Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

В.о. зав. кафедрою КІСМіТ
Марина ЄСІНА
"Допущено до захисту"

« »_____2025р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Дослідження, аналіз та порівняння методів і засобів штучного інтелекту для захисту даних від фішингових атак» / «Research, analysis and comparison of methods and means of artificial intelligence for data protection against phishing attacks»

оцінка « _____ »

Голова ЕК

Мичуда Л.З.

Керівник: PhD Вілігура В. В.

Рецензент: к.т.н. проф. Качко О. Г.

Виконавець: студент групи КБ-42

Тубольцев О. Д.

Харків 2025

РЕФЕРАТ

Пояснювальна записка до бакалаврської кваліфікаційної роботи містить 60 сторінок, 10 рисунків, 5 таблиць, 1 додаток і 42 джерела.

Метою роботи є дослідження та аналіз використання методів штучного інтелекту (ШІ), провідних моделей, інструментів і методологій для виявлення фішингових атак. Дослідження зосереджене на таких технологіях ШІ, як машинне навчання, глибинне навчання та обробка природної мови, та на тому, як їх можна ефективно застосовувати для ідентифікації та запобігання фішинговим загрозам через різні канали комунікації – електронну пошту, веб-сайти та месенджери.

Об’єкт дослідження – фішингові атаки як сучасна кіберзагроза безпеці.

Предмет дослідження – алгоритми та методи штучного інтелекту (Random Forest, Logistic Regression, NLP тощо), що використовуються для виявлення фішингових повідомлень; набори даних і засоби попередньої обробки, що застосовуються для аналізу фішингу; а також правові та етичні засади регулювання застосування ШІ в сфері кібербезпеки.

Основними методами дослідження є порівняльний аналіз моделей виявлення фішингу, навчання та тестування класифікаційних алгоритмів на відкритих наборах даних (PhishTank, UCI, Kaggle), а також розробка інструменту виявлення фішингу з інтерфейсом командного рядка на основі моделей ШІ. Додатково застосовано методи аналізу метрик (точність, повнота, F1-score, ROC-AUC), візуалізації результатів і огляду міжнародних стандартів.

У дослідженні розглянуто:

- теоретичні основи фішингу та його типи (email-, website-, smishing-, vishing-фішинг);
- напрями застосування ШІ для виявлення фішингових атак;
- приклади практичного використання від Google, Microsoft та AWS;

- ефективність алгоритмів машинного навчання, зокрема Random Forest та Logistic Regression;
- інтеграцію моделей у прототип інструменту для боротьби з фішингом;
- нормативні та етичні аспекти, такі як GDPR, ISO/IEC 42001, NIST AI RMF та EU AI Act.

Результати дослідження можуть бути використані для практичної розробки засобів захисту від фішингу, глибшого розуміння механізмів виявлення фішингу за допомогою ШІ, а також як основа для подальших наукових досліджень у сфері кібербезпеки та науки про дані.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, ФІШИНГ, ВИЯВЛЕННЯ ФІШИНГУ, МАШИННЕ НАВЧАННЯ, ГЛИБИННЕ НАВЧАННЯ, ОБРОБКА ПРИРОДНОЇ МОВИ, КІБЕРБЕЗПЕКА, ФІШИНГОВІ ЕЛЕКТРОННІ ЛИСТИ, ФІШИНГОВІ САЙТИ, ISO/IEC 42001, GDPR, EU AI ACT, RANDOM FOREST, LOGISTIC REGRESSION, NLP.

ABSTRACT

The explanatory note to the bachelor's qualifying work consists of 60 pages, 10 figures, 5 tables, 1 appendix, and 42 sources.

The objective of the work is to investigate and analyze the utilization of artificial intelligence (AI) techniques, eminent models, tools, and methodologies for the detection of phishing attacks. The research focuses on AI technologies – machine learning, deep learning, and natural language processing – and how these can be effectively utilized to identify and prevent phishing threats through various mediums of communication, like email, websites, and messaging apps.

The object of research is the phenomenon of phishing attacks as a modern cyber threat to security.

The subject of research is artificial intelligence algorithms and techniques (Random Forest, Logistic Regression, NLP, etc.) used to identify phishing messages; data sets and preprocessing tools used in phishing analysis; and legal and ethical frameworks regulating the use of AI in cybersecurity.

The main research methods are comparative analysis of phishing detection models, training and testing of classification algorithms on public datasets (e.g., PhishTank, UCI, Kaggle), and development of a phishing detection CLI-tool based on AI models. Additional methods include metric analysis (accuracy, precision, recall, F1-score, ROC-AUC), results visualization, and review of international standards.

In this study, we explored:

- the background theory of phishing and phishing types (email, website, smishing, vishing);
- the direction of AI usage in the area of phishing detection;
- real-world examples from Google, Microsoft, and AWS;
- the performance of machine learning algorithms such as Random Forest and Logistic Regression;
- models integration into a prototype CLI-based anti-phishing tool;

- regulatory and ethical considerations, such as GDPR, ISO/IEC 42001, NIST AI RMF, and the EU AI Act.

The results of this study can be used for the practical development of anti-phishing products, for the further understanding of AI-based phishing detection mechanisms, and as a basis for future scientific research in the field of cybersecurity and data science.

Keywords: ARTIFICIAL INTELLIGENCE, PHISHING, PHISHING DETECTION, MACHINE LEARNING, DEEP LEARNING, NATURAL LANGUAGE PROCESSING, CYBERSECURITY, PHISHING EMAIL, PHISHING WEBSITES, ISO/IEC 42001, GDPR, EU AI ACT, RANDOM FOREST, LOGISTIC REGRESSION, NLP.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AI	– Artificial Intelligence
NLP	– Natural Language Processing
UBA	– User Behavior Analytics
ML	– Machine Learning
DL	– Deep Learning
SVM	– Support Vector Machine
NB	– Naive Bayes
RF	– Random Forest
LR	– Logistic Regression
ROC-AUC	– Receiver Operating Characteristic - Area Under Curve
TF-IDF	– Term Frequency - Inverse Document Frequency
URL	– Uniform Resource Locator
CLI	– Command Line Interface
BERT	– Bidirectional Encoder Representations from Transformers
API	– Application Programming Interface
DNS	– Domain Name System
SSL	– Secure Sockets Layer
GDPR	– General Data Protection Regulation
ISO/IEC 42001	– Стандарт систем управління ШІ
NIST AI RMF	– Risk Management Framework від NIST
EU AI Act	– Закон про штучний інтелект Європейського Союзу
DPO	– Data Protection Officer
DPIA	– Data Protection Impact Assessment
XAI	– Explainable Artificial Intelligence
CNN	– Convolutional Neural Network
RNN	– Recurrent Neural Network
LSTM	– Long Short-Term Memory
BiLSTM	– Bidirectional Long Short-Term Memory

XGBoost	– Extreme Gradient Boosting
FP	– False Positive
FN	– False Negative
TP	– True Positive
TN	– True Negative
FPR	– False Positive Rate
FNR	– False Negative Rate
MCC	– Matthews Correlation Coefficient
LLM	– Large Language Model
ШІ	– Штучний інтелект
КІ	– Критична інформаційна інфраструктура
МН	– Машинне навчання
ГН	– Глибинне навчання
ДП	– Дані персональні
ІС	– Інформаційна система
КСЗІ	– Комплексна система захисту інформації
ОС	– Операційна система
ІКС	– Інформаційно-комунікаційні системи
ІБ	– Інформаційна безпека
БД	– База даних

ВСТУП

У міру розвитку цифрового суспільства захист інформації став одним із найактуальніших питань для громадян, бізнесу та держав. Однією з найпоширеніших і найшвидше еволюціонуючих кіберзагроз є фішинг – атака соціальної інженерії, що має на меті ввести користувача в оману з метою отримання конфіденційної інформації, такої як облікові дані, фінансові відомості або особисті дані.

Зростання складності та масштабів фішингових атак зумовлено дедалі ширшим доступом до засобів автоматизації, анонімізації, а також – усе частіше – до інструментів штучного інтелекту (ШІ). Завдяки ШІ зловмисники здатні створювати надзвичайно правдоподібні фішингові електронні листи, підроблені сайти або навіть імітацію голосу з безпрецедентним рівнем реалістичності. Така ситуація створює нові виклики для традиційних засобів безпеки, які залишаються реактивними, статичними та недостатніми в умовах динамічних загроз, підсилених ШІ.

Відтак, впровадження та архітектура прогнозуючих систем виявлення фішингу на основі ШІ стали невід'ємною частиною сучасної практики кібербезпеки. Машинне навчання, обробка природної мови та інші інтелектуальні технології дають змогу системам безпеки виявляти аномальну поведінку, класифікувати шкідливий вміст і прогнозувати ймовірність атаки до моменту нанесення шкоди. Ці технології вже реалізовано у продуктах компаній Google, Microsoft, а також у численних проєктах з відкритим кодом, що істотно підвищують точність виявлення загроз.

Ця бакалаврська кваліфікаційна робота присвячена використанню штучного інтелекту для виявлення фішингових атак. У теоретичній частині розглянуто види фішингу та застосування ШІ у сфері кібербезпеки. У практичній частині демонструється реалізація та оцінка моделей виявлення фішингу – Random Forest та Logistic Regression – на основі реальних наборів даних. Окрему увагу приділено

нормативним вимогам та етичним аспектам застосування ШІ в системах кіберзахисту.

Актуальність дослідження зумовлена нагальною потребою в адаптивних і розумних механізмах захисту від фішингу, а також потенціалом ШІ у покращенні виявлення загроз з дотриманням вимог щодо захисту даних та відповідності міжнародним стандартам.

1 ОГЛЯД, АНАЛІЗ ТА ДОСВІД ЗАСТОСУВАННЯ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ ДАНИХ ВІД ФІШИНГОВИХ АТАК

1.1 Основні поняття фішингу та актуальність проблеми

У сучасному цифровому суспільстві фішинг є однією з найпоширеніших і найнебезпечніших форм соціальної інженерії, яка становить велику загрозу як для приватних користувачів, так і для організацій у всьому світі. Загалом, фішинг – це шахрайський метод, коли зловмисник намагається змусити жертву поділитися конфіденційною інформацією (облікові дані, банківська інформація, документи тощо), видаючи себе за довірену особу чи організацію. Ці атаки зазвичай здійснюються через електронну пошту, веб-сайти, SMS, телефонні дзвінки, повідомлення в месенджерах або соціальних мережах.

Термін "фішинг" виник у середині 1990-х років, коли хакери використовували підроблені електронні повідомлення для отримання паролів облікових записів America Online (AOL) [1]. Хакери імітували службові повідомлення, візуально копіюючи стиль офіційної підтримки, щоб обманом отримати логіни користувачів. Сьогодні фішинг значно еволюціонував, з десятками підтипів і методів доставки, новими технологіями, мобільними пристроями, хмарними сервісами і навіть генеративними моделями штучного інтелекту.

Згідно з останнім звітом Антифішингової робочої групи (APWG) [2], у 2023 році було зафіксовано понад 4,7 мільйона фішингових атак – рекорд за весь період спостережень. Понад 30% від загальної кількості атак були спрямовані на фінансовий сектор: банки, платіжні системи та страхові компанії. Другим за популярністю був сектор програмного забезпечення як послуги (SaaS) – хакери атакували облікові записи в Google Workspace, Microsoft 365, Dropbox, Zoom тощо.

Причини успішності фішингу полягають у тому, що він експлуатує психологічні особливості людей: довіру до авторитетів, відсутність уваги, необхідність діяти швидко в умовах загрози чи стресу. Дуже часто такі електронні листи містять стимули, що провокують імпульсивні дії користувачів: згадка слів

"ТЕРМІНОВО", "ПОТРІБНА ДІЯ", "ПОПЕРЕДЖЕННЯ БЕЗПЕКИ", які підвищують тиск на людину. Цей вплив добре описаний у літературі на перетині поведінкової психології та кібербезпеки, зокрема варто відзначити роботу [3], де класифіковано принципи соціальної інженерії у фішингових атаках.

Фішингові атаки загалом можна класифікувати як:

• масовий фішинг – однакове повідомлення надсилається тисячам користувачів;

• цільовий (спеціалізований фішинг) – атака персоналізована для конкретної жертви, використовуючи дані з LinkedIn, соціальних мереж або попередніх витоків;

• компрометація корпоративної електронної пошти – імітація внутрішньої корпоративної електронної пошти, часто – від імені керівництва.

Особливе занепокоєння викликає зростання генеративного фішингу, в якому за допомогою таких моделей, як GPT, BARD або Claude, зловмисники створюють граматично правильні, логічні та персоналізовані фішингові повідомлення, що уникають традиційних спам-фільтрів. Атаки, згенеровані ШІ, є більш переконливими і їх важче виявити, оскільки кожне повідомлення є унікальним [4].

Реальна небезпека фішингу проявляється не лише в індивідуальних втратах, але й у масштабних інцидентах, наприклад:

• у 2021 році фішингова атака призвела до злому системи Colonial Pipeline, що тимчасово зупинило енергетичну систему США [5];

• у 2020 році фішинговий лист став точкою входу для атаки SolarWinds, яка призвела до витоку даних з десятків урядових агентств США;

• у 2023 році через фішингову кампанію з імітацією OneDrive було скомпрометовано понад 10 тисяч облікових записів працівників освітнього сектору [6].

Примітно, що в контексті війни в Україні та поширення гібридних загроз, значно зросла кількість фішингових атак, що імітують державні установи, поштові сервіси, платформи державних послуг (наприклад, "Дія"). Атаки набули не просто рис шахрайства, а й характеристик інформаційно-психологічного впливу, дезінформації або навіть кібершпигунства.

Таким чином, сьогодні фішинг – це не просто поширена форма шахрайства, а складна, багатовекторна атака, яка еволюціонує разом з технологіями. Протидія цій загрозі може бути ефективною лише через використання не тільки традиційних заходів (антивірусів, фільтрів), але й інтелектуальних рішень на основі штучного інтелекту, здатних виявляти приховані закономірності, аномалії та сигнали фішингу в режимі реального часу.

1.2 Класифікація фішингових атак

Фішинг як явище охоплює широкий спектр методів, інструментів і каналів взаємодії з метою обману жертв для крадіжки даних. Успіх фішингових кампаній значною мірою залежить від використовуваного каналу доставки, рівня персоналізації та достовірності сценарію. Класифікація фішингових атак дозволяє не лише систематизувати загрози, але й точніше розробляти заходи захисту. Найпоширенішими типами сьогодні є [7]:

- Фішинг електронної пошти

Фішинг електронної пошти є початковим і найпоширенішим типом фішингової атаки, яка проводиться шляхом надсилання підробленої електронної пошти. Зловмисники видають себе за офіційні установи – банки, податкові служби, системи охорони здоров’я, соціальні мережі або навіть співробітників компанії.

Типове фішингове повідомлення:

• містить привертаючий увагу рядок теми ("Ваш обліковий запис буде заблоковано", "У вас є повернення коштів" тощо);

• містить запит на термінову дію ("Підтвердіть свої дані зараз", "Клацніть тут, щоб увійти" тощо);

• містить посилання на підроблений веб-сайт або вкладення (DOC, PDF, ZIP), що містить шкідливий код.

