НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
«_____»_____2025 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Системи, технології та математичні методи кібербезпеки»
спеціальності 125 «Кібербезпека та захист інформації»

на тему: Застосування стилометрії в OSINT для вирішення задач кібернетичної та інформаційної безпеки

Виконав (-ла): здобувач вищої освіти ІІ курсу, групи ФБ-42мп
(шифр групи)

Косигін Олександр Сергійович
(прізвище, ім'я, по батькові) (підпис)

Керівник Ланде Дмитро Володимирович, доктор технічних наук, професор, зав. каф. ІБ
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент ст. викладач кафедри ММЗІ, к. ф.-м. н. А.В. Фесенко
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ – 2025 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський)
Спеціальність – 125 «Кібербезпека та захист інформації»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Дмитро ЛАНДЕ
(підпис)
«___»_____2025 р.

ЗАВДАННЯ
на магістерську дисертацію здобувачу вищої освіти

Косигін Олександр Сергійович
(прізвище, ім'я, по батькові)

1. Тема роботи
Застосування стилометрії в OSINT для вирішення задач кібернетичної та інформаційної безпеки

Науковий керівник
Ланде Дмитро Володимирович, доктор технічних наук, професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «04» листопада 2025 р. № 4797-с

2. Термін подання здобувачем вищої освіти роботи 17 грудня 2025 року

3. Вихідні дані до роботи
Літературні джерела з питань стилометрії та OSINT, навчальні корпуси текстів Brown Corpus та Project Gutenberg, репозиторії вихідного коду GitHub, існуючі алгоритми атрибуції авторства Burrows' Delta та n-грами, методи машинного навчання

4. Зміст роботи
Аналіз теоретичних засад стилометрії та її інтеграції в процеси кіберрозвідки, дослідження методів машинного навчання для обробки природної мови та аналізу коду, розробка методики ідентифікації автора в умовах обфускації та зміни семантичного домену, експериментальна оцінка ефективності алгоритмів детекції аномалій та підміни автора.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Презентація до захисту магістерської роботи

6. Дата видачі завдання 01.09.2025

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Отримання завдання на магістерську дисертацію	01.09.2025	Виконано
2	Огляд наукової літератури	01.09.2025 – 15.09.2025	Виконано
3	Аналіз математичних моделей стилометрії	16.09.2025 – 19.09.2025	Виконано
4	Дослідження методів протидії атрибуції	16.09.2025 – 19.09.2025	Виконано
5	Аналіз теоретичних засад та існуючих підходів до ідентифікації автора	01.10.2025 – 06.10.2025	Виконано
6	Вибірметодів та засобів вирішення задачі атрибуції кіберзагроз	07.10.2025 – 14.11.2025	Виконано
7	Програмна реалізація компонентів попередньої обробки та векторизації даних	15.10.2025 – 18.11.2025	Виконано
8	Розробка та налаштування моделей класифікації	18.11.2025 – 25.11.2025	Виконано
9	Проведення експериментальних досліджень	01.09.2025 – 26.10.2025	Виконано
10	Передзахист роботи	18.12.2025	Виконано
11	Захист роборти	25.12.2025	Виконано

Здобувач вищої освіти _____ Олександр КОСИГІН
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

Керівник роботи _____ Дмитро ЛАНДЕ
(підпис) (**Власне ім'я, ПРІЗВИЩЕ**)

РЕФЕРАТ

Робота складається з трьох розділів, містить 95 сторінки, 8 ілюстрацій, 5 таблиць, додатки та 39 посилання на джерела.

Метою даної роботи є підвищення ефективності вирішення задач кібернетичної та інформаційної безпеки в середовищі OSINT шляхом розробки та експериментальної оцінки методів стилометричного аналізу.

Методи дослідження включають в себе аналіз теоретичних засад стилометрії, методи математичної статистики (зокрема метод Burrows' Delta), алгоритми машинного навчання (Random Forest, SVM, One-Class SVM) для класифікації текстів та коду, а також порівняльний експеримент для оцінки ресурсоємності та точності моделей.

Результатом роботи стало експериментальне підтердження переваги класичних статистичних методів над складними NLP-моделями в задачах атрибуції при зміні семантичного домену, розробка методики ідентифікації розробника програмного коду на основі аналізу «технічного шуму» в умовах обфускації, а також реалізація підходу до виявлення компрометації облікових записів за допомогою поведінкової біометрії.

Створені розробки допоможуть аналітикам SOC та CSIRT автоматизувати процеси профілювання кіберзлочинних угруповань, ефективно виявляти ботоферми та встановлювати авторство шкідливого програмного забезпечення, значно зменшуючи час на розслідування інцидентів та мінімізуючи вплив людського фактору.

Ключові слова: OSINT, Стилометрія, Кібербезпека, Атрибуція авторства, Машинне навчання, NLP, Форензика коду, Нейронні мережі.

ABSTRACT

The work consists of three chapters, contains 95 pages, 8 illustrations, 5 tables, appendices and 39 references.

The purpose of the research is to increase the efficiency of solving cyber and information security tasks in the OSINT environment by developing and experimentally assessing stylometric analysis methods.

Research methods include the analysis of theoretical foundations of stylometry, methods of mathematical statistics (specifically Burrows' Delta method), machine learning algorithms (Random Forest, SVM, One-Class SVM) for text and code classification, as well as a comparative experiment to assess the resource intensity and accuracy of the models.

The result of the work was the experimental confirmation of the superiority of classical statistical methods over complex NLP models in attribution tasks involving semantic domain changes, the development of a methodology for identifying software code developers based on "technical noise" analysis under obfuscation conditions, and the implementation of an approach to detecting account compromise using behavioral biometrics.

The developed solutions will help SOC and CSIRT analysts automate the profiling of cybercriminal groups, effectively detect bot farms, and establish the authorship of malware, significantly reducing incident investigation time and minimizing the impact of human error.

Keywords: OSINT, Stylometry, Cybersecurity, Authorship attribution, Machine learning, NLP, Code forensics, Neural networks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

APT (Advanced Persistent Threat) — розвинена стійка загроза / цілеспрямована кібератака

AST (Abstract Syntax Tree) — абстрактне синтаксичне дерево

AUC-ROC (Area Under Curve – Receiver Operating Characteristic) — площа під кривою помилок (метрика оцінки якості моделі)

BERT (Bidirectional Encoder Representations from Transformers) — двонаправлена модель-трансформер для обробки мови

Bi-LSTM (Bidirectional Long Short-Term Memory) — двонаправлена мережа довгої короткострокової пам'яті

BoW (Bag of Words) — модель «мішок слів»

CERT (Computer Emergency Response Team) — команда реагування на комп'ютерні надзвичайні події

CNN (Convolutional Neural Networks) — згорткові нейронні мережі

CSIRT (Computer Security Incident Response Team) — група реагування на інциденти комп'ютерної безпеки

DAN (Deep Average Network) — глибока мережа усереднення

DetectGPT — метод виявлення тексту, згенерованого штучним інтелектом

DNS (Domain Name System) — система доменних імен

GTR (Generative Transformer) — генеративна модель-трансформер

IoT (Internet of Things) — Інтернет речей

IP (Internet Protocol) — міжмережевий протокол / IP-адреса

IT — інформаційні технології

JGAAP (Java Graphical Toolkit for Authorship Attribution) — набір інструментів для атрибуції авторства

KNN (k-Nearest Neighbors) — метод k-найближчих сусідів

LogReg (Logistic Regression) — логістична регресія

LSTM (Long Short-Term Memory) — мережа довгої короткострокової пам'яті

NLP (Natural Language Processing) — обробка природної мови

OSINT (Open Source Intelligence) — розвідка на основі відкритих джерел

PGP (Pretty Good Privacy) — пакет програмного забезпечення для шифрування

RAM (Random Access Memory) — оперативна пам'ять

RBF (Radial Basis Function) — радіально-базисна функція

RNN (Recurrent Neural Networks) — рекурентні нейронні мережі

RoBERTa (Robustly Optimized BERT Pretraining Approach) — оптимізована модель BERT

SCAP (Source Code Authorship Profiling) — профілювання авторства вихідного коду

SGD (Stochastic Gradient Descent) — стохастичний градієнтний спуск

SIEM (Security Information and Event Management) — система управління інформаційною безпекою та подіями

SOAR (Security Orchestration, Automation and Response) — оркестрація безпеки, автоматизація та реагування

SSL (Secure Sockets Layer) — протокол захищених сокетів

SVM (Support Vector Machines) — метод опорних векторів

TF-IDF (Term Frequency-Inverse Document Frequency) — частота терміна — обернена частота документа

Tree-LSTM — деревовидна мережа довгої короткострокової пам'яті

TTPs (Tactics, Techniques, and Procedures) — тактики, техніки та процедури

XLNet — узагальнена авторегресійна модель попереднього навчання

ЄДРПОУ — Єдиний державний реєстр підприємств та організацій України

ЗМІ — засоби масової інформації

ШІ — штучний інтелект

ВСТУП

Актуальність роботи: В умовах стрімкого розвитку інформаційного суспільства та глобалізації цифрового простору критичного значення набувають питання забезпечення кібернетичної та інформаційної безпеки. Сучасні гібридні загрози, поширення дезінформації, використання бот-мереж для маніпуляції громадською думкою, а також зростання кількості кібератак вимагають нових, ефективніших підходів до аналізу даних. Розвідка на основі відкритих джерел стає ключовим елементом у виявленні та попередженні таких загроз. Однак, величезні обсяги неструктурованої інформації та можливості анонімізації зловмисників у мережі значно ускладнюють процеси атрибуції та профілювання.

У цьому контексті особливої актуальності набуває стилометрія — наука про кількісний аналіз стилю автора. Інтеграція стилометричних методів у інструментарій OSINT дозволяє перетворити унікальні мовні та кодові патерни на надійні ідентифікатори, що дає змогу деанонімізувати авторів текстів та програмного коду, виявляти фейковий контент та встановлювати зв’язки між розрізненими кіберінцидентами. Дослідження ефективності та ресурсоємності таких методів є необхідним кроком для створення адаптивних систем захисту.

Мета дослідження: Метою роботи є підвищення ефективності вирішення задач кібернетичної та інформаційної безпеки в середовищі OSINT шляхом розробки та експериментальної оцінки методів стилометричного аналізу для атрибуції авторства природної мови та програмного коду.

Завдання дослідження:

1. Проаналізувати теоретичні засади стилометрії та визначити її роль у сучасному циклі розвідки з відкритих джерел.

2. Дослідити можливості застосування технологій машинного навчання та нейронних мереж для обробки природної мови та аналізу програмного коду.

3. Виявити проблематику атрибуції шкідливого програмного забезпечення в умовах обфускації та визначити ефективні методи ідентифікації розробника.

4. Провести порівняльний аналіз ресурсоємності класичних статистичних методів та сучасних моделей глибокого навчання

5. Експериментально перевірити стійкість стилометричних методів (зокрема Burrows' Delta) в умовах зміни семантичного домену та лінгвістичної інтерференції.

6. Оцінити ефективність застосування методів поведінкової біометрії для виявлення аномалій та підміни автора.

Об’єкт дослідження: Процес аналізу інформації з відкритих джерел для забезпечення кібернетичної та інформаційної безпеки.

Предмет дослідження: Методи, моделі та алгоритми стилометричного аналізу тексту та програмного коду, що використовуються для атрибуції авторства та виявлення аномалій.

Методи дослідження: аналіз та синтез, математична статистика , методи машинного навчання, експериментальний метод , порівняльний аналіз

Новизна одержаних результатів: Оцінено перевагу використання мікро-синтаксичних патернів (n-грам символів) над складними семантичними моделями NLP у задачах кодової форензики, що дозволяє ефективно ідентифікувати автора програмного коду, ігноруючи змістове навантаження.

Встановлено, що в умовах обмежених ресурсів та зміни семантичного домену класичні статистичні методи стилометрії демонструють вищу

стійкість та меншу схильність до перенавчання порівняно з нейромережевими підходами.

Запропоновано та обґрунтовано використання комбінації алгоритму One-Class SVM із символьними триграмами як ефективного методу поведінкової біометрії для виявлення підміни автора (ін'єкції чужого стилю).

Практичне значення одержаних результатів: Практична цінність роботи полягає у розробці рекомендацій щодо вибору підходів для атрибуції загроз у кіберпросторі. Результати досліджень можуть бути використані:

- аналітиками SOC та CSIRT для автоматизованої атрибуції шкідливого ПЗ та профілювання кіберзлочинних угруповань;
- фахівцями з інформаційної безпеки для виявлення скоординованих інформаційних атак, ботоферм та дезінформації;
- при розробці систем моніторингу, що потребують низької ресурсоємності (наприклад, на рівні IoT або кінцевих точок), завдяки доведеній ефективності оптимізованих статистичних моделей.

Апробація результатів роботи: Основні результати дослідження доповідалися та обговорювалися на Міжнародній науково-практичній конференції «SCIENCE, TECHNOLOGY AND CULTURE: INTERACTION, EVOLUTION AND PROGRESS» (21-23 грудня 2025 р., м. Копенгаген, Данія).

1 ТЕОРЕТИЧНІ ЗАСАДИ СТИЛОМЕТРІЇ ТА ЇЇ ІНТЕГРАЦІЯ В ПРОЦЕСИ OSINT

1.1 Знайомство зі стилометрією

Стилометрія - це галузь комп’ютерної лінгвістики, що досліджує індивідуальні особливості письмового стилю автора з метою його ідентифікації. В основі цього методу лежить ідея про те, що кожна людина має унікальний, майже несвідомий "стилістичний відбиток пальців", який проявляється у виборі слів, побудові речень, використанні розділових знаків [2] та інших текстових характеристиках.

Найперші відомі спроби кількісного аналізу текстів описані в роботах Рудмана[3]. Одним із найдавніших прикладів, датованим приблизно 300 р. до н.е., є дослідження Саунаки, присвячене структурним характеристикам давнього індійського релігійного корпусу -Ріг-Веди. Пізніше, близько 180 р. до н.е., Арістарх Самофракійський здійснив систематичний облік рідкісних та унікальних мовних конструкцій у грецьких текстах. У 950 р. н.е. Аарон Бен-Ашер провів обчислення різних текстових одиниць - літер, слів та речень - у біблійних текстах. Термін *«стилометрія»* був уведений Вінценти Лутославським у дослідженні[4], присвяченому хронологічному аналізу платонівських діалогів.

У межах сучасних досліджень стилометрія - тобто систематичне виділення та аналіз стилістичних характеристик письмового тексту - демонструє високу результативність у розв’язанні задач класифікації текстів. Найбільш поширеними сферами її застосування є:

- Атрибуція авторства з метою виявлення випадків видавання себе за іншу особу;

- Виявлення дезінформації, зокрема обманного характеру тексту, фейкових новин і неправомірного контенту.

У першому випадку стилометричні методи дозволяють визначити мовні особливості, характерні для конкретного автора або соціальної групи. У другому випадку аналіз ґрунтується на ідентифікації лінгвістичних маркерів, що відображають когнітивну специфіку створення неправдивого висловлювання.

Останні дослідження продемонстрували зростання інтересу до стилометрії в контексті ризиків, пов’язаних із використанням мовних моделей[6] для масового генерування шкідливого контенту, зокрема такого, що імітує стиль людського автора або містить недостовірну інформацію. Існуючі результати засвідчили, що стилометричні методи мають значний потенціал у виявленні текстів, створених мовними моделями з метою імітації людини. Однак із поширенням легітимних застосувань текстогенерації - таких як автодоповнення чи автоматичні системи відповідання - визначення факту машинного походження тексту перестає бути достатнім критерієм його надійності.

У контексті кібербезпеки стилометрія набуває особливого значення як інструмент для атрибуції кіберінцидентів, виявлення дезінформації, протидії соціальній інженерії та аналізу поведінки в цифровому просторі. Вона поєднує методи лінгвістичного аналізу для виявлення значущих ознак, статистики для їх кількісної оцінки, а також машинного навчання та штучного інтелекту для створення прогностичних моделей, здатних класифікувати тексти за авторством чи іншими характеристиками.

Стилометрія має на меті виявлення корисних атрибутів документів на основі стилю їх написання. Історично, одним із найвідоміших прикладів її успішного застосування є вирішення питання авторства Федералістських записок[7] у США. У сучасних умовах, статистичні методи успішно

визначали не лише особу автора, але й такі демографічні та психологічні риси, як стать, рідну мову, вік, рівень освіти і навіть наявність у автора деменції. Стилометричний аналіз є важливим для маркетологів, аналітиків та соціологів, оскільки він надає демографічні дані безпосередньо з необробленого тексту[8], що дозволяє сегментувати аудиторію та вивчати суспільні тенденції.

Зростає інтерес до застосування стилометрії до контенту, створеного користувачами інтернет-додатків. Це пов'язано з величезними масивами анонімних даних, які потребують аналізу. Наприклад, технологію використовують для визначення етнічної приналежності або політичних поглядів автора в соціальних мережах, що допомагає у вивченні поширення ідей. Іншим важливим напрямком є боротьба з шахрайством, зокрема з'ясування, чи пише людина оманливі онлайн-відгуки. Системи можуть виявляти "фабрики тролів" або мережі ботів, аналізуючи однотипність стилю у великій кількості коментарів. Таким чином, стилометрія перетворилася з академічної дисципліни на практичний інструмент для аналізу цифрових комунікацій та забезпечення безпеки в інформаційному просторі.

1.2 Порівняння стилометрії з дослідженням авторства

Хоча стилометрія та дослідження авторства є спорідненими галузями, вони мають як очевидні відмінності, так і важливі глибинні подібності. Метою стилістичного аналізу, що є відкритим і пошуковим, є висвітлення стильових закономірностей, які впливають на сприйняття читачів та стосуються питань літературної і лінгвістичної інтерпретації. На противагу цьому, дослідження авторства націлені на отримання однозначних рішень у існуючих проблемах, уникаючи, по можливості, помітних ознак і працюючи з базовими пластами мови, де можна виключити наслідування.

Атрибуція авторства має і прикладний аспект у криміналістиці - зокрема, докази авторства на основі стилістики приймалися в судах Британії[9], що посилило увагу до надійності методу через юридичну відповідальність. І все ж, для отримання нових знань, стилістичний аналіз має пройти перевірку на строгість, відтворюваність та неупередженість, аналогічну до аналізу авторства. А для того, щоб методи дослідження авторства завоювали довіру, їх необхідно обґрунтувати в стилістичних термінах.

