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: «Захист від фішингових атак за допомогою штучного інтелекту»

оцінка « _____ »

Голова ЕК
Мичуда Л.З.

Керівник: Ph.D Шеханін К.Ю.

Рецензент: к.ф.-м.н. Карась І.В.

Виконавець: студент групи КБ-41
Безсмольна А.Р.

Харків 2025

РЕФЕРАТ

Пояснювальна записка містить 60 сторінок, 2 таблиці та 23 посилання на джерела.

Мета роботи полягає у розробці та аналізі ефективних методів захисту від фішингових атак на основі технологій штучного інтелекту. Робота присвячена вивченню типів фішингових атак, аналізу традиційних методів захисту та дослідженню можливостей використання алгоритмів ШІ для виявлення та попередження фішингу.

Об’єктом дослідження дипломної роботи є процеси виявлення та запобігання фішинговим атакам у цифрових середовищах.

Предметом дослідження виступають методи та алгоритми штучного інтелекту, що застосовуються для захисту від фішингових атак.

Основні методи дослідження полягають в розробці та тестуванні моделі ШІ, яка поєднує кілька підходів машинного навчання для підвищення точності виявлення фішингових атак, а також в аналізі їхньої ефективності в умовах сучасних кіберзагроз.

У роботі досліджено види та техніки фішингу, сучасні традиційні методи виявлення атак, проблеми їхньої ефективності та обмеження, розглянуто роль ШІ у сфері кіберзахисту, особливо методи машинного та глибокого навчання, аналіз поведінки користувачів, а також практичне впровадження ШІ-моделі.

Результати роботи можуть бути використані у наукових публікаціях, практичних розробках систем безпеки, а також для підвищення обізнаності про ефективні методи захисту від фішингових атак із застосуванням сучасних технологій.

Ключові слова: ФІШИНГ, ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, ФІШИНГОВА АТАКА, ВИЯВЛЕННЯ АНОМАЛІЙ, МОДЕЛЬ ШІ, ОБРОБКА ПРИРОДНОЇ МОВИ, ЗАХИСТ ІНФОРМАЦІЇ.

ABSTRACT

The explanatory note contains 60 pages, 2 tables and 23 sources.

The purpose of this thesis is to develop and analyze effective methods of protection against phishing attacks using artificial intelligence technologies. The work focuses on studying the types of phishing attacks, analyzing traditional protection methods, and exploring the potential of AI algorithms for detecting and preventing phishing.

The object of the research is the processes of detecting and preventing phishing attacks in digital environments.

The subject of the research is the methods and algorithms of artificial intelligence applied to phishing protection.

The main research methods involve the development and testing of an AI model that combines multiple machine learning approaches to improve the accuracy of phishing detection, as well as analyzing its effectiveness under modern cyber threat conditions.

This work explores various types and techniques of phishing, traditional attack detection methods, their effectiveness and limitations, and the role of AI in cybersecurity — particularly machine learning, deep learning, user behavior analysis, and the practical implementation of the AI-based model.

The results of this thesis can be used in academic publications, practical development of security systems, and to raise awareness about effective phishing protection methods using modern technologies.

Keywords: PHISHING, ARTIFICIAL INTELLIGENCE, NEURAL NETWORKS, MACHINE LEARNING, PHISHING ATTACK, ANOMALY DETECTION, AI MODEL, NATURAL LANGUAGE PROCESSING, INFORMATION SECURITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

Accuracy	– Точність класифікації
API	– Application Programming Interface
AWS	– Amazon Web Services
BERT	– Bidirectional Encoder Representations from Transformers
Big Data	– Великі дані
CNN	– Convolutional Neural Network
DKIM	– DomainKeys Identified Mail
DL	– Deep Learning
DMARC	– Domain-based Message Authentication, Reporting, and Conformance
EDR	– Endpoint Detection and Response
F1-Score	– Гармонійне середнє між точністю передбачення та повнотою
GDPR	– General Data Protection Regulation
HTML	– HyperText Markup Language
HTTPS	– HyperText Transfer Protocol Secure
Kaggle	– Платформа для змагань із даних і машинного навчання
LSTM	– Long Short-Term Memory
ML	– Machine Learning
NLP	– Natural Language Processing
Precision	– Точність передбачення
Recall	– Повнота класифікації
RNN	– Recurrent Neural Network
SIEM	– Security Information and Event Management
SMTP	– Simple Mail Transfer Protocol

SPF	– Sender Policy Framework
SVM	– Support Vector Machine
URL	– Uniform Resource Locator
UBA	– User Behavior Analysis
XGBoost	– Extreme Gradient Boosting
ШІ	– Штучний інтелект

ВСТУП

У сучасному цифровому світі, де інформаційні технології стрімко розвиваються, кіберзагрози стають дедалі складнішими та масштабнішими. Одним із найпоширеніших і найнебезпечніших видів кібератак є фішингові атаки, спрямовані на викрадення конфіденційної інформації, такої як паролі, фінансові дані чи персональні відомості. Фішинг ґрунтується на соціальній інженерії, використовуючи методи обману для маніпулювання поведінкою користувачів. За даними звіту Verizon Data Breach Investigations Report 2024 року, близько 36% усіх кібератак пов’язані з фішингом, що свідчить про його значну поширеність і ефективність. Водночас, із розвитком штучного інтелекту (ШІ) зловмисники також удосконалюють свої методи, створюючи більш персоналізовані та важковидимі атаки, зокрема за допомогою генеративних моделей ШІ.