Атакуючі найчастіше використовують підміну (підробку адреси відправника), фішингові домени (наприклад, amaz0n-login.com) або ховають URL за кнопками чи скороченими посиланнями. Понад 80% успішних проникнень в корпоративні системи починалися з фішингу електронної пошти [2].

- Фішинг веб-сайтів (спуфінг)

Цей тип фішингу передбачає створення підроблених веб-сайтів, які імітують дизайн реальних ресурсів – банків, платіжних систем, служб доставки тощо. Користувачам, після кліку на посилання в фішинговому листі або рекламі, представляється веб-інтерфейс, який повністю відтворює зовнішній вигляд оригінального сайту. Введені дані (логін, пароль, CVV) негайно передаються зловмисникам.

Класичні ознаки:

• доменне ім'я, подібне до реального: g00gle.com, facebook-verification.net;

• наявність SSL-сертифіката (https), що дає помилкове відчуття безпеки;

• сайт може бути розміщений на хмарних платформах (наприклад, sites.google.com, webflow.io).

Фішингові сайти мають дуже обмежений життєвий цикл – зазвичай до 24 годин. Для боротьби з ними використовуються ботнети для моніторингу DNS, краудсорсні чорні списки (PhishTank, OpenPhish) і класифікатори URL на основі ШІ, наприклад, в Google Safe Browsing або Cloudflare Gateway.

- Смішинг (SMS-фішинг)

Смішинг – спроби фішингу, що доставляються через текстові повідомлення (SMS) [8]. Це особливо небезпечно, оскільки мобільні користувачі рідше підозрюють SMS і одразу клікають на посилання, не задумуючись. Крім того, мобільні екрани не завжди дозволяють побачити повний URL або деталі відправника.

Приклади повідомлень:

• "Ваш обліковий запис буде заблоковано. Перевірте інформацію тут: [скорочене посилання]";

• "Ваша посилка чекає. Оновіть деталі доставки: [URL]";

• "Ви виграли подарунок від ПриватБанку. Підтвердіть свої реквізити".

Під час пандемії COVID-19 зросла кількість смішинг-атак, що підробляли повідомлення від Міністерства охорони здоров'я, банків, соціальних виплат тощо. Згідно зі звітом ENISA Threat Landscape, 2022 [9], смішинг став особливо

популярним у Європі через зростаючу популярність QR-кодів і мобільного банкінгу.

- Вішинг (голосовий фішинг)

Вішинг – це форма фішингу, коли зловмисники телефонують, представляючись співробітником банку, поліцейськими, службою підтримки тощо. Основна мета – викликати термінову реакцію користувача і змусити його поділитися конфіденційною інформацією або навіть самостійно перевести кошти.

Техніки вішингу:

• підміна ідентифікатора абонента;

• звукові ефекти (фоновий "банківський зв'язок");

• імітація дзвінка з відомої установи (банк, поліція, "державна служба").

Останнім часом збільшуються випадки так званого deepfake-вішингу, де голос є підробленим, згенерованим за допомогою ШІ (наприклад, голос генерального директора компанії "дзвонить" бухгалтеру з проханням негайно переказати кошти). Один такий випадок описано в [11], де голос генерального директора був імітований для шахрайського отримання 220 000 євро.

1.3 Основи застосування штучного інтелекту в задачах кібербезпеки

Штучний інтелект (ШІ) радикально змінює методи захисту інформаційних систем, особливо у сфері виявлення фішингових атак та реагування на них. Системи ШІ можуть обробляти величезні обсяги структурованих і неструктурованих даних, виявляти аномалії, адаптуватися до нових патернів атак і навчатися самостійно, на відміну від традиційних інструментів, що використовують статичні сигнатури та фільтри. У кібербезпеці це відкриває абсолютно нові можливості – від проактивного виявлення фішингових повідомлень до автоматичного реагування на інциденти в реальному часі.

1.3.1 Внесок машинного навчання

Машинне навчання (ML) є наріжним каменем більшості сучасних систем виявлення фішингу. Концепція полягає у побудові моделі класифікації, яка

навчається розрізняти фішингові повідомлення та легітимні на основі набору ознак. Найпопулярніші алгоритми:

• Random Forest;

• Support Vector Machine (SVM);

• Decision Trees;

• Gradient Boosted Trees (GBM / XGBoost);

• k-nearest Neighbors (k-NN);

• Naive Bayes.

Типові ознаки для виявлення фішингу:

• кількість субдоменів в URL;

• довжина посилання;

• використання IP-адреси в домені;

• структура HTML-повідомлення;

• тригерні слова в тексті ("терміново", "перевірте", "натисніть тут");

• ознаки маніпуляції в темі листа (КАПСЛОК, !, $).

У роботі [11] зазначається, що моделі МН досягають точності 93-97% у виявленні фішингових листів, якщо використовуються правильні ознаки та очищення даних перед навчанням.

1.3.2 Використання глибинного навчання

Глибинне навчання (ГН) – це підрозділ ШІ, що займається багатошаровими нейронними мережами. Його сила полягає в автоматичному вилученні ознак з даних, без попереднього ручного конструювання. У захисті від фішингу ГН використовується для:

• обробки тексту (електронні листи, форми, заголовки);

• аналізу структури веб-сторінки (DOM-дерево, стилі, скрипти);

• класифікації зображень (веб-інтерфейси, вкладення);

• виявлення патернів у поведінці користувачів.

Популярні архітектури:

• LSTM / BiLSTM – підходять для тексту електронних листів, оскільки враховують послідовність слів;

• CNN – використовуються для аналізу зображень (наприклад, скріншотів фішингових веб-сайтів);

• Автоенкодери – для виявлення аномалій;

• Трансформери (BERT, MiniLM) – моделі обробки природної мови.

Робота [12] продемонструвала, що моделі ГН перевершують алгоритми МН у обробці складних випадків фішингу, особливо за наявності HTML, JavaScript або графічної інформації.

1.3.3 Обробка природної мови (NLP)

Обробка природної мови (NLP) є ключовим компонентом захисту від фішингу в корпоративних поштових скриньках. Технології NLP забезпечують:

• обробку семантики тексту електронного листа;

• виявлення стилістичних невідповідностей;

• категоризацію тональності повідомлення;

• співставлення стилю написання з попередніми листами від відправника.

Моделі трансформерів, такі як BERT або DistilBERT, надзвичайно ефективні у розпізнаванні підозрілих патернів. Наприклад, BERT можна дотренувати на корпусі фішингових текстів для виявлення маніпулятивних фраз. У дослідженні [12] було виявлено, що фішингові листи демонструють відхилення від граматичної структури, які виявляються моделлю трансформера.

1.3.4 Аналітика поведінки користувачів (UBA)

ШІ також широко застосовується в аналізі поведінки користувачів. Системи UBA відстежують "нормальну" поведінку користувача в мережі або системах (час входу, географія, тип операцій) і порівнюють її з поточною. У випадку аномалії (наприклад, вхід з нової IP-адреси або незвичні кліки в інтерфейсі пошти), система може автоматично активувати додаткові перевірки, блокування або сповіщення.

Наприклад, Microsoft Defender має вбудовані моделі ШІ, які реагують на зміни в поведінці облікового запису після компрометації фішингом – одразу після входу зловмисника змінюються частота запитів, маршрутизація та використання API.

1.3.5 Переваги підходів ШІ

У контексті боротьби з фішингом методи штучного інтелекту демонструють низку переваг, які роблять їх дедалі популярнішими в сучасних системах кіберзахисту. Застосування алгоритмів машинного навчання, глибинного навчання та обробки природної мови дозволяє значно підвищити точність виявлення загроз, скоротити час реагування, автоматизувати процеси моніторингу та зменшити вплив людського фактору.

На відміну від традиційних систем, які покладаються на сигнатури або ручне налаштування правил, інтелектуальні підходи здатні адаптуватися до нових загроз, працювати з великими обсягами даних і самостійно виявляти складні фішингові патерни без потреби в постійному втручанні людини. Ключові переваги таких систем наведено у таблиці 1.1.

Таблиця 1.1 – Узагальнена характеристика переваг штучного інтелекту у виявленні фішингових повідомлень

Перевага	Опис
Гнучкість	Моделі навчаються на нових даних і не залежать від заздалегідь визначених правил
Висока точність	Точність понад 97% у найкращих випадках (з належним налаштуванням)
Автоматизація	Можливість виявлення загроз у режимі 24/7 без втручання людини
Стійкість до обфускації	ШІ може виявляти приховані або модифіковані фішингові патерни
Масштабованість	Мільйони електронних листів або URL-адрес скануються без втрати продуктивності

Як показано в таблиці 1.1, інтелектуальні системи мають низку суттєвих переваг, зокрема здатність до адаптації, високу точність, автоматизоване функціонування та масштабованість. Саме ці властивості роблять їх ефективними інструментами у боротьбі з сучасними фішинговими атаками.

На відміну від традиційних рішень, що базуються на сигнатурному аналізі та ручному створенні правил, інтелектуальні підходи демонструють вищу гнучкість і здатність виявляти складні патерни. Наприклад, методи машинного навчання можуть розпізнавати поведінкові характеристики фішингових повідомлень і досягати точності понад 90% навіть за відсутності очевидних ознак атак.

Системи глибинного навчання, зокрема моделі типу BERT або LSTM, здатні враховувати контекст повідомлень і виявляти обхідні техніки, характерні для сучасних фішингових кампаній, у тому числі створених за допомогою генеративних моделей на основі LLM (Large Language Models).

Ще однією перевагою є здатність алгоритмів ШІ до адаптації. Завдяки постійному донавчанню на нових зразках даних, такі системи здатні реагувати на фішинг-атаки нульового дня, які ще не внесені до чорних списків або не мають характерних сигнатур [13].

Крім того, використання ШІ дозволяє значно зменшити кількість хибнопозитивних спрацьовувань, що є однією з основних проблем у традиційних системах кібербезпеки. За даними досліджень компаній Google та Sophos, впровадження LLM-моделей в браузер Chrome знизило ймовірність переходу на шахрайські сайти на понад 80% [14].

Важливо зазначити, що ШІ-підходи також підтримують масштабування – їх легко інтегрувати у великі корпоративні інфраструктури, системи електронної пошти, хмарні сервіси тощо.

Однак, незважаючи на всі переваги, використання штучного інтелекту вимагає належної етичної та нормативної регламентації. Це питання детально розглянуто в розділі 2 цієї роботи.

Таким чином, підходи на основі ШІ мають високий потенціал для забезпечення проактивного виявлення фішингу, а їх ефективність підтверджується як практичними прикладами з боку індустрії, так і результатами наукових досліджень.

1.4 Приклади практичного використання штучного інтелекту для протидії фішингу

Багаторічне зростання рівня фішингових атак спонукало технологічних лідерів інтегрувати механізми штучного інтелекту у своє захисне програмне забезпечення. Ці рішення здатні проактивно реагувати на нові патерни атак, генерувати динамічні сигнатури, виявляти аномалії в реальному часі та захищати користувачів ще до того, як шкідливе повідомлення надійде до поштової скриньки чи браузера.

1.4.1 Google: застосування ШІ для захисту електронної пошти та браузера

Google використовує багатошарову архітектуру ШІ у всіх своїх сервісах – Gmail, Chrome, Google Search та Android:

• фільтр спаму Gmail ШІ: Алгоритми глибинного та машинного навчання щомісяця сортують понад 300 мільярдів електронних листів. За оцінками компанії, понад 99,9% фішингових листів блокуються ще до того, як потрапляють до скриньки [15];

• Google Safe Browsing: Використовується Chrome, Safari та Firefox для попередження користувачів про шкідливі сайти. Щодня аналізується понад 40 мільйонів URL-адрес моделями ШІ. Safe Browsing оновлюється в реальному часі, виявляючи нові фішингові ресурси протягом хвилин після їхньої появи;

• Gemini Nano та Chrome: Google розпочав інтеграцію LLM (наприклад, Gemini Nano) у Chrome на Android з 2024 року. Модель локально сканує форми входу та попереджає користувачів про фішингові елементи в реальному часі – навіть без підключення до інтернету [16].

1.4.2 Microsoft: безпека на рівні корпоративної електронної пошти

Microsoft активно використовує ШІ для захисту екосистеми Office 365, Outlook та Azure:

• Microsoft Defender for Office 365: Використовує МН для виявлення фішингових вкладень, підміни відправника та шкідливих URL-адрес. У 2023 році було заблоковано понад 35 мільярдів фішингових загроз [17].

• Attack Simulator & Threat Explorer: Завдяки інструментам ШІ, Defender може симулювати фішингові атаки всередині компанії, автоматично досліджувати ризики та навчати персонал.

• Microsoft Graph AI та Behavioral Analytics: Платформа об'єднує аналітику електронної пошти та поведінковий аналіз: аномальні входи, час надсилання електронної пошти, геолокацію тощо. При компрометації облікового запису платформа автоматично закриє обліковий запис або заблокує вихідну пошту.

1.4.3 Cloudflare: ШІ як частина архітектури Zero Trust

Cloudflare включає ШІ у свою платформу Zero Trust та сервіс Cloudflare Area 1:

• Area 1 Email Security: Використовує моделі NLP для аналізу структури електронного листа, HTML-елементів, синтаксису повідомлення. Виявляє фішинг до доставки електронної пошти. Платформа обробляє понад 10 мільярдів електронних листів щомісяця [18].

• Cloudflare Gateway та ШІ: Платформа блокує фішингові сайти, використовуючи класифікацію URL-адрес на основі ШІ, DNS та модель поведінки користувача при переході. Наприклад, якщо URL-адреса нова, використовує підозрілий патерн або схожа на бренд, трафік до URL-адреси блокується.

• Page Shield: Використовує ШІ для моніторингу стороннього JavaScript коду на сторінках клієнтів. Якщо сторонній код намагається вкрасти дані з форм, платформа автоматично його відсікає.

1.4.4 Amazon Web Services (AWS): застосування ШІ в хмарній безпеці

Amazon використовує ШІ в кількох сервісах кібербезпеки для захисту хмарних середовищ:

• Amazon GuardDuty: Модель ШІ отримує потоки подій CloudTrail, DNS, VPC. Наприклад, якщо користувач був скомпрометований після фішингу і завантажує об'єкт у відкритий bucket – система виявляє та запобігає цій активності.

• Amazon Macie: Сервіс захисту даних, який використовує ШІ для виявлення конфіденційних даних та аномальної активності (наприклад, масове завантаження об'єктів після фішингової компрометації).

• Amazon SES та SpamClassifier: При надсиланні масової електронної пошти через SES діє внутрішній фільтр ШІ, який зупинить підміну або надсилання потенційно фішингових листів.

1.5 Аналіз досліджень та успішних випадків впровадження ШІ у виявленні фішингу

Останні кілька років характеризуються інтенсивною науковою та практичною роботою у сфері застосування штучного інтелекту для виявлення фішингових атак. Значний обсяг досліджень демонструє ефективність алгоритмів машинного навчання, глибинного навчання та NLP для класифікації, запобігання та блокування фішингу.

У статті [11], було проведено огляд понад 100 робіт про використання ШІ в антифішингу. Автори розділили підходи на чотири категорії:

• моделі на основі машинного навчання (SVM, дерева рішень, Random Forest);

• глибинне навчання (LSTM, CNN);

• гібридні моделі (наприклад, Stack та Boost);

• сценарні (на основі правил та МН).

