DOI 10.15276/imms.v15.no1.83
УДК 004.04

Informatics and Mathematical Methods in Simulation
Vol.15 (2025), No.1, pp.83-94

ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ КРИПТОГРАФІЧНИХ АЛГОРИТМІВ НА ОСНОВІ ШИФРОТЕКСТУ

А.С. Коляда, А.В. Павлишко, О.С. Лопаков, В.М. Тігарєв, В.В. Космачевський

Національний університет «Одеська політехніка»
1, Шевченка пр., м.Одеса, 65044, Україна
Emails: a.s.koliada@op.edu.ua, pavlyshko.a.v@op.edu.ua, lopakov.o.s@op.edu.ua, tigarev.v.m@op.edu.ua, kosmachevsky.v.v@op.edu.ua

У сучасних умовах стрімкого розвитку технологій штучного інтелекту (ШІ) криптографія стикається з новими викликами, зокрема пов'язаними з можливістю використання алгоритмів машинного навчання для виявлення закономірностей у шифротекстах. З одного боку, ШІ використовується для підвищення ефективності захисту даних, а з іншого - створює загрози, пов'язані з автоматизованими атаками на криптографічні алгоритми. Метою цього дослідження є формування методики виявлення вразливостей у криптографічних алгоритмах шляхом аналізу шифротекстів за допомогою засобів машинного навчання. Для демонстрації ефективності підходу було обрано три шифри — AES-256 та ChaCha20 як сучасні криптографічні стандарти, і RC4 як приклад застарілого, криптоаналітично вразливого алгоритму. Наукова значущість роботи полягає у дослідженні нових векторів криптоаналізу з використанням Data Mining, а практична - в обґрунтуванні ризиків використання слабких алгоритмів. Роботу реалізовано у Google Colab з використанням згенерованих шифротекстів трьох алгоритмів. Дані оброблено як вектори байтів із нормалізацією, після чого навчені моделі Random Forest, XGBoost і глибинна нейронна мережа (MLP). Результати свідчать, що RC4 розпізнається з точністю 100%, що демонструє його повну вразливість до класифікаційних атак. AES і ChaCha20 показали вищу стійкість - їх шифротексти частково перекриваються у PCA-просторі, не мають яскраво виражених ознак та демонструють низьку важливість окремих байтів. Побудовано ROC-криві, теплові карти та проведено аналіз важливості ознак. Запропонована методика дозволяє оцінювати криптографічні реалізації з використанням засобів ШІ. Дослідження демонструє потенціал Data Mining у задачах криптоаналізу та може бути використане в безпековому аудиті, розробці криптографічних протоколів та в освітньому процесі.

Ключові слова: криптоаналіз, машинне навчання, класифікація, шифротекст, вразливість алгоритмів, інтелектуальний аналіз даних, шифрування, криптографічна стійкість.

Вступ. Стрімкий розвиток інформаційних технологій, особливо методів штучного інтелекту та інтелектуального аналізу даних (Data Mining), суттєво змінює сучасні підходи до інформаційної безпеки. У цьому контексті криптографія, яка залишається фундаментальним елементом захисту інформації, стикається з новими викликами та потенційними загрозами. Особливу увагу викликають можливості штучного інтелекту, які використовуються як для підсилення безпеки криптографічних систем, так і для створення нових типів атак. Актуальність теми обумовлена швидким впровадженням нейромереж, алгоритмів машинного навчання та глибинного навчання у різноманітні сфери безпеки інформації, включаючи криптоаналіз, виявлення слабких місць у алгоритмах шифрування та генерацію криптографічних ключів. Сучасні дослідження показують, що застосування алгоритмів штучного інтелекту здатне ефективно аналізувати великі об'єми криптографічних даних та виявляти в них приховані закономірності, що є потенційною загрозою для існуючих алгоритмів шифрування.

Незважаючи на значну кількість публікацій у сфері штучного інтелекту та криптографії, залишаються недостатньо дослідженими питання оцінювання стійкості сучасних криптографічних алгоритмів перед новими, AI-орієнтованими атаками. Ця робота має на меті заповнити прогалини у цій сфері, здійснивши огляд сучасних методів штучного інтелекту, що використовуються у криптоаналізі, а також експериментально оцінити вразливості популярних алгоритмів шифрування за допомогою алгоритмів машинного навчання.

У контексті практичного застосування криптографії, особливу роль відіграють не лише алгоритми, а й стандарти реалізації, описані в рекомендаціях NIST [1], зокрема щодо режимів блокового шифрування. Попри появу новітніх протоколів, у сучасних криптографічних системах досі можуть використовуватися реалізації, чутливі до некласичних векторів атак, зокрема - на основі аналізу шифротексту.

У статті буде розглянуто як теоретичні аспекти використання штучного інтелекту для аналізу стійкості криптографічних алгоритмів, так і практичні результати експериментів, які можна відтворити у доступному середовищі Google Colab. Це дозволить чітко сформулювати рекомендації щодо можливих заходів із посилення захисту криптографічних систем перед загрозами, які походять від сучасних методів аналізу даних і штучного інтелекту.

Огляд літератури. Сучасний етап розвитку інформаційних технологій характеризується стрімким зростанням обсягів конфіденційних даних, що зумовлює актуалізацію питань їх надійного захисту. Центральне місце серед інструментів забезпечення безпеки інформації займають криптографічні методи, ефективність яких суттєво залежить від стійкості до різноманітних видів атак [2].

