skuteczności algorytmów sztucznej inteligencji w klasyfikacji metod szyfrowania plików. Proces realizacji badań obejmuje:

- Wygenerowanie różnego rodzaju plików w formatach takich jak: txt, png, pdf, docx, xlsx (wygenerowano 154 pliki),
- Zaszyfrowanie wygenerowanych plików różnymi metodami według ustalonych wcześniej proporcji – AES256 (35%), RSA (10%), DES (20%) i ChaCha20 (35%),
- Przygotowanie zbioru danych zawierającego cechy plików zaszyfrowanych różnymi metodami (entropia Shannon’a, średnia, odchylenie standardowe, skośność, kurtoza, rozmiar pliku, rozmiar pliku modulo 16, test chi-kwadrat, histogram bajtów),
- Wstępną analizę i przetworzenie danych,
- Trening oraz ewaluację kilku algorytmów sztucznej inteligencji,
- Analizę wyników i wyciągnięcie wniosków.

Do oceny skuteczności modeli wykorzystano miary takie jak dokładność (accuracy), precyzja (precision), czułość (recall) i średnia F1-score.

## 2. Zbiór danych

Do eksperymentu wykorzystano zbiór danych zawierający cechy plików szyfrowanych przy użyciu następujących algorytmów:

- AES-256 (Advanced Encryption Standard),
- RSA (Rivest-Shamir-Adelman),
- DES (Data Encryption Standard),
- ChaCha20.

Dane zostały podzielone na zbiór treningowy (80%) i testowy (20%) z zachowaniem proporcji klas.

## 3. Narzędzia

Do realizacji projektu wykorzystano:

- Język Python,
- Bibliotekę pandas,
- Bibliotekę numpy,
- Bibliotekę matplotlib,
- Bibliotekę pycryptodome,
- Bibliotekę PIL,
- Bibliotekę fpdf,
- Bibliotekę docx,
- Bibliotekę sklearn.

Scikit-learn była źródłem modeli sztucznej inteligencji, z których korzystano w ramach przeprowadzenia eksperymentu.

Za pomocą bibliotek PIL, fpdf i docx wygenerowano odpowiednio pliki graficzne png, dokumenty tekstowe pdf, dokumenty w formacie docx i xslx.

Za pomocą biblioteki pycryptodome dokonano zaszyfrowania wygenerowanych wcześniej plików.

## 4. Algorytmy

W eksperymencie przetestowano następujące modele uczenia maszynowego:

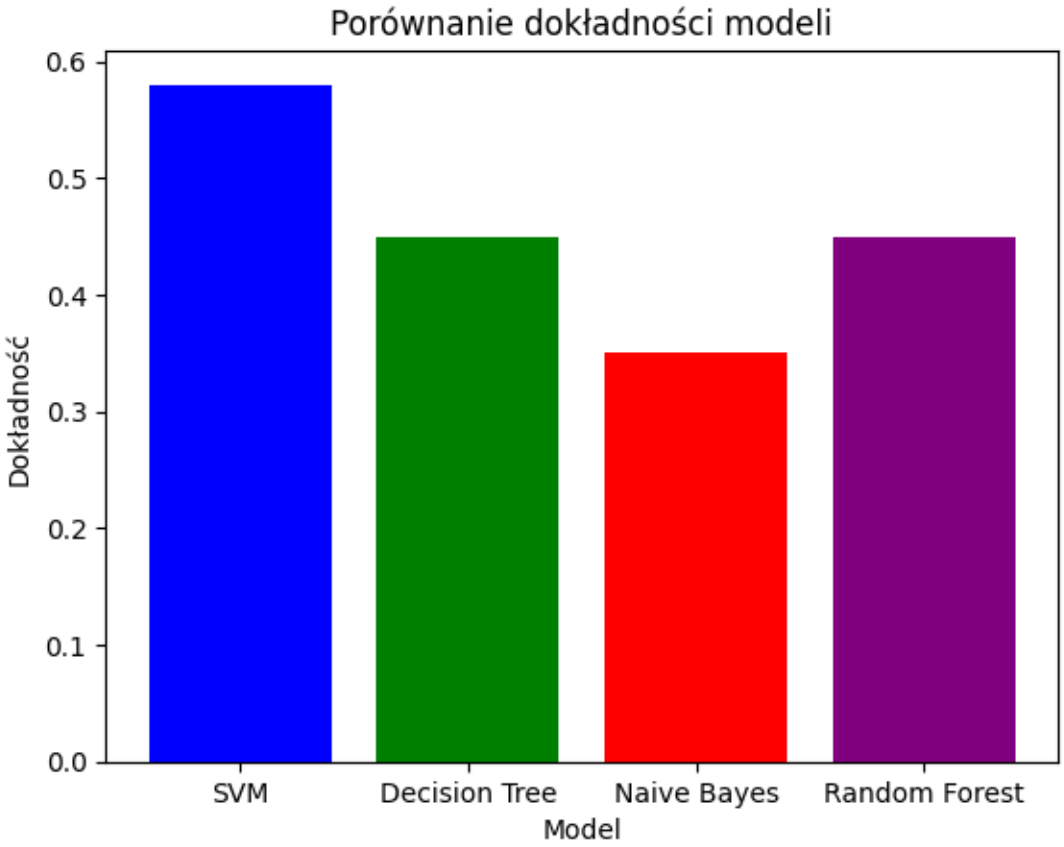
- SVM (Support Vector Machine) z jądrem RBF,
- Drzewo decyzyjne (Decision Tree),
- Naive Bayes,
- Las Losowy (Random Forest).