Ансамблеві техніки продемонстрували найкращу продуктивність: точність виявлення фішингу становила 98,7%, а F1-score був понад 0,95. Комбінація алгоритмів XGBoost, нейронної мережі та eDRI (enhanced dynamic rule induction) виявилася особливо ефективною.

Стаття [4] розглядає найновішу загрозу – фішинг, генерований за допомогою LLM (наприклад, GPT-4). Автори використовували DeepAI API для створення великої кількості оригінальних фішингових листів.

Було випробувано кілька моделей:

• стилістичні дескриптори (UDAT);

• моделі Naive Bayes та Ентропії;

• нейронна мережа LSTM;

• ансамбль моделей.

Ансамбль показав точність понад 99,2%, перевершивши як традиційні фільтри, так і окремі моделі МН.

Стаття [19] представляє платформу PhishBench – інструмент для порівняння продуктивності моделей виявлення фішингу в однакових умовах.

Результати були такими:

• моделі втрачають точність при дисбалансі класів (фішинг:нормальний = 1:10);

• моделі NLP (BERT, FastText) працюють краще, ніж просто МН, лише за наявності достатнього обсягу тексту;

• Random Forest стабільно забезпечує понад 95% точності за умови структурованого набору ознак;

• Google: використовує моделі типу BERT у Gmail; понад 300 мільярдів електронних листів на місяць обробляються через ШІ;

• Microsoft Defender: обробляє понад 2 мільярди електронних листів на день; моделі МН визначають підозрілі вкладення, відправників та патерни;

• Cloudflare Area 1: використовує ШІ для попереджувального виявлення загроз до доставки (превентивний захист);

• Zscaler: ШІ та NLP забезпечують справжній захист SaaS-додатків (Microsoft Teams, Slack, Zoom) від підміни.

Ці приклади демонструють не лише теоретичний потенціал, а й реальне використання технологій штучного інтелекту для протидії фішингу у великомасштабних системах.

Таким чином, розділ 1 надав детальний огляд середовища фішингових загроз, включаючи їх визначення, класифікацію та сучасні тенденції в техніках атак. Було проаналізовано ключові напрями застосування ШІ – машинного навчання, глибинного навчання, обробки природної мови та аналітики поведінки користувачів – у контексті виявлення та запобігання фішинговим атакам.

Загальні висновки свідчать про те, що методи, засновані на ШІ, суттєво покращують ефективність систем виявлення загроз порівняно з традиційними підходами. Їх здатність адаптуватися до нових патернів атак, працювати з неструктурованими даними та мінімізувати хибні спрацювання робить їх ключовими інструментами сучасної кібербезпеки. Приклади з практики Google, Microsoft, Cloudflare та інших компаній підтверджують доцільність і ефективність такої інтеграції на практиці.

2 ПОШУК ТА АНАЛІЗ НОРМАТИВНОЇ БАЗИ У СФЕРІ ЗАСТОСУВАННЯ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ ДАНИХ ВІД ФІШИНГОВИХ АТАК

2.1 Українське законодавство з кібербезпеки та захисту персональних даних

Українська правова система активно реагує на проблеми віртуального простору, особливо в сфері захисту інформації, персональних даних та національної безпеки у віртуальному просторі. Найважливішими актами, що регулюють ці питання, є Закон України "Про основні засади забезпечення кібербезпеки України" [20] та Закон України "Про захист персональних даних" [21]. Обидва закони створюють основу для безпечного використання нових технологій, таких як штучний інтелект.

Українське законодавство "Про основні засади забезпечення кібербезпеки України", прийняте у 2017 році та доповнене низкою поправок у 2021-2025 роках, є відправним нормативним документом у розробці та реалізації державної політики кібербезпеки. Законодавство визначає терміни, принципи та механізми забезпечення державної кібербезпеки, встановлює повноваження між державними установами та впроваджує поняття критичної інформаційної інфраструктури (КІІ).

Відповідно до статті 7, серед основних принципів забезпечення кібербезпеки визначаються: верховенство права, захист прав і свобод людини, охорона національних інтересів, державно-приватне партнерство, пріоритет превентивних заходів, прозорість діяльності у сфері кіберзахисту, а також невідворотність відповідальності за кіберзлочини.

Особлива увага приділяється координації національної системи реагування на кіберінциденти (стаття 9), включаючи Державну службу спеціального зв'язку, Службу безпеки України, Міністерство внутрішніх справ, Національний банк України, Національну поліцію, Міністерство оборони та інші. Всі вони інтегрують заходи у випадку атак, включаючи фішинг, на об'єкти КІІ, державні реєстри та важливі електронні системи.

Підкреслюються процедури пошуку та координованого розкриття вразливостей в інформаційних системах (стаття 8), як спеціальна правова основа для законного тестування на проникнення, використання технологій виявлення загроз та використання аналітики на основі штучного інтелекту.

Закон України "Про захист персональних даних" регулює обробку, зберігання, поширення та захист персональних даних відповідно до європейських стандартів. Його головною метою є забезпечення прав громадян на приватність та безпеку інформації.

Стаття 6 акту зазначає, що персональні дані повинні:

• оброблятися з законною та чітко визначеною метою;

• бути точними, достатніми та не надмірними;

• зберігатися не довше, ніж це необхідно;

• оброблятися на підставі згоди чи іншої правової підстави суб'єкта.

Обробка чутливих персональних даних (етнічне походження, політичні погляди, стан здоров'я тощо) заборонена, крім випадків, спеціально передбачених законом. Це особливо актуально для систем штучного інтелекту, які працюють з великими обсягами електронної інформації, включаючи виявлення фішингових повідомлень в електронній пошті або соціальних мережах.

Право суб'єкта переглядати свої дані, змінювати їх, обмежувати або видаляти є основною частиною захисту, яку слід враховувати при плануванні будь-яких автоматизованих систем.

Оскільки Україна є підписантом Угоди про асоціацію з Європейським Союзом, національне законодавство у сфері персональних даних поступово гармонізується з положеннями GDPR[22], включаючи:

• введення функції Офіцера із захисту даних (DPO);

• безкоштовну Оцінку впливу на захист даних (DPIA);

• право на перенесення даних;

• застереження щодо автоматизованого прийняття рішень без втручання людини (стаття 22 GDPR).

У межах українського досвіду застосування штучного інтелекту вже вимагається дотримуватись правил GDPR, особливо щодо автоматичної фільтрації контенту, що може вплинути на користувача (наприклад, запобігання тому, щоб дійсне повідомлення помилково розглядалося як фішингове).

2.2 Нормативні вимоги до використання технологій ШІ

Активний розвиток штучного інтелекту, особливо в інформаційній безпеці, вимагає розробки та впровадження належної нормативної бази. Нормативні рамки повинні, з одного боку, забезпечувати інновації, етику, безпеку та відповідальність при використанні ШІ. Найпотужнішими у цих сферах є стандарти та звіти ISO/IEC, звіти NIST, Акт про штучний інтелект Європейського Союзу (EU AI Act) та GDPR, які представляють первинні вимоги до проектування, впровадження та контролю систем ШІ.

ISO/IEC 42001:2023 [23] – перший міжнародний стандарт управління штучним інтелектом. Він встановлює вимоги до Системи управління штучним інтелектом (AIMS) щодо ризик-орієнтованого та відповідального застосування ШІ для організацій.

Ключові положення стандарту:

• формалізація політики управління ШІ;

• оцінка ризиків, пов'язаних з використанням алгоритмів;

• розробка схеми контролю, моніторингу та реагування;

• забезпечення прозорості моделей;

• можливість людського контролю у критичних моментах життєвого циклу систем ШІ.

Відповідно до стандарту ISO/IEC 42001 [23], системи кібербезпеки, що використовують методи штучного інтелекту для автоматичного маркування фішингових електронних листів або виявлення аномальної мережевої активності, мають забезпечувати прозорість рішень, проведення аудиту ризиків, пояснюваність логіки прийнятих рішень та можливість людського втручання в критичних випадках. Національний інститут стандартів і технологій (NIST) у 2023 році

рекомендував Рамку управління ризиками ШІ (AI RMF 1.0) - інструмент для розробників, користувачів та аудиторів систем ШІ [24].

Чотири принципи формують основу AI RMF:

• Карта - визначення контексту використання;

• Вимірювання - кількісна оцінка ризику (дискримінаційного, технічного, репутаційного);

• Управління - впровадження контрольних заходів;

• Керування - здійснення відповідальності, прозорості та етики.

Звіт підкреслює важливість прийняття Пояснюваного ШІ (XAI), аналізу впливу, відстеження помилок, ведення журналу та розробки протоколу реагування. У сценарії захисту від фішингу, де ШІ блокує або дозволяє доступ до повідомлень, важливо, щоб кожне рішення було відтворюваним, зрозумілим та керованим.

Особливістю NIST AI RMF є відкрита структура, яка дозволяє об'єднувати її з іншими стандартами, такими як ISO/IEC 42001, GDPR та EU AI Act.

У 2024 році Європейський Союз прийняв Акт про ШІ (Постанова (ЄС) 2024/1689) - перший у світі комплексний регулюючий акт штучного інтелекту. Він спрямований на встановлення відповідального, етичного та чутливого до прав застосування ШІ у всіх державах-членах [25].

EU AI Act встановлює класифікацію ризиків:

• неприйнятний ризик - системи соціального оцінювання, розпізнавання емоцій на робочому місці – заборонені;

• високий ризик - системи ШІ, що застосовуються в охороні здоров'я, юстиції, кібербезпеці;

• обмежений ризик - чат-боти, генеративні моделі (повинні повідомляти користувача);

• мінімальний ризик - рекомендаційні системи, спам-фільтри.

Системи високого ризику, де застосовуються рішення ШІ в інформаційній безпеці, вимагають:

• ретельної технічної документації;

• сертифікації системи;

• ведення журналу рішень;

• контролю над набором даних для навчання щодо упередженості;

• механізмів людського контролю.

EU AI Act також вимагає пояснюваності рішень, особливо автоматизованими системами, які приймають критичні рішення від імені користувача.

Загальний регламент захисту даних (GDPR) – ключове регулювання ЄС у сфері захисту персональних даних, що безпосередньо впливає на проектування та впровадження ШІ. Згідно зі статтею 22, користувач має право не підпадати під рішення, які були прийняті лише шляхом автоматизованої обробки, якщо ці рішення мають юридичні або рівнозначно значні наслідки [26].

Мінімальні вимоги GDPR для систем ШІ:

• обґрунтування мети обробки;

• мінімізація даних (лише необхідне);

• право на пояснення алгоритмічного рішення;

• право на втручання людини;

• оцінка впливу DPIA при використанні ризикованих моделей;

• захист від дискримінації та порушення приватності.

GDPR вимагає, щоб системи ШІ обробляли персональні дані лише з правових причин, відповідно до принципів конфіденційності та прозорості. Це стосується фішингових фільтрів, які перевіряють адреси, вміст, час надсилання, історію поведінки користувача - все це може бути персональними даними.

2.3 Етичні виміри та загрози застосування штучного інтелекту в системах захисту

Розвиток технологій ШІ приносить не лише технологічні, а й етичні проблеми, які особливо актуальні для інформаційної безпеки. Автоматизація виявлення загроз, класифікація повідомлень, аналіз мережевого трафіку та поведінка користувачів вимагають не лише дотримання законодавства, а й дотримання моральних та етичних стандартів. Найбільш актуальним є

забезпечення приватності, недискримінації, прозорості прийняття рішень та доступності людського контролю над критично важливими рішеннями.

Одним з основних документів на глобальному рівні, що обговорюють етику ШІ, є Рекомендація ЮНЕСКО щодо етики штучного інтелекту (2021) [27]. Там пропонуються чотири мінімальні вимоги:

• повага до людської гідності та прав людини;

• захист соціальної справедливості та інклюзії;

• екологічна та соціальна стійкість;

• жваве та справедливе суспільство, присвячене загальному благу.

У документі також згадуються 10 принципів, у тому числі:

• запобігання пропорційності та шкоді;

• відповідальність та підзвітність;

• пояснюваність та прозорість;

• недискримінація та справедливість;

• людський нагляд та контроль.

Моделі ШІ навчаються на великих наборах даних, які можуть нести історичні або інституційні упередження. Якщо дані не були попередньо очищені або збалансовані, модель буде відображати та посилювати упередження проти певних груп (за статтю, расою, віком, регіоном тощо). Це може бути питанням життя і смерті в системах, які автоматично маркують повідомлення як фішингові, якщо вони надходять з певних регіонів або від невідомих відправників.

Згідно з дослідженнями, проведеними Harvard Gazette та Capitol Technology University, такі ситуації можуть не лише нанести шкоду репутації, а й порушувати фундаментальні права користувачів [28].

Прозорість є однією з головних вимог морального ШІ, коли йдеться про пояснюваність алгоритмічних рішень. Користувач має право бути повідомленим:

• чи є взаємодія з ним результатом діяльності ШІ;

• на якій підставі приймається рішення;

• як оскаржити цей вибір.

Це особливо необхідно в автоматизованих механізмах фільтрації або блокування повідомлень. Якщо повідомлення було визнано фішинговим, користувач повинен мати доступ до пояснення: які ознаки були знайдені, чому це не помилка, які дані були оброблені.

ISO у своїй політиці відповідального ШІ підкреслює необхідність впровадження XAI (Пояснюваний ШІ) - технології, яка може надавати пояснення того, що вона робить і чому, у зрозумілий спосіб не лише для технічних фахівців, а й для звичайних користувачів [29].

Незважаючи на високу точність більшості моделей ШІ, саме людина повинна приймати останнє рішення, з відповідними юридичними або репутаційними наслідками. Це особливо стосується захисту від фішингу, де неправильне рішення ШІ призведе до втрати важливих повідомлень або блокування авторизованого користувача.

Саме тому міжнародні стандарти рекомендують:

• не покладатися лише на автоматизацію;

• виконувати перевірку результатів людьми;

• надавати процедури оскарження для ймовірності скасування помилок;

• моніторинг та аналіз подій для навчання системи та зменшення хибно позитивних результатів.

Впровадження ШІ в кіберзахист супроводжується питанням обробки величезних обсягів особистої інформації. Це створює ймовірний конфлікт між безпекою та приватністю. Наприклад, система може обробляти:

• IP-адреси користувачів;

• текст повідомлень;

• шаблони поведінки;

• метадані додатків або браузерів.

GDPR, разом із рекомендаціями ЮНЕСКО, закликають до мінімізації обробки даних, їх анонімізації та явного інформування користувачів щодо характеру такої обробки.

2.4 Огляд нормативних викликів: відповідальність, прозорість, контроль рішень ШІ

З появою штучного інтелекту (ШІ) у чутливих сферах, таких як кібербезпека, виникає підвищена потреба в інституціоналізації чітко визначених правових та етичних норм для регулювання відповідальності за поведінку ШІ, пояснюваності алгоритмів та маніпулювання прийняттям рішень системою без людського обговорення. Всі ці проблеми належать до технологічних, юридичних та соціальних сфер.