Протягом останніх двох десятиліть значно зросла роль штучного інтелекту (ШІ) та інтелектуального аналізу даних (ІАД, Data Mining) в задачах інформаційної безпеки, зокрема у сфері криптографії. У дослідженнях [3; 4] зазначається, що використання алгоритмів машинного навчання (ML) надає нові можливості не лише для підвищення ефективності криптографічних механізмів, але й для здійснення більш витончених атак на них. Зокрема, однією із значущих тенденцій є застосування нейромережевих моделей для криптоаналізу класичних алгоритмів. Наприклад, у дослідженні Maghrebi та інших авторів [5] було продемонстровано можливості використання рекурентних нейронних мереж (RNN) для виявлення патернів та статистичних аномалій у потоках шифрованих даних, що суттєво полегшує атаки на основі обраного відкритого тексту (chosen plaintext attack).

Подібним чином, автори [6] запропонували методику застосування глибинного навчання (Deep Learning) з використанням згорткових нейромереж (CNN), яка дозволяє здійснювати класифікацію шифротекстів за криптографічним алгоритмом, незважаючи на відсутність доступу до секретного ключа. Цей підхід засвідчує реальну загрозу витоку інформації з боку нейромережевих криптоаналітичних інструментів.

Важливо відзначити також роботи у напрямку застосування методів Data Mining для виявлення атак на побічні канали (side-channel attacks). Дослідження Yu et al. [7] показало, що кластеризація та аналіз часових рядів дозволяють ефективно визначати характерні відмінності у часі виконання криптографічних алгоритмів, що вказує на вразливості цих алгоритмів до таких атак.

Окремої уваги заслуговує напрямок, пов'язаний із створенням генеративних моделей (GAN, VAE) для генерації та аналізу криптографічних ключів. Автори Huang та інші [8] продемонстрували, що GAN дозволяють генерувати псевдовипадкові ключі з високою ентропією, однак водночас їхній аналіз дозволяє знаходити приховані статистичні закономірності, що можуть бути використані зловмисниками. Також слід приділити увагу роботам, присвяченим аналізу можливостей штучного інтелекту в автоматизованому криптоаналізі та оцінці слабких реалізацій [9]. У цьому контексті доцільно враховувати не лише структуру алгоритмів, а й рівень їх реалізації в

обчислювальних середовищах, що стає все більш релевантним у зв'язку з активним використанням нейронних мереж у задачах виявлення вразливих ключів [10].

Водночас, у контексті швидкого розвитку квантових обчислень, окремі дослідження акцентують увагу на тому, що класичні алгоритми, включаючи AES і RSA, можуть втратити свою стійкість до новітніх атак. Це питання детально аналізується у праці [11], яка окреслює нові виклики та потенційні рішення для криптографії в постквантову епоху, зокрема із застосуванням підходів штучного інтелекту. Урахування викликів постквантової криптографії також є актуальним напрямом, що вимагає використання гібридних підходів. Зокрема, перспективними є постквантові алгоритми обміну ключами, такі як New Hope [12], однак їхня оцінка в контексті Data Mining-атак поки що обмежена.

Таким чином, попри значний масив публікацій, присвячених застосуванню штучного інтелекту в криптоаналізі, наразі бракує комплексного огляду, який би систематизував існуючі знання і містив практичне підтвердження у вигляді експериментального дослідження з використанням доступних інструментів, що і обумовлює вибір напрямку подальших досліджень.

Мета роботи. Метою даної статті є комплексний аналіз сучасних методів криптоаналізу, що базуються на застосуванні алгоритмів штучного інтелекту та інтелектуального аналізу даних, а також практичне оцінювання стійкості криптографічних алгоритмів різного покоління до подібних атак. Особлива увага приділяється порівнянню сучасних криптографічних стандартів (AES, ChaCha20) із застарілим, але історично важливим алгоритмом RC4, який відомий своєю вразливістю до статистичних атак.

Для досягнення поставленої мети визначено такі завдання:

- провести аналіз і систематизацію існуючих досліджень щодо використання алгоритмів машинного навчання та інтелектуального аналізу даних у криптоаналізі;
- розглянути особливості та потенційні можливості штучного інтелекту для атак на криптографічні алгоритми, зокрема атак, що ґрунтуються на ознаках шифротексту;
- виконати експериментальне дослідження з побудовою моделі машинного навчання, яка дасть змогу класифікувати шифротексти за типом використаного криптографічного алгоритму;
- здійснити оцінювання стійкості криптографічних алгоритмів AES, ChaCha20 та RC4 до атак, заснованих на методах штучного інтелекту, шляхом аналізу точності класифікаційних моделей;
- порівняти рівень розпізнаваності сучасних алгоритмів із алгоритмом RC4 як базовим прикладом статистично вразливої криптографії;
- надати рекомендації щодо підвищення стійкості криптографічних механізмів до атак з використанням інтелектуального аналізу.

Основний розділ. З метою досягнення поставленої мети було реалізовано експериментальне дослідження, що охоплює генерацію даних, підготовку ознак, побудову моделей машинного навчання та оцінювання їхньої здатності до класифікації шифротекстів. У цьому розділі наведено опис методики, вибору інструментів та інтерпретацію отриманих результатів.