## 5. Ogólne wyniki badań

Poniższa tabela przedstawia uśrednione wyniki dokładności dla poszczególnych modeli:

| Model | Dokładność |
|---|---|
| SVM | 58.06% |
| Decision Tree | 45.16% |
| Naive Bayes | 35.48% |
| Random Forest | 45.16% |

Wykres uśrednionych wyników:




## 6. Wyniki klasyfikacji poszczególnych metod szyfrowania

AES-256

- Najlepszy model: SVM (dokładność ~ 82%)
- Interpretacja: AES-256 był najlepiej rozpoznawalnym algorytmem. Może to wynikać z charakterystycznych cech w ekstraktowanych parametrach, np. rozkład entropii lub wzorzec bitowy w plikach zaszyfrowanych AES.

- Problemy: Modele mogły mylić AES z innymi szyframi blokowymi (np. DES), ale ogólna skuteczność klasyfikacji była wysoka.

ChaCha20

- Najlepszy model: SVM (~ 55% recall, 50% precision)
- Interpretacja: ChaCha20, jako szyfr strumieniowy, może mieć mniej wyraźne cechy w porównaniu do szyfrów blokowych. Z tego powodu klasyfikacja jest trudniejsza.
- Problemy: Algorytm był często mylony z AES lub danymi plain, co sugeruje podobieństwa w generowanych cechach.

DES

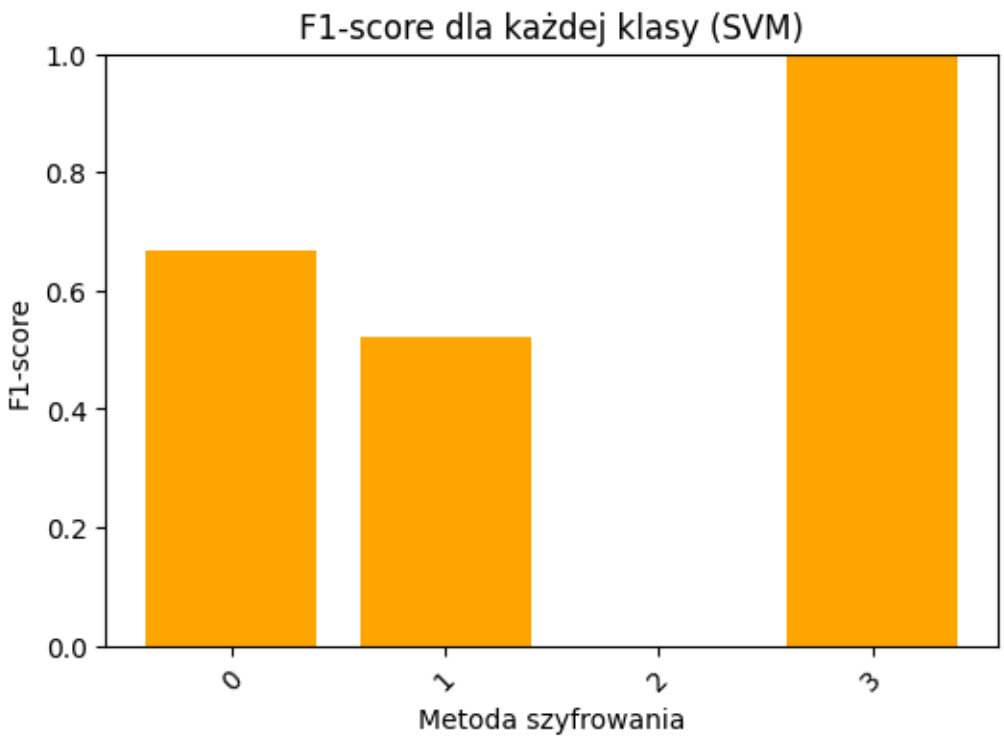
- Najlepszy model: Random Forest (~ 33% recall, 25% precision)
- Interpretacja: DES był najsłabiej klasyfikowanym algorytmem. Może to wynikać z jego mniejszego rozmiaru klucza i struktury blokowej, co generuje bardziej chaotyczne wzorce cech.
- Problemy: Modele myliły DES z AES-256 oraz plain, co wskazuje na problemy z wyodrębnieniem cech charakterystycznych dla tego algorytmu.