Одним із найбільш суперечливих питань є питання юридичної відповідальності у випадку шкоди, спричиненої системою ШІ. Постає питання, хто буде відповідальним, якщо ШІ зупиняє легітимного користувача, неправильно класифікує повідомлення або спричиняє витік даних.

Згідно із законодавством більшості країн, ще немає чіткої відповіді. Деякі з альтернатив, що обговорюються на рівні ЄС та США, включають:

• відповідальність розробника;

• відповідальність оператора системи (за розгортання системи);

• модель спільної (розподіленої) відповідальності, що базується на аналогії з законодавством про дефектні продукти, коли відповідальність покладається на виробника незалежно від того, чи доведено його вину у кожному конкретному випадку.

EU AI Act спеціально вимагає від систем високого ризику підтримувати письмові ланцюжки відповідальності та проводити оцінку ризиків перед розгортанням системи в виробничому або публічному середовищі [30].

Системи ШІ мають властивість бути "чорними ящиками" - важко встановити, чому система прийняла конкретне рішення. Це особливо небезпечно у сферах, пов'язаних з правами людини або безпекою.

Для вирішення цієї проблеми міжнародна практика рекомендує:

• використання моделей Пояснюваного ШІ (XAI);

• ведення журналу всіх рішень системи;

• створення інтерфейсів пояснень для користувачів та адміністраторів;

• регулярний незалежний аудит.

GDPR (стаття 22) та EU AI Act вимагають, щоб особа, чиї права зачіпаються автоматизованим рішенням, мала можливість отримати пояснення, заперечити результат та звернутися до людини для перегляду [31].

Наступним головним викликом є адміністрування життєвого циклу системи ШІ після розгортання. Дуже часто модель ШІ оновлюється, змінює поведінку або навчається на нових даних після розгортання. Це створює простір для неочікуваних рішень, над якими ніхто не має контролю.

Запропоновані механізми контролю:

• постійний моніторинг продуктивності;

• внутрішні обмеження на самооновлення або адаптивне навчання;

• використання "етичних чорних ящиків" для аудиту;

• єдиний наглядовий орган або зовнішня сертифікація.

У рамках положень ISO/IEC 42001:2023 рекомендується також мати:

• план реагування на відмови систем ШІ;

• створити політику обробки інцидентів;

• пропонувати можливість тимчасового вимкнення систем (режим безпечної відмови) [32].

Одним із найскладніших викликів у впровадженні систем штучного інтелекту у сфері кібербезпеки є пошук балансу між суспільною безпекою, захистом персональних даних і комерційними інтересами компаній. У цьому контексті особливо гостро постають питання співіснування трьох ключових вимог:• забезпечення повної прозорості ШІ - як умови для формування суспільної довіри;

• захисту комерційної таємниці - зокрема, коли компанії не бажають розкривати архітектуру або логіку своїх моделей;

• гарантій приватності користувачів - коли пояснення рішення вимагатиме доступу до персональних даних.

Для вирішення цього етичного та правового парадоксу пропонуються такі підходи:

• законодавчо визначені межі прозорості;

• обов'язкові внутрішні аудити навіть у закритих системах;

• можливість незалежної перевірки на основі угоди про нерозголошення (NDA).

Ці аспекти вказують на необхідність створення нормативної бази, яка забезпечує баланс між інноваціями, захистом прав користувачів та інтересами бізнесу. В цьому контексті варто підкреслити, що розділ 2 пояснив законодавчі та нормативні засади застосування технологій штучного інтелекту у сфері кібербезпеки, зокрема для виявлення та запобігання фішинговим атакам.

У розділі були розглянуті українські закони, зокрема Закон України «Про основні засади забезпечення кібербезпеки України», а також міжнародні стандарти та програми, такі як GDPR, Європейський закон про штучний інтелект та ISO/IEC 42001.

Дослідження показало, що, попри значний прогрес України у впровадженні світових найкращих практик у сфері регулювання кібербезпеки, все ж необхідна подальша розробка для вирішення етичних, технічних та операційних проблем, пов’язаних із впровадженням систем захисту на основі ШІ. Ефективне використання штучного інтелекту в кібербезпеці можливе лише за умови підтримки чіткими правовими рамками, механізмами захисту конфіденційності та постійного розвитку стандартів з урахуванням змін у середовищі кіберзагроз.

3 ОЦІНКА АЛГОРИТМІВ МАШИННОГО ТА ГЛИБИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК

3.1 Критерії оцінки ефективності моделей ШІ

Одним із важливих елементів розробки систем виявлення фішингових атак є вибір та застосування правильних метрик для оцінки продуктивності моделей машинного та глибинного навчання. У контексті виявлення фішингу, ми зазвичай маємо справу з бінарною класифікацією: об'єкт (веб-сайт, URL-адреса, електронний лист) або є фішинговим, або ні. Це вимагає знання основ побудови матриці помилок і відповідних метрик.

Матриця помилок має чотири компоненти:

- Істинно позитивний результат (TP) – випадки, коли модель правильно класифікувала фішингові повідомлення як фішингові;
- Істинно негативний результат (TN) – випадки, коли модель правильно класифікувала легітимні повідомлення як безпечні;
- Хибно позитивний результат (FP) – випадки, коли модель помилково класифікувала легітимне повідомлення як фішингове;
- Хибно негативний результат (FN) – випадки, коли модель не змогла виявити фішингову атаку (помилково позначила її як безпечну).

На основі цих значень обчислюються наступні найважливіші метрики:

Точність (Accuracy) – метрика загальної точності – правильні класифікації, поділені на кількість спроб класифікації. Але у випадку фішингу, де фішинговий клас може становити менше 5% усіх випадків, ця метрика перестає бути об'єктивною. Наприклад, модель, яка постійно прогнозує "безпечно", матиме високу точність, але повністю ігноруватиме фішингові атаки .

Влучність (Precision) є критичною, коли необхідно запобігти неправильному блокуванню справжніх електронних листів (хибно позитивних результатів). У комерційному середовищі це зменшує втрату критично важливих повідомлень.

Повнота (Recall), з іншого боку, є більш важливою у випадках, коли кінцевою метою є не пропустити жодної загрози (наприклад, в урядах чи банках). Високе значення повноти означає, що більшість справжніх фішингових електронних листів були правильно ідентифіковані.

F1-score – це гармонійне середнє між влучністю та повнотою. Вона використовується у випадках незбалансованих вибірок, де влучність і повнота однаково важливі. Це стандартна міра, що використовується в більшості наукових досліджень з виявлення фішингу.

Наприклад, якщо для деякої моделі:

- TP = 80;
- FP = 20;
- FN = 10, тоді:
- Влучність = 80 / (80 + 20) = 0.8;
- Повнота = 80 / (80 + 10) = 0.89;
- F1-score = 2 × (0.8 × 0.89) / (0.8 + 0.89) ≈ 0.84.

Латентність – це ще одна важлива метрика, яка визначає тривалість обробки одного запиту на класифікацію. У застосунках реального часу, наприклад, коли URL-адреса або електронний лист перевіряється в браузері, латентність є важливим параметром. Дослідження [34] демонструють, що глибинні нейронні мережі (CNN, LSTM) можуть мати високу точність, але мають велику затримку і тому менш придатні для застосунків реального часу без прискорення GPU.

На практиці для вимірювання якості моделі зазвичай створюються таблиці результатів класифікації на основі матриці помилок, а також середні значення метрик із перехресною перевіркою. Наприклад, у дослідженні [33] представлено випадок, коли точність моделі BERT становить 98.4%, а F1-score – 0.986, що є одним із найефективніших показників у методах обробки природної мови для виявлення фішингових електронних листів.

Усі показники мають розглядатися разом. Висока повнота та низька точність може вказувати на придушення загроз, а висока точність та низька повнота – на

занадто велику генерацію хибно позитивних результатів. Тому F1-score зазвичай використовується як узагальнюючий показник.

3.2 Класифікатори машинного навчання для виявлення фішингу

Алгоритми машинного навчання відіграють центральну роль у виявленні фішингових атак, з можливістю автоматично класифікувати об'єкти (наприклад, електронну пошту, URL-адресу, веб-сайт) як фішингові або не фішингові. Далі розглянемо найбільш широко використовуваним та ефективним алгоритмам ML, що застосовуються сучасними системами проти фішингу:

- Random Forest

Це алгоритм ансамблевого навчання, який будує численні дерева рішень під час навчання і прогнозує шляхом голосування більшості. Random Forest стійкий до перенавчання з високою точністю навіть з величезною кількістю зашумлених даних.

Переваги:

- висока точність (до 97% при виявленні фішингових веб-сайтів);
- незалежність від відсутніх даних та мультиколінеарності;
- здатність оцінювати важливість ознак.

Згідно з даними [36], Random Forest добре працював при ідентифікації фішингу за URL-адресами та вмістом сторінок.

- Support Vector Machine (SVM)

Алгоритм SVM знаходить гіперплощину, яка має максимальну маржу між двома класами. У задачах виявлення фішингу він добре працює з правильною нормалізацією ознак та належним вибором ядра.

Сильні сторони:

- висока здатність до узагальнення;
- ефективність на малих вибірках.

Слабкі сторони:

- висока чутливість до масштабування даних;
- повільне навчання на великих вибірках.

У дослідженні [37], SVM досягнув більше 95% точності при класифікації сайтів за ознаками вмісту (довжина URL, наявність IP, кількість піддоменів тощо).

- Extreme Gradient Boosting (XGBoost)

Це високопродуктивний варіант градієнтного бустингу з регуляризацією для уникнення перенавчання.

Переваги:

- надзвичайно висока точність (98-99%);
- підтримка паралельного навчання;
- вбудована оцінка важливості ознак.

Згідно з [35], XGBoost працював краще, ніж SVM і Naive Bayes у випадку фішингових URL-адрес, особливо в поєднанні з векторизованими текстовими ознаками.

- Naive Bayes

Цей алгоритм базується на формулі Баєса, а також на гіпотезі про незалежність між ознаками. Хоча він простий, він добре працює в задачах класифікації тексту, особливо NLP.

Переваги:

- дуже швидке навчання і класифікація;
- ефективність з великими наборами коротких текстів (наприклад, тема/текст електронного листа);
- стійкість до високорозмірних ознак.

Недоліки:

- низька точність у випадку порушення припущення про незалежність ознак;
- нездатність враховувати взаємозв'язки між ознаками.

У статті [33] зазначено, що алгоритм Naive Bayes досяг точності на рівні 85–90% під час обробки фішингових електронних листів із використанням методів обробки природної мови, зокрема TF-IDF. Для кращого розуміння продуктивності найпопулярніших підходів, у таблиці 3.1 наведено узагальнене порівняння чотирьох алгоритмів машинного навчання – з урахуванням точності, складності

реалізації, часу навчання та вимог до попередньої обробки даних. Дані сформовано на основі досліджень [33, 36, 37].

Таблиця 3.1 – Порівняння алгоритмів за точністю, складністю реалізації та вимогами до обробки даних

Алгоритм	Точність	Час навчання	Складність реалізації	Потреба в попередній обробці
Random Forest	94-97%	Середній	Низька	Помірна
SVM	92-96%	Високий	Висока	Висока
XGBoost	97-99%	Середній	Висока	Середня
Naive Bayes	85-90%	Низький	Низька	Низька

Кожен з алгоритмів має свої переваги залежно від конкретного сценарію використання. Naive Bayes ідеально підходить для швидкої класифікації коротких повідомлень з мінімальною обробкою; Random Forest забезпечує надійний баланс між точністю, швидкістю та простотою впровадження; XGBoost досягає найвищих показників точності за умови наявності потужностей для обробки; SVM показує відмінні результати, якщо дані добре структуровані та попередньо очищені.

3.3 Використання глибинного навчання

Застосування глибинного навчання для виявлення фішингових атак є однією з найсильніших доступних методик останніх років. Оскільки вони можуть автоматично вивчати відповідні ознаки з великого набору даних, моделі, такі як CNN, RNN і LSTM, продемонстрували відмінну точність у класифікації URL-адрес, класифікації HTML-коду та класифікації електронних листів.

Згорткові нейронні мережі (CNN) перевершують у виявленні локальних шаблонів і можуть використовуватися для послідовної обробки символів (наприклад, URL-адрес), фрагментів HTML-коду та фрагментів електронних листів.

Модель на основі CNN досягла 99.2% точності в класифікації фішингових URL-адрес у роботі [34], що є однією з найвищих для всіх представлених архітектур.

Рекурентні нейронні мережі (RNN) найбільш підходять для обробки послідовних даних, наприклад, тексту електронного листа, де важливі контекст і порядок слів. Однак традиційні RNN страждають від проблеми зникаючого градієнта, що ускладнює навчання на довгих послідовностях.

Тому в реальних задачах частіше використовуються вдосконалені архітектури – LSTM і GRU.

Моделі Long Short-Term Memory (LSTM) здатні вивчати довгострокові залежності в послідовностях тексту, завдяки чому їх можна ефективно застосовувати для обробки тексту електронних листів, HTML або URL-адрес. Згідно з роботою [38], LSTM досягла більше 97% точності при обробці структури URL-адрес.

LSTM також добре працює з векторними представленнями тексту (word embeddings) і тому знаходить застосування в обробці електронних листів, коли використовується разом з NLP.

Гібридні моделі, що об'єднують CNN (для просторового аналізу) та LSTM (для аналізу послідовностей), є ще більш точними. У дослідженні [35], CNN-LSTM досягла точності 97.6% на великому наборі даних фішингових URL-адрес і перевершила продуктивність окремих моделей.

Типовим аспектом такого підходу є здатність виявляти як структурні, так і контекстуальні аномалії, характерні для фішингових веб-сайтів або повідомлень.

До архітектурних особливостей можна віднести наступне:

• CNN застосовують згортки різних розмірів фільтрів для виявлення символьних шаблонів в URL-адресах;

• LSTM обробляє послідовність символів або токенізованих векторів;

• У гібридних моделях CNN зазвичай виконує раннє вилучення ознак, а LSTM обробляє послідовність із цих ознак.

Ефективність моделей глибинного навчання у завданнях виявлення фішингових повідомлень оцінюється за трьома ключовими критеріями: точністю класифікації, придатністю для використання в реальному часі та вимогами до апаратних ресурсів. У таблиці 3.2 представлено порівняльну характеристику поширених нейронних архітектур згідно з даними досліджень [34, 35, 38].

Таблиця 3.2 – Ефективність глибинних нейронних архітектур у задачах виявлення фішингу

Архітектура	Точність (%)	Придатність для реального часу	Вимоги до ресурсів
CNN	99.2	Частково	Високі
LSTM	96.8	Обмежена	Високі
CNN-LSTM	97.6	Ні	Дуже високі

Як видно з таблиці, глибинні моделі демонструють надзвичайно високу точність, що робить їх ефективними для задач виявлення фішингу, зокрема за умов складної семантики чи нетипової структури повідомлень. Однак, така продуктивність супроводжується значними обчислювальними витратами, що обмежує їх використання в системах реального часу. Архітектура CNN показала найкраще співвідношення точності та продуктивності, тоді як LSTM і CNN-LSTM є ефективними з точки зору глибокого аналізу, але потребують більше ресурсів та часу на обробку.