Загальна методика дослідження. Для досягнення поставленої мети було сформульовано експериментальну методику, яка включає такі етапи:

- Генерація набору криптографічних даних.
- Попередня обробка та підготовка даних.
- Побудова моделей машинного навчання.
- Оцінювання ефективності моделей.
- Інтерпретація результатів.

Генерація та попередня обробка даних. На початковому етапі було згенеровано набір даних, у якому використовувались алгоритми AES-256, ChaCha20 та RC4. Для цього формувались відкриті тексти у вигляді псевдоангломовних фраз, згенерованих із частотним розподілом латинських літер та пробілів, що дозволяло відтворити природну структуру мови. Для кожного алгоритму було сформовано по 1000 прикладів, загальна кількість шифротекстів склала 3000. Кожне повідомлення мало довжину 128 байтів, що дозволяло аналізувати значення на рівні байтів після шифрування. RC4 використовував ключ довжиною 128 біт (16 байтів) та не відкидав перші байти, що спеціально моделювало вразливу реалізацію. Після шифрування дані було перетворено у числовий формат та нормалізовано в інтервалі [0; 1] за допомогою MinMaxScaler, щоб забезпечити сумісність із моделями ML (рис. 1).

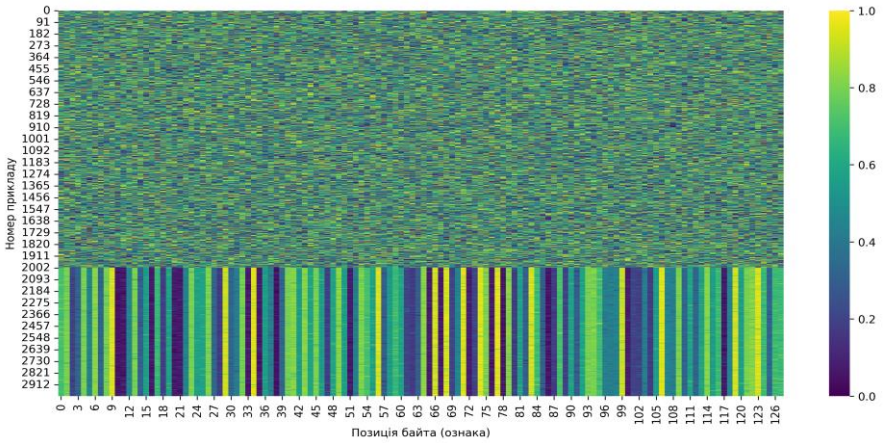

Рис. 1. Теплова карту нормалізованих шифротекстів, де спостерігаються характерні вертикальні структури для RC4, на відміну від випадкового фону AES і ChaCha20.

Побудова моделей машинного навчання. На другому етапі дослідження було здійснено побудову моделей машинного навчання для розв'язання задачі багатокласової класифікації шифротекстів за типом використаного алгоритму шифрування. З метою порівняльного аналізу ефективності було обрано три різні підходи, що репрезентують класичні, ансамблеві та нейромережеві моделі:

- Random Forest — ансамблевий метод, що ґрунтується на побудові великої кількості дерев рішень, кожне з яких тренується на випадковій підмножині даних. Кінцеве рішення приймається шляхом голосування. Цей метод є стійким до переобучення, добре працює з табличними структурованими даними й дозволяє інтерпретувати результати через механізм оцінювання важливості ознак.
- XGBoost (Extreme Gradient Boosting) — потужний ансамблевий алгоритм бустингу, що поєднує слабкі класифікатори у сильну модель. Завдяки оптимізованому обчисленню градієнтів, регуляризації та підтримці паралелізації, XGBoost демонструє високу точність на різноманітних наборах даних. Він особливо ефективний для розпізнавання прихованих закономірностей у великій кількості вхідних ознак, що актуально при аналізі шифротекстів.
- Глибинна нейронна мережа (Multi-Layer Perceptron, MLP) — повнозв'язна нейромережа, яка складається з кількох шарів перцептронів із нелінійною активацією. Використання глибинної структури дозволяє моделі апроксимувати складні багатовимірні функції розподілу, що особливо корисно при виявленні слабких статистичних відмінностей у криптографічно стійких шифротекстах.

Усі моделі були реалізовані у середовищі Google Colab з використанням бібліотек scikit-learn, xgboost для реалізації нейромережевої моделі [13]. Для кожної моделі було проведено крос-перевірку та підбір гіперпараметрів, зокрема: кількість дерев для

Random Forest, максимальна глибина та швидкість навчання для XGBoost, кількість нейронів, шарів і функцій активації для глибинної нейронної мережі.