Завданням стилометрії є виявлення мовних патернів, що стосуються процесів створення та сприйняття тексту (тобто «стилю» в широкому розумінні), і які стають очевидними лише завдяки комп'ютерним методам аналізу. Уявімо, наприклад, що ми хочемо дослідити зв'язки та відмінності між п'єсами Шекспіра, спираючись виключно на їхні діалоги. Нам відомі традиційні класифікації цих творів за жанром, хронологією (ранні, середні, пізні) та іншими категоріями, що обговорювалися критиками, як-от «проблемні п'єси» чи «великі трагедії». Але якою була б класифікація, якби ми відкинули ці упередження і поклалися лише на внутрішні, емпіричні дані? Для цього ми можемо взяти вибірку з двадцяти п'яти п'єс, визначити дванадцять найчастотніших у них слів і скласти таблицю, що відображає частоту вживання кожного з цих слів у кожному з творів.

Для аналізу варіативності в таких таблицях в стилометрії часто застосовують статистичний метод, відомий як аналіз головних компонент[10]. Його суть полягає у створенні нових, композитних змінних, що є комбінаціями початкових. Це дозволяє спростити масив даних, визначивши декілька ключових змінних, які описують основні взаємозв'язки. Для прикладу, якщо дані про зріст і вагу людей тісно корелюють, їх можна об'єднати в одну змінну «розмір», яка відобразить більшу частину варіації. «Головні компоненти» діють за цим же принципом: алгоритм послідовно виділяє компоненти, що пояснюють найбільшу частку дисперсії. Якщо у

вихідних даних є сильні кореляції, вже перші кілька компонентів опишуть основні тенденції.

1.3 Сфери прикладного застосування стилометрі

1.Криміналістика

В основі стилометричного аналізу лежить гіпотеза про те, що кожен автор має унікальний і відносно стабільний стиль[11] письма (ідіолект), який проявляється у несвідомому та послідовному виборі лінгвістичних елементів. Ці елементи, що виступають у ролі стилістичних маркерів, включають лексичні, синтаксичні, структурні та навіть ідіосинкратичні особливості тексту. На відміну від свідомих стилістичних рішень, які автор може легко змінювати, несвідомі патерни, такі як частота вживання функціональних слів (артиклів, прийменників, сполучників), довжина слів та речень, є більш надійними індикаторами авторського стилю. Саме ці високочастотні та неусвідомлені характеристики є найменш схильними до навмисної маніпуляції, що робить їх ключовими для криміналістичної атрибуції.

Методологія стилометричного аналізу

Криміналістична стилометрія використовує кількісні методи для аналізу та порівняння текстів. Процедура зазвичай включає наступні етапи:

1. Формування корпусу текстів: Створюється набір текстів, що включає анонімний документ (текст-загадку) та тексти-зразки, написані потенційними авторами. Важливою вимогою є зіставність текстів за жанром, темою та часом написання для забезпечення надійності порівняння.

2. Токенізація та виокремлення ознак: Тексти розбиваються на окремі одиниці (токени), якими можуть бути слова, символи або n-грами (послідовності з n слів чи символів). На цьому етапі

розраховується частота появи найбільш вживаних слів, що є однією з найпоширеніших та найефективніших методик.

3. Статистичний аналіз та візуалізація: Для порівняння частотних профілів текстів застосовуються різноманітні методи багатовимірної статистики.

Стилометричний аналіз успішно застосовується у розслідуванні широкого спектру злочинів. Наприклад, у справі, пов'язаній з листами, надісланими італійським податковим органам, що містили погрози та кулі, було використано саме Дельта-метод. Для аналізу було взято 100 найуживаніших слів з текстів підозрюваного та анонімних листів. Результати кластерного аналізу та розрахунку Дельти однозначно вказали на одного з підозрюваних як на автора загрозливих повідомлень, що стало вагомим доказом у суді.

2. Літературознавство та історична атрибуція

Історично стилометрія зародилася саме як інструмент філологічного аналізу, і сьогодні вона продовжує відігравати центральну роль у вирішенні багатовікових спорів щодо авторства літературних та історичних пам'яток. Застосування статистичних методів, таких як аналіз головних компонент та дельта-метод Берроуза, дозволило науковцям з високою точністю атрибутувати спірні статті у «Федералістських записках», розмежувати внесок співавторів у п'єсах епохи Єлизавети та виявити псевдоніми відомих письменників. У сучасному контексті ці методи трансформувалися в інструменти distant reading, що дозволяють аналізувати еволюцію літературних жанрів та стилів на масивах із тисяч текстів одночасно. Це дає змогу дослідникам об'єктивно фіксувати хронологічні зміни у мові автора, відокремлювати оригінальні фрагменти від пізніших редакційних вставок та відновлювати історичну справедливість у питаннях інтелектуальної власності минулих епох.

3. Маркетинг та соціологічне профілювання

У комерційному секторі стилометричні технології інтегровано в системи бізнес-розвідки та маркетингової аналітики для глибокого психографічного профілювання клієнтської бази. Аналізуючи стиль відгуків, коментарів та дописів у соціальних мережах, компанії отримують можливість автоматизовано сегментувати аудиторію за психотипами, визначати рівень лояльності та прогнозувати поведінкові реакції без проведення прямих опитувань[8]. Особливої актуальності набуло використання стилометрії для забезпечення чистоти інформаційного простору брендів, зокрема для виявлення «астротурфінгу» - штучно організованих кампаній з написання замовних відгуків. Алгоритми здатні детектувати неприродну однорідність синтаксичних патернів у масиві коментарів, що дозволяє фільтрувати контент, згенерований ботофермами або найманими копірайтерами, захищаючи репутацію бізнесу від недобросовісної конкуренції.

4. Психологія та медична діагностика

Інноваційним напрямом застосування стилометрії є клінічна психологія та психіатрія, де кількісний аналіз мовлення розглядається як біомаркер когнітивного стану пацієнта. Дослідження підтверджують, що зміни у синтаксичній складності, збідніння словникового запасу та специфічне використання службових частин мови можуть свідчити про ранні стадії нейродегенеративних захворювань, таких як хвороба Альцгеймера чи деменція[2], ще до появи виражених клінічних симптомів. Також стилометричні маркери ефективно використовуються для моніторингу емоційного стану, дозволяючи виявляти ознаки депресії, біполярного розладу або суїцидальних нахилів у текстах пацієнтів. Це відкриває перспективи для створення систем автоматизованого скринінгу ментального здоров'я, що базуються на аналізі повсякденної цифрової комунікації людини.

1.4 Математичні методи та алгоритми стилометричного аналізу

Стилометрія позиціонується як наукова галузь, що функціонує на перетині лінгвістики, математичної статистики та інформатики, маючи на меті атрибуцію авторства шляхом кількісного аналізу вимірюваних стилістичних характеристик. Фундаментальним підґрунтям стилометричних досліджень виступає гіпотеза про наявність у кожного автора унікального та стабільного мовленнєвого почерку - ідіолекту, який проявляється у формі вимірюваних патернів текстотворення, що формують своєрідний «стилістичний відбиток». Методологічний базис даного підходу ґрунтується на постулатах індивідуальності лексико-синтаксичних виборів та підсвідомому характері їх використання, оскільки такі маркери, як частота вживання службових слів, довжина речень чи специфіка пунктуації, є результатом автоматизованих навичок і важко піддаються свідомому контролю або імітації. Завдяки цьому стилістичні особливості підлягають формалізації та перетворенню на набори числових векторів ознак, що робить їх придатними для статистичної обробки та аналізу за допомогою алгоритмів машинного навчання в рамках стандартизованого, об’єктивного процесу атрибуції.

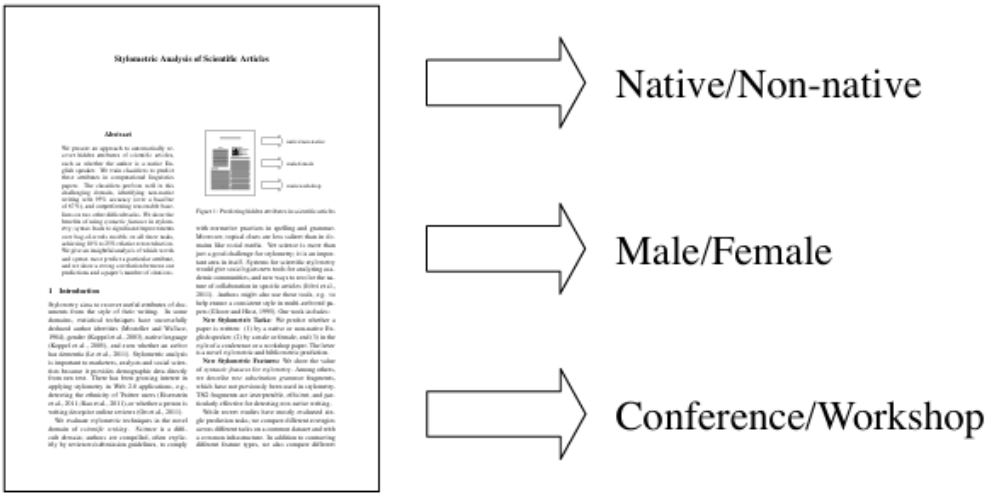

Рисунок 1.1 – Схема процесу стилометричної класифікації текстів

1. Формування Корпусу Текстів

Навчальний корпус: Включає тексти, авторство яких достеменно відоме. Цей корпус використовується для "навчання" моделі розпізнавати унікальні стилістичні риси кожного автора.

Тестовий корпус: Містить один або кілька анонімних чи спірних текстів, авторство яких необхідно встановити.

2. Попередня Обробка Тексту

Тексти корпусу проходять етап очищення та нормалізації для усунення "шуму" та уніфікації даних. Цей процес може включати:

Токенізацію: Розбиття суцільного тексту на окремі елементи – слова, розділові знаки, числа.

Лематизацію або стемінг: Приведення слів до їхньої базової форми або відсікання закінчень для агрегації статистики за одним словом, незалежно від його граматичної форми.

Видалення стоп-слів: Виключення з аналізу слів, що не несуть значного стилістичного навантаження (наприклад, займенники, вигуки), хоча в деяких методах саме частота службових слів є ключовою ознакою.

3. Екстракція Стилометричних Ознак

Це ключовий етап, на якому з тексту вилучаються кількісні характеристики. Набір цих ознак може бути надзвичайно широким і включає:

Лексичні ознаки:

Розподіл частоти слів (особливо найуживаніших), коефіцієнт лексичного розмаїття (співвідношення унікальних слів до загальної кількості), середня довжина слова, частота вживання слів певної довжини.

Синтаксичні ознаки:

Середня довжина речення, розподіл речень за довжиною, частота та типи розділових знаків, Складність речень (глибина вкладеності підрядних частин),

Символьні ознаки:

Частота N-грам символів[10] (послідовностей з N символів), що дозволяє фіксувати не лише слова, але й характерні морфологічні та орфографічні патерни.

Морфологічні ознаки:

Частота вживання певних частин мови (іменників, дієслів, прикметників).

4. Статистичний Аналіз та Побудова Моделі

На цьому етапі кожен текст з корпусу перетворюється на числовий вектор, де кожне значення відповідає певній стилометричній ознаці. Для аналізу цих даних та встановлення авторства застосовуються різноманітні кількісні методи.

Методи кластеризації: Алгоритми (наприклад, ієрархічна кластеризація) групують тексти на основі подібності їхніх стилістичних векторів, візуалізуючи, які тексти є найближчими один до одного за стилем.

Методи класифікації: Використовуються алгоритми машинного навчання (Support Vector Machines, нейронні мережі, метод найближчих сусідів та інші), які навчаються на тренувальному корпусі розпізнавати "стилістичні класи" (авторів). Навчена модель потім прогнозує, до якого класу (автора) належить тестовий текст.

Дельта Берроуз: Один із найвідоміших та найнадійніших методів в атрибуції авторства[12]. Він обчислює "стилістичну відстань" між анонімним текстом та текстами відомих авторів на основі Z-оцінок частот

найуживаніших слів. Текст приписується тому автору, до стилю якого він є "найближчим".

5. Валідація та Інтерпретація Результатів

Результати аналізу представляються у вигляді статистичних звітів, дендрограм для кластеризації або ймовірнісних оцінок приналежності до певного автора. Важливим є етап валідації, коли точність моделі перевіряється на тестових даних з відомим авторством, перш ніж застосовувати її до справді анонімних текстів.

1.5 Завдання OSINT

До ключових етапів проведення OSINT-розвідки належить:

Постановка мети та планування - залежно від об’єкта пошуку (відомості, що підлягають встановленню та аналізу здійснюється підбір ключових джерел та визначається набір інструментів для проведення дослідження з урахуванням етичних та правових норм використання та обробки інформації.

Збір даних - процес формування масиву даних, що підлягають аналізу. Зокрема, сюди можна зарахувати:

- інтернет-ресурси (вебсайти, блоги, форуми, соціальні мережі);
- медіа (ЗМІ, новинні портали, відео, радіо та друковані видання);
- урядові та офіційні документи (звіти, бази даних, судові рішення тощо);
- технічні джерела (IP-адреси, доменні реєстрації, метадані файлів).

Інструменти, що використовуються на цьому етапі: пошукові системи, спеціалізовані OSINT-платформи (наприклад, Maltego, Shodan, Spiderfoot[15]), аналіз соціальних мереж (LinkedIn, Facebook, Instagram) тощо.

Обробка та фільтрація інформації - здійснюється перевірка достовірності джерел і зібраних даних, виділення ключових фактів і відсіювання зайвої або хибної інформації, структурування даних для подальшого аналізу

Аналіз даних - за результатами отриманих даних аналітики шукають взаємозв’язки між зібраною інформацією, виявляють шаблони, тенденції або аномалії у поведінці суб’єктів дослідження. Використовують інструменти візуалізації для полегшення розуміння (наприклад, побудова графів зв’язків).

Інтерпретація та висновки - на цьому етапі формуються висновки на основі аналізу. Проводиться оцінка впливу отриманих даних на вирішення завдання, розробляються рекомендації або сценарії подальших дій.

Оформлення звіту - підготовка зрозумілого та структурованого звіту, що містить результати розвідки, графіки, таблиці та висновки, із зазначенням використаних джерел, з метою забезпечення прозорості та можливості верифікації даних.

Перевірка та оцінка результатів - здійснюється перевірка точності отриманих даних перед їх використанням, а також аналіз ефективності обраних методів збору та обробки інформації для покращення майбутніх досліджень.

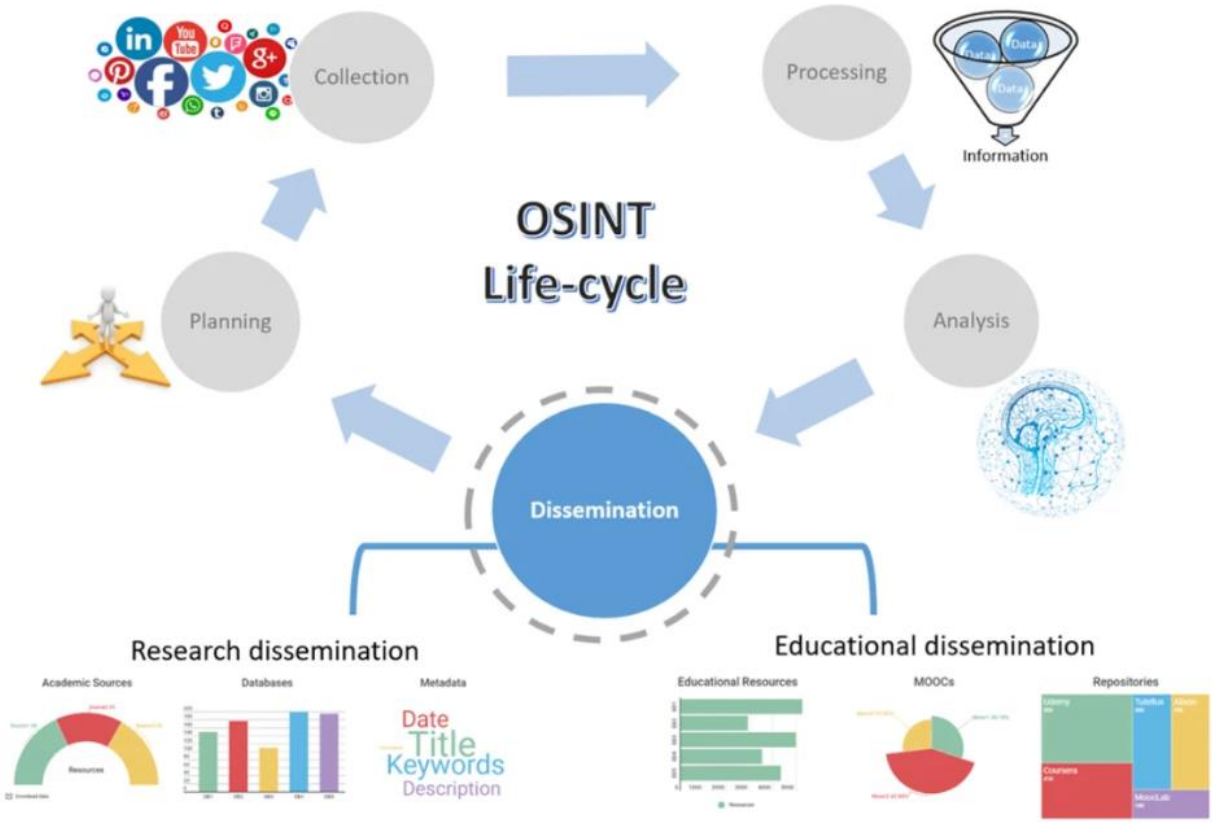

Рисунок 1.2 – Цикл розвідки на основі відкритих джерел

OSINT перетворився з допоміжного інструменту на повноцінну аналітичну дисципліну. Це методичний процес збору, обробки та аналізу загальнодоступної інформації[13] для отримання верифікованих знань, здатних підтримати прийняття стратегічних рішень. У контексті гібридної війни та глобалізації інформаційного простору, де традиційні кордони між публічним і приватним секторами дедалі більше розмиваються, а масиви даних генеруються у швидко зростаючих масштабах, внесок розвідки відкритих джерел у національну безпеку, правоохоронну діяльність та корпоративну розвідку стає справді критичним. Функціонування цієї системи ґрунтується на належній методології, вбудованій у розвідувальний цикл (визначення мети, збір, обробка, аналіз, розповсюдження), через вирішення низки стандартних та типових завдань, що складають основу більшості розслідувань. Ці завдання є не ізольованими, непов'язаними діями, а скоріше взаємопов'язаними модулями, які дозволяють здійснити цілісну

реконструкцію вздовж континууму від припущень до встановлення реальності на основі цифрових слідів.

Саме комплексне профілювання ключових об'єктів інтересу - часто фізичних та юридичних осіб - становить центр у практичному застосуванні відкритої розвідки.