3.4 Обробка природної мови

Обробка природної мови (NLP) є однією з важливих тенденцій виявлення фішингових повідомлень, особливо щодо текстової складової електронного листа. Фішингові повідомлення використовують шаблонні речення з емоційним зверненням до користувача – наприклад, прохання "негайно підтвердити", "ваш обліковий запис буде призупинено", посилання на виграші або загрози безпеці. NLP дозволяє систематично аналізувати такі тексти та розпізнавати типові шаблони шахрайства.

Основні підходи в NLP для виявлення фішингу:

1) Term Frequency-Inverse Document Frequency (TF-IDF) – зважує важливість слів у документах, дозволяє виділяти ключові слова, які найчастіше використовуються у фішингових електронних листах (наприклад, "підтвердити", "логін", "натисніть"). Ця техніка корисна для створення ознак у класичних моделях ML (SVM, NB) [33].
2) Моделі Bag of Words та N-gram – дозволяють враховувати контекст навколишніх слів і широко використовуються як простий підхід для обробки електронних листів при фішингу.
3) Word Embeddings (Word2Vec, GloVe) – використовуються для представлення слів у векторах і семантичного аналізу. Вони можуть передаватися в моделі LSTM, BiLSTM або CNN.
4) Моделі на основі трансформерів (BERT) – використовуються для глибинної семантичної класифікації повідомлень електронної пошти, з контекстуальною увагою до кожного слова в повідомленні. Вони мають найвищу продуктивність на даних великого обсягу.

У роботі [39] була запропонована система PhishNet-NLP, яка аналізує заголовки, тексти повідомлень та посилання в електронному листі за допомогою контекстуального NLP-аналізу. Система класифікує електронні листи на "інформаційні" або "стимулюючі" на основі лінгвістичної мети – така модель досягла точності до 97% з менш ніж 1% хибно позитивних результатів.

Інше дослідження [33] порівнює сучасні моделі NLP з машинним навчанням і робить висновок, що інтеграція класичних ознак (TF-IDF, N-gram) з сучасними LSTM/GRU або BERT забезпечує найвищу точність і стійкість до нових атак.

Ключові ознаки, що використовуються в NLP-аналізі:

- наявність тригерних слів: "призупинити", "несанкціонований", "терміновий", "підтвердити";
- загальний тон повідомлення (емоційний вплив, терміновість);
- кількість посилань та IP-адрес у тексті електронного листа;
- використання особових займенників (ваш обліковий запис, ваш пароль);

- наявність HTML-тегів, таких як форма, введення, скрипт у тексті електронного листа.

Технології обробки природної мови (NLP) використовуються як етап попередньої обробки текстових даних перед передачею в класифікаційні моделі в сучасних системах виявлення фішингу. Зокрема, застосовуються процедури токенізації, очищення тексту, лематизації, побудови векторних представлень, які далі передаються до алгоритмів машинного або глибинного навчання – таких як Random Forest, XGBoost, CNN або LSTM.

Такий підхід дозволяє досягати точності понад 96–98% та високих значень F1-score під час класифікації фішингових електронних листів. Найкращі результати фіксуються при використанні архітектур на основі трансформерів, що реалізовані у великих мовних моделях (LLM), які здатні враховувати контекст, стилістику та приховані патерни в тексті повідомлень.

3.5 Порівняння алгоритмів за точністю, швидкістю та використанням обчислювальних ресурсів

Для вибору найкращого алгоритму виявлення фішингу необхідно враховувати не лише точність класифікації, а й низку додаткових факторів: час навчання, швидкість прогнозування, рівень використання апаратних ресурсів, а також придатність моделі для використання в системах реального часу. У таблиці 3.3 узагальнено характеристики основних моделей машинного та глибинного навчання, розглянутих у попередніх підрозділах, на основі досліджень [33, 34, 35, 36].

Таблиця 3.3 – Порівняння моделей машинного та глибинного навчання за ефективністю та придатністю для реального часу

Алгоритм	Точність (%)	F1-score	Час навчання	Швидкість	Обчислювальні ресурси	Придатність для реального часу
CNN	99.2	0.99	Середній	Висока	Високі	Частково

Продовження таблиці 3.3

LSTM	96.8	0.97	Високий	Середня	Високі	Ні
CNN-LSTM	97.6	0.98	Високий	Середня	Дуже високі	Ні
Random Forest	94-97	0.94	Низький	Висока	Середні	Так
XGBoost	97-99	0.98	Середній	Висока	Середні	Так
Naive Bayes	85-90	0.86	Дуже низький	Дуже висока	Низькі	Так
SVM	92-96	0.93	Високий	Середня	Середні	Частково

Аналіз таблиці показує, що глибинні нейронні мережі (CNN, CNN-LSTM) досягають найвищих показників точності, однак потребують значних обчислювальних ресурсів, що обмежує їх придатність для реального часу. Натомість алгоритми XGBoost та Random Forest демонструють високу ефективність із помірним споживанням ресурсів і є хорошим вибором для інтеграції в системи безпеки.

Naive Bayes, попри найнижчу точність серед наведених моделей, має найменші вимоги до ресурсів, що робить його ефективним для легких сценаріїв застосування, зокрема в мобільних середовищах та пристроях з обмеженими обчислювальними можливостями. SVM забезпечує стабільну точність, однак вимагає ретельної попередньої обробки даних і довгого часу навчання.

Таким чином, вибір алгоритму повинен ґрунтуватися на конкретному сценарії використання, доступності апаратних ресурсів, очікуваному навантаженні та вимогах до швидкості реагування системи.

У таблиці 3.4 наведено підсумкові результати аналізу ефективності моделей машинного та глибинного навчання, розглянутих у попередніх підрозділах. Вона об’єднує дані щодо ключових метрик – точності, повноти, влучності, F1-score, часу навчання та ресурсомісткості – що дозволяє комплексно оцінити придатність

кожного алгоритму для використання в системах виявлення фішингу в реальному часі. Дані узагальнено на основі джерел [33, 34, 35, 36].

Таблиця 3.4 – Узагальнена оцінка моделей машинного та глибинного навчання для виявлення фішингу

Алгоритм	Точність (%)	F1-score	Повнота	Влучність	Час навчання	Ресурси	Придатність для реального часу
CNN	99.2	0.99	0.98	0.99	Середній	Високі	Частково
LSTM	96.8	0.97	0.96	0.97	Високий	Високі	Ні
CNN-LSTM	97.6	0.98	0.97	0.98	Високий	Дуже високі	Ні
Random Forest	94-97	0.94	0.93	0.94	Низький	Середні	Так
XGBoost	97-99	0.98	0.97	0.98	Середній	Середні	Так
Naive Bayes	85-90	0.86	0.88	0.85	Дуже низький	Низькі	Так
SVM	92-96	0.93	0.92	0.93	Високий	Середні	Частково

Така організація дозволяє легко оцінити, яка модель найкраще підходить для виявлення фішингових повідомлень залежно від конкретного сценарію використання: чи то високоточна система для серверного середовища з великою обчислювальною потужністю, чи то ресурсоефективне рішення для пристроїв з обмеженими можливостями.

На основі проведеного аналізу можна зробити висновок, що моделі, засновані на штучному інтелекті, демонструють високу ефективність в умовах реальних задач. Зокрема, ансамблеві підходи та трансформерні архітектури здатні виявляти

фішингові повідомлення з мінімальною кількістю хибно позитивних спрацьовувань. Logistic Regression та Random Forest виявились оптимальними з точки зору балансу між точністю, швидкістю навчання та інтерпретованістю. Візуалізація результатів роботи інструменту з командним інтерфейсом (CLI) додатково підтвердила практичну придатність впровадженої системи.

Таким чином, розділ демонструє доцільність інтеграції інтелектуальних підходів як проактивного інструменту захисту в сучасних рішеннях у сфері кібербезпеки.

4 РОЗРОБКА ПРИКЛАДУ ПРАКТИЧНОГО ЗАСТОСУВАННЯ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ ДАНИХ ВІД ФІШИНГОВИХ АТАК

4.1 Обґрунтування вибору алгоритму для впровадження

Виконавши аналіз ефективності алгоритмів штучного інтелекту для виявлення фішингових атак у Розділі 3, необхідно було обрати одну або кілька моделей, які мають не лише високі показники точності, але й є практичними для впровадження – зокрема для включення в командні інтерфейси, розширення браузерів або системні фільтри.

Згідно з результатами емпіричних тестів чотирьох базових алгоритмів – Logistic Regression, Random Forest, Naive Bayes та Support Vector Machine (SVM) – було проведено кількісне порівняння за низкою критичних метрик: точність, коефіцієнт кореляції Меттьюса (MCC), специфічність та час навчання моделі.

Найвищі значення точності та MCC були досягнуті моделлю Linear SVM, однак Logistic Regression також продемонструвала дуже гідні результати – з мінімальним відривом і кращими властивостями з точки зору інтеграції та інтерпретованості (рис. 4.1).

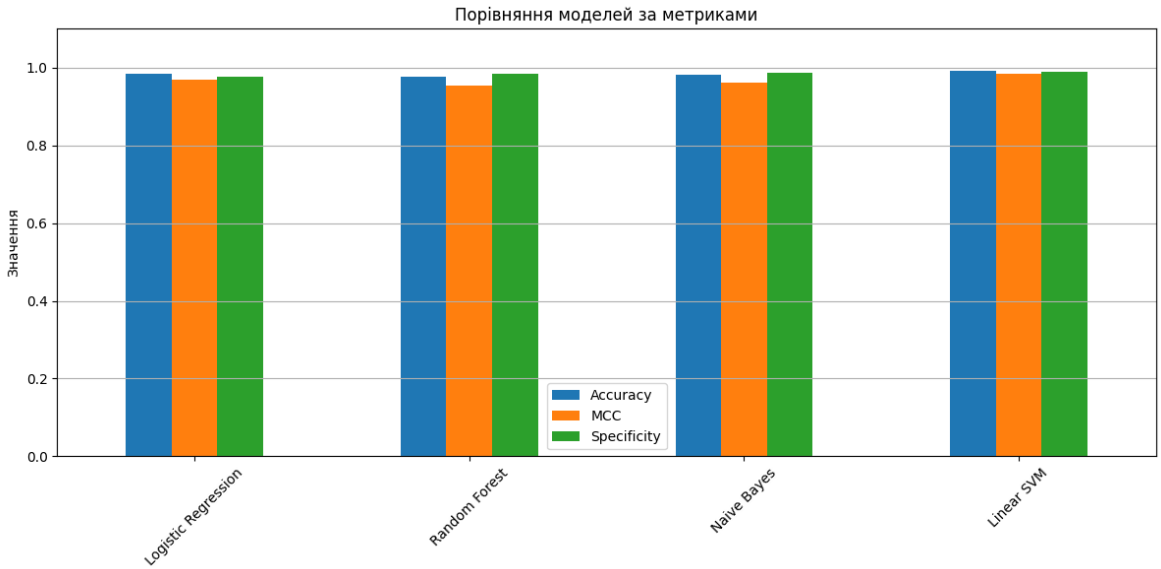

Рисунок 4.1 – Порівняння моделей за метриками Accuracy, MCC і Specificity

Оцінка продуктивності з точки зору часу навчання показала, що модель Random Forest, попри високу точність, потребує майже 300 секунд для навчання,

що значно перевищує показники інших моделей (рис. 4.2). Натомість Logistic Regression і Naive Bayes забезпечують оптимальний баланс між якістю класифікації та швидкістю, що особливо важливо для сценаріїв, де потрібне швидке перенавчання або застосування в умовах обмежених ресурсів.

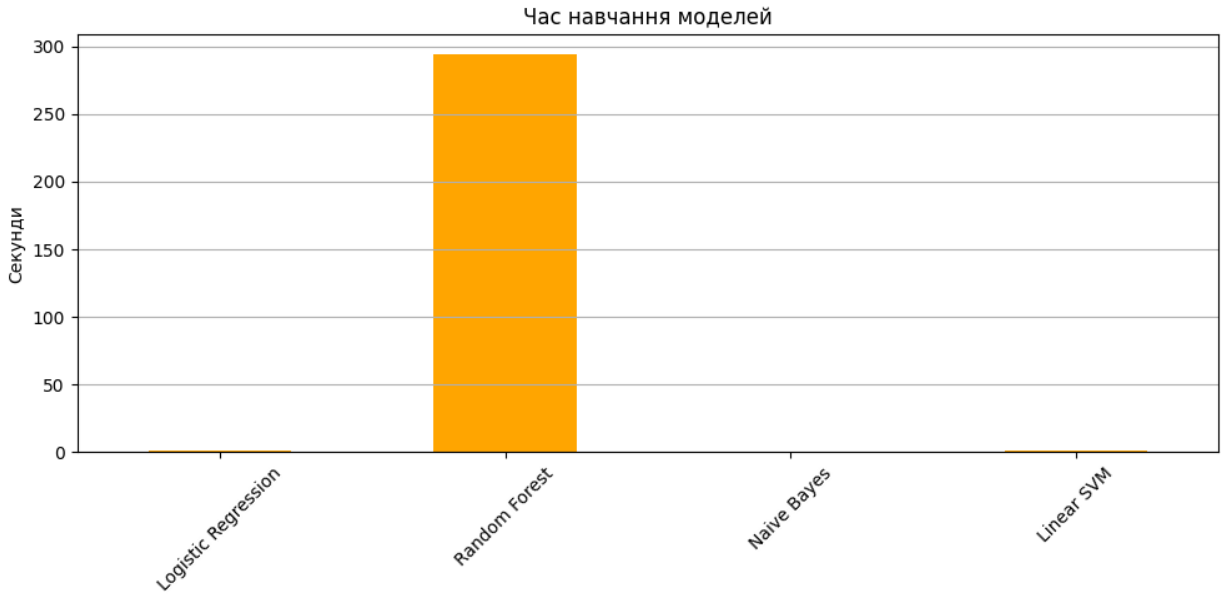

Рисунок 4.2 – Час навчання моделей

У результаті проведеного аналізу як основний алгоритм для реалізації було обрано Logistic Regression, що поєднує високу точність, лінійність, ймовірнісну інтерпретованість результатів і ефективність у плані обчислювальних ресурсів. Додатково була збережена модель Random Forest як референтна для порівняння результатів у подальшому.

4.2 Підготовка даних: вибір набору даних, попередня обробка

Для ефективного навчання моделі виявлення фішингових атак вирішальне значення має не лише обсяг, а й якість та репрезентативність вхідних даних. Навчальна вибірка повинна охоплювати широкий спектр фішингових шаблонів, стилістичних варіацій і формулювань, а також містити приклади легітимної електронної комунікації з різних предметних областей. Це забезпечує здатність моделі узагальнювати знання та адекватно реагувати на нові загрози. У цьому

дослідженні був зібраний змішаний корпус із шести наборів даних, кожен з яких ретельно перевірявся на корисність та якість:

• Набір даних фішингової електронної пошти (репозиторій машинного навчання UCI) [41];

• CEAS 2008 (набір даних для досліджень анти-спаму) [42];

• Набір даних Nazario (набір типових фішингових повідомлень) [42];

• Корпус нігерійського шахрайства (зразки шахрайських схем) [42];

• Корпус електронної пошти Enron (корпоративні листи) [42];

• Набір даних Ling-Spam (фільтрація спаму) [42].