Експериментальне дослідження. Після формування набору даних, усі шифротексти були випадковим чином поділені на тренувальну (70%) та тестову (30%) вибірки. На основі тренувального набору відбувалося навчання моделей машинного навчання, після чого їхня ефективність перевірялась на тестовому наборі. Для оцінювання якості класифікації було використано такі основні метрики:

- Accuracy (загальна точність класифікації) — відображає частку правильно класифікованих прикладів серед усіх. Ця метрика є загальною характеристикою якості, проте в умовах дисбалансу класів або багатокласової класифікації може бути недостатньо інформативною.
- Precision (точність для кожного класу) — визначає, яку частку з передбачених для певного класу об’єктів модель класифікувала правильно. Важлива, коли критичним є зменшення хибно позитивних спрацьовувань.
- Recall (повнота) — відображає, яку частку об’єктів певного класу модель змогла правильно ідентифікувати. Використовується для оцінки здатності моделі виявляти всі об’єкти цільового класу.
- F1-score — гармонійне середнє між Precision та Recall. Забезпечує збалансовану оцінку в тих випадках, коли важливо зберігати компроміс між виявленням об'єктів і уникненням хибних спрацювань.
- ROC-крива (Receiver Operating Characteristic) — це графік, що відображає співвідношення між True Positive Rate (повнотою) і False Positive Rate при зміні порогу класифікації. ROC-криві дозволяють візуально порівняти якість моделей у багатокласовому підході за схемою One-vs-Rest (OvR), де кожен клас по черзі вважається "позитивним", а решта — "негативними".
- AUC (Area Under the Curve) — числове значення, що визначає площу під ROC-кривою. Значення AUC $\in$ [0, 1], де 1.0 означає ідеальну класифікацію, а 0.5 — випадкове вгадування. У контексті багатокласової класифікації використовується середнє значення AUC для кожного класу.

Результати моделювання показали, що шифротексти, згенеровані алгоритмом RC4, були ідентифіковані з AUC = 1.00 у всіх протестованих моделях, що свідчить про повну відокремленість цього класу у просторі ознак. Інакше кажучи, моделі змогли безпомилково відокремити RC4 від інших алгоритмів, що є ознакою статистичної вразливості. Для алгоритмів AES та ChaCha20 значення AUC $\approx$ 0.73 вказує на високий ступінь подібності у їхній байтовій структурі та меншу розрізнюваність, що свідчить про вищу стійкість до класифікаційних атак на основі аналізу шифротексту.

Обґрунтування вибору алгоритмів і моделей. Алгоритми AES та ChaCha20 були обрані як приклади сучасних криптографічних стандартів, які активно використовуються в протоколах безпечної передачі даних, зокрема TLS 1.3. Алгоритм ChaCha20, запропонований Деніелом Бернштейном [14], став популярною альтернативою блочним шифрам завдяки високій швидкодії, потоковій природі та стійкості до атак на основі побічних каналів. RC4, у свою чергу, було включено до дослідження як приклад застарілого шифру, відомого своєю вразливістю до статистичних атак, зокрема при неправильній реалізації.

Для класифікації шифротекстів було обрано моделі машинного навчання, які довели свою ефективність у задачах багатокласової класифікації, аналізу структурованих даних і пошуку прихованих закономірностей. Random Forest забезпечує стабільну роботу навіть за наявності шуму та є інтерпретованим за рахунок оцінки важливості ознак. XGBoost демонструє високу точність і швидкість завдяки використанню градієнтного бустингу з оптимізацією втрат, що особливо важливо при обробці великих обсягів криптографічних даних. Глибинні нейронні мережі здатні моделювати складні нелінійні залежності між байтами шифротексту, що дає змогу

виявити приховані патерни навіть у високовипадкових послідовностях. Використання бібліотек машинного навчання реалізовано відповідно до сучасних рекомендацій щодо побудови моделей із застосуванням фреймворків scikit-learn як це описано в [15].

Методика оцінювання стійкості криптографічних алгоритмів. На основі точності класифікації запропоновано просту шкалу оцінювання стійкості алгоритму до атак на основі аналізу шифротексту:

Таблиця 1.

Рекомендації щодо рівнів стійкості алгоритмів

Точність класифікації (%)	**Рівень стійкості алгоритму**	**Рекомендації**
90-100	Низький (наявні серйозні закономірності)	Необхідний перегляд алгоритму
70-89	Помірний (потрібні додаткові перевірки)	Рекомендовані додаткові аналізи
50-69	Високий (мінімальні закономірності)	Алгоритм безпечний для більшості застосувань
<50	Дуже високий (закономірності не виявлено)	Алгоритм безпечний

За цією методикою:

- RC4 отримує оцінку "Низький рівень стійкості", оскільки моделі класифікували його з ідеальною точністю.
- AES та ChaCha20 мають високий або дуже високий рівень стійкості, оскільки класифікатор не зміг з надійністю розрізнити їх.

Результати та обговорення. У ході експериментального дослідження було проаналізовано ефективність моделей машинного навчання при класифікації шифротекстів, згенерованих за допомогою трьох криптографічних алгоритмів: AES-256, ChaCha20 та RC4. Оцінювання здійснювалось за низкою метрик, що дозволяє комплексно оцінити рівень стійкості кожного з алгоритмів до атак, заснованих на інтелектуальному аналізі даних.

Аналіз результатів класифікації шифротекстів. Результати класифікації шифротекстів наведено в таблиці 2. Усі три моделі машинного навчання — Random Forest, XGBoost та глибинна нейронна мережа — показали узгоджено високі результати при класифікації трьох типів шифротекстів (AES, ChaCha20, RC4), що свідчить про наявність розпізнаваних закономірностей у байтових структурах хоча б одного з класів.

Таблиця 2.