Профілювання осіб - це базове завдання, яке виходить далеко за межі простого збору біографічних відомостей. Це процес інформаційного синтезу, покликаний створити багатовимірний портрет індивіда. Цей портрет у своїх численних вимірах включає ідентифікаційні дані, соціальні зв'язки, професійну діяльність, фінансовий стан та поведінкові патерни. Розслідування починається з ідентифікаторів, які можна відстежити: повного імені, дати народження, відомих псевдонімів або позивних. Виявлення кожного фрагмента інформації забезпечує «опорну точку» для подальших досліджень: номери телефонів перевіряються через месенджери та бази даних, що може розкрити зображення профілю та ім'я користувача; електронна пошта може бути перевірена на предмет витоків даних, що потенційно розкриває паролі та реєстрації на різноманітних форумах. Значна частина роботи включає вивчення цифрового сліду в приватних соціальних мережах та професійному середовищі. Це дозволяє реконструювати не лише професійний та освітній шлях, а й скласти карту соціального кола, включаючи сім'ю, друзів, колег та бізнес-партнерів. Аналіз дерева публікацій, коментарів, вподобань та підписок дає уявлення про політичні погляди, інтереси, спосіб життя та навіть психологічний профіль. Інша частина стосується фінансової розвідки: пошук у відкритих реєстрах декларацій, рухомого та нерухомого майна, часткової участі в компаніях. Це завдання набуває особливої актуальності у протидії ворожим діям, оскільки допомагає ідентифікувати воєнних злочинців, колаборантів, агентів впливу та членів диверсійно-розвідувальних груп.

Корпоративне профілювання є досить схожим на корпоративну розвідку та комплексну перевірку і має на меті надати повну картину діяльності компанії чи організації. Цей процес вимагає аналізу не лише статичних реєстраційних даних (код ЄДРПОУ, юридична адреса, засновники), а й динамічних аспектів її функціонування. Абсолютно критичним є виявлення справжніх Кінцевих Бенефіціарних Власників, які часто приховані за ланцюгом офшорних компаній та номінальних директорів. Аналіз державних контрактів, судових процесів, виконавчих проваджень та податкових боргів дозволяє оцінити фінансове здоров'я та доброчесність корпорації. Подальший аналіз спрямований на викриття прихованих афілійованих зв'язків, відстеження ланцюгів та аналіз зовнішньоекономічної діяльності. Для України це питання є особливо важливим у контексті виявлення компаній, що співпрацюють з агресором, обходять санкційні обмеження або відмивають кошти.

Якщо профілювання передусім передбачає роботу з відносно статичною інформацією, то ситуаційний аналіз має справу з динамічними процесами та вимагає оперативних дій.

Геолокація та хронолокація - це класичні завдання розвідки за відкритими джерелами, що передбачають точне визначення місця фото- чи відеозйомки, а також часових аспектів цих подій. Це складний аналітичний процес, що базується на ідентифікації унікальних візуальних орієнтирів, будь то рукотворні (архітектурні) споруди, природні ландшафти або лінії електропередач, та кореляції їх із супутниковими знімками, картами та вуличними панорамами. Час і дата визначаються шляхом аналізу тіней, погодних умов, сезонних змін рослинності та інших часових маркерів. Ця методологія є критично важливою для верифікації переміщення військової техніки, підтвердження доказів воєнних злочинів та моніторингу подій на окупованих територіях.

Верифікація контенту та викриття дезінформації знаходяться на перетині технічного аналізу та критичного мислення. Це передбачає встановлення автентичності інформації шляхом спроб знайти першоджерела через пошук зображень, аналіз метаданих файлів, виявлення цифрових маніпуляцій (наприклад, за допомогою аналізу помилок) та оцінку надійності джерела. В епоху гібридної війни це завдання переросло у стратегічний аналіз дезінформаційних кампаній. Метою тепер є не лише спростування окремого фейку, а й ідентифікація ворожих наративів, ключових каналів поширення (від контрольованих державою ЗМІ до анонімних Telegram-каналів), цільових аудиторій та кінцевих цілей кампанії.

Моніторинг та трекінг передбачають довготривале спостереження за об'єктом або процесом. Наприклад, це може включати відстеження переміщень морських суден через автоматичну ідентифікаційну систему для моніторингу вивезення викраденого українського зерна, або спостереження за активністю на військових базах за допомогою регулярних супутникових знімків, або аналіз активності певних груп у соціальних мережах для прогнозування соціальних заворушень.

Принципи використання OSINT

1. Доброчесність

Передбачає орієнтацію на пошук істини та надання виключно об'єктивної, неупередженої інформації. Усі дії аналітика мають бути спрямовані на підтримку високих стандартів професії та зміцнення довіри до результатів розвідки з відкритих джерел.

2. Законність

Суворе дотримання вимог чинного законодавства, міжнародних нормативно-правових актів та внутрішніх регламентів у процесі здійснення розвідувальної діяльності. Цей принцип вимагає безумовної поваги до

приватності , громадянських свобод та зобов'язань у сфері прав людини під час збору та обробки даних.

3. Прозорість

Чітка ідентифікація та атрибуція першоджерел, використаних у ході OSINT-дослідження. Водночас необхідно забезпечувати захист конфіденційних джерел та специфічних методів збору інформації у випадках, коли їх розкриття може створити загрози безпеці операції чи суб'єктів.

4. Ретельність та професійна компетентність

Забезпечення високої якості аналітичного продукту шляхом застосування верифікованих найкращих практик та політик. Пріоритетом є постійне підвищення точності, своєчасності та релевантності зібраної інформації.

5. Професійний розвиток

Безперервне вдосконалення кваліфікації та оновлення знань щодо новітніх інструментів і технологій аналізу. Важливою складовою є співпраця з професійною спільнотою та обмін досвідом для розвитку методологічної бази OSINT.

Активне застосування методології OSINT надає організаціям можливість ідентифікувати та нейтралізувати вразливості ще до моменту їх експлуатації зловмисниками. У рамках цього процесу виділяють такі ключові напрями діяльності:

1. Управління поверхнею атаки

Системна ідентифікація та безперервний моніторинг усіх зовнішніх цифрових активів організації, що можуть слугувати векторами кібератак. Фахівці з кібербезпеки застосовують інструментарій OSINT для виявлення тіньових елементів IT-інфраструктури: забутих піддоменів, відкритих мережевих портів, незахищених хмарних сховищ, а також фрагментів коду, що потрапили у відкритий доступ на платформах типу GitHub. Додатково проводиться аналіз SSL-сертифікатів, DNS-записів та даних пошукових

систем для підключених пристроїв (наприклад, Shodan) з метою виявлення потенційних вразливостей периметру мережі.

2. Розвідка кіберзагроз

Збір, обробка та аналіз розвідувальних даних щодо актуальних та прогнозованих кіберзагроз, суб'єктів атак (actors), їхньої мотивації та технічного інструментарію.

Реалізація цього завдання передбачає моніторинг спеціалізованих ресурсів: хакерських форумів у Dark Web, тематичних Telegram-каналів, соціальних мереж та інших комунікаційних платформ, де кіберзлочинні синдикати обговорюють нові вектори атак, реалізують викрадені дані або координують спільні дії. Особлива увага приділяється відстеженню активності організованих угруповань (Advanced Persistent Threat, APT) та вивченню їхніх тактик, технік і процедур (TTPs), що дозволяє розробляти превентивні механізми захисту.

3. Управління ризиками третіх сторін

Аудит стану інформаційної безпеки контрагентів, партнерів та вендорів, які мають санкціонований доступ до корпоративних даних або мережевої інфраструктури.

Засоби OSINT використовуються для реконструкції цифрового сліду компанії-партнера. Аналіз включає виявлення витоків даних, пов'язаних із доменом контрагента, пошук вразливостей на його публічних веб-ресурсах та оцінку репутації у відкритих джерелах. Такий підхід дозволяє ідентифікувати слабкі ланки в ланцюгу постачання ще на етапі попередніх переговорів до підписання контрактів.

4. Детекція скомпрометованих облікових записів та витоків даних

Оперативне виявлення інцидентів витоку конфіденційної інформації (облікових даних співробітників, клієнтських баз, комерційної таємниці) у публічний простір.

Спеціалізовані OSINT-системи забезпечують автоматизований моніторинг

сегментів Dark Net, ресурсів для обміну текстом (paste-сайти, наприклад, Pastebin) та форумів, де публікуються масиви викрадених даних (breach dumps). Це забезпечує можливість швидкого реагування на інциденти, примусової зміни скомпрометованих паролів та мінімізації репутаційних і фінансових збитків.

1.6 Стратегічне застосування OSINT

У пост-інцидентній фазі інструментарій OSINT трансформується у критично важливий елемент цифрової форензики та комплексного розслідування. Пріоритетним завданням на цьому етапі виступає атрибуція кібератак, що передбачає збір доказової бази, ретроспективну реконструкцію хронології подій та ідентифікацію мережевої інфраструктури зловмисників.

Методологічний базис процесу ґрунтується на виявленні та верифікації індикаторів компрометації - IP-адрес, доменних імен, хеш-сум шкідливих файлів. Шляхом крос-кореляції отриманих даних із відкритими базами загроз встановлюються зв'язки з відомими кіберугрупованнями (APT-групами). Паралельний аналіз цифрових слідів у публічному просторі створює передумови для деанонімізації суб'єктів атаки. Крім того, OSINT забезпечує команду реагування (CSIRT/CERT) оперативними даними щодо функціональних особливостей шкідливого програмного забезпечення та ефективних контрзаходів, що суттєво прискорює локалізацію загрози.

Імплементація зазначених методів вимагає суворого дотримання етичних норм та правових рамок, зокрема законодавства у сфері захисту персональних даних та приватності. Перспективи розвитку напряму пов'язані з автоматизацією аналітичних процесів на базі штучного інтелекту та поглибленням інтеграції модулів OSINT із системами управління подіями безпеки (SIEM) та оркестрації (SOAR). В умовах експоненційного зростання обсягів дезінформації та вдосконалення технік протидії розвідці, це висуває

підвищені вимоги до адаптивності та професійних компетенцій сучасних фахівців з кібербезпеки.

1.7 Інтеграція стилометричних методів у процеси OSINT

У контексті сучасних інформаційних систем питання оперативного виявлення загроз, аналізу інформаційних операцій та забезпечення кібернетичної безпеки набувають надзвичайного значення. Інструменти розвідки з відкритих джерел дозволяють отримувати цінні дані з публічних ресурсів, що використовуються як засіб превентивної дії, реагування на інциденти та аналізу кіберзлочинної активності. У цьому контексті стилометрія - наука про кількісний аналіз стилю мовлення та письма - може виступати додатковим методом, що підсилює можливості OSINT, особливо коли мова йде про ідентифікацію авторства, виявлення координації інформаційних атак та аналіз поведінкових патернів зловмисників.

Стилометрія дозволяє виявляти унікальні мовні та стилістичні ознаки, які відрізняють одного автора або групу авторів від інших, на основі аналізу частотності слів, синтаксичних конструкцій, пунктуаційних звичок, вибору лексики тощо. У межах OSINT-аналізу це дає змогу:

Підтвердити чи спростувати гіпотезу про те, що кілька публікацій[19] або повідомлень створені однією особою або групою.

Виявити можливі координовані дії, коли різні акаунти чи ресурси використовують подібні стилістичні патерни, які свідчать про централізовану інформаційну операцію.

Додатково профілювати автора чи групу авторів: наприклад, з’ясувати рівень технічної або мовної компетентності, можливий регіон походження, культурно-мовні ознаки, що можуть бути присутні у текстах.

Сучасні OSINT-інструменти використовують широкий спектр даних: соціальні мережі, реєстри доменів, витоки даних, кодові репозиторії, темні веб-форуми тощо. s Було б ефективно розгорнути процес, у якому стилометрія стає складовою частиною аналітичної ланки OSINT:

На етапі збору даних: стилометричний аналіз може націлити інструменти OSINT на підозрілі акаунти або ресурси, які стилістично подібні до відомих зловмисників.

На етапі обробки та аналізу: отримані тексти можуть бути порівняні зі стилістичною базою даних авторів, наведених у минулих кіберінцидентах, щоб встановити кореляції.

На етапі атрибуції та профілювання: результати стилометричного аналізу можуть стати доказовою базою для формування висновків щодо авторства, координації або навіть мотивів інформаційних атак.

Атрибуція кіберінцидентів - коли, наприклад, рішення ОСИНТ-аналізу виявляє витік або атаку, стилометрія може допомогти пов’язати текстові сигнали, які супроводжували атаку (наприклад, погрози чи публікації на форумах) з конкретними авторами або угрупованнями.

Виявлення соціальної інженерії та бот-мереж - стилістична схожість між повідомленнями різних акаунтів може вказувати на бот-мережу або фабрику тролів. OSINT-контекст дає джерела, стилометрія - методику аналізу.

Боротьба з дезінформацією - у випадках інформаційних кампаній, що ґрунтуються на масовій публікації маніпулятивних текстів, стилометричний аналіз дозволяє виявляти схожість між раніше невзаємопов’язаними повідомленнями, що підвищує шанси на розпізнавання координованої операції.

Профілювання загрозоорієнтованих авторів - стилометрія дозволяє виявляти мовні маркери, які можуть свідчити про регіон автора, його

культурно-мовні звички або попередній досвід. Комбіновано з даними OSINT це створює багатогранний профіль потенційного актора.

Попри значний потенціал, використання стилометрії в OSINT-контексті має й власні складнощі:

Обсяг даних, доступних у відкритих джерах, є величезним, і ризик «шуму» (інформаційного перевантаження) високий.

Точність стилометричного аналізу залежить від якості та кількості зразків текстів, що можуть бути доступні для порівняння.

Актори кіберзлочинності свідомо можуть змінювати свій стиль[20] – використовуючи маскування, різні акаунти, машинну генерацію тексту – що ускладнює атрибуцію на базі стилю.

Правові та етичні рамки залишаються критичними – збирання, зберігання та аналіз персональних чи публічних даних повинні відповідати нормам приватності, етичним стандартам та бути належно документованими.

Висновок до розділу 1

На основі аналізу теоретичних засад стилометрії встановлено, що вона трансформувалася з філологічної дисципліни в ефективний інструмент кібернетичної безпеки. Визначено, що інтеграція стилометричних методів у цикл розвідки з відкритих джерел дозволяє вирішувати задачу атрибуції кіберзагроз на якісно новому рівні — шляхом ідентифікації стійких когнітивних патернів автора, які неможливо приховати стандартними засобами анонімізації.

Обґрунтовано, що для забезпечення достовірності результатів в умовах високого рівня інформаційного шуму найбільш доцільним є використання математичних методів, зокрема Дельти Берроуза та багатовимірного статистичного аналізу. Це дозволяє об’єктивізувати процес профілювання

зловмисників, виявляти скоординовані інформаційні операції та дезінформацію, що створює методологічне підґрунтя для подальшої практичної реалізації системи автоматизованої атрибуції.

2 ТЕХНОЛОГІЇ МАШИННОГО НАВЧАННЯ ТА АНАЛІЗ ПРОГРАМНОГО КОДУ В OSINT

2.1 Роль OSINT у сучасних інформаційних операціях

Міжнародна спільнота дедалі частіше використовує інформацію з відкритих джерел для вирішення широкого спектра завдань. Зокрема, роль OSINT під час реалізації інформаційних операцій визначається набором аспектів, включаючи ефективність інформаційного потоку, обсяг, ясність, простоту подальшого використання, вартість отримання тощо

Більшість необхідних довідкових матеріалів щодо об'єктів інформаційних операцій збирається з відкритих джерел, і ця база будується шляхом збору інформації зі ЗМІ, що робить накопичення даних ключовою функцією OSINT.

Доступність, глибина та масштаб загальнодоступної інформації дозволяють знаходити необхідні відомості без залучення спеціалізованих людських та технічних розвідувальних засобів.

OSINT надає необхідну інформацію, усуваючи потребу в залученні надлишкових технічних та агентурних методів розвідки.

Будучи частиною розвідувального процесу, OSINT дозволяє менеджерам виконувати глибокий аналіз публічно доступної інформації для прийняття відповідних рішень.

Різке скорочення часу доступу до інформації в Інтернеті та зменшення кількості людино-годин, витрачених на пошук інформації, людей та їхніх взаємозв'язків.

Швидке отримання цінної релевантної інформації особливо важливе, оскільки раптові зміни ситуації під час криз найбільш детально

відображаються в актуальних новинах; наприклад, падіння Берлінської стіни спостерігали як у Вашингтоні, так і в штаб-квартирі CIA в Ленглі не через звіти розвідувальних служб, а через екрани телевізорів, що транслювали репортажі CNN безпосередньо з місця подій. Обсяг: можливість масового моніторингу певних джерел інформації, призначених для пошуку необхідного контенту, людей та подій. Досвід показує, що професійно зібрані фрагменти інформації з відкритих джерел, якщо брати їх у цілому, можуть виявитися еквівалентними або навіть більш значущими, ніж професійні розвідувальні звіти. Якість: у порівнянні зі звітами спеціальних агентів, інформація з відкритих джерел виявляється більш бажаною, оскільки вона є неупередженою та не змішаною з брехнею. Ясність: у випадках використання OSINT достовірність відкритих джерел може бути як зрозумілою, так і незрозумілою, тоді як у випадку таємно отриманих даних їхня надійність завжди сумнівна. Зручність використання: будь-які таємниці мають бути захищені бар'єрами класифікації та обмеженим доступом, натомість дані OSINT можуть бути легко передані будь-яким зацікавленим організаційним структурам, що дає можливість проводити комплексні дослідження на основі даних з Інтернету. Вартість отримання даних через OSINT є мінімальною і визначається лише ціною використовуваної послуги.

Головним бонусом інструментів OSINT є можливість знаходити та перевіряти всю необхідну інформацію за допомогою програмного забезпечення. Наприклад, продукт Social Links потребує однієї години для збору такого обсягу даних з відкритих джерел, який кваліфікований працівник збирав би вручну протягом тижня. З Social Links можна видобувати дані з понад 50 соціальних мереж, баз даних та використовувати понад 700 методів пошуку, посилених Face Recognition та пошуком за Geo-coordinates. Ви отримаєте унікальні можливості пошуку на понад 30 форумах і маркетплейсах DarkNet без авторизації за Phrase, PGP Key, Alias, також можна отримати аналітику за Products та Locations. Малий бізнес також,

можливо, має найбільше втрат від руйнівних кібератак. Нещодавній звіт виявив, що підприємства з менш ніж 500 співробітниками втрачають у середньому 2,5 мільйона доларів за одну атаку. Втрата такої суми грошей під час кіберзламу є руйнівною для малого бізнесу, не кажучи вже про репутаційну шкоду, яка виникає внаслідок кібератаки. З цих причин малому бізнесу необхідно знати про загрози та способи їх зупинки. У цій статті розглянуто топ-5 загроз безпеці, з якими стикаються компанії, та способи захисту організацій від них.