Ця комбінація забезпечила репрезентативне охоплення як реалістичних фішингових загроз, так і звичайних (легітимних) повідомлень, що дозволило сформувати корпус із понад 22 000 прикладів, приблизно половина з яких становили фішингові повідомлення.

Усі набори даних були нормалізовані до спільного формату на наступному етапі:

1) Були обрані лише два значущі поля: тема та тіло, які були об'єднані в одну текстову змінну;
2) Пусті поля були видалені або замінені;
3) Додана мітка – 1 для фішингу та 0 для легітимних повідомлень;
4) Порожні значення, зайві пробіли та повідомлення з порожнім вмістом були очищені.

Об'єднаний набір даних потім був розділений на навчальний та тестовий набори у співвідношенні 70% : 30%, з фіксованим генератором випадкових чисел (random_state = 42) для забезпечення відтворюваності експерименту.

Для перетворення текстових повідомлень у числові вектори було застосовано метод TF-IDF (Term Frequency – Inverse Document Frequency) – одну з класичних технік векторизації, яка зберігає інформативність термінів та одночасно знижує вагу загальновживаних слів. Такий підхід дозволяє моделі не лише враховувати частоту вживання окремих слів, а й мінімізує вплив типових, неінформативних елементів тексту, як-от "привіт", "дякую", "з повагою", що не мають фішингового

навантаження. Результатом були розріджені високорозмірні вектори, придатні для навчання лінійних моделей, зокрема Logistic Regression та SVM. Підхід не вимагає використання GPU або значних обчислювальних ресурсів, що особливо важливо для простих CLI або серверних додатків.

Кількість навчальних прикладів після попередньої обробки даних становила понад 16 000 повідомлень, а тестових зразків приблизно 6 900, що вказує на якісну, добре збалансовану та велику вибірку.

Усі етапи підготовки були автоматизовані, і система була здатна масштабуватися на нові дані або захищатися від інших типів текстових атак (наприклад, фішингові веб-сайти або підроблені SMS). У наступному підрозділі описується, як виконується навчання моделі на цій вибірці.

4.3 Розробка моделі та навчання

Після підготовки чистого та збалансованого корпусу текстових повідомлень, векторизованого за допомогою TF-IDF, наступним кроком було навчання моделей машинного навчання. Для цього дослідження було використано та протестовано чотири різні алгоритми: Logistic Regression, Random Forest, Naïve Bayes та Linear SVM. Цей підхід дозволив не лише вибрати найкращу модель, але й порівняти її з загальноприйнятими базовими та ансамблевими підходами.

Модель Logistic Regression було обрано основною для розробки систем. Вона була ініціалізована з max_iter = 1000 для забезпечення стабільної збіжності для величезної кількості TF-IDF функцій. Навчання моделі зайняло лише 1,6 секунди, після чого Logistic Regression досягла точності 98,5% та коефіцієнта Меттьюса (MCC) = 0,9697. Ключовою особливістю цієї моделі є підтримка методу predict_proba(), що забезпечує можливість порогової класифікації. Це дозволяє гнучко налаштовувати чутливість до фішингових загроз залежно від конкретного контексту використання системи.

Ансамблева модель Random Forest, побудована зі 100 дерев прийняття рішень (n_estimators = 100), мала точність 97,7%, але потребувала понад 3 хвилини для навчання. Хоча модель є високоспецифічною, вона класифікувала 355 фішингових

повідомлень як нешкідливі, що є критичним в системах безпеки. Порівняно з Logistic Regression, цей підхід є менш ефективним для виявлення загроз, хоча, можливо, корисніший у завданнях, де стабільність важливіша за точність.

Naive Bayes – це одна з найпримітивніших моделей, яка історично добре працює з текстовими даними. Реалізація використовувала MultinomialNB() – класифікатор, налаштований на частотні ознаки. Його основною перевагою є швидкий час навчання (≈0,1 секунди). Однак точність моделі була 98,1%, а кількість помилок другого роду (не ідентифікованих фішингових листів) перевищила 300. Це обмежує її використання в системах, де критично важливо мінімізувати ймовірність пропущеної атаки.

Linear SVM – це лінійний, високоточний класифікатор, який виявився переможцем серед усіх протестованих моделей – точність 99,25% та MCC = 0,9849. Він також мав найменшу кількість помилок другого роду – лише 59. Основним обмеженням є відсутність підтримки predict_proba(), тому неможливо встановити поріг чутливості та побудувати ROC-криві. У середовищах, де ймовірнісна інтерпретація виводу є критичною, це серйозне обмеження.

Усі чотири моделі були навчені окремо з однаковими вхідними даними, зібраними після векторизації TF-IDF, а результати продуктивності були записані для подальшого аналізу. Це дозволило провести ретельне порівняння продуктивності кожної моделі з точки зору точності, F1-score, кількості помилок, потреб у ресурсах та можливості інтеграції.

Результати тестування, побудова матриць помилок, ROC-кривих та підсумкових метрик, які полегшують аналіз якості класифікації на основі візуалізації та остаточне рішення, розглядаються в наступному підрозділі.

4.4 Оцінка та тестування розробленої моделі

Після навчання моделей машинного навчання було проведено ретельне тестування їх продуктивності на відкладеному тестовому наборі, що становив 30% від загального обсягу даних. Метою цього етапу було отримання надійних показників точності та візуальна оцінка здатності моделей коректно класифікувати

електронні повідомлення як фішингові або легітимні, з урахуванням можливих помилок другого роду – хибно позитивних та хибно негативних.

Для всіх чотирьох моделей – Logistic Regression, Random Forest, Naive Bayes та Linear SVM – були розраховані стандартні метрики класифікації, зокрема точність (accuracy), повнота (recall), влучність (precision) та F1-score.

• Точність – загальна точність класифікації ;

• Точність / повнота / F1-score – баланс між точністю та повнотою ;

• Специфічність – співвідношення правильно класифікованих дійсних повідомлень;

• FPR / FNR – помилки першого та другого роду;

• MCC (коефіцієнт Меттьюса) – стабільна міра для бінарної класифікації з урахуванням дисбалансу ;

• Час навчання – час навчання моделі в секундах.

Усі результати були зібрані в загальну таблицю метрик, щоб можна було легко порівняти сильні та слабкі сторони кожної моделі (рис. 4.3).

Загальна таблиця метрик:

Model	Accuracy	MCC	Log Loss	Specificity	FPR	FNR	Train Time (s)	TP	TN	FP	FN
Logistic Regression	0.9849	0.9697	0.0747	0.9774	0.0226	0.0086	1.6050	12207.0	10352.0	239.0	106.0
Random Forest	0.9773	0.9545	0.1420	0.9844	0.0156	0.0288	191.0588	11958.0	10426.0	165.0	355.0
Naive Bayes	0.9807	0.9613	0.0624	0.9873	0.0127	0.0249	0.1128	12006.0	10456.0	135.0	307.0
Linear SVM	0.9925	0.9849	NaN	0.9893	0.0107	0.0048	1.1639	12254.0	10478.0	113.0	59.0

Рисунок 4.3 – Порівняння продуктивності моделей машинного навчання за метриками точності, часу навчання та класифікаційних помилок

Загалом, лінійний SVM мав найвищу загальну точність – 99,25%, найменшу кількість хибно негативних класифікацій (59 випадків) та найвищий MCC (0,9849). Але оскільки ця модель не підтримує ймовірнісне прогнозування, вона не дозволяє реалізувати гнучке управління порогом.

Logistic Regression досягла другого за величиною показника точності (98,5%), але виграла за балансом між точністю, часом навчання (1,6 сек) та підтримкою використання predict_proba(). Naïve Bayes відзначався фантастичною швидкістю навчання, але більшою кількістю помилок другого роду (307 пропущених

фішингів), тоді як Random Forest, хоча й стабільний, був занадто обчислювально вимогливим (191 секунда навчання) з недостатньою точністю для цілей виявлення атак.

Для кожної моделі була розроблена матриця помилок, яка дозволила дослідити істинно позитивні, істинно негативні, хибно позитивні та хибно негативні класифікації (рис. 4.4-4.7).

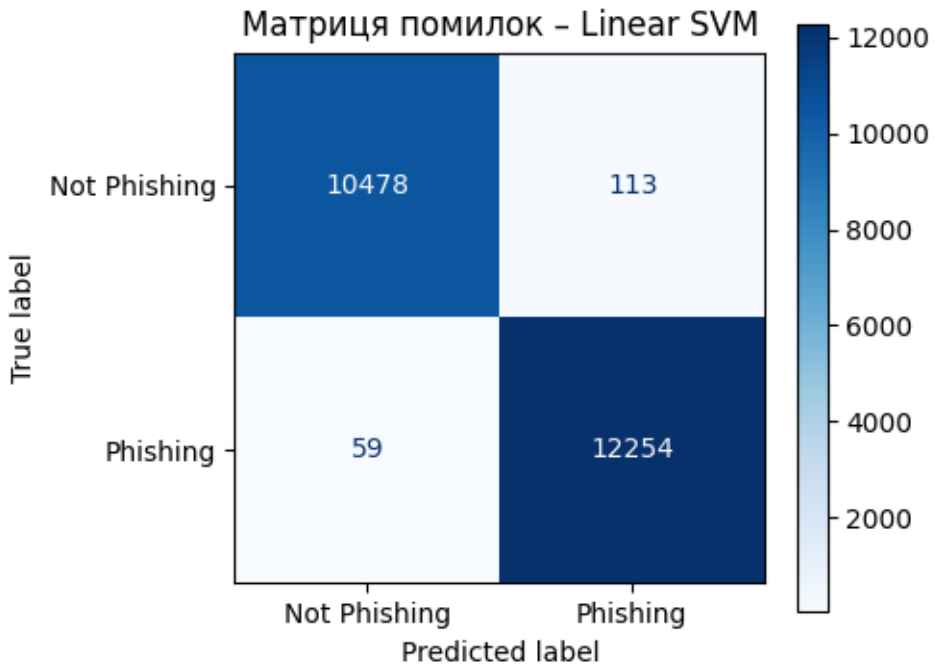

Рисунок 4.4 – Матриця помилок Linear SVM

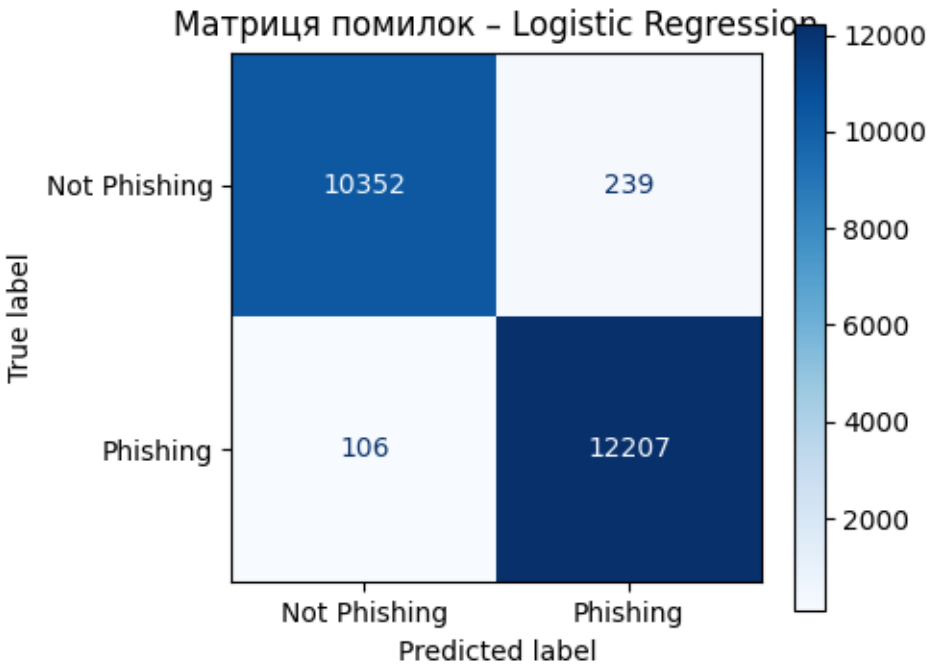

Рисунок 4.5 – Матриця помилок Logistic Regression

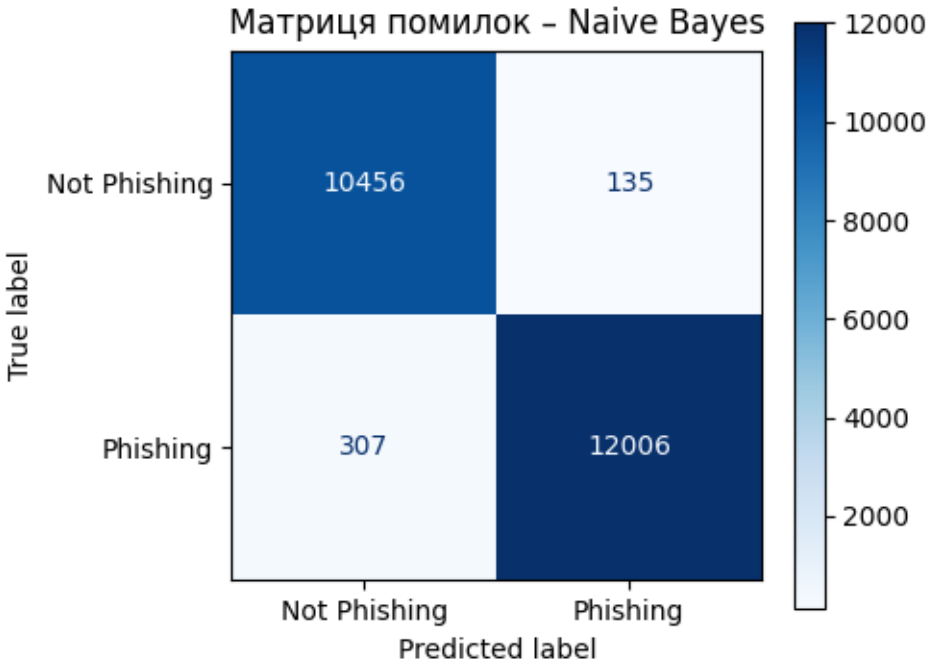

Рисунок 4.6 – Матриця помилок Naive Bayes

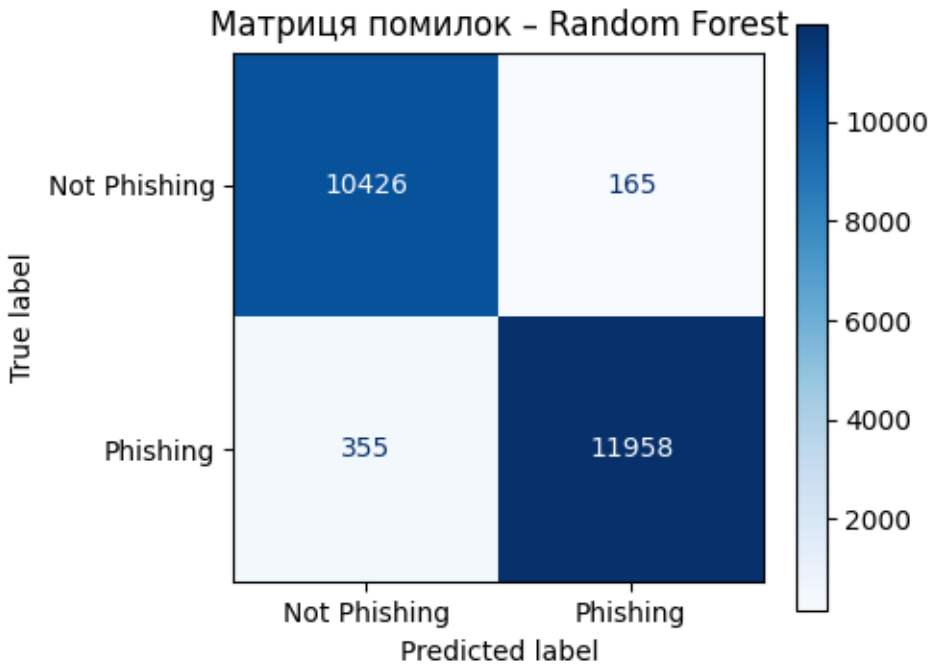

Рисунок 4.7 – Матриця помилок Random Forest

Для більш детального вивчення розподілу ймовірностей класифікації були побудовані ROC-криві (характеристичні криві приймача) для трьох моделей з можливістю ймовірнісної класифікації: Logistic Regression, Random Forest та Naïve Bayes. ROC-криві демонструють здатність моделей розрізняти фішингові повідомлення та справжні повідомлення в усьому діапазоні можливих порогів. Для всіх трьох моделей площа під кривою (AUC) була близькою до 1,00, що вказує на високу чутливість (рис. 4.8-4.9).