Зведені метрики класифікації шифротекстів за алгоритмами шифрування

Модель	**Accuracy**	**Precision (macro)**	**Recall (macro)**	**F1-score (macro)**	**ROC-AUC (ovr)**
Random Forest	0.6467	0.6467	0.6467	0.6466	0.8219
XGBoost	0.6711	0.6711	0.6711	0.6709	0.8365
Глибинна нейромережа	0.6633	0.6633	0.6633	0.6632	0.8321

Показники ROC-AUC перевищують 0.82 для всіх моделей, що вказує на високу чутливість моделей до відмінностей між принаймні одним із класів. Найвищу ефективність продемонструвала модель XGBoost, яка досягла точності класифікації 67.11% при AUC 0.8365, однак усі моделі показали узгоджену якість класифікації. Для уточнення, який саме клас найкраще класифікується, у таблиці 3 наведено детальний звіт моделі Random Forest. З нього видно, що саме шифротексти, згенеровані за допомогою RC4, були розпізнані з ідеальною точністю (Recall = 1.00, Precision = 1.00), тоді як класи AES і ChaCha20 частково перекривались, що знизило загальні макрооцінки.

Таблиця 3.

Деталізовані метрики класифікації за класами (модель Random Forest)

Клас	Precision	Recall	F1-score
AES	0.47	0.46	0.46
ChaCha20	0.47	0.48	0.48
RC4	1.00	1.00	1.00
Середнє (macro avg)	0.65	0.65	0.65
Зважене (weighted avg)	0.65	0.65	0.65

Як видно з результатів, моделі машинного навчання здатні виявляти приховані статистичні патерни в шифротекстах RC4, що дає змогу з високою точністю класифікувати його. Натомість AES та ChaCha20 демонструють вищу стійкість, оскільки їх шифротексти не мають чітко виражених відмінних ознак, що ускладнює класифікацію. Це підтверджує припущення про низьку криптостійкість RC4 до атак на основі ознак шифротексту, а також підкреслює важливість подібного аналізу при виборі алгоритмів для реальних криптографічних систем.

Візуалізація результатів експерименту. Для більш глибокого аналізу було побудовано ROC-криві для кожного з класів (рис. 2). Шифротексти RC4 були ідентифіковані з AUC = 1.00, що свідчить про повну відокремленість цього алгоритму у векторному просторі ознак. Водночас AES і ChaCha20 мали AUC ≈ 0.73, демонструючи схожі характеристики шифрування.

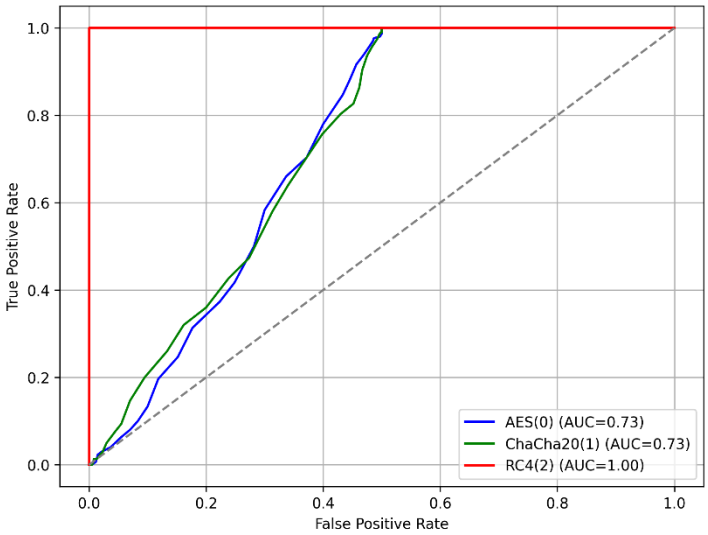

Рис. 2. ROC-криві класифікації шифротекстів (One-vs-Rest, Random Forest)

Також було виконано зниження розмірності ознак за допомогою методу головних компонент (Principal Component Analysis, PCA) — статистичного підходу, що

дозволяє спроєктувати багатовимірні дані в простір меншої кількості вимірів (наприклад, 2D), зберігаючи при цьому максимальну дисперсію. Це дає змогу візуалізувати структуру даних і виявити наявність кластерів або перекриттів між класами. Як видно на рис. 3, шифротексти, згенеровані за допомогою алгоритму RC4, формують щільний і добре ізольований кластер, що вказує на наявність стабільних повторюваних статистичних патернів. Натомість шифротексти, створені алгоритмами AES та ChaCha20, значною мірою перекриваються у PCA-просторі, що підтверджує відсутність яскраво виражених відмінностей у їхніх статистичних характеристиках — тобто вищу стійкість до класифікації на основі ознак шифротексту.