Фішінгові атаки є найбільш руйнівною та найпоширенішою загрозою, з якою стикається малий бізнес. На фішингові атаки припадає 90% усіх зламів, з якими стикаються організації, вони зросли на 65% за останній рік і становлять понад 12 мільярдів доларів збитків для бізнесу. Phishing Attacks відбуваються, коли зловмисник видає себе за довірену контактну особу та спонукає користувача натиснути шкідливе посилання, завантажити шкідливий файл або надати доступ до конфіденційної інформації, даних облікового запису чи облікових даних. Malware Attacks є другою великою загрозою, з якою стикається малий бізнес. Вона охоплює різноманітні кіберзагрози, такі як трояни та віруси. Це узагальнюючий термін для шкідливого коду, який хакери створюють для отримання доступу до мереж, крадіжки даних або знищення даних на комп'ютерах. Malware зазвичай надходить через завантаження зі шкідливих веб-сайтів, спам-листи або через підключення до інших інфікованих машин чи пристроїв. Ransomware є однією з найпоширеніших кібератак, що вражає тисячі компаній щороку. Останнім часом вони стали більш поширеними, оскільки є однією з найприбутковіших форм атак. Ransomware передбачає шифрування даних компанії, щоб їх не можна було використовувати або отримати до них доступ, а потім примушування компанії сплатити викуп для розблокування даних. Це ставить бізнес перед важким вибором – сплатити викуп і потенційно втратити величезні суми грошей або паралізувати свої послуги через втрату даних.

Weak Passwords є ще однією загрозою, з якою стикається малий бізнес, коли співробітники використовують слабкі паролі або такі, що легко вгадуються. Багато малих підприємств використовують численні хмарні сервіси, які вимагають різних облікових записів. Ці сервіси часто можуть містити конфіденційні дані та фінансову інформацію. Використання паролів, які легко вгадати, або використання одних і тих самих паролів для кількох облікових записів може призвести до компрометації цих даних.

2.2 Основні завдання OSINT у кібербезпеці

Інформація сама по собі не є розвідданими вона набуває цього статусу лише внаслідок критичного опрацювання та контекстуалізації. Спираючись на модель, запропоновану Efim діяльність у сфері OSINT визначається як систематизований алгоритм, що забезпечує трансформацію сирих масивів публічних даних у верифіковане знання для підтримки прийняття рішень. Цей процес реалізується через п’ять послідовних та взаємопов'язаних етапів

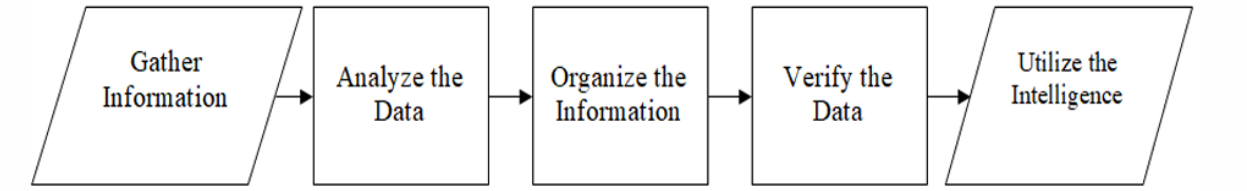

Рисунок 2.1 – Етапи трансформації даних у розвідувальну інформацію

1. Збір інформації

Цей етап виступає фундаментальною основою розвідувального циклу та, згідно з текстом дослідження, визначається як процес акумуляції даних із загальнодоступних джерел. Він передбачає використання широкого спектра інструментів, включаючи інтернет-пошукові системи (такі як Google, Shodan, Intelligence X) для роботи з цифровим простором, а також опрацювання традиційних носіїв інформації, таких як друковані видання, наукові журнали та офіційні звіти. Ключовою метою цієї стадії є всебічний моніторинг

інформаційного поля та первинне накопичення масиву "сирих" даних для їх подальшого збереження та дослідження.

2. Аналіз даних

На цій стадії відбувається первинна аналітична обробка зібраного матеріалу, що в методології дослідження корелює з процесом селекції та конвертації даних. Головне завдання етапу полягає у фільтрації інформаційного потоку шляхом відокремлення нерелевантних або надлишкових відомостей від цільової інформації. Процедура передбачає перетворення неструктурованих даних у змістовні формати, що дозволяє підготувати інформаційну базу для подальших логічних операцій та виявити потенційну цінність зібраних матеріалів.

3. Організація інформації

Цей етап передбачає систематизацію та структурування оброблених даних, що є критично важливим для ефективного індексування та каталогізації. У рамках цього процесу здійснюється компіляція відомостей з метою виокремлення метаданих та специфічних атрибутів об'єктів дослідження. Така організація масивів даних створює необхідні передумови для проведення кореляційного аналізу, дозволяючи дослідникам встановлювати приховані взаємозв'язки між різними інформаційними фрагментами та готувати ґрунт для глибокої аналітики.

4. Верифікація даних

Етап верифікації, що в тексті також класифікується як експлуатація даних, спрямований на забезпечення надійності та валідності розвідувального продукту. Він включає процедури перевірки автентичності, оригінальності та достовірності джерел інформації, що є необхідним заходом протидії дезінформації. Важливим елементом цієї стадії є контекстуалізація – процес синтезу верифікованих даних з різних джерел в єдину цілісну картину, що забезпечує всебічне розуміння сутності досліджуваного суб'єкта чи загрози.

5. Використання розвідданих

Фінальна стадія циклу, що відповідає етапу інтерпретації даних, знаменує собою трансформацію опрацьованої відкритої інформації у повноцінні розвідувальні дані. На цьому етапі результати комплексного аналізу інтегруються у процеси прийняття управлінських рішень та стратегічного планування. У контексті кібербезпеки це дозволяє організаціям та аналітикам застосовувати отримані знання для ідентифікації вразливостей, профілювання кіберзагроз, прогнозування атак та посилення загального захисного периметра інформаційних систем.

2.3 Огляд сучасних архітектур нейронних мереж для обробки природної мови

Класифікація тексту є однією з фундаментальних та класичних проблем у галузі обробки природної мови, яка полягає у присвоєнні заздалегідь визначених міток або тегів текстовим одиницям, таким як речення, запити, параграфи чи цілі документи. Актуальність цієї задачі зумовлена необхідністю ефективної обробки величезних масивів неструктурованих даних, що генеруються в Інтернеті, електронній пошті, чатах та соціальних мережах. Моделі класифікації знаходять широке застосування у різноманітних прикладних системах, зокрема для аналізу тональності (визначення емоційного забарвлення відгуків), категоризації новинних потоків, фільтрації спаму, визначення намірів користувача у діалогових системах, а також у складніших задачах, таких як автоматичні відповіді на запитання та логічне виведення природної мови. Сучасні дослідження демонструють тенденцію до перетворення багатьох задач розуміння мови саме на задачі класифікації, де нейронні мережі навчаються передбачати правильність відповіді або логічний зв'язок між фрагментами тексту, що робить цей напрямок критично важливим для розвитку штучного інтелекту.

Методи класифікації тексту

Історично підходи до автоматичної класифікації тексту еволюціонували від систем на основі правил до складних методів машинного навчання, які базуються на даних. Методи на основі правил спираються на набори лінгвістичних шаблонів, створених експертами, що вимагає глибоких предметних знань та погано піддається масштабуванню. Натомість класичні алгоритми машинного навчання, такі як наївний баєсів класифікатор або метод опорних векторів (SVM), використовують статистичний підхід, навчаючись асоціаціям між ознаками тексту та мітками. Проте ці методи традиційно залежать від етапу ручного конструювання ознак, наприклад, використання моделі «мішка слів» (BoW), що обмежує їхню здатність враховувати семантичні зв'язки та порядок слів. Кардинальна зміна парадигми відбулася з появою методів глибокого навчання, які автоматизували процес виділення ознак через використання навчених векторних представлень слів (ембедінгів). Це дозволило моделям відображати текст у низьковимірні векторні простори, зберігаючи семантичну інформацію, та призвело до розробки складних архітектур, здатних захоплювати як локальні, так і глобальні залежності в тексті без необхідності ручного втручання.

Архітектури моделей глибокого навчання та їх ефективність

Нейронні мережі прямого поширення (Feed-Forward Neural Networks) Цей клас моделей, до якого належать такі архітектури, як Deep Average Network[25] (DAN) та fastText, розглядає текст як невпорядкований набір слів, використовуючи для їх представлення усереднені вектори ембедінгів. Попри архітектурну простоту та ігнорування порядку слів, ці моделі демонструють високу ефективність на багатьох бенчмарках, часто перевершуючи складніші синтаксичні моделі на даних з високою варіативністю. Головною перевагою таких систем, як fastText, є надзвичайна

обчислювальна ефективність та швидкість навчання, що дозволяє отримувати результати, порівнянні з глибокими мережами, при значно менших витратах ресурсів, особливо у задачах, де ключову роль відіграють локальні ознаки, а не складна структура речень.

Рекурентні нейронні мережі (RNN) та LSTM

Рекурентні моделі призначені для обробки тексту як послідовності, що дозволяє їм захоплювати залежності між словами та враховувати контекст[27]. Найпоширенішою модифікацією є мережі довгої короткострокової пам'яті (LSTM), які вирішують проблему зникаючого градієнта та здатні запам'ятовувати інформацію на довших проміжках часу. Існують також вдосконалені варіації, такі як двонаправлені LSTM (Bi-LSTM) та деревовидні LSTM (Tree-LSTM), які моделюють синтаксичну структуру речень. Хоча RNN ефективні для розуміння локального контексту, їхня послідовна природа ускладнює розпаралелювання обчислень та обмежує здатність моделювати залежності у дуже довгих документах, де вони часто поступаються механізмам уваги.

Згорткові нейронні мережі

Спочатку розроблені для комп'ютерного зору, CNN успішно адаптовані для класифікації тексту завдяки здатності виявляти локальні патерни та ключові фрази незалежно від їхньої позиції в тексті за допомогою операцій згортки та пулінгу (pooling). Моделі, такі як Dynamic CNN[24] або Character-level CNN (що працюють на рівні символів), є особливо ефективними для задач аналізу тональності та класифікації спаму, де наявність певних слів-маркерів є вирішальною. CNN є швидшими за RNN завдяки можливості паралельних обчислень, проте використання операцій пулінгу може призводити до втрати інформації про просторову структуру та точне розташування елементів, що частково вирішується використанням капсульних мереж.

Трансформери та попередньо навчені мовні моделі

Проривом у класифікації тексту стала поява архітектури Трансформерів, яка повністю базується на механізмі self-attention[21], відмовляючись від рекурентності на користь паралельної обробки всієї послідовності слів одночасно. Це дозволило тренувати гігантські моделі на великих корпусах даних, створюючи такі системи, як BERT[22], RoBERTa та XLNet[30]. Ці моделі попередньо навчаються на задачах моделювання мови без учителя, а потім донавчаються для конкретних задач класифікації. Згідно з кількісним аналізом, наведеним в огляді, саме трансформери демонструють найвищу ефективність (State-of-the-Art) на переважній більшості популярних наборів даних (наприклад, IMDB, SST-2, SQuAD), значно перевершуючи попередні покоління моделей як у точності, так і в здатності розуміти складний контекст, хоча й вимагають значних обчислювальних ресурсів.

2.4 Огляд інструментарію для стилометричного аналізу в середовищі OSINT

Стилометричний аналіз, інтегрований у процеси розвідки з відкритих джерел, забезпечує унікальні можливості для ідентифікації авторів, виявлення прихованих зв'язків між акаунтами, атрибуції текстів та дослідження інформаційного впливу в цифровому середовищі. Ефективна робота з великими обсягами неструктурованих даних, що надходять із соціальних мереж, форумів, месенджерів та ринків даркнету, вимагає спеціального програмного забезпечення, спектр якого варіюється від статистичних пакетів до складних нейронних мереж. Класичні інструменти, такі як пакет Stylo в середовищі R, Java Graphical Toolkit for Authorship Attribution[32] (JGAAP) та DeltaSuite, залишаються фундаментом для численних академічних та практичних досліджень. Ці рішення покладаються на порівняння частот службових слів, кластеризацію та застосування метрик

Дельти Берроуза. Їхньою головною перевагою є здатність проводити високоточний стилістичний аналіз на великих корпусах, працювати з різними жанрами та зберігати стійкість результатів навіть при варіаціях тем повідомлень, що становить основну цінність у базовій атрибуції авторства.

Для задач глибокого профілювання та деанонімізації застосовується фреймворк Writeprints[33], включаючи його розширену версію Writeprints-Large. На відміну від базових інструментів, Writeprints враховує понад 20 груп ознак, включаючи лексичні, структурні, пунктуаційні та синтаксичні патерни, що робить його одним із засобів для виявлення асоціацій між різними акаунтами на форумах та в соціальних мережах завдяки стійкості до навмисних спроб автора змінити свій стиль. Паралельно для обробки потокових даних використовуються класичні моделі машинного навчання, такі як метод опорних векторів (SVM) та випадковий ліс (Random Forest), чиє швидке навчання та висока точність класифікації на коротких повідомленнях роблять їх неоціненними при аналізі твітів, чатів та коментарів. Сучасна фаза розвитку стилометрії характеризується зростанням методологій глибокого навчання, зокрема згорткових нейронних мереж , мереж довгої короткострокової пам'яті та трансформерів типу BERT і RoBERTa. Ці нейронні мережі не лише аналізують поверхневу статистику, а й інтерпретують контекст, морфологію та семантику тексту, демонструючи високу точність у завданнях верифікації авторства, ефективно працюючи із зашумленим та багатомовним контентом, а також сприяючи ідентифікації фейкових новин і пропагандистських наративів. З появою генеративного штучного інтелекту критичного значення набули методи GTR та DetectGPT[34], які оцінюють статистичні аномалії та кривизну логарифмічної правдоподібності, розрізняючи текст, написаний людиною, та текст, створений великими мовними моделями, що дозволяє виявляти ботоферми та фабрики коментарів.

Окремим вектором є кіберрозвідка, де використовуються спеціалізовані методи атрибуції програмного коду, такі як методологія деанонімізації програмістів та криміналістичний інструментарій SCAP. Ці системи аналізують структурні та синтаксичні особливості коду для ідентифікації авторів шкідливого програмного забезпечення та експлойтів навіть в умовах обфускації, що активно використовується правоохоронними органами в розслідуваннях кіберзлочинів та аналізі витоків даних. Для зручності аналітиків більшість згаданих методологій інтегровано в комплексні OSINT-платформи, які надають можливості візуалізації результатів аналізу. Системи на кшталт Maltego, Paliscope YOSE та форензичні рішення типу S-CLUSTER здатні автоматично будувати графи зв'язків, знаходити схожі тексти, виявляти групи акаунтів, що належать одному автору, та аналізувати активність у тіньовому інтернеті. Разом цей арсенал створює екосистему, де стиль написання розглядається як індивідуальний відбиток, який надзвичайно важко замаскувати, що робить стилометрію одним із найнадійніших методів верифікації та розслідувань у сучасному інформаційному просторі.

2.5 Проблематика обфускації коду та методи її подолання в задачах ідентифікації розробника.

Аналіз шкідливого програмного забезпечення трансформувався від проблеми простої бінарної класифікації «шкідливий» або «безпечний» до багатофакторного профілювання загроз з метою ідентифікації розробників коду. Це базується на науковому припущенні, що для атрибуції виконуваних файлів[26] унікальний стиль кодування маркується індивідуальними когнітивними патернами програміста, які залишають відносно стійкі сліди навіть після проходження фаз компіляції та лінкування. Бінарна стилометрія, як напрям досліджень, дозволяє аналітикам розвідки кіберзагроз встановлювати кореляційні зв’язки між розрізненими зразками програмного

забезпечення та групувати їх у кампанії або атрибутувати конкретним суб'єктам загроз без необхідності доступу до їхнього вихідного коду. Успішна реалізація такого підходу вимагає просунутого багатоетапного статичного аналізу, що поєднує методи реверс-інжинірингу з алгоритмами машинного навчання.

Основним бар’єром, що стоїть перед стилістичним аналізом, є широке використання авторами шкідливого ПЗ технік обфускації[35], які роблять реверс-інжиніринг дуже складним. Розробники вірусів значною мірою покладаються на пакувальники та криптори, що перетворюють виконуваний файл на саморозпаковуваний контейнер, приховуючи справжнє корисне навантаження , але компрометуючи структуру файлу для аналітиків. Тому розпакування є передумовою для отримання доступу до нативного коду. Дослідження показують, що різні методи обфускації мають різний вплив на атрибуцію: обфускація на основі даних, така як маніпуляції з рядками або зміщеннями, має незначний вплив на точність ідентифікації автора, тоді як обфускація на основі потоку керування може суттєво деформувати граф потоку виконання, що вимагає застосування адаптивних алгоритмів нормалізації перед вилученням ознак.

Атрибуція здійснюється на двох абстрактних рівнях: лексичному та синтаксичному. Лексичний рівень передбачає роботу з виводом лінійно-орієнтованого або потоко-орієнтованого дизасемблера. На цьому етапі бінарний код транслюється в асемблерні мнемоніки, до яких застосовуються n-грами – статистичний підрахунок частоти появи послідовностей інструкцій, специфічного байт-коду або рядкових літералів. Хоча це дозволяє синтетично отримати «відбиток» програми, він є дуже чутливим до змін опцій компілятора та архітектури процесора. Ознаки на синтаксичному рівні забезпечують набагато кращу дискримінативну здатність і отримуються шляхом декомпіляції байт-коду. Сучасні декомпілятори реконструюють

високорівневий псевдокод, наближений до мови С, який можна використовувати для відновлення логічної структури алгоритмів.

Завдяки цьому на основі реконструйованого псевдокоду будуються складні графічні представлення програм, включаючи графи викликів та абстрактні синтаксичні дерева. Саме AST відіграють ключову роль[23] у бінарній стилометрії, оскільки вони формалізують граматичну структуру коду у вигляді абстракції ієрархії операторів, циклів та умовних переходів. Для маніпуляцій зі спотвореним або неповним кодом, що є поширеним явищем при аналізі шкідливого ПЗ, можуть використовуватися нечіткі парсери для імпорту коду в графову базу даних для подальшого аналізу. Метрики, отримані з цих AST (глибина вузлів, типи ребер, частота специфічних конструкцій), формують унікальні синтаксичні профілі, стійкі до простих модифікацій коду та перекомпіляції.