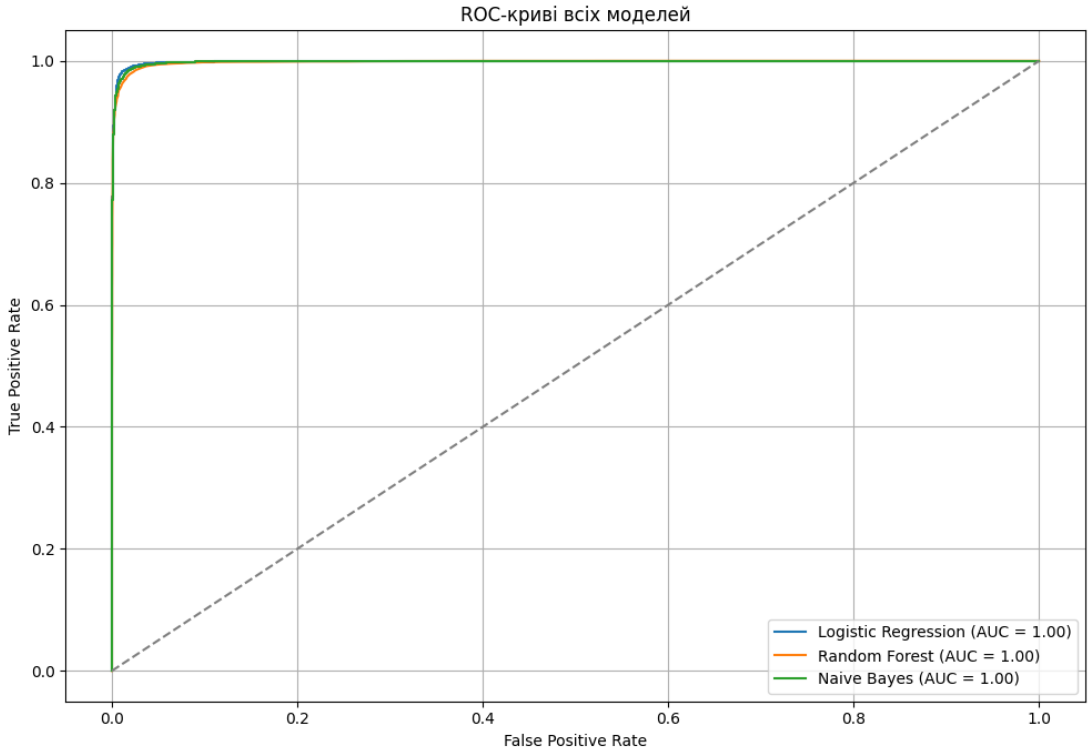

Рисунок 4.8 – ROC-Криві для всіх моделей

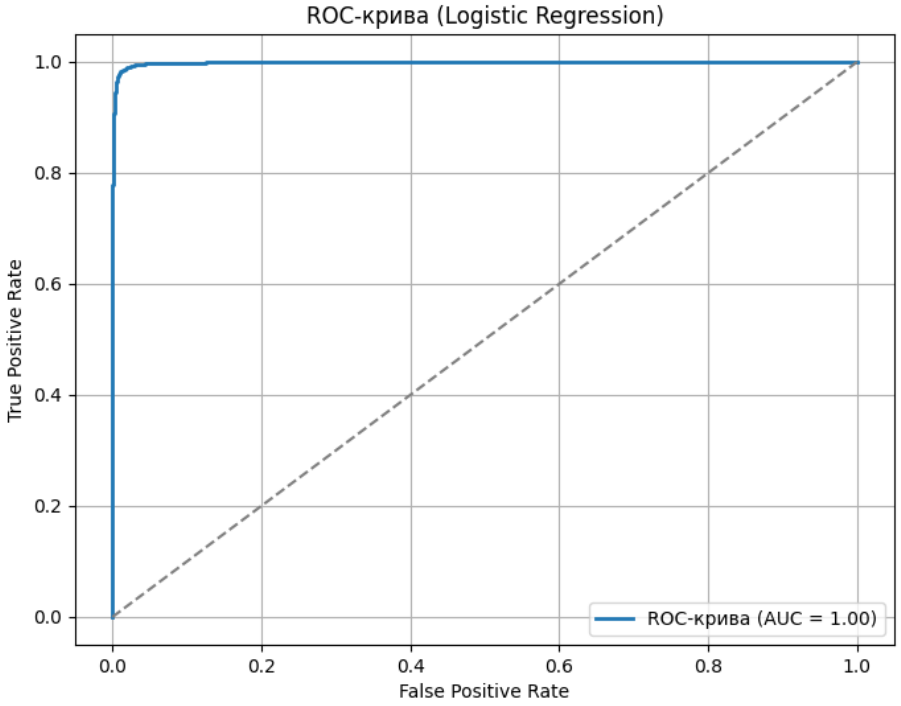

Рисунок 4.9 – ROC-Крива для Logistic Regression

Для перевірки продуктивності моделі в реальному світі також було проведено розширене тестування CLI на 20 реалістичних прикладах, де система правильно

класифікувала основні фішингові фрази, такі як: "оновіть платіжну інформацію", "перевірте свій обліковий запис", "натисніть тут, щоб отримати свій приз".

Результати тестування підтвердили високу ефективність розробленої системи, зокрема моделі Logistic Regression, яка продемонструвала найкращі показники точності та практичну придатність для застосування в режимі реального часу. У наступному підрозділі буде представлено приклад інтеграції цієї моделі у вигляді утиліти командного рядка для антифішингового аналізу електронних повідомлень.

4.5 Приклад інтеграції розробленої моделі

Для демонстрації ефективності розгорнутої системи виявлення фішингових атак був реалізований простий, але функціональний інструмент CLI (інтерфейс командного рядка) для полегшення миттєвої перевірки тексту вмісту повідомлень на фішингові індикатори.

Інструмент CLI розроблений на основі навченої моделі Logistic Regression, яка також була збережена разом із векторизатором TF-IDF. Користувачу надається можливість ввести фрагмент тексту – скажімо, електронний лист або повідомлення – на якому система позначає його як "Фішинг" або "Не фішинг”.

Продуктивність утиліти була випробувана на 20 типових повідомленнях (рис. 4.10), як підозрілих шаблонах, так і звичайних ділових листах. Модель змогла виявити всі приклади фішингу, навіть стандартні фрази, такі як "Натисніть тут, щоб отримати свій приз", "Оновіть свою платіжну інформацію", "Ваш обліковий запис був скомпрометований".

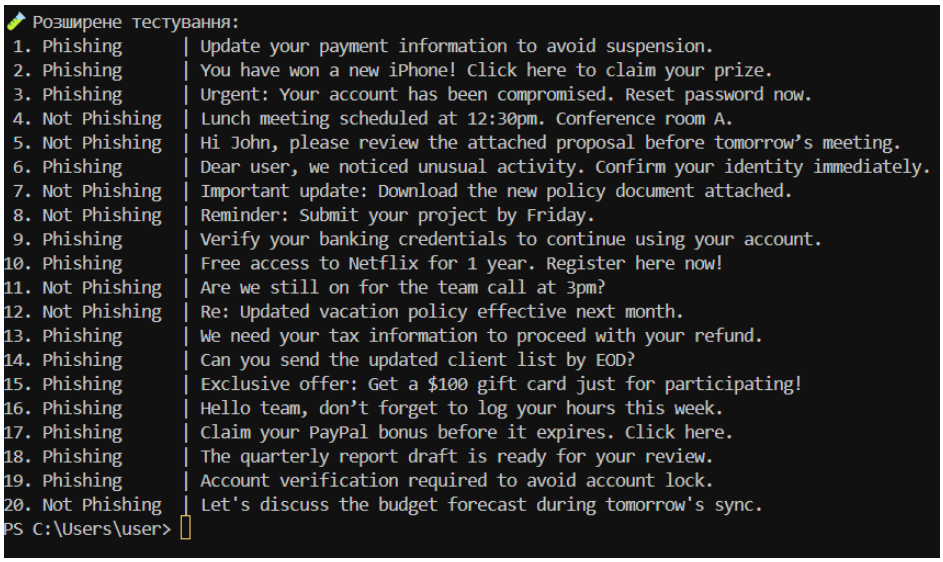

Рисунок 4.10 – Результати розширеного тестування класифікатора на прикладах текстів фішингових і нефішингових повідомлень

Окрім безпосереднього використання терміналу, ця реалізація CLI миттєво масштабується і може використовуватися як компонент інших систем:

• Шлюзи електронної пошти: попередня фільтрація вхідних повідомлень до доставки користувачам;

• Розширення браузера: сканування форм авторизації або зловмисних URL-адрес;

• Системи SIEM: автоматизовані плани реагування на інциденти;

• Служби технічної підтримки: швидка перевірка тексту на фішинг без додаткових сервісів.

Іншою перевагою є можливість аналізу порогових значень: замість бінарного результату можна використовувати ймовірнісний вихід (наприклад, через predict_proba()), що дозволяє гнучкіше оцінювати ступінь ризику – наприклад, позначати повідомлення як потенційно фішингові лише за умови перевищення 85% ймовірності. Це розширює функціональність системи та дозволяє застосовувати її в реальних умовах.

Загалом, результати реалізації підтверджують, що створене рішення не обмежується освітнім прикладом, а має практичну цінність і може бути інтегроване в інформаційні системи різного масштабу.

Розділ завершується демонстрацією того, що навіть компактна модель, належним чином навчена та валідована, здатна ефективно виявляти фішингові повідомлення в режимі реального часу. У поєднанні з CLI-інтерфейсом вона демонструє потенціал до інтеграції в існуючі процеси кіберзахисту та розширення в рамках корпоративних або хмарних платформ.

ВИСНОВКИ

Ця робота містить детальний аналіз і реалізацію методів штучного інтелекту для виявлення фішингових атак. Дослідження демонструє, що застосування машинного навчання, глибинного навчання та обробки природної мови суттєво підвищує здатність виявляти та запобігати фішинговим загрозам у сучасних комунікаційних системах.

Теоретична частина дослідження охоплює огляд типів фішингу, векторів атак і сучасних тенденцій, з особливим акцентом на зростаючу складність фішингових кампаній, зокрема тих, які генеруються за допомогою великих мовних моделей. Широкий огляд рішень на базі ШІ показав, що алгоритми, такі як Random Forest, XGBoost, Logistic Regression та нейронні мережі (CNN, LSTM і гібридні моделі), демонструють високу точність, особливо при навчанні на репрезентативних і змістовних наборах даних.

Практична частина демонструє повний цикл створення системи виявлення фішингу: від попередньої обробки даних і виділення ознак за допомогою TF-IDF до навчання, оцінки моделей та впровадження у вигляді утиліти командного рядка. Найвищу точність серед протестованих моделей показав Linear SVM (99,25%), однак Logistic Regression була обрана основною моделлю через її збалансованість між точністю, інтерпретованістю та простотою інтеграції в режимі реального часу.

Експерименти підтвердили, що налаштовані моделі ШІ здатні з високою точністю виявляти фішингові повідомлення, мають низький рівень хибно позитивних спрацьовувань і стійкі до змін у шаблонах атак. Розроблений інструмент CLI, що базується на навчальній моделі, успішно класифікував реальні приклади фішингових повідомлень, що підтверджує доцільність цього підходу для ширшого застосування в кібербезпеці.

Крім того, у роботі розглянуто етичні та правові аспекти використання ШІ в галузі кібербезпеки, зокрема вимоги GDPR, стандарт ISO/IEC 42001 та EU AI Act. Результати підкреслюють необхідність пояснюваності рішень ШІ, захисту прав

користувачів та надійного управління моделями, особливо у системах, які автоматично фільтрують або блокують комунікацію.

У підсумку, запропонована система виявлення фішингу на основі ШІ є не лише теоретично обґрунтованою, але й має практичне застосування. Вона може слугувати основою для подальших досліджень інтелектуальних захисних механізмів і бути інтегрована у шлюзи електронної пошти, розширення браузерів або системи SIEM. Викладені у роботі методи та результати можуть бути розширені для протидії ще складнішим загрозам, таким як цільовий фішинг (spear-phishing), smishing і соціальна інженерія на основі deepfake.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Basit A., Qamar F., Zafar M., Jalil Z., Alazab M. A Comprehensive Survey of AI-enabled Phishing Attacks Detection Techniques // *Journal of Network and Systems Management*. – 2020. – Vol. 28. – P. 1125–1153. – DOI: 10.1007/s11235-020-00733-5.

2. APWG Report 2023 – Global Phishing Trends [Електронний ресурс]. – Режим доступу: https://apwg.org/trendsreports/

3. Ferreira A., Teles A. Social Engineering Principles in Phishing // *Journal of Internet Services and Applications*. – 2019. – Vol. 10. – № 12. – DOI: 10.1007/s41060-024-00579-w.

4. Eze K., Shamir L. Analysis and Prevention of AI-based Phishing Email Attacks [Електронний ресурс] // *arXiv preprint*. – 2024. – arXiv:2405.05435. – Режим доступу: https://arxiv.org/abs/2405.05435.