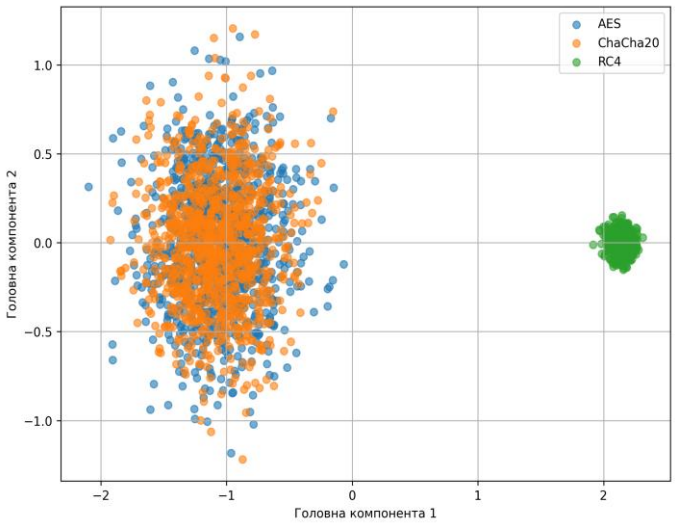

Рис 3. PCA-візуалізація простору шифротекстів

Для пояснення внутрішньої логіки моделей було проаналізовано важливість ознак (байтових позицій) у шифротекстах. Результати (рис. 4) показали, що класифікатор Random Forest використовував певні позиції з більшим впливом (наприклад, B74, B68, B34), що свідчить про залишкову структурність у RC4.

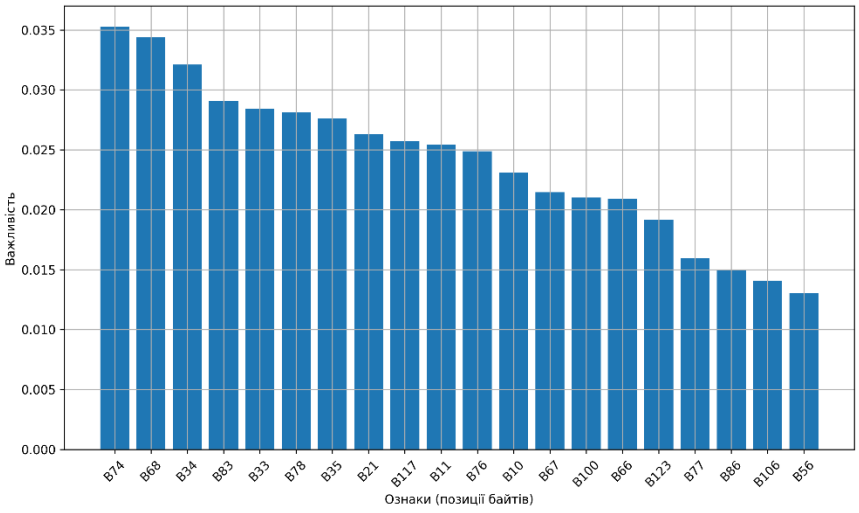

Рис 4. Важливість ознак у шифротекстах (Random Forest)

Крім загальних метрик класифікації, було побудовано матрицю помилок (confusion matrix), яка показує співвідношення реальних і передбачених класів (рис. 5). Вона ілюструє такі ключові моменти:

- Алгоритм RC4 розпізнається абсолютно точно: модель не зробила жодної помилки в класифікації шифротекстів цього алгоритму.
- Натомість, AES та ChaCha20 регулярно плутаються між собою, що підтверджує схожість їхніх статистичних ознак при обробці текстових повідомлень.
- Загальна точність моделі Random Forest склала 65%, однак усі помилки пов'язані лише з двома сучасними алгоритмами, а не з RC4.