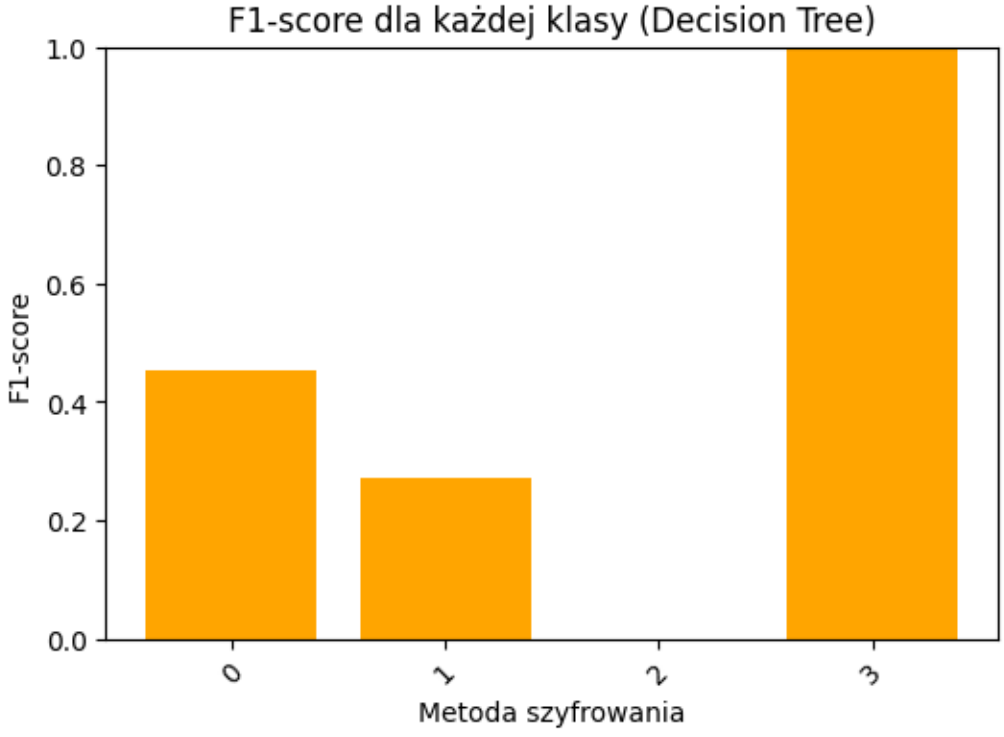
RSA

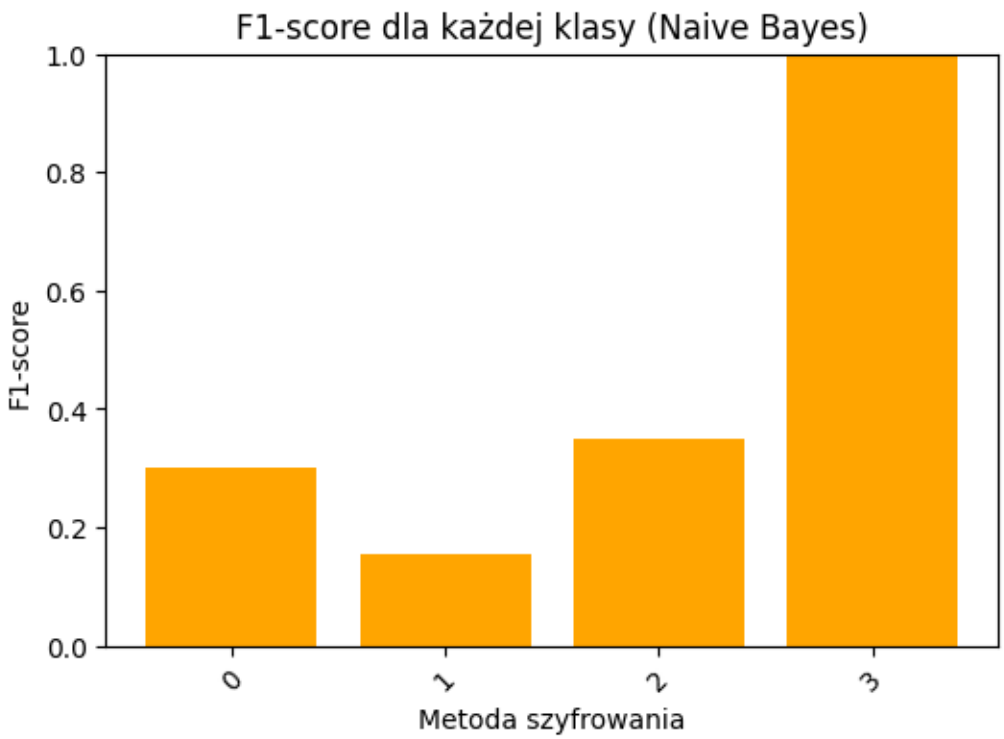
- Najlepszy model: Wszystkie (~ 100% recall, 100% precision)
- Interpretacja: RSA był klasyfikowany prawidłowo w każdym przypadku. Wynika to prawdopodobnie z dużej różnicy między szyfrowaniem asymetrycznym a symetrycznym.
- Problemy: Brak.