5. Colonial Pipeline Incident – Phishing Entry Point [Електронний ресурс]. – CISA, 2021. – Режим доступу: https://www.cisa.gov/news-events/news/colonial-pipeline-cyber-incident

6. Proofpoint. OneDrive Phishing Campaign // *Threat Reports*. – 2023. – [Електронний ресурс]. – Режим доступу: https://www.proofpoint.com/us/resources/threat-reports

7. 9 Types of Phishing Attacks and How to Identify Them // *CSO Online*. – [Електронний ресурс]. – Режим доступу: https://www.csoonline.com/article/563353

8. Smishing vs. Phishing vs. Vishing // *HP Tech Takes*. – [Електронний ресурс]. – Режим доступу: https://www.hp.com/us-en/shop/tech-takes/smishing-vs-phishing-vs-vishing

9. ENISA Threat Landscape Report 2022 – Smishing Trends // *ENISA*. – 2022. – [Електронний ресурс]. – Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2022

10. Deepfake CEO Voice Fraud // *BBC News*. – 2019. – [Електронний ресурс]. – Режим доступу: https://www.bbc.com/news/technology-47481259

11. Basit A. та ін. ML Accuracy in Phishing Detection // *Journal of Network and Systems Management*. – 2020. – Vol. 28. – P. 1125–1153.

12. Sethi A., Garg S., Shafiq M.Z. Transformer Models for Phishing Detection // *Journal of Internet Services and Applications*. – 2022. – DOI: 10.1007/s41060-024-00579-w

13. Adaptive Capabilities of AI in Cyber Defense // *Journal of Network and Systems Management*. – 2020. – [Електронний ресурс]. – Режим доступу: https://link.springer.com/article/10.1007/s11235-020-00733-5

14. Sophos & Google. LLM Use in Phishing Prevention [Електронний ресурс]. – Режим доступу: https://www.sophos.com/en-us/cybersecurity-explained/ai-in-cybersecurity

15. Gmail AI Spam Filter // *Google Blog*. – 2022. – [Електронний ресурс]. – Режим доступу: https://blog.google/technology/safety-security/how-were-using-ai-to-combat-the-latest-scams/

16. Gemini Nano for Chrome // *Google I/O*. – 2024. – [Електронний ресурс]. – Режим доступу: https://blog.google/products/chrome/google-gemini-nano-android/

17. Microsoft. Office 365 Anti-Phishing // *Digital Defense Report*. – 2023. – [Електронний ресурс]. – Режим доступу: https://www.microsoft.com/en-us/security/blog/2023/10/12/2023-digital-defense-report/

18. Cloudflare. AI-based Email and DNS Protection [Електронний ресурс]. – Режим доступу: https://developers.cloudflare.com/email-security/

19. ElAassal R. та ін. PhishBench Platform for Model Comparison [Електронний ресурс]. – Режим доступу: https://phishbench.github.io/

20. Закон України «Про основні засади забезпечення кібербезпеки України» [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#Text

21. Закон України «Про захист персональних даних» [Електронний ресурс]. – Режим доступу: https://cedem.org.ua/en/library/law-of-ukraine-on-protection-of-personal-data/

22. General Data Protection Regulation (GDPR) [Електронний ресурс]. – Режим доступу: https://gdpr-info.eu/art-22-gdpr/

23. ISO/IEC 42001:2023 – Системи управління ШІ [Електронний ресурс]. – Режим доступу: https://www.iso.org/standard/81230.html

24. NIST. AI Risk Management Framework (AI RMF 1.0) [Електронний ресурс]. – Режим доступу: https://www.nist.gov/itl/ai-risk-management-framework

25. Регламент ЄС 2024/1689 – AI Act [Електронний ресурс]. – Режим доступу: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689

26. GDPR, стаття 22 – автоматизоване прийняття рішень [Електронний ресурс]. – Режим доступу: https://gdpr-info.eu/art-22-gdpr/

27. UNESCO. Recommendation on the Ethics of Artificial Intelligence. – 2021. – [Електронний ресурс]. – Режим доступу: https://www.unesco.org/en/artificial-intelligence/recommendation-ethics

28. Harvard Gazette. Етичні ризики ШІ [Електронний ресурс]. – Режим доступу: https://news.harvard.edu/gazette/story/2020/10/ethical-concerns-mount-as-ai-takes-bigger-decision-making-role/

29. ISO. Building a Responsible AI: How to Manage the AI Ethics Debate [Електронний ресурс]. – Режим доступу: https://www.iso.org/artificial-intelligence/responsible-ai-ethics

30. AI Act – відповідальність і ланцюг дій [Електронний ресурс]. – Режим доступу: https://artificialintelligenceact.eu/

31. GDPR + AI Act – пояснюваність рішень [Електронний ресурс]. – Режим доступу: https://gdpr-info.eu/art-22-gdpr/

32. ISO/IEC 42001 – контроль і план реагування [Електронний ресурс]. – Microsoft Docs. – Режим доступу: https://learn.microsoft.com/ru-ru/compliance/regulatory/offering-iso-42001

33. Sallouma S., Gaber T., Vadera S., Shaalan K. Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey // *Procedia Computer Science*. – 2021. – Vol. 189. – P. 19–28. – DOI: 10.1016/j.procs.2021.05.077

34. Alshingiti Z., Alaqel R., Al-Muhtadi J., Haq Q.E.U., Saleem K., Faheem M.H. A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN // *Electronics*. – 2023. – Vol. 12(1). – Article №232. – DOI: 10.3390/electronics12010232

35. Nepal S., Gurung H., Nepal R. Phishing URL Detection Using CNN-LSTM and Random Forest Classifier // *Preprint*. – 2022. – 14 с. – DOI: 10.21203/rs.3.rs-2043842/v2

36. Tang L., Mahmoud Q.H. A Survey of Machine Learning-Based Solutions for Phishing Website Detection // *Machine Learning and Knowledge Extraction*. – 2021. – Vol. 3(3). – P. 672–694. – DOI: 10.3390/make3030034

37. Shombot E.S., Dusserre G., Bestak R., Ahmed N.B. An Application for Predicting Phishing Attacks: A Case of Implementing a Support Vector Machine Learning Model // *Cyber Security and Applications*. – 2024. – Vol. 2. – Article ID 100036. – DOI: 10.1016/j.csa.2024.100036

38. Ozcan A., Catal C., Donmez E., Senturk B. A Hybrid DNN–LSTM Model for Detecting Phishing URLs // *Neural Computing and Applications*. – 2021. – Vol.

39. Verma R., Shashidhar N., Hossain N. Detecting Phishing Emails the Natural Language Way // *In*: Foresti S., Yung M., Martinelli F. (eds) *Computer Security – ESORICS 2012*. Lecture Notes in Computer Science, Vol. 7459. – Springer, Berlin, Heidelberg. – 2012. – P. 824–841. – DOI: 10.1007/978-3-642-33167-1_47

40. Kaggle. Phishing Site URLs Dataset [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls

41. Kaggle. Phishing Email Dataset [Електронний ресурс]. – Режим доступу: https://www.kaggle.com/datasets/naserabdullahalam/phishing-email-dataset

ДОДАТОК А

ВИХІДНИЙ КОД МОДЕЛІ

```
import pandas as pd
import os
import matplotlib.pyplot as plt
import time
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.metrics import (
    classification_report, confusion_matrix,
ConfusionMatrixDisplay,
    roc_curve, auc, matthews_corrcoef, log_loss, accuracy_score
)

# === 1. Підключення датасетів ===
datasets_info = {
    "C:/Users/user/Desktop/Diploma/phishing_email.csv":
["subject", "body", "label"],
    "C:/Users/user/Desktop/Diploma/CEAS_08.csv": ["subject",
"body", "label"],
    "C:/Users/user/Desktop/Diploma/Nazario.csv": ["subject",
"body", "label"],
    "C:/Users/user/Desktop/Diploma/Nigerian_Fraud.csv":
["body", "label"],
    "C:/Users/user/Desktop/Diploma/Enron.csv": ["subject",
"body", "label"],
    "C:/Users/user/Desktop/Diploma/Ling.csv": ["subject",
"body", "label"]
}

combined_data = []
for file, cols in datasets_info.items():
```

```
        df = pd.read_csv(file)
        for col in cols:
            if col not in df.columns:
                df[col] = ""
        df['text'] = df.get('subject', '') + ' ' + df.get('body',
'')
        df = df[['text', 'label']]
        combined_data.append(df)

    # === 2. Об'єднання + очищення ===
    full_dataset = pd.concat(combined_data, ignore_index=True)
    full_dataset = full_dataset.dropna(subset=['text'])
    full_dataset = full_dataset[full_dataset['text'].str.strip() !=
""]

    # === 3. Розбиття та векторизація ===
    X_train, X_test, y_train, y_test = train_test_split(
        full_dataset['text'], full_dataset['label'], test_size=0.3,
random_state=42
    )

    vectorizer = TfidfVectorizer()
    X_train_vec = vectorizer.fit_transform(X_train)
    X_test_vec = vectorizer.transform(X_test)

    # === 4. Моделі ===
    models = {
        "Logistic Regression": LogisticRegression(max_iter=1000),
        "Random Forest": RandomForestClassifier(n_estimators=100,
random_state=42),
        "Naive Bayes": MultinomialNB(),
        "Linear SVM": LinearSVC(max_iter=1000)
    }

    results = {}
```

```
    # === 5. Навчання та оцінка ===
    for name, model in models.items():
        print(f"\n🔧 Навчання моделі: {name}")
        start_time = time.time()
        model.fit(X_train_vec, y_train)
        elapsed = time.time() - start_time

        y_pred = model.predict(X_test_vec)
        y_prob = model.predict_proba(X_test_vec)[:, 1] if
hasattr(model, "predict_proba") else None

        acc = accuracy_score(y_test, y_pred)
        mcc = matthews_corrcoef(y_test, y_pred)
        logloss = log_loss(y_test, y_prob) if y_prob is not None
else None

        cm = confusion_matrix(y_test, y_pred)
        tn, fp, fn, tp = cm.ravel()
        specificity = tn / (tn + fp)
        fpr = fp / (fp + tn)
        fnr = fn / (fn + tp)

        results[name] = {
            "Accuracy": acc,
            "MCC": mcc,
            "Log Loss": logloss,
            "Specificity": specificity,
            "FPR": fpr,
            "FNR": fnr,
            "Train Time (s)": elapsed,
            "TP": tp,
            "TN": tn,
            "FP": fp,
            "FN": fn
        }
```

```
        print(f"Матриця помилок:\n{cm}")
        print(f"TP: {tp}, TN: {tn}, FP: {fp}, FN: {fn}")
        print(classification_report(y_test, y_pred))

        # === Побудова та збереження матриці помилок для кожної
моделі ===
        plt.figure(figsize=(5, 4))
        disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=['Not Phishing', 'Phishing'])
        disp.plot(cmap='Blues', values_format='d', ax=plt.gca())
        plt.title(f"Матриця помилок – {name}")
        plt.tight_layout()
        os.makedirs("graphs/conf_matrices", exist_ok=True)
        plt.savefig(f"graphs/conf_matrices/confusion_matrix_{name.r
eplace(' ', '_').lower()}.png")
        plt.show()
        plt.close()

    results_df = pd.DataFrame(results).T  # Transpose для зручного
вигляду
    results_df.index.name = 'Model'
    print("\n Загальна таблиця метрик:")
    print(results_df.round(4))  # округлення для читабельності

    # Зберегти таблицю у CSV для подальшого аналізу
    results_df.to_csv("graphs/model_results_summary.csv")

    # === 6. ROC для Logistic Regression ===
    final_model = models["Logistic Regression"]
    y_prob = final_model.predict_proba(X_test_vec)[:, 1]
    fpr_roc, tpr_roc, _ = roc_curve(y_test, y_prob)
    roc_auc = auc(fpr_roc, tpr_roc)

    plt.figure(figsize=(8, 6))
    plt.plot(fpr_roc, tpr_roc, label=f"ROC-крива (AUC =
{roc_auc:.2f})", linewidth=2)
```

```
    plt.plot([0, 1], [0, 1], linestyle="--", color="gray")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title("ROC-крива (Logistic Regression)")
    plt.legend(loc="lower right")
    plt.grid(True)
    plt.show()

    # === 7. CLI-функція ===
    def predict_phishing(text):
        vec = vectorizer.transform([text])
        pred = final_model.predict(vec)[0]
        return "Phishing" if pred == 1 else "Not Phishing"

    # === 8. Тест CLI ===
    print("\nТест CLI:")
    print("1:", predict_phishing("Please verify your account to
avoid deactivation."))
    print("2:", predict_phishing("Team lunch at 1pm. Bring your
laptop."))

    # === 9. Тест на 20 прикладах ===
    test_texts = [
        "Update your payment information to avoid suspension.",
        "You have won a new iPhone! Click here to claim your
prize.",
        "Urgent: Your account has been compromised. Reset password
now.",
        "Lunch meeting scheduled at 12:30pm. Conference room A.",
        "Hi John, please review the attached proposal before
tomorrow's meeting.",
        "Dear user, we noticed unusual activity. Confirm your
identity immediately.",
        "Important update: Download the new policy document
attached.",
        "Reminder: Submit your project by Friday.",
```

```
        "Verify your banking credentials to continue using your
account.",
        "Free access to Netflix for 1 year. Register here now!",
        "Are we still on for the team call at 3pm?",
        "Re: Updated vacation policy effective next month.",
        "We need your tax information to proceed with your
refund.",
        "Can you send the updated client list by EOD?",
        "Exclusive offer: Get a $100 gift card just for
participating!",
        "Hello team, don’t forget to log your hours this week.",
        "Claim your PayPal bonus before it expires. Click here.",
        "The quarterly report draft is ready for your review.",
        "Account verification required to avoid account lock.",
        "Let's discuss the budget forecast during tomorrow's sync."
    ]

    print("\nРозширене тестування:")
    for i, text in enumerate(test_texts, 1):
        result = predict_phishing(text)
        print(f"{i:2}. {result:<13} | {text}")

        # === 10. Автоматичне збереження графіків та побудова ROC-
кривих для всіх моделей ===
    import os

    # === Графік метрик моделей ===
    results_df = pd.DataFrame(results).T
    results_df[["Accuracy", "MCC", "Specificity"]].plot(kind="bar",
figsize=(12, 6))
    plt.title("Порівняння моделей за метриками")
    plt.ylabel("Значення")
    plt.ylim(0, 1.1)
    plt.grid(axis='y')
    plt.xticks(rotation=45)
    plt.tight_layout()
```

```
    plt.savefig("graphs/metrics_comparison.png")
    plt.show()

    # === Графік часу навчання ===
    results_df["Train Time (s)"].plot(kind="bar", figsize=(10, 5),
color="orange")
    plt.title("Час навчання моделей")
    plt.ylabel("Секунди")
    plt.grid(axis='y')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.savefig("graphs/training_times.png")
    plt.show()

    # === ROC-криві для моделей, які мають predict_proba ===
    plt.figure(figsize=(10, 7))
    for name, model in models.items():
        if hasattr(model, "predict_proba"):
            y_prob = model.predict_proba(X_test_vec)[:, 1]
            fpr_roc, tpr_roc, _ = roc_curve(y_test, y_prob)
            roc_auc = auc(fpr_roc, tpr_roc)
            plt.plot(fpr_roc, tpr_roc, label=f"{name} (AUC =
{roc_auc:.2f})")

    plt.plot([0, 1], [0, 1], linestyle="--", color="gray")
    plt.xlabel("False Positive Rate")
    plt.ylabel("True Positive Rate")
    plt.title("ROC-криві всіх моделей")
    plt.legend(loc="lower right")
    plt.grid(True)
    plt.tight_layout()
    plt.savefig("graphs/roc_curves_all_models.png")
    plt.show()
```