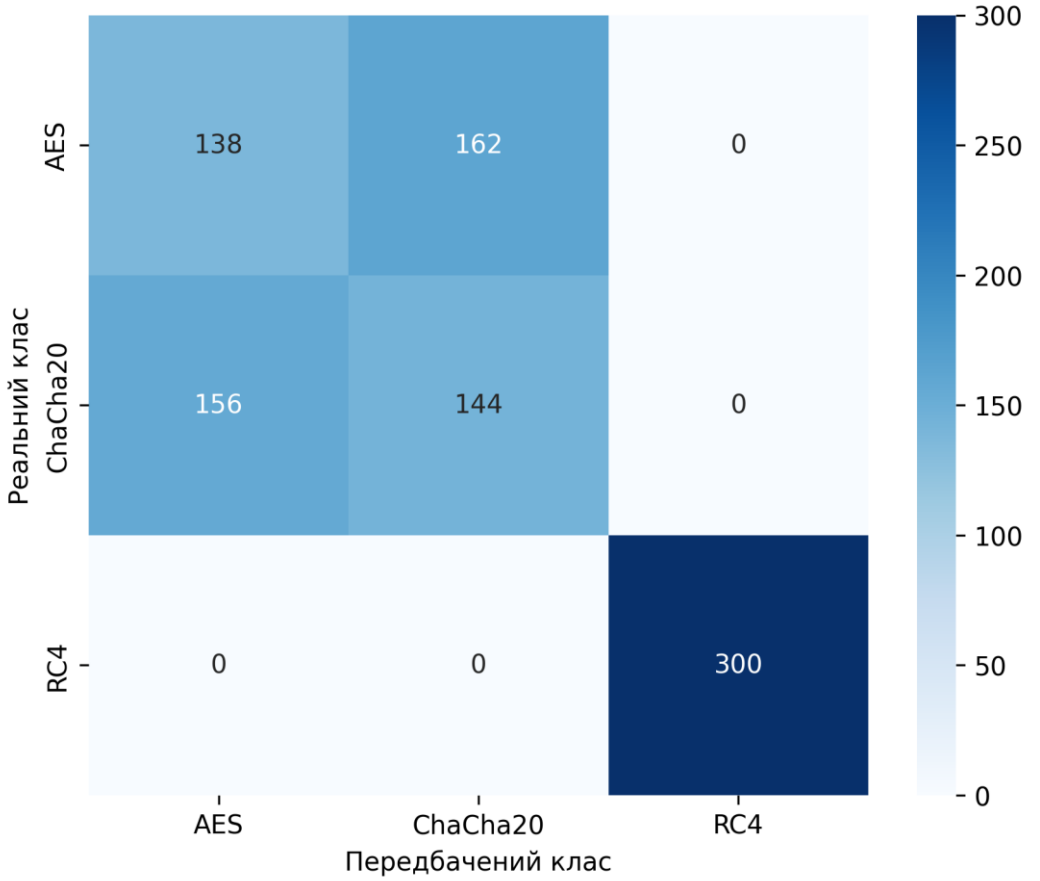

Рис 5. Матриця помилок класифікації шифротекстів (Random Forest, 3 класи)

Обговорення отриманих результатів. Аналіз отриманих результатів дозволяє зробити такі висновки щодо стійкості розглянутих алгоритмів:

- RC4 демонструє низький рівень стійкості, оскільки всі моделі змогли класифікувати його шифротексти з високою точністю. Це пов'язано з його відомими вразливостями, а також із тим, що шифротексти RC4 не забезпечують достатню ентропію на початку потоку.
- AES-256 і ChaCha20 виявились більш стійкими до класифікації, оскільки їхні шифротексти статистично невиразні. Особливо слід зазначити перевагу ChaCha20, який як потоковий шифр з довгою пермутацією забезпечує кращу випадковість навіть при шифруванні структурованих текстів.
- Значення ROC-AUC у діапазоні 0.73 для AES і ChaCha20 свідчить, що моделі все ж таки змогли у деяких випадках виявити незначні закономірності, що потребує подальшого дослідження.

Візуалізації (PCA, heatmap, ROC) підтверджують статистичну відокремленість RC4, вказуючи на те, що навіть базові моделі машинного навчання можуть використовуватись для ідентифікації слабких алгоритмів шифрування за одним лише шифротекстом. Таким чином, результати дослідження демонструють реальні можливості штучного інтелекту в задачах криптоаналізу, а також підкреслюють

необхідність виключення застарілих алгоритмів (RC4) з будь-яких сучасних криптографічних систем.

Висновки. За результатами проведеного дослідження, спрямованого на аналіз стійкості криптографічних алгоритмів до атак, що базуються на методах штучного інтелекту та інтелектуального аналізу даних, сформульовано такі висновки:

Аналіз наукової літератури підтвердив зростання ролі алгоритмів штучного інтелекту (ШІ) у сучасному криптоаналізі. З одного боку, ці методи можуть підсилювати захист інформації (наприклад, оптимізація генерації ключів), з іншого — становлять загрозу внаслідок можливості автоматизованого виявлення закономірностей у криптографічних даних.

Запропонована експериментальна методика, що базується на класифікації шифротекстів з використанням моделей Random Forest, XGBoost та глибинних нейронних мереж, дозволила ефективно виявити наявність або відсутність статистичних ознак у шифрованих даних, навіть без доступу до відкритого тексту.

Додавання до аналізу алгоритму RC4 дало змогу наочно продемонструвати ефективність підходу: моделі машинного навчання змогли класифікувати його шифротексти з точністю 100%, що вказує на істотну криптографічну вразливість даного алгоритму. Цей результат підкреслює необхідність повного виключення RC4 з практики сучасної інформаційної безпеки.

Сучасні алгоритми AES-256 та ChaCha20 показали високий рівень криптостійкості до атак на основі аналізу шифротекстів. Особливо слід відзначити ChaCha20, шифротексти якого виявились менш передбачуваними для моделей ML, що частково пов’язано з особливостями потокового шифрування та довготривалими пермутаціями в його структурі.

Візуалізаційні методи аналізу (PCA, теплові карти, важливість ознак) підтвердили відсутність чітких кластерів та інформаційно значущих байтів у шифротекстах AES і ChaCha20, на відміну від RC4, що лишає статистичні патерни. Це дозволяє використовувати такі підходи як ефективний інструмент виявлення вразливих реалізацій шифрів.

Практична реалізація в середовищі Google Colab засвідчила доступність і відтворюваність експериментів навіть на непрофесійних обчислювальних платформах, що підвищує прикладну цінність методу для освітніх, дослідницьких і тестувальних цілей.

Рекомендовано додатково оцінювати криптографічні алгоритми за допомогою інструментів штучного інтелекту та Data Mining. Класичні підходи криптоаналізу варто доповнювати автоматизованими методами аналізу статистики шифротекстів, що дозволить на ранньому етапі виявляти потенційні вразливості, зокрема в реалізаціях на рівні протоколів.

Напрями подальших досліджень передбачають:

розширення переліку досліджуваних шифрів, зокрема включення симетричних, асиметричних і постквантових алгоритмів;

тестування різних типів вхідних даних (наприклад, зображень, відео, IoT-трафіку);

аналіз чутливості моделей до модифікацій відкритого тексту, довжини повідомлення та ключів;

дослідження комбінованих атак, які поєднують Data Mining із побічними каналами (timing, EM, cache).