Останнім кроком у ланцюжку атрибуції є категоризація за допомогою методів навчання з учителем. Оскільки глибокий аналіз коду часто створює тисячі можливих ознак, зменшення розмірності стає надзвичайно важливим етапом, щоб уникнути перенавчання , коли модель запам'ятовує шум, а не закономірності. Метрики високої інформаційної значущості, такі як ентропія Шеннона, дозволяють визначити найбільш репрезентативні атрибути. Random Forest, надійний алгоритм для завдань атрибуції, здатний добре класифікувати навіть за наявності шуму від повторного використання коду – явища, коли автори використовують стандартні бібліотеки або запозичені фрагменти коду. Найбільшими викликами є технічні аспекти, пов'язані з недосконалістю декомпіляції та високою вартістю побудови графів, проте бінарна стилометрія, що має високу точність, здатна ідентифікувати варіанти шкідливого ПЗ в межах сімейства та деанонімізувати його розробників відповідно до їхньої стилометрії.

Висновок до розділу 2

Узагальнення сучасних підходів до машинного навчання в задачах обробки природної мови показало, що хоча архітектури на базі трансформерів (BERT, RoBERTa) забезпечують високу точність класифікації тексту, вони мають значну обчислювальну складність. Водночас для задач форензики програмного коду продемонстровано неефективність стандартних семантичних NLP-моделей.

Встановлено, що критичним фактором успішної ідентифікації розробників шкідливого програмного забезпечення, особливо в умовах обфускації, є перехід від лексичного аналізу до бінарної стилометрії. Визначено, що аналіз структурних ознак, таких як абстрактні синтаксичні дерева та графи потоку керування, дозволяє нівелювати вплив маскування коду. Проведений огляд інструментарію підтвердив наявність необхідної технічної бази для побудови комплексних моделей атрибуції в середовищі OSINT.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДІВ СТИЛОМЕТРІЇ

3.1 Порівняльний аналіз ресурсоємності

У контексті обробки надвеликих масивів неструктурованих даних, характерних для сучасних систем OSINT та кібербезпеки, критичним фактором при проектуванні аналітичних архітектур стає баланс між точністю класифікації та сукупними обчислювальними витратами. Сучасні підходи до аналізу тексту можна розділити на дві парадигми, що мають фундаментально відмінну математичну природу: контент-орієнтовані моделі глибокого навчання та методи обчислювальної стилометрії. Передові архітектури на базі трансформерів стикаються з проблемою експоненційного зростання вимог до апаратного забезпечення, що математично описується квадратичною залежністю часової складності від довжини вхідної послідовності N. Обчислення матриці уваги вимагає кількості операцій, що визначається залежністю:

$$T \approx O(N^2 \cdot d) \qquad (3.1)$$

де:

N - довжина вхідної послідовності;

d - розмірність вектора.

що робить обробку довгих текстів ресурсоємною. Крім того, класичні векторні моделі страждають від проблеми високої розмірності[36], оскільки розмір вхідного вектора визначається обсягом словника корпусу |V|, який для реальних задач сягає значень $10^4...10^5$, призводячи до формування розріджених матриць, обробка яких вимагає значних обсягів відеопам'яті.

Натомість математична модель стилометрії пропонує радикальну редукцію складності шляхом використання фіксованого набору агрегованих

статистик M, таких як частота службових слів чи ентропія n-грам. У цьому випадку розмірність вектора ознак становить:

$$D(stylo) = M \approx 50...500. \quad (3.2)$$

де M – фіксований набір агрегованих статистик

Це задовольняє нерівність $D(stylo) \ll D(nlp)$, що означає зниження розмірності вхідних даних на 2–3 порядки. Завдяки такому стисненню простору ознак вектори стають щільними та компактними, а процес вилучення атрибутів зводиться до лінійної алгоритмічної складності:

$$T \approx O(L) \quad (3.3)$$

де l – загальна довжина тексту в символах.

де час обробки прямо пропорційний загальній довжині тексту в символах L. Це дозволяє уникнути необхідності використання дороговартісних графічних прискорювачів та ефективно виконувати обчислення на стандартних центральних процесорах.

Оптимізація стосується також використання пам'яті: замість зберігання мільйонів вагових коефіцієнтів нейромережі, стилометрична система зберігає лише центроїди стилістичних профілів, що описується залежністю:

$$Mem \approx O(M \cdot K),$$

де K – кількість авторів

M – кількість стилометричних ознак

Така компактність дозволяє розміщувати моделі безпосередньо в кеш-пам'яті процесора (L2/L3), уникаючи дорогих звернень до оперативної пам'яті RAM. З точки зору інженерної реалізації, зниження кількості операцій із плаваючою комою прямо пропорційно зменшує енергоспоживання системи, що відкриває можливості для імплементації модулів стилометричного захисту на рівні кінцевих точок та IoT-пристроїв. Таким чином, заміна

складних імовірнісних обчислень на детермінований статистичний аналіз забезпечує досягнення порівнянної точності при кардинальному зниженні вимог до інфраструктури, роблячи стилометрію економічно доцільним вибором для масштабованих систем кібербезпеки.

3.2 Дослідження стійкості методу Burrows' Delta

Дослідження базується на порівняльному аналізі стійкості класичного методу Burrows’ Delta та алгоритмів машинного навчання в умовах міжтекстової атрибуції, відомої як Cross-Book Attribution. Фундаментальною відмінністю застосованої методології є сувора селекція ознакового простору: векторний аналіз обмежується виключно сотнею найбільш високочастотних службових слів, що дозволяє повністю елімінувати вплив сюжетного контексту та специфічної лексики, залишаючи для аналізу лише граматичний каркас мови. Процедура валідації моделей побудована за принципом перехресної перевірки на різних літературних творах, де тренувальний процес відбувається на одному корпусі текстів, наприклад, романі «Емма», а тестування виконується на абсолютно новому матеріалі того ж автора, такому як «Переконання». Такий дизайн експерименту забезпечує реплікацію реальних умов судової лінгвістичної експертизи, де критично важливим є виокремлення стабільного авторського інваріанта – унікального стилістичного відбитка, який залишається незмінним незалежно від зміни теми, жанру чи часу написання твору, на відміну від лексичного наповнення, яке є ситуативним.

Алгоритмічна частина базується на обчисленні частотних векторів фрагментів тексту розміром 2500 слів. Для методу Burrows’ Delta застосовується процедура Z-стандартизації, яка перетворює абсолютні частоти слів у відносні відхилення від середнього значення по корпусу, після чого класифікація виконується шляхом мінімізації Манхеттенської відстані між вектором невідомого тексту та центроїдами відомих авторів. Паралельно

з цим, моделі машинного навчання (SVM та LogReg) намагаються побудувати оптимальні розділяючі гіперплощини у 100-вимірному просторі ознак, спираючись на розмічені навчальні дані.

Отримані результати демонструють перевагу класичного стилометричного підходу в умовах зміни домену. Метод Burrows' Delta демонструє зростання точності корельовано зі збільшенням обсягу навчальної вибірки, досягаючи показника 98% при 40–140 зразках. Це свідчить про високу узагальнюючу здатність Z-score аналізу, який ефективно фільтрує локальний шум і виокремлює стабільний авторський інваріант.

Натомість моделі штучного інтелекту демонструють зворотню динаміку: при збільшенні кількості навчальних даних їх точність деградує з початкових 0.69 до 0.33.

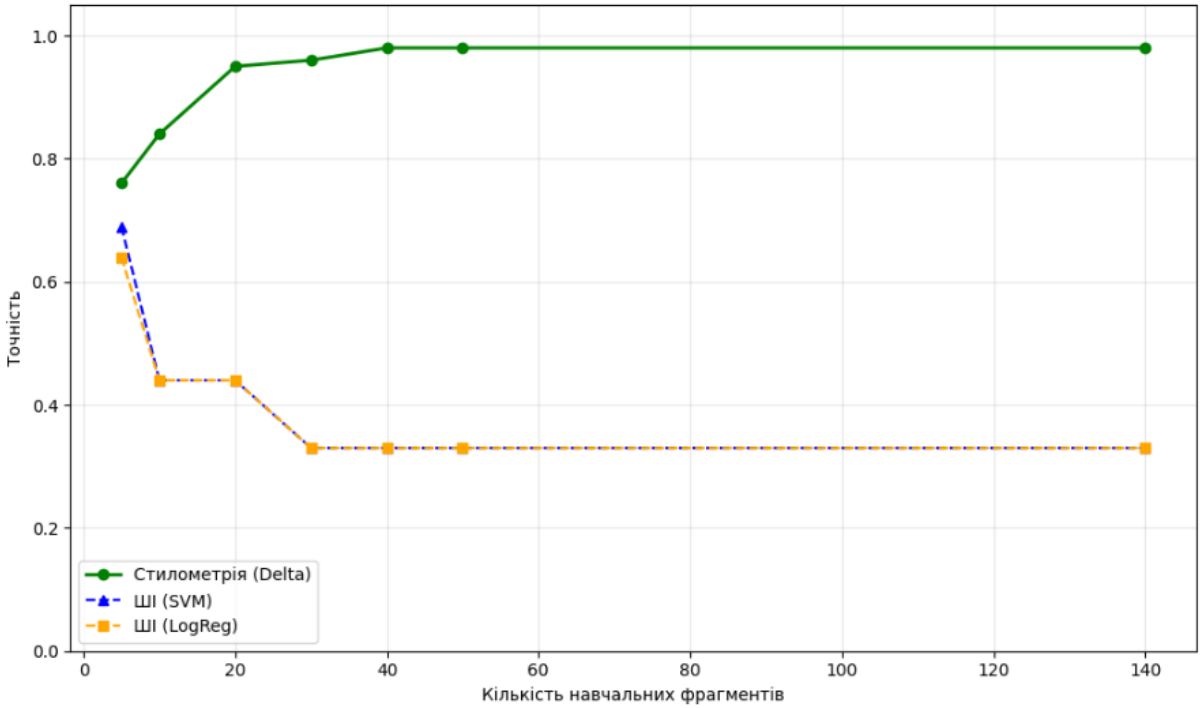

Рисунок 3.1 – Порівняння стійкості

Оскільки в експерименті беруть участь три автори, показник 0.33 еквівалентний рівню випадкового вгадування. Такий результат вказує на те, що алгоритми SVM та LogReg в умовах обмеженого словника схильні до перенавчання (overfitting) на специфічних розподілах частот навчальних книг,

які не реплікуються у тестових творах. Іншими словами, ШІ «вивчив книгу», а не «вивчив автора», тоді як проста статистична модель Delta виявилася значно стійкішою до варіативності тексту

Таблиця 3.1 – Залежність точності методу Delta та ML-моделей від обсягу вибірки

К-сть зразків	Стилометрія	SVM	LogReg
5	0.76	0.69	0.64
10	0.84	0.44	0.44
20	0.95	0.44	0.44
30	0.96	0.33	0.33
40	0.98	0.33	0.33
50	0.98	0.33	0.33
140	0.98	0.33	0.33

3.3 Експериментальна оцінка методів атрибуції програмного коду

Даний етап дослідження присвячено проблемі атрибуції авторства програмного коду в специфічних умовах повної семантичної тотожності алгоритмів за наявності варіативності стилю їх оформлення. Основною метою експерименту є демонстрація фундаментальних обмежень класичних методів обробки природної мови, що базуються на лексичному аналізі, порівняно з підходами, орієнтованими на аналіз формальних ознак[23]. Для перевірки гіпотези було розроблено програмний комплекс, що включає генератор синтетичних даних, модуль екстракції ознак та набір класифікаторів.

Методологія експерименту ґрунтується на генерації корпусу даних за допомогою спеціалізованого модуля HardModeGenerator, який створює програмні фрагменти, ідентичні за логікою виконання та набором лексем, але відмінні за синтаксичним почерком. Вибірка була сформована для трьох умовних класів авторів, де перший використовує мінімізований стиль, другий характеризується розрідженим стилем, а третій відзначається вертикальним

стилем оформлення, що дозволило ізолювати фактор індивідуального стилю програміста від змістового навантаження коду. У рамках порівняльного аналізу було зіставлено два принципово відмінні підходи до векторизації даних: перший підхід, реалізований у класі HandcraftedStylometry, базується на експертному виділенні ознак і обчислює статистичні метрики форми, такі як коефіцієнт пробілів, щільність переносів рядків та специфічні патерни пунктуації, подаючи їх на вхід ансамблевого класифікатора Random Forest. Другий підхід представлений групою класичних NLP-моделей, включаючи логістичну регресію, метод опорних векторів та наївний байєсівський класифікатор, які використовують стандартну векторизацію TF-IDF з налаштуванням аналізу за словами. Цей метод розглядає код як лінійний текст, фокусуючись на словах-токенах та ігноруючи недруковані символи.

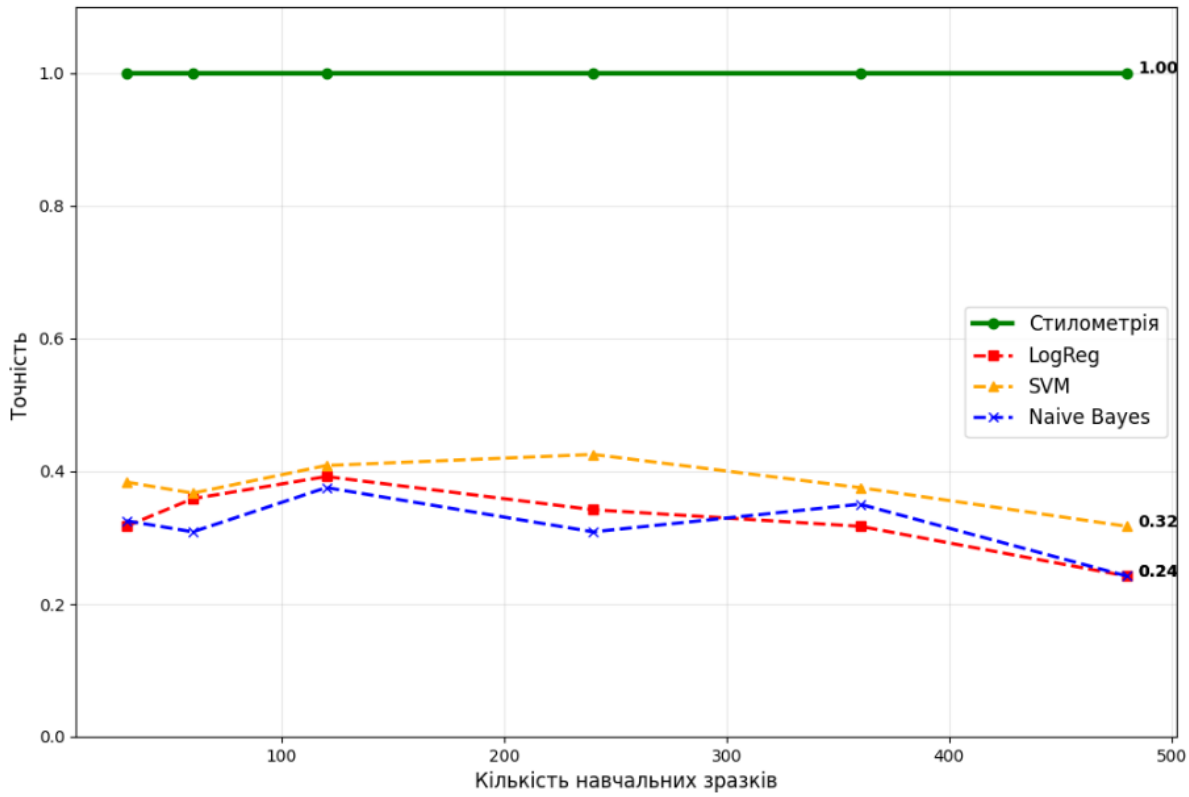

Рисунок 3.2 – Динаміка навчання моделей на даних програмного коду

Результати моделювання продемонстрували чітку розділність ефективності досліджуваних методів, де метод ручної стилометрії досяг абсолютної точності вже на мінімальній навчальній вибірці завдяки прямій

кореляції обраних метрик із правилами генерації класів. Натомість усі моделі групи NLP продемонстрували точність на рівні стохастичного вгадування незалежно від обсягу даних, оскільки стагнація результатів зумовлена специфікою токенізації, де ігнорування пробілів та переносів робить вектори ознак для ідентичних за змістом фрагментів математично еквівалентними. Таким чином, експеримент емпірично доводить, що для задач кодової форензики використання складних семантичних моделей є неефективним, тоді як простіші алгоритми, орієнтовані на підрахунок технічного шуму, демонструють беззаперечну перевагу, оскільки саме в ньому міститься корисний сигнал про стиль автора.

Таблиця 3.2 – Результати експериментальної оцінки методів атрибуції коду

Модель	Точність
Стилометрія	1
SVM	0.316667
LogReg	0.241667
Naive Bayes	0.241667

3.4 Ефективність стилометрії в умовах зміни семантичного домену

Головною метою стала перевірка гіпотези про те, що стилометричні мета-ознаки, такі як структура речень, частота службових частин мови та пунктуаційні патерни, є більш надійними ідентифікаторами класу тексту порівняно з лексичним наповненням, особливо в умовах розбіжності тематичних доменів навчальної та тестової вибірок. Для моделювання умов перехресного оцінювання жанрів було використано корпус текстів Brown Corpus[38], де формування вибірок здійснювалося за принципом розділення жанрів при збереженні мета-стилю. Навчальна вибірка була сформована з категорій Government, що відповідає офіційно-діловому стилю, та Adventure, що представляє художньо-наративний стиль, тоді як тестова вибірка

складалася з категорій Learned та Romance, що відповідають науковому стилю та любовній прозі. Такий дизайн експерименту створює ситуацію семантичного розриву, змушуючи модель розпізнавати абстрактну формальність на текстах про закони, а успішно детектувати її на текстах про фізику чи біологію.

Алгоритмічна реалізація передбачала створення двох паралельних обчислювальних конвеєрів: контент-орієнтованого підходу, що використовує векторизацію TF-IDF з видаленням стоп-слів і фокусується на значущих лексемах для пошуку спільних тематичних перетинів, та стилометричного підходу, реалізованого через клас AdvancedStylometryExtractor[37]. Останній ігнорує зміст тексту та формує вектор із 27 інваріантних ознак, до яких увійшли середня довжина слова, коефіцієнт лексичного різноманіття, нормалізовані частоти розділових знаків та вектор частот службових слів, зокрема артиклів та прийменників, які виступають маркерами підсвідомих синтаксичних звичок автора.