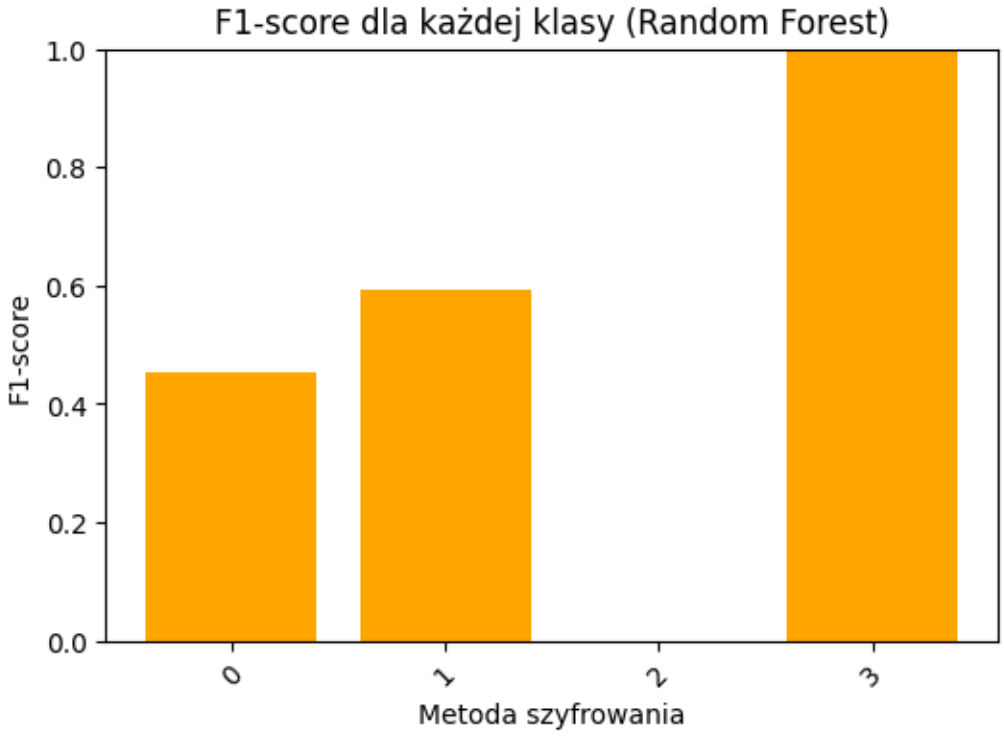
Wnioski częściowe

- SVM najlepiej klasyfikuje AES-256 i ChaCha20, ale ma problem z DES.
- Random Forest ma lepsze wyniki dla DES, ale ogólnie jest mniej skuteczny niż SVM.
- RSA jest rozpoznawane idealnie, jednak wynika to głównie z różnicy w charakterystyce szyfrowania asymetrycznego.
- DES i ChaCha20 są najtrudniejsze do klasyfikacji, co sugeruje konieczność ulepszenia cech wejściowych lub zmiany modelu.


F1-score dla każdej klasy (SVM)
1.0
0.8
0.6
0.4
0.2
0.0
F1-score
0
1
2
3
Metoda szyfrowania


F1-score dla każdej klasy (Decision Tree)
1.0
0.8
0.6
0.4
0.2
0.0
F1-score
0
1
2
3
Metoda szyfrowania




Metoda szyfrowania: 1 – AES-256, 2 – ChaCha20, 3 – DES, 4 – RSA.

## 7. Wnioski ogólne **Wnioski**

SVM uzyskał najwyższą dokładność, co sugeruje, że jest najlepiej dopasowany do tego typu danych.

Naive **Bayes** wypadł najgorzej, co może wynikać z założeń modelu (niezależność cech), które nie są spełnione w tym przypadku.

Random Forest i Drzewo decyzyjne osiągnęły podobne wyniki, ale mogą wymagać lepszego dostrajania hiperparametrów.

Słabe wyniki klasy DES sugerują, że cechy wybrane do klasyfikacji mogą być niewystarczające lub dane są zbyt nierównomiernie rozłożone.