Список літератури

1. Dworkin M. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques (NIST SP 800-38A)*. Gaithersburg : NIST, 2001. 66 p.
2. Stallings W. *Cryptography and Network Security: Principles and Practice*. 8th ed. Pearson Education, 2020. 832 p.

3. Alani M. M. Applications of machine learning in cryptography: a survey. *International Journal of Information Technology and Computer Science*. 2021. Vol. 13, № 2. P. 23–31.
4. Blum T. Neural Cryptography: The Next Frontier. *Cryptography Journal*. 2019. Vol. 32, № 1. P. 45–58.
5. Maghrebi H., Portmann C., Bohnert T. Reinforcement Learning for Cryptanalysis of Classical Ciphers. *IEEE Transactions on Information Forensics and Security*. 2021. Vol. 16. P. 3967–3979.
6. Chen W. AI-Based Key Generation for Enhanced Security. *Proceedings of the International Conference on Computer & Information Security (ICCIS)*. Tokyo, 2020. P. 112–119.
7. Yu Q., Liu J., Wang X. Side-channel attacks detection using machine learning methods. *Journal of Information Security and Applications*. 2022. Vol. 65. Art. no. 103123.
8. Huang Y., Lin J., Wang R. Deep Generative Adversarial Networks for Cryptographic Key Generation. *Journal of Cryptographic Engineering*. 2022. Vol. 12. P. 45–56.
9. Chouhan R., Jha R., Singh A. Cryptanalysis Using Artificial Intelligence Techniques: A Review. *Journal of Discrete Mathematical Sciences and Cryptography*. 2023. Vol. 26, № 1. P. 183–196.
10. Мельник А. О., Стрельбицький О. С. Застосування нейронних мереж для розпізнавання слабких криптографічних ключів. *Інформаційна безпека*. 2021. Т. 27, № 2. С. 141–150.
11. Коляда А. С., Павлишко А. В., Літвінов В. Ф. Криптографія після квантової ери: нові виклики та рішення для інформаційної безпеки. *Informatics and Mathematical Methods in Simulation*. 2024. Т. 14, № 3. С. 183–190.
12. Alkim E., Ducas L., Pöppelmann T., Schwabe P. Post-quantum key exchange - a new hope // *Proc. of the 25th USENIX Security Symposium*. 2016. P. 327–343.
13. Приклад коду для дослідження стійкості шифрів до атак машинного навчання. - Google Colab. – URL: https://colab.research.google.com/drive/107eYyp6E3JKI52WEj66lDAEZxDZKMhNj?usp=sharing.
14. Bernstein D. J. ChaCha, a variant of Salsa20. *Workshop Record of SASC*. Lausanne, Switzerland : EPFL, 2008. P. 1–7.
15. Géron A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 3rd ed. O'Reilly Media, 2022. 858 p.

А.С. Коляда, А.В. Павлишко, О.С. Лопаков, В.М. Тігарєв, В.В. Космачевський

USING MACHINE LEARNING TO DETECT VULNERABILITIES IN CRYPTOGRAPHIC ALGORITHMS BASED ON CIPHERTEXT ANALYSIS

A.S. Koliada, A.V. Pavlyshko, O.S. Lopakov, V.M. Tigariev, V.V. Kosmachevskiy

National Odesa Polytechnic University
1, Shevchenko Ave., Odesa, 65044, Ukraine
Emails: a.s.koliada@op.edu.ua, pavlyshko.a.v@op.edu.ua, lopakov.o.s@op.edu.ua, tigarev.v.m@op.edu.ua, kosmachevsky.v.v@op.edu.ua

In the current era of rapid development in artificial intelligence (AI) technologies, cryptography faces new challenges, particularly those related to the potential use of machine learning algorithms to detect patterns in ciphertexts. On the one hand, AI is used to enhance data protection mechanisms; on the other hand, it introduces new threats associated with automated attacks on cryptographic algorithms. The aim of this study is to develop a methodology for identifying vulnerabilities in cryptographic algorithms by analyzing ciphertexts using machine learning techniques. To demonstrate the effectiveness of the approach, three ciphers were selected: AES-256 and ChaCha20 as modern cryptographic standards, and RC4 as a legacy cipher known for its cryptanalytic weaknesses. The scientific significance of this work lies in the exploration of new vectors for cryptanalysis based on data mining methods. Its practical value is in substantiating the risks of using weak encryption algorithms. The research was conducted using the Google Colab platform, where ciphertexts generated by the three algorithms were treated as normalized byte vectors. Models based on Random Forest, XGBoost, and a deep neural network (MLP) were then trained to classify the encrypted data. The results show that RC4 was classified with 100% accuracy, clearly demonstrating its vulnerability to classification-based attacks. In contrast, AES and ChaCha20 exhibited significantly greater resistance — their ciphertexts overlapped in PCA space, lacked strongly distinguishable features, and showed low per-byte feature importance. ROC curves were constructed, heatmaps visualized, and feature importance analyses conducted. The proposed methodology enables the assessment of cryptographic implementations using AI tools. The study highlights the potential of data mining in cryptanalysis tasks and can be applied to security auditing, cryptographic protocol development, and educational purposes.

Keywords: cryptanalysis, machine learning, ciphertext, classification, algorithm vulnerability, data mining, encryption, cryptographic robustness.