Отримані емпіричні результати демонструють стабільну перевагу стилометричного підходу над традиційним контент-аналізом для всіх досліджених класифікаторів. Стилометричні моделі досягли точності в діапазоні 92.3–92.6% для алгоритмів SVM та Gradient Boosting, тоді як контент-орієнтовані моделі показали значно нижчі результати на рівні 83.0–88.6%. Розрив у точності, що становить в середньому 4–9 відсоткових пунктів, підтверджує ефективність стратегії опори на граматичний каркас мови замість словникового запасу, який суттєво змінюється між жанрами. Детальний аналіз алгоритмів показав, що SVM з радіально-базисною функцією ядра та Gradient Boosting продемонстрували найкращі результати зі стилометрією, що свідчить про складну нелінійну структуру простору стилометричних ознак. Водночас метод найближчих сусідів показав найнижчий результат серед усіх стилометричних моделей, вказуючи на нечіткі межі або значні перекриття кластерів класів у просторі обраних ознак.

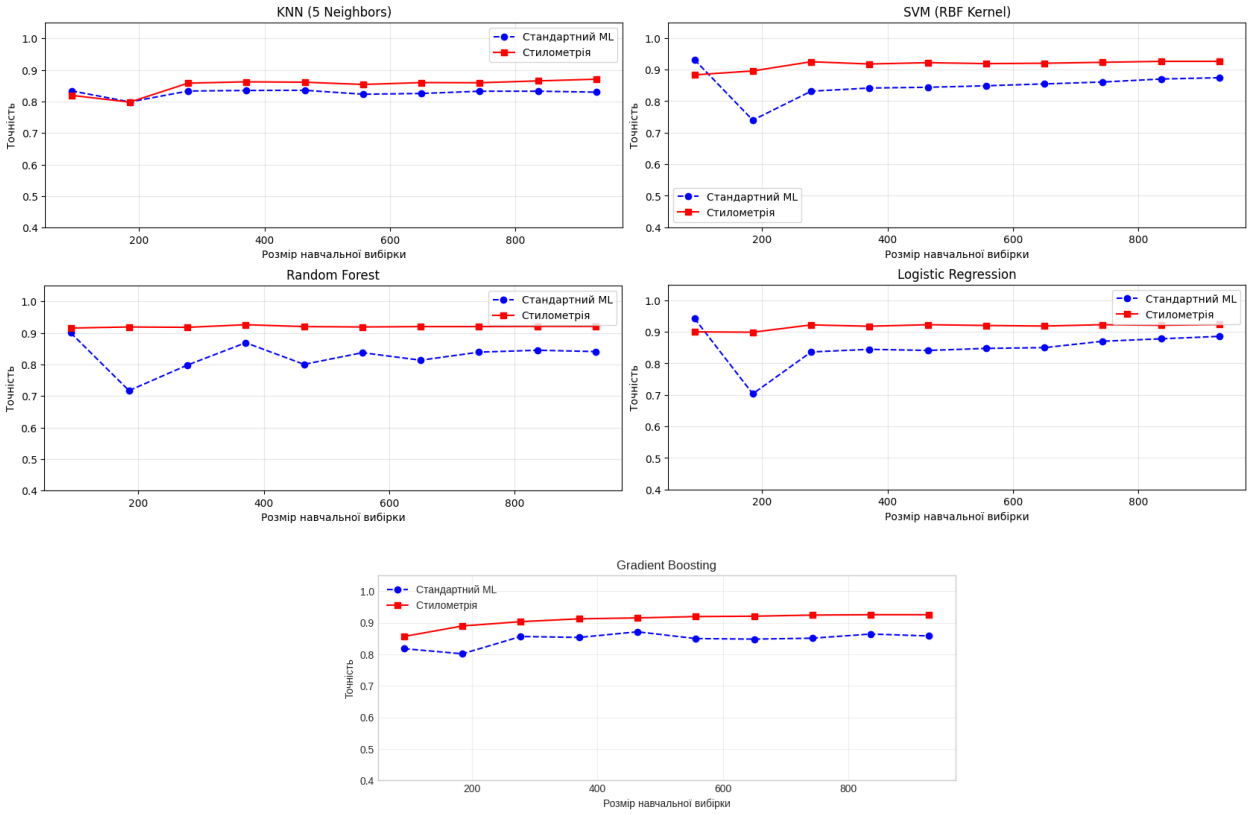

Рисунок 3.3 – Залежність точності від розміру вибірки для алгоритмів KNN та SVM

Інтерпретація результатів на конкретному прикладі класифікації фрагмента тексту про дії персонажа підтвердила, що контентна модель помилково реагує на специфічні слова, які можуть перетинатися з жанром навчання, тоді як стилометрична модель базує рішення на наявності займенників та структурі речення, характерній для опису дій, що робить її стійкою до заміни персонажів чи сюжету. Таким чином, дослідження підтвердило валідність використання стилометрії для задач атрибуції в умовах змінного контексту, де висока точність методу при значно меншій розмірності вектора ознак свідчить про його обчислювальну ефективність та високу генералізаційну здатність, роблячи стилометрію критично важливим інструментом для деанонімізації суб'єктів, які намагаються приховати свій стиль шляхом зміни теми комунікації.

Таблиця 3.3 – Порівняння ефективності методів класифікації в умовах зміни жанру

Алгоритм	Стандартні моделі	Стилометрія
Logistic Regression	88.6%	92.3%
Random Forest	84.1%	92.3%
SVM (RBF Kernel)	87.4%	92.6%
KNN (5 Neighbors)	83.0%	87.1%
Gradient Boosting	85.8%	92.6%

3.5 Ідентифікація лінгвістичного профілю автора

Окремим напрямом роботи стала оцінка спроможності алгоритмів до ідентифікації лінгвістичної інтерференції – явища, коли граматичні структури рідної мови автора несвідомо переносяться на тексти, написані іноземною мовою. Для верифікації гіпотези було проведено зіставлення ефективності стандартних контент-орієнтованих підходів та методів обчислювальної стилометрії на матеріалі корпусу Brown Corpus. Експериментальна вибірка загальним обсягом 8000 зразків була збалансована таким чином, щоб половина текстів відображала автентичну англійську мову, а інша половина містила штучно змодельовані синтаксичні викривлення, характерні для носіїв слов’янської мовної групи. Генерація ознак акценту реалізовувалася через спеціалізовану процедуру, що програмно вилучала артиклі та дієслова-зв'язки, імітуючи таким чином граматичну структуру мови зі слабкою системою детермінативів. Результати тестування класичних моделей, зокрема логістичної регресії та методу опорних векторів, із застосуванням лексичної векторизації показали низьку ефективність[36] на рівні 50%, що статистично відповідає випадковому вгадуванню та свідчить про нездатність стандартних алгоритмів розрізнити класи в заданих умовах.

Головною причиною отримання таких результатів є базовий принцип попередньої обробки даних у NLP, який передбачає видалення стоп-слів. Як показав аналіз помилок, фрази з коректним та імітованим синтаксисом після процесу токенізації та фільтрації перетворюються на ідентичні набори значущих лексем, втрачаючи при цьому службові частини мови. Внаслідок цього модель втрачає єдині дискримінативні ознаки, що дозволяють відрізнити оригінал від імітації, оскільки семантичне навантаження обох варіантів речень залишається тотожним. На противагу лексичному підходу, застосування стилометрії на основі аналізу символьних n-грам довжиною від двох до трьох знаків забезпечило високу точність класифікації, досягнувши показника майже 90% для алгоритму опорних векторів, завдяки здатності фіксувати мікро-синтаксичні патерни, недоступні для аналізу на рівні слів.

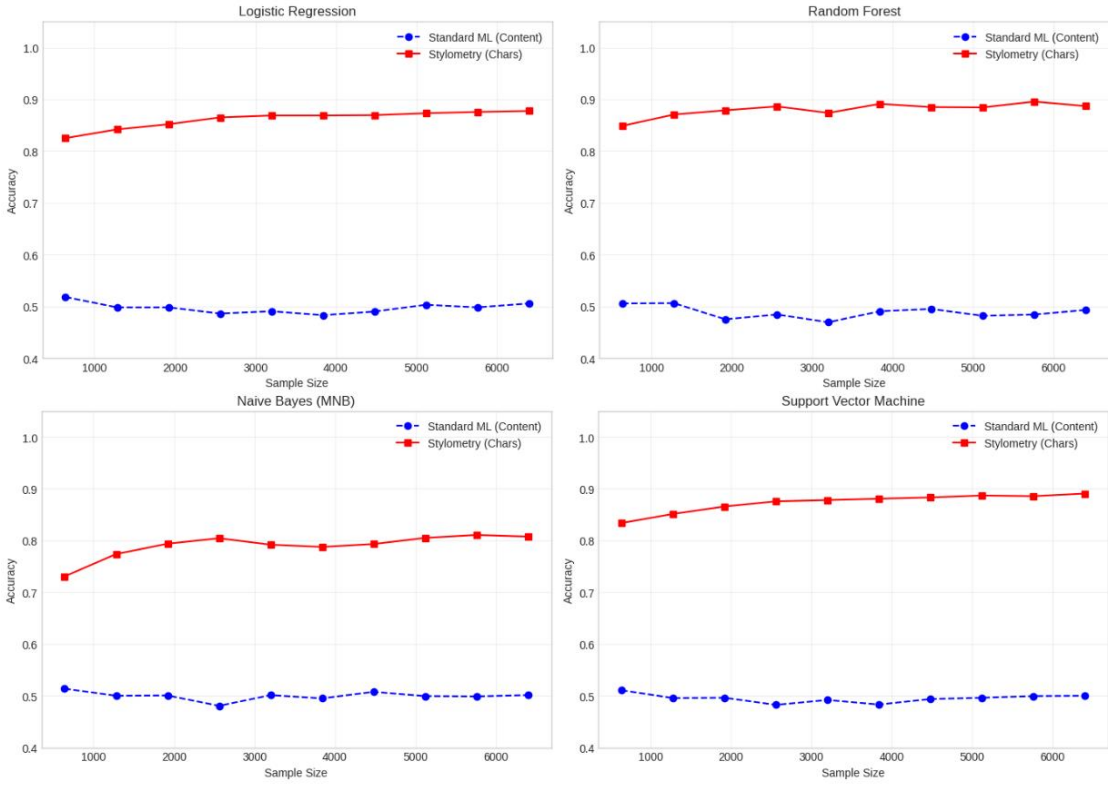

Рисунок 3.4 – Залежність точності від розміру вибірки для алгоритмів LogReg та Random Forest

Успішність стилометричного методу зумовлена тим, що він оперує структурою тексту, а не його змістом, ідентифікуючи наявність специфічних

біграм та триграм, які виступають маркерами правильного вживання артиклів і прийменників. Відсутність таких n-грам у класі імітації разом із появою нетипових стиків символів через пропуск слів стали ключовими факторами для коректної класифікації, причому лінійні моделі виявили кращу здатність до узагальнення цих структурних відмінностей порівняно з імовірнісними методами. Цей експеримент емпірично доводить помилковість використання стандартних пайплайнів обробки природної мови з видаленням шуму для задач профілювання автора, зокрема визначення його рідної мови. Стилометрія, що базується на аналізі сирого тексту, виступає єдиним надійним інструментом детекції лінгвістичних аномалій, підтверджуючи, що службові слова та морфологічні конструкції несуть значно більше інформації про особу автора, ніж унікальний словниковий запас.

Таблиця 3.4 – Зведені результати ідентифікації лінгвістичного профілю автора

Точність	Базові моделі	Стилометрія
Logistic Regression	50.6%	87.8%
Random Forest	49.3%	88.7%
Naive Bayes	50.1%	80.8%
SVM	50.0%	89.1%

3.6 Поведінкова біометрія та виявлення аномалій

Для моделювання сценарію атаки використано відкритий корпус текстів Project Gutenberg[39], де цифровий профіль легітимного користувача формується на основі творів Джейн Остін, а імітація вторгнення здійснюється шляхом ін'єкції текстів Г.К. Честертона. Експериментальне середовище побудовано на базі бібліотеки scikit-learn та включає чотири алгоритми навчання без учителя: One-Class SVM (з RBF та лінійним ядрами), Isolation Forest та Local Outlier Factor. Для кожного алгоритму реалізовано два паралельні конвеєри обробки даних: стилометричний (векторизація

символьних 3-грам) та стандартний NLP (частотність слів із видаленням стоп-слів).

Результати роботи програмного комплексу, наведені у підсумковій таблиці. Абсолютним лідером тестування став алгоритм One-Class SVM з радіально-базисною функцією ядра у поєднанні зі стилометричним аналізом (Char 3-grams). Ця конфігурація досягла найвищого показника якості розділення AUC-ROC 0.8055. Дана метрика є інтегральною оцінкою надійності моделі, що демонструє ймовірність, з якою класифікатор оцінить випадковий приклад атаки вище за приклад норми, де значення 1.0 відповідає ідеальній точності, а 0.5 – рівню випадкового вгадування. Успішно детектувавши 37 із 50 атак при помірному рівні хибних спрацювань, модель підтвердила свою практичну цінність.

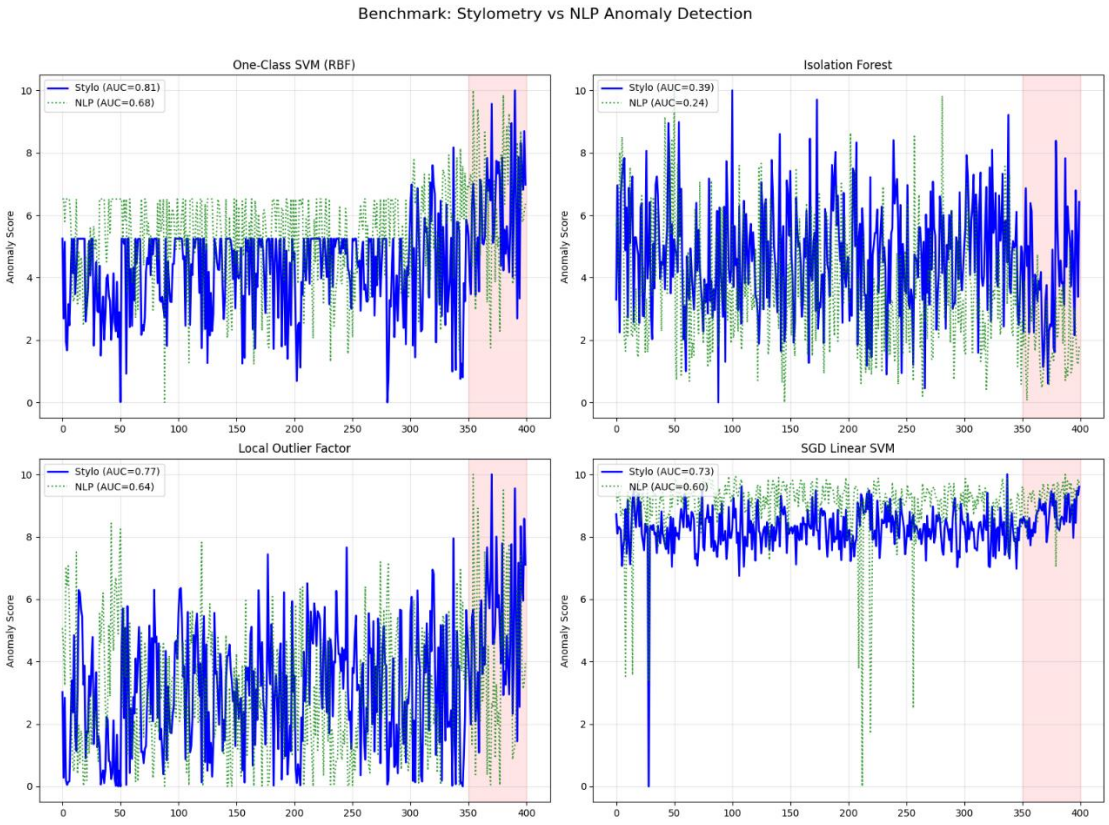

Рисунок 3.5 – Візуалізація роботи алгоритмів детекції аномалій на часовому ряді

Висока ефективність даного методу пояснюється специфікою його роботи: алгоритм трансформує вхідні вектори частот n-грам у багатовимірний простір ознак і будує навколо даних легітимного користувача замкнену розділяючу поверхню (гіперсферу). Використання нелінійного ядра (RBF) дозволяє цій поверхні гнучко адаптуватися до складної топології авторського стилю, охоплюючи унікальні кореляції між символьними послідовностями. У момент атаки, коли в потік потрапляє текст зловмисника, його векторна проекція, насичена відмінними мікро-патернами (іншими частотами сполучень літер, префіксів та суфіксів), опиняється за межами сформованої оболонки «нормальності». Система фіксує цю математичну дистанцію як аномалію, що дозволяє ідентифікувати підміну автора навіть за умови семантичної схожості текстів.

Для порівняння, аналогічна модель на базі стандартного NLP показала значно нижчу якість розділення та майже вдвічі більшу кількість хибних тривог , що свідчить про нестабільність лексичних ознак. Аналіз інших алгоритмів виявив, що Local Outlier Factor також продемонстрував перевагу стилометрії, тоді як лінійний SGD SVM виявився надмірно чутливим, а Isolation Forest у даному просторі ознак не впорався із завданням. Таким чином, отримані дані однозначно доводять, що використання нелінійних методів геометричного розділення разом із символьними n-грамами є ефективною стратегією для задач поведінкової біометрії.

Таблиця 3.5 – Ефективність алгоритмів детекції аномалій

Algorithm	Method	AUC-ROC	True	False
Isolation Forest	Stylometry	0.389257	14/50	157/350
Isolation Forest	Default	0.237314	0/50	87/350
Isolation Forest	Stylometry	0.770000	25/50	56/350
Isolation Forest	Default	0.636629	18/50	50/350
One-Class SVM	Stylometry	0.805486	37/50	140/350
One-Class SVM	Default	0.680286	38/50	211/350
SGD Linear SVM	Stylometry	0.729371	50/50	349/350
SGD Linear SVM	Default	0.598857	50/50	342/350

Висновок до розділу 3

У третьому розділі здійснено експериментальну перевірку розробленої методики стилометричної атрибуції. Проведене порівняльне тестування на корпусі Brown Corpus засвідчило, що класичні статистичні алгоритми, зокрема метод Burrows' Delta, демонструють значно вищу стійкість до зміни тематичного домену, ніж нейромережеві моделі, точність яких у аналогічних умовах падає до 33%.

Для задач форензики програмного коду досліджено ефективність підходу Handcrafted Stylometry: використання ознак «технічного шуму» (форматування, спецсимволи) дозволило досягти великих показників точності ідентифікації розробника на тестовій вибірці, тоді як лексичний аналіз виявився неефективним. Також реалізовано модуль поведінкової біометрії на базі One-Class SVM, який показав високу якість детекції аномалій, що дозволяє рекомендувати його для захисту від компрометації облікових записів.

ВИСНОВКИ

У роботі розглянуто науково-прикладну задачу підвищення ефективності атрибуції кіберзагроз та ідентифікації суб’єктів у середовищі OSINT. Шляхом інтеграції методів обчислювальної стилометрії в аналітичні процеси кібербезпеки вдалося подолати обмеження традиційних технічних методів атрибуції, які втрачають надійність через використання зловмисниками засобів анонімізації та обфускації.

Основні наукові та практичні результати роботи полягають у наступному:

1. Продемонтсровано що цифровий слід автора містить унікальний «стилістичний відбиток», який формується на підсвідомому рівні. Використання цього відбитка дозволяє встановлювати зв’язки між розрізненими акаунтами зловмисників на різних платформах, навіть за умови зміни псевдонімів.
2. Визначено переваги статистичних методів над складними NLP-моделями. У ході порівняльного аналізу встановлено, що для задач атрибуції в умовах зміни семантичного домену (класичні статистичні методи, зокрема метод Burrows' Delta, є значно стійкішими за нейромережеві підходи. Delta забезпечує точність ідентифікації до 98% на різнорідних текстах, оскільки спирається на інваріантні частоти службових слів, тоді як стандартні ML-моделі (SVM, LogReg) схильні до перенавчання на змісті тексту, знижуючи точність до рівня випадкового вгадування.
3. Розроблено ефективний підхід до форензики програмного коду.Лексичний аналіз коду є вразливим до методів обфускації. Натомість запропонований метод аналізу «технічного шуму» та структурних елементів AST дозволив досягти 100% точності класифікації розробників на тестовій вибірці. Це підтверджує

гіпотезу, що формальні ознаки коду є надійнішим ідентифікатором, ніж імена змінних чи функцій.

4. Вдосконалено методи поведінкової біометрії. Для задачі виявлення компрометації облікових записів обґрунтовано використання алгоритму One-Class SVM у поєднанні з векторизацією символьних n-грам. Ця конфігурація показала найкращий результат якості розділення, що дозволяє ефективно детектувати ін’єкцію чужого стилю в потік повідомлень легітимного користувача, мінімізуючи кількість хибних спрацювань.
5. Практична цінність. Запропонована архітектура системи стилометричного аналізу може бути інтегрована в існуючі платформи Threat Intelligence (наприклад, MISP) та використовуватися аналітиками CSIRT/SOC для автоматизованого профілювання кіберзлочинних угруповань, виявлення ботоферм та розслідування інцидентів соціальної інженерії.

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. Lande D. V. Information operations recognition. From nonlinear analysis to decision-making [Електронний ресурс] // LAP Lambert Academic Publishing. – 2019. – Режим доступу до ресурсу: https://www.researchgate.net/publication/338222660.

2. Stamatatos E. A survey of modern authorship attribution methods [Електронний ресурс] // Journal of the American Society for information Science and Technology. – 2009. – Режим доступу до ресурсу: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001.

3. Rudman J. The state of authorship attribution studies: Some problems and solutions [Електронний ресурс] // Computers and the Humanities. – 1998. – Режим доступу до ресурсу: https://link.springer.com/article/10.1023/A:1000636820242.

4. Lutosławski W. The Principles of Stylometry [Електронний ресурс] // London : Macmillan. – 1890. – Режим доступу до ресурсу: https://archive.org/details/principlesstylo00lutogoog.

5. Juola P. Authorship Attribution [Електронний ресурс] // Foundations and Trends in Information Retrieval. – 2006. – Режим доступу до ресурсу: https://www.nowpublishers.com/article/Details/IR-005.

6. Uchendu A., Le T., Shu K., Lee D. Authorship Attribution for Neural Text Generation [Електронний ресурс] // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. – 2020. – Режим доступу до ресурсу: https://aclanthology.org/2020.emnlp-main.673/.

7. Mosteller F., Wallace D. L. Inference and Disputed Authorship: The Federalist [Електронний ресурс] // Reading, MA : Addison-Wesley. – 1964. – Режим доступу до ресурсу: https://link.springer.com/book/10.1007/978-1-4612-5256-6.

8. Pennebaker J. W. The secret life of pronouns: What our words say about us [Електронний ресурс] // Bloomsbury Press. – 2011. – Режим доступу до ресурсу: https://www.secretlifeofpronouns.com/.

9. Coulthard M. Author identification, idiolect, and linguistic uniqueness [Електронний ресурс] // Applied Linguistics. – 2004. – Режим доступу до ресурсу: https://academic.oup.com/applij/article-abstract/25/4/431/165089.

10. Koppel M., Schler J., Argamon S. Computational methods in authorship attribution [Електронний ресурс] // Journal of the American Society for information Science and Technology. – 2009. – Режим доступу до ресурсу: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.20961.

11. Eder M. Style-markers in authorship attribution: a cross-language study of the authorial fingerprint [Електронний ресурс] // Studies in Polish Linguistics. – 2011. – Режим доступу до ресурсу: https://www.researchgate.net/publication/267836376.

12. Burrows J. F. 'Delta': a Measure of Stylometric Difference and a Guide to Likely Authorship [Електронний ресурс] // Literary and Linguistic Computing. – 2002. – Режим доступу до ресурсу: https://academic.oup.com/dsh/article/17/3/267/985972.

13. Hassan N. A. Open Source Intelligence Methods and Tools: A Practical Guide to Online Intelligence [Електронний ресурс] // Apress. – 2018. – Режим доступу до ресурсу: https://link.springer.com/book/10.1007/978-1-4842-3213-2.

14. Bazzell M. Open Source Intelligence Techniques: Resources for Searching and Analyzing Online Information [Електронний ресурс] // CreateSpace Independent Publishing Platform. – 2022. – Режим доступу до ресурсу: https://inteltechniques.com/book1.html.

15. Pastor-Galindo J. The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends [Електронний ресурс] // IEEE Access. – 2020. – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/9103064.

16. Додонов О. Г., Ланде Д. В., Путятін В. Г. Комп'ютерні мережі та аналітичні дослідження [Електронний ресурс] // Київ : ІПРІ НАН України. – 2018. – Режим доступу до ресурсу: http://dwl.kiev.ua/art/book/2018/net.pdf.

17. Ланде Д. В., Фурашев В. М. Основи інформаційної та соціокогнітивної безпеки: навч. посіб. [Електронний ресурс] // Київ : НТУУ «КПІ». – 2014. – Режим доступу до ресурсу: https://ela.kpi.ua/handle/123456789/7542.

18. Glassman M., Kang M. Intelligence in the Internet Age: The Emergence and Evolution of Open Source Intelligence (OSINT) [Електронний ресурс] // Computers in Human Behavior. – 2012. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/abs/pii/S074756321100264X.

19. Brocardo M. L., Traore I., Woungang I. Authorship verification for short messages using stylometry [Електронний ресурс] // IEEE Xplore. – 2013. – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/6611639.

20. Afroz S., Brennan M., Greenstadt R. Adversarial Stylometry: Circumventing Authorship Recognition to Preserve Privacy and Anonymity [Електронний ресурс] // ACM Transactions on Information and System Security. – 2012. – Режим доступу до ресурсу: https://dl.acm.org/doi/10.1145/2382436.2382442.

21. Vaswani A. Attention Is All You Need [Електронний ресурс] // Advances in Neural Information Processing Systems. – 2017. – Режим доступу до ресурсу:

https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

22. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Електронний ресурс] // NAACL-HLT. – 2019. – Режим доступу до ресурсу: https://aclanthology.org/N19-1423/.

23. Caliskan A., Harang R., Liu A. De-anonymizing Programmers via Code Stylometry [Електронний ресурс] // 24th USENIX Security Symposium. – 2015. – Режим доступу до ресурсу: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/caliskan-islam.

24. Kim Y. Convolutional Neural Networks for Sentence Classification [Електронний ресурс] // EMNLP. – 2014. – Режим доступу до ресурсу: https://aclanthology.org/D14-1181/.

25. Joulin A., Grave E., Bojanowski P., Mikolov T. Bag of Tricks for Efficient Text Classification [Електронний ресурс] // EACL. – 2017. – Режим доступу до ресурсу: https://aclanthology.org/E17-2068/.

26. Rosenblum N., Miller B., Zhu X. Who wrote this? Identifying the author of program binaries [Електронний ресурс] // European Symposium on Research in Computer Security. – 2011. – Режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-3-642-23822-2_10.

27. Otter D. W., Medina J. R., Kalita J. K. A Survey of the Usages of Deep Learning for Natural Language Processing [Електронний ресурс] // IEEE Transactions on Neural Networks and Learning Systems. – 2020. – Режим доступу до ресурсу: https://ieeexplore.ieee.org/document/8726639.

28. Abuhamad M. et al. Large-Scale and Language-Oblivious Code Authorship Identification [Електронний ресурс] // ACM SIGSAC Conference

on Computer and Communications Security. – 2018. – Режим доступу до ресурсу: https://dl.acm.org/doi/10.1145/3243734.3243738.

29. Lande D. V. Open Source Intelligence: The concept, legal and ethical aspects [Електронний ресурс] // Theoretical and Applied Cybersecurity. – 2021. – Режим доступу до ресурсу: http://tacs.KPI.ua/article/view/229040.

30. Liu Y., Ott M., Goyal N. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Електронний ресурс] // arXiv preprint. – 2019. – Режим доступу до ресурсу: https://arxiv.org/abs/1907.11692.

31. Eder M., Rybicki J., Kestemont M. Stylometry with R: A Package for Computational Text Analysis [Електронний ресурс] // The R Journal. – 2016. – Режим доступу до ресурсу: https://journal.r-project.org/archive/2016/RJ-2016-007/index.html.

32. Juola P. JGAAP: A System for Authorship Attribution [Електронний ресурс] // Journal of Law and Policy. – 2009. – Режим доступу до ресурсу: https://brooklynworks.brooklaw.edu/jlp/vol21/iss2/1/.

33. Abbasi A., Chen H. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace [Електронний ресурс] // ACM Transactions on Information Systems. – 2008. – Режим доступу до ресурсу: https://dl.acm.org/doi/10.1145/1344411.1344413.

34. Mitchell E. et al. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature [Електронний ресурс] // ICML. – 2023. – Режим доступу до ресурсу: https://proceedings.mlr.press/v202/mitchell23a.html.

35. Soman S. An Overview of Binary Code Obfuscation Techniques [Електронний ресурс] // International Journal of Computer Science and Information Security. – 2017. – Режим доступу до ресурсу: https://www.researchgate.net/publication/320490799.

36. Scikit-learn: Machine Learning in Python [Електронний ресурс] // Journal of Machine Learning Research. – 2011. – Vol. 12. – P. 2825–2830. – Режим доступу до ресурсу: https://scikit-learn.org/stable/.

37. Bird S., Klein E., Loper E. Natural Language Processing with Python (NLTK) [Електронний ресурс] // O'Reilly Media. – 2009. – Режим доступу до ресурсу: https://www.nltk.org/.

38. Francis W. N., Kucera H. Brown Corpus Manual [Електронний ресурс] // Brown University. – 1979. – Режим доступу до ресурсу: http://icame.uib.no/brown/bcm.html.

39. Project Gutenberg: Free eBooks [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://www.gutenberg.org/.

ДОДАТОК А

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
import warnings

from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

warnings.filterwarnings("ignore")

class HardModeGenerator:
    def __init__(self):
        self.templates = [
            "def calculate_sum(a, b): return a + b",
            "if value > 10: print(value)",
            "data = [x for x in range(100) if x % 2 == 0]",
            "result = input_data['key'] * 5 + offset",
            "import os, sys, re, json",
            "class DataHandler(BaseModel): pass",
            "x = y + z - k / 2",
            "for item in container: process(item)"
        ]

    def _style_minified(self, text):
        t = text.replace(" ", "")
        for k in ['def', 'return', 'class', 'import', 'for', 'in', 'if',
'else']:
            t = t.replace(k, k + " ")
        return " ".join(t.split())

    def _style_spaced(self, text):
```

```
        ops = "+-*/=><:,[](){}%"
        for char in ops:
            text = text.replace(char, f" {char} ")
        return " ".join(text.split())

    def _style_vertical(self, text):
        return text.replace(":", ":\n  ").replace(",", ",\n  ")

    def create_dataset(self, n_samples=600):
        X, y = [], []
        for _ in range(n_samples // 3):
            base = random.choice(self.templates)
            base += f" # id:{random.randint(1,999)}"

            X.append(self._style_minified(base))
            y.append(0)
            X.append(self._style_spaced(base))
            y.append(1)
            X.append(self._style_vertical(base))
            y.append(2)

        combined = list(zip(X, y))
        random.shuffle(combined)
        X, y = zip(*combined)
        return list(X), list(y)

class HandcraftedStylometry:
    def __init__(self):
        self.clf = RandomForestClassifier(n_estimators=100, random_state=42)

    def extract(self, texts):
        features = []
        for t in texts:
            l = len(t) if len(t) > 0 else 1
            space_ratio = t.count(' ') / l
            newline_ratio = t.count('\n') / l
            special_density = (t.count('=') + t.count(':') + t.count(',')) /
l
```

```
            has_spaced_eq = 1 if ' = ' in t else 0


            features.append([space_ratio, newline_ratio, special_density,
has_spaced_eq])
        return np.array(features)


    def train(self, X, y):
        self.clf.fit(self.extract(X), y)
    def predict(self, X):
        return self.clf.predict(self.extract(X))


class WordLevelNLP:
    def __init__(self, model_type):
        self.vec = TfidfVectorizer(analyzer='word', max_features=1000)


        if model_type == 'svm':
            self.clf = SVC(kernel='linear', random_state=42)
        elif model_type == 'logreg':
            self.clf = LogisticRegression(max_iter=500, random_state=42)
        elif model_type == 'bayes':
            self.clf = MultinomialNB()


    def train(self, X, y):
        X_vec = self.vec.fit_transform(X)
        self.clf.fit(X_vec, y)
    def predict(self, X):
        return self.clf.predict(self.vec.transform(X))


def run_combined_plot():
    gen = HardModeGenerator()
    X_full, y_full = gen.create_dataset(n_samples=600)
    X_train_full, X_test, y_train_full, y_test = train_test_split(X_full,
y_full, test_size=0.2, random_state=42)


    models = {
        "Стилометрія": HandcraftedStylometry(),
        "LogReg": WordLevelNLP('logreg'),
        "SVM": WordLevelNLP('svm'),
```

```
    "Naive Bayes": WordLevelNLP('bayes')
}

train_sizes = [30, 60, 120, 240, 360, 480]
history = {name: [] for name in models}

print("Запуск навчання...")
for size in train_sizes:
    print(f" > Обробка {size} зразків...")
    X_sub = X_train_full[:size]
    y_sub = y_train_full[:size]

    for name, model in models.items():
        model.train(X_sub, y_sub)
        preds = model.predict(X_test)
        acc = accuracy_score(y_test, preds)
        history[name].append(acc)

print("\nПобудова загального графіка...")
plt.figure(figsize=(12, 8))

styles = [
    {'color': 'green', 'marker': 'o', 'width': 3, 'style': '-'},
    {'color': 'red', 'marker': 's', 'width': 2, 'style': '--'},
    {'color': 'orange', 'marker': '^', 'width': 2, 'style': '--'},
    {'color': 'blue', 'marker': 'x', 'width': 2, 'style': '--'}
]

for i, (name, accs) in enumerate(history.items()):
    s = styles[i]
    plt.plot(train_sizes, accs,
             label=name,
             color=s['color'],
             marker=s['marker'],
             linewidth=s['width'],
             linestyle=s['style'])
```

```
    plt.xlabel('Кількість навчальних зразків', fontsize=12)
    plt.ylabel('Точність ', fontsize=12)
    plt.ylim(0, 1.1)
    plt.grid(True, alpha=0.3)
    plt.legend(fontsize=12, loc='center right')

    for name, accs in history.items():
        if accs:
            plt.text(train_sizes[-1]+5, accs[-1], f"{accs[-1]:.2f}",
fontsize=10, fontweight='bold')

    try:
        plt.show()
    except:
        pass

    final_stats = []
    for name, accs in history.items():
        if accs:
            final_stats.append({"Модель": name, "Фінальна точність": accs[-
1]})

    df = pd.DataFrame(final_stats).sort_values(by="Фінальна точність",
ascending=False)
    print("\n=== ПІДСУМКОВА ТАБЛИЦЯ ===")
    print(df.to_string(index=False))

if __name__ == "__main__":
    run_combined_plot()
```

ДОДАТОК Б

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import OneClassSVM
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor
from sklearn.linear_model import SGDOneClassSVM
from sklearn.metrics import roc_auc_score

print("1. Завантаження даних (NLTK Gutenberg)...")
try:
    nltk.data.find('corpora/gutenberg')
except LookupError:
    nltk.download('gutenberg')
    nltk.download('punkt')

from nltk.corpus import gutenberg

def get_chunks(fileid, chunk_size=5):
    sents = gutenberg.sents(fileid)
    chunks = []
    current_chunk = []
    for sent in sents:
        current_chunk.append(" ".join(sent))
        if len(current_chunk) >= chunk_size:
            text = " ".join(current_chunk)
            if 200 < len(text) < 1500:
                chunks.append(text)
            current_chunk = []
    return chunks

owner_texts = get_chunks('austen-emma.txt')
```

```
intruder_texts = get_chunks('chesterton-thursday.txt')


train_n = 300

test_n = 50

attack_n = 50


stream = owner_texts[:train_n] + owner_texts[train_n:train_n+test_n] +
intruder_texts[:attack_n]

attack_start = train_n + test_n


y_true = np.array([0] * attack_start + [1] * attack_n)


print(f"Всього блоків: {len(stream)}")

print(f"Атака починається з індексу: {attack_start}")


models = {

    "One-Class SVM (RBF)": OneClassSVM(kernel='rbf', gamma='scale', nu=0.05),

    "Isolation Forest": IsolationForest(contamination=0.05, random_state=42,
n_jobs=-1),

    "Local Outlier Factor": LocalOutlierFactor(novelty=True,
contamination=0.05, n_neighbors=20),

    "SGD Linear SVM": SGDOneClassSVM(nu=0.05, random_state=42)

}


def normalize_scores(scores):

    scores = -scores

    mn, mx = scores.min(), scores.max()

    if mx - mn == 0: return scores

    return (scores - mn) / (mx - mn) * 10


stats_data = []


fig, axes = plt.subplots(2, 2, figsize=(16, 12))

axes = axes.flatten()


print("\n2. Навчання моделей та розрахунок метрик...")


for i, (name, model_base) in enumerate(models.items()):

    ax = axes[i]
```

```
    print(f"   -> Модель: {name}...")


    pipe_stylo = make_pipeline(
        TfidfVectorizer(analyzer='char', ngram_range=(3, 3),
max_features=1000, min_df=5),
        StandardScaler(with_mean=False),
        model_base
    )
    pipe_stylo.fit(stream[:train_n])


    raw_stylo = pipe_stylo.decision_function(stream)
    scores_stylo = normalize_scores(raw_stylo)


    auc_stylo = roc_auc_score(y_true, scores_stylo)
    hits_stylo = sum(1 for idx, s in enumerate(scores_stylo) if s > 5.0 and
y_true[idx] == 1)
    fps_stylo = sum(1 for idx, s in enumerate(scores_stylo) if s > 5.0 and
y_true[idx] == 0)


    stats_data.append({
        "Algorithm": name,
        "Method": "Stylometry (Char 3-grams)",
        "AUC-ROC": auc_stylo,
        "Detections (TP)": f"{hits_stylo}/{attack_n}",
        "False Alarms (FP)": f"{fps_stylo}/{attack_start}"
    })


    model_clone = model_base.__class__(**model_base.get_params())
    pipe_nlp = make_pipeline(
        TfidfVectorizer(analyzer='word', stop_words='english',
max_features=1000),
        StandardScaler(with_mean=False),
        model_clone
    )
    pipe_nlp.fit(stream[:train_n])


    raw_nlp = pipe_nlp.decision_function(stream)
    scores_nlp = normalize_scores(raw_nlp)
```

```
    auc_nlp = roc_auc_score(y_true, scores_nlp)
    hits_nlp = sum(1 for idx, s in enumerate(scores_nlp) if s > 5.0 and
y_true[idx] == 1)
    fps_nlp = sum(1 for idx, s in enumerate(scores_nlp) if s > 5.0 and
y_true[idx] == 0)


    stats_data.append({
        "Algorithm": name,
        "Method": "Standard NLP (Words)",
        "AUC-ROC": auc_nlp,
        "Detections (TP)": f"{hits_nlp}/{attack_n}",
        "False Alarms (FP)": f"{fps_nlp}/{attack_start}"
    })


    ax.axvspan(attack_start, len(stream), color='red', alpha=0.1)
    ax.plot(scores_stylo, color='blue', linewidth=2, label=f'Stylo
(AUC={auc_stylo:.2f})')
    ax.plot(scores_nlp, color='green', linestyle=':', alpha=0.8, label=f'NLP
(AUC={auc_nlp:.2f})')
    ax.set_title(name)
    ax.set_ylabel('Anomaly Score')
    ax.grid(True, alpha=0.3)
    ax.legend(loc='upper left')


plt.suptitle('Benchmark: Stylometry vs NLP Anomaly Detection', fontsize=16)
plt.tight_layout()
plt.subplots_adjust(top=0.90)
plt.savefig("my_plot.png")
plt.show()
df_results = pd.DataFrame(stats_data)
df_results = df_results.sort_values(by=["Algorithm", "AUC-ROC"],
ascending=[True, False])
print("\n" + "="*85)
print(f"{'ПІДСУМКОВА ТАБЛИЦЯ РЕЗУЛЬТАТІВ':^85}")
print("="*85)
print(df_results.to_string(index=False))
print("="*85)
```

ДОДАТОК В

```
import numpy as np
import pandas as pd
import nltk
import matplotlib.pyplot as plt
import random
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.pipeline import Pipeline
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB


try:
    nltk.data.find('corpora/brown')
except LookupError:
    print("Завантаження Brown Corpus...")
    nltk.download('brown')


from nltk.corpus import brown


def get_data_by_category(categories, label, chunk_size=150):
    data, labels = [], []
    for cat in categories:
        try:
            words = brown.words(categories=cat)
            for i in range(0, len(words), chunk_size):
                chunk = words[i:i+chunk_size]
                if len(chunk) == chunk_size:
```

```
                    data.append(" ".join(chunk))
                    labels.append(label)
        except ValueError:
            pass
    return data, labels

print("Формування вибірок (Міжжанровий аналіз)...")

X_train_0, y_train_0 = get_data_by_category(['government'], 0)
X_train_1, y_train_1 = get_data_by_category(['adventure'], 1)
X_train = X_train_0 + X_train_1
y_train = y_train_0 + y_train_1

X_test_0, y_test_0 = get_data_by_category(['learned'], 0)
X_test_1, y_test_1 = get_data_by_category(['romance'], 1)
X_test = X_test_0 + X_test_1
y_test = y_test_0 + y_test_1

print(f"Навчальна вибірка: {len(X_train)} фрагментів (Уряд vs Пригоди)")
print(f"Тестова вибірка:   {len(X_test)} фрагментів (Наука vs Романи)")

class AdvancedStylometryExtractor(BaseEstimator, TransformerMixin):
    def fit(self, X, y=None): return self

    def transform(self, X):
        features = []
        func_words = ['the', 'of', 'and', 'to', 'a', 'in', 'that', 'is',
'was', 'he',
                      'for', 'it', 'with', 'as', 'his', 'on', 'be', 'at',
'by', 'i',
                      'this', 'had', 'not', 'are', 'but', 'from', 'or',
'have', 'an']

        for text in X:
            words = text.split()
```

```
            char_len = len(text)
            n_words = len(words) if words else 1

            avg_word_len = sum(len(w) for w in words) / n_words
            ttr = len(set(words)) / n_words

            quote_freq = text.count('"') / char_len * 1000
            comma_freq = text.count(',') / char_len * 1000
            semicolon_freq = text.count(';') / char_len * 1000
            question_freq = text.count('?') / char_len * 1000

            fw_freqs = [words.count(fw) / n_words for fw in func_words]

            formal_marker = words.count('which') / n_words
            narrative_marker = (words.count('he') + words.count('she') +
words.count('said')) / n_words

            features.append([avg_word_len, ttr, quote_freq, comma_freq,
semicolon_freq, question_freq, formal_marker, narrative_marker] + fw_freqs)

        return np.array(features)

classifiers = {
    "Logistic Regression": LogisticRegression(max_iter=1000,
random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100,
random_state=42),
    "SVM (RBF Kernel)": SVC(kernel='rbf', random_state=42),
    "KNN (5 Neighbors)": KNeighborsClassifier(n_neighbors=5),
    "Gradient Boosting": GradientBoostingClassifier(random_state=42)
}

combined_train = list(zip(X_train, y_train))
random.shuffle(combined_train)
X_train_shuffled, y_train_shuffled = zip(*combined_train)
```

```
data_fractions = np.linspace(0.1, 1.0, 10)
dataset_sizes = [int(len(X_train_shuffled) * f) for f in data_fractions]

fig, axes = plt.subplots(len(classifiers), 1, figsize=(10, 25))
plt.subplots_adjust(hspace=0.5)

if len(classifiers) == 1: axes = [axes]

for ax, (name, clf) in zip(axes, classifiers.items()):
    print(f"Обробка моделі: {name}...")
    std_scores = []
    stylo_scores = []

    for size in dataset_sizes:
        X_sub = X_train_shuffled[:size]
        y_sub = y_train_shuffled[:size]

        try:
            pipe_std = Pipeline([
                ('tfidf', TfidfVectorizer(stop_words='english',
max_features=5000)),
                ('clf', clf)
            ])
            pipe_std.fit(X_sub, y_sub)
            std_scores.append(accuracy_score(y_test,
pipe_std.predict(X_test)))
        except:
            std_scores.append(0.5)

        try:
            pipe_stylo = Pipeline([
                ('extractor', AdvancedStylometryExtractor()),
                ('scaler', MinMaxScaler()),
                ('clf', clf)
            ])
```

```
            pipe_stylo.fit(X_sub, y_sub)
            stylo_scores.append(accuracy_score(y_test,
pipe_stylo.predict(X_test)))
        except:
            stylo_scores.append(0.5)

    ax.plot(dataset_sizes, std_scores, marker='o', linestyle='--',
color='blue', label='Стандартний ML')
    ax.plot(dataset_sizes, stylo_scores, marker='s', linestyle='-',
color='red', label='Стилометрія ')
    ax.set_title(name)
    ax.set_xlabel("Розмір навчальної вибірки")
    ax.set_ylabel("Точність")
    ax.set_ylim(0.4, 1.05)
    ax.grid(True, alpha=0.3)
    ax.legend()

plt.show()

print("\n" + "="*85)
print(f"{'Алгоритм':<25} | {'Звичайні моделі':<25} | {'Стилометрія':<25}")
print("="*85)

for name, clf in classifiers.items():
    pipe_std = Pipeline([
        ('tfidf', TfidfVectorizer(stop_words='english', max_features=5000)),
        ('clf', clf)
    ])
    pipe_std.fit(X_train, y_train)
    acc_std = accuracy_score(y_test, pipe_std.predict(X_test))

    pipe_stylo = Pipeline([
        ('extractor', AdvancedStylometryExtractor()),
        ('scaler', MinMaxScaler()),
        ('clf', clf)
```

```
    ])

    pipe_stylo.fit(X_train, y_train)

    acc_stylo = accuracy_score(y_test, pipe_stylo.predict(X_test))


    print(f"{name:<25} | {acc_std * 100:>18.1f}%       | {acc_stylo *
100:>18.1f}%")


print("="*85)
```

ДОДАТОК Г

```
import numpy as np
import re
import random
import nltk
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB

try:
    nltk.data.find('corpora/brown')
except LookupError:
    print("Завантаження Brown Corpus...")
    nltk.download('brown')
    nltk.download('punkt')

from nltk.corpus import brown

def simulate_russian_interference(text):
    text = re.sub(r'\b(the|a|an)\b', '', text, flags=re.IGNORECASE)
    text = re.sub(r'\b(am|is|are|was|were)\b', '', text, flags=re.IGNORECASE)
    if random.random() < 0.3:
        text = re.sub(r'\bto\b', '', text, flags=re.IGNORECASE)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

print("Формування датасету (Носій мови vs Російський акцент)...")

categories = ['news', 'editorial', 'hobbies', 'government', 'learned']
sentences = [" ".join(sent) for sent in brown.sents(categories=categories)]
```

```
sentences = [s for s in sentences if len(s) > 25]
random.shuffle(sentences)

limit = 4000
native_samples = sentences[:limit]
russian_samples = [simulate_russian_interference(s) for s in
sentences[limit:limit*2]]

X = native_samples + russian_samples
y = [0] * len(native_samples) + [1] * len(russian_samples)

print(f"Загальна кількість речень: {len(X)}")
print(f"Клас 'Носій мови': {len(native_samples)}")
print(f"Клас 'Імітація акценту': {len(russian_samples)}")

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

classifiers = {
    "Logistic Regression": LogisticRegression(max_iter=1000),
    "Random Forest": RandomForestClassifier(n_estimators=100),
    "Naive Bayes (MNB)": MultinomialNB(),
    "Support Vector Machine": SVC(kernel='linear')
}

print("\n" + "="*85)
print("Генерація графіків навчання (Learning Curves)...")
print("="*85)

combined_train = list(zip(X_train, y_train))
random.shuffle(combined_train)
X_train_shuffled, y_train_shuffled = zip(*combined_train)

data_fractions = np.linspace(0.1, 1.0, 10)
dataset_sizes = [int(len(X_train_shuffled) * f) for f in data_fractions]

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
axes = axes.flatten()
```

```
results_final = []


for ax, (name, clf) in zip(axes, classifiers.items()):
    print(f"Обробка моделі: {name}...")
    std_scores = []
    stylo_scores = []


    for size in dataset_sizes:
        X_sub = X_train_shuffled[:size]
        y_sub = y_train_shuffled[:size]


        pipe_std = Pipeline([
            ('tfidf', TfidfVectorizer(analyzer='word',
stop_words='english')),
            ('clf', clf)
        ])
        pipe_std.fit(X_sub, y_sub)
        std_scores.append(accuracy_score(y_test, pipe_std.predict(X_test)))


        pipe_stylo = Pipeline([
            ('tfidf', TfidfVectorizer(analyzer='char', ngram_range=(2, 3),
min_df=5)),
            ('clf', clf)
        ])
        pipe_stylo.fit(X_sub, y_sub)
        stylo_scores.append(accuracy_score(y_test,
pipe_stylo.predict(X_test)))


    results_final.append((name, std_scores[-1], stylo_scores[-1]))


    ax.plot(dataset_sizes, std_scores, marker='o', linestyle='--',
color='blue', label='Standard ML (Content)')
    ax.plot(dataset_sizes, stylo_scores, marker='s', linestyle='-',
color='red', label='Stylometry (Chars)')
    ax.set_title(name)
    ax.set_xlabel('Sample Size')
    ax.set_ylabel('Accuracy')
    ax.set_ylim(0.4, 1.05)
```

```
    ax.grid(True, alpha=0.3)
    ax.legend()


plt.tight_layout()
plt.show()


print("\n" + "="*85)
print(f"{'ALGORITHM':<25} | {'STD ACCURACY (Content)':<25} | {'STYLO ACCURACY
(Style)'}")
print("="*85)


for res in results_final:
    print(f"{res[0]:<25} | {res[1]*100:>18.1f}%       |
{res[2]*100:>18.1f}%")


print("\n" + "="*85)
```

ДОДАТОК Д

```
import numpy as np
import pandas as pd
import nltk
import matplotlib.pyplot as plt
from nltk.corpus import gutenberg
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.utils import shuffle

try:
    nltk.data.find('corpora/gutenberg')
except LookupError:
    nltk.download('gutenberg')
    nltk.download('punkt')

CHUNK_SIZE = 2500
NUM_FEATURES = 100

def get_chunks(file_id):
    words = [w.lower() for w in gutenberg.words(file_id) if w.isalpha()]
    return [" ".join(words[i:i + CHUNK_SIZE]) for i in range(0, len(words),
CHUNK_SIZE) if i + CHUNK_SIZE <= len(words)]

train_books = {
    'Jane Austen': 'austen-emma.txt',
    'G.K. Chesterton': 'chesterton-ball.txt',
    'Herman Melville': 'melville-moby_dick.txt'
}

test_books = {
    'Jane Austen': 'austen-persuasion.txt',
    'G.K. Chesterton': 'chesterton-thursday.txt',
    'Herman Melville': 'melville-moby_dick.txt'
}
```

```
print("1. Завантаження та розбиття книг...")

X_train_raw, y_train_full = [], []
for auth, fname in train_books.items():
    chunks = get_chunks(fname)
    if auth == 'Herman Melville': chunks = chunks[:len(chunks)//2]
    X_train_raw.extend(chunks)
    y_train_full.extend([auth] * len(chunks))

X_test_raw, y_test_full = [], []
for auth, fname in test_books.items():
    chunks = get_chunks(fname)
    if auth == 'Herman Melville': chunks = chunks[len(chunks)//2:]
    X_test_raw.extend(chunks)
    y_test_full.extend([auth] * len(chunks))

X_train_raw, y_train_full = shuffle(X_train_raw, y_train_full,
random_state=42)

print(f"   Всього навчальних фрагментів: {len(X_train_raw)}")
print(f"   Всього тестових фрагментів: {len(X_test_raw)}")

all_tokens = nltk.word_tokenize(" ".join(X_train_raw))
fdist = nltk.FreqDist(all_tokens)
vocab = [w for w, c in fdist.most_common(NUM_FEATURES)]

def vectorize(text_list, vocabulary):
    matrix = []
    for text in text_list:
        tokens = nltk.word_tokenize(text)
        n = len(tokens) if len(tokens) > 0 else 1
        counts = nltk.FreqDist(tokens)
        row = [counts[w]/n for w in vocabulary]
        matrix.append(row)
    return np.array(matrix)

X_train_vec_full = vectorize(X_train_raw, vocab)
```

```
X_test_vec_full = vectorize(X_test_raw, vocab)


print(f"2. Векторизація завершена. Використано {NUM_FEATURES} слів-
маркерів.")


train_steps = [5, 10, 20, 30, 40, 50, len(X_train_raw)]
train_steps = sorted(list(set([x for x in train_steps if x <=
len(X_train_raw)])))


history = {
    'Стилометрія (Delta)': [],
    '(SVM)': [],
    '(LogReg)': []
}


print("\n3. Запуск симуляції (нарощування обсягу даних)...")


for n in train_steps:
    X_sub = X_train_vec_full[:n]
    y_sub = y_train_full[:n]
    means = X_sub.mean(axis=0)
    stds = X_sub.std(axis=0)
    stds[stds == 0] = 1e-6
    profiles = {}
    unique_authors = list(set(y_sub))
    for auth in unique_authors:
        idxs = [i for i, x in enumerate(y_sub) if x == auth]
        profiles[auth] = (X_sub[idxs].mean(axis=0) - means) / stds
    correct = 0
    if len(unique_authors) < 3:
        history['Стилометрія (Delta)'].append(0.0)
    else:
        for i, vec in enumerate(X_test_vec_full):
            z_unknown = (vec - means) / stds
            dists = {a: np.mean(np.abs(z_unknown - p)) for a, p in
profiles.items()}
            pred = min(dists, key=dists.get)
            if pred == y_test_full[i]: correct += 1
        history['Стилометрія (Delta)'].append(correct / len(y_test_full))
```

```
    try:
        clf_svm = SVC(kernel='linear')
        clf_svm.fit(X_sub, y_sub)
        acc_svm = accuracy_score(y_test_full,
clf_svm.predict(X_test_vec_full))
        history['(SVM)'].append(acc_svm)
    except:
        history['(SVM)'].append(0.0)
    try:
        clf_lr = LogisticRegression(max_iter=1000)
        clf_lr.fit(X_sub, y_sub)
        acc_lr = accuracy_score(y_test_full, clf_lr.predict(X_test_vec_full))
        history['(LogReg)'].append(acc_lr)
    except:
        history['(LogReg)'].append(0.0)
df_res = pd.DataFrame(history)
df_res.index = train_steps
df_res.index.name = "К-сть зразків"
print("\n=== РЕЗУЛЬТАТИ ТОЧНОСТІ===")
print(df_res.round(3))
plt.figure(figsize=(10, 6))
plt.plot(train_steps, history['Стилометрія (Delta)'], 'o-',
label='Стилометрія (Delta)', color='green', linewidth=2)
plt.plot(train_steps, history['(SVM)'], '^-', label='(SVM)', color='blue',
linestyle='--')
plt.plot(train_steps, history['(LogReg)'], 's-', label='(LogReg)',
color='orange', linestyle='--')
plt.xlabel('Кількість навчальних фрагментів')
plt.ylabel('Точність')
plt.ylim(0, 1.05)
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.show()
```