Традиційні методи захисту від фішингу, такі як чорні списки, сигнатурний аналіз або фільтрація на основі правил, уже не здатні повною мірою протистояти новим загрозам через їхню статичність і обмежену адаптивність. У цьому контексті штучний інтелект відкриває нові можливості для створення динамічних і проактивних систем захисту. Алгоритми машинного навчання, нейронні мережі та обробка природної мови дозволяють аналізувати великі обсяги даних, виявляти приховані закономірності та прогнозувати потенційні загрози з високою точністю.

Актуальність теми бакалаврської роботи зумовлена кількома ключовими факторами. По-перше, зростання кількості фішингових атак становить серйозну загрозу для безпеки як індивідуальних користувачів, так і організацій. По-друге, швидкий розвиток технологій ШІ створює нові можливості для автоматизації процесів виявлення та запобігання кібератакам. По-третє, в Україні, де цифровізація активно набирає обертів, питання кібербезпеки набувають особливої ваги в контексті захисту критичної інфраструктури та особистих даних

громадян. Таким чином, дослідження методів захисту від фішингових атак за допомогою ШІ є важливим внеском у розвиток сучасних систем кібербезпеки.

Метою роботи є розробка та аналіз ефективних методів захисту від фішингових атак на основі технологій штучного інтелекту. Для досягнення цієї мети поставлено такі завдання:

1. Провести аналіз сучасних фішингових атак, їхньої класифікації, технік і методів виявлення.
2. Дослідити можливості застосування алгоритмів машинного навчання та нейронних мереж для виявлення фішингових загроз.
3. Розробити модель ШІ для автоматичного виявлення фішингових атак і оцінити її ефективність на основі реальних і синтетичних даних.
4. Визначити перспективи впровадження ШІ-рішень у практичні системи кібербезпеки.

Об’єктом дослідження є процеси виявлення та запобігання фішинговим атакам у цифрових середовищах. Предметом дослідження виступають методи та алгоритми штучного інтелекту, що застосовуються для захисту від фішингових атак.

Наукова новизна роботи полягає в розробці та тестуванні моделі ШІ, яка поєднує кілька підходів машинного навчання для підвищення точності виявлення фішингових атак, а також в аналізі їхньої ефективності в умовах сучасних кіберзагроз. Практична цінність роботи полягає в можливості використання запропонованих рішень для створення автоматизованих систем захисту, які можуть бути інтегровані в корпоративні та індивідуальні середовища.

Дослідження базується на аналізі сучасних наукових публікацій, практичних посібників і звітів у сфері кібербезпеки, а також на експериментальній перевірці розробленої моделі. Очікується, що результати роботи сприятимуть підвищенню рівня захисту від фішингових атак і стануть основою для подальших досліджень у цій сфері.

1 АНАЛІЗ ФІШИНГОВИХ АТАК ТА СУЧАСНИХ МЕТОДІВ ЇХ ВИЯВЛЕННЯ

1.1 Визначення та класифікація фішингових атак.

Фішингові атаки є одним із найпоширеніших видів кіберзлочинів, які використовують методи соціальної інженерії для обману користувачів та отримання їхньої конфіденційної інформації, такої як паролі, банківські дані чи особисті відомості. Термін "фішинг" походить від англійського слова "fishing" (риболовля), що метафорично відображає процес "виловлювання" даних користувачів за допомогою приманок. Згідно з визначенням, фішинг — це шахрайська практика, спрямована на імітацію легітимних джерел (вебсайтів, електронних листів, повідомлень) для маніпуляції жертвою з метою отримання доступу до її даних або фінансових ресурсів [1]. Такі атаки часто використовують психологічні методи, щоб змусити користувача виконати певні дії, наприклад, перейти за шкідливим посиланням чи ввести дані на фальшивій сторінці.

Основною характеристикою фішингових атак є їхня спрямованість на людський фактор. Зловмисники експлуатують довіру користувачів до відомих брендів, організацій або осіб, створюючи переконливі копії офіційних ресурсів. Наприклад, фішинговий електронний лист може імітувати повідомлення від банку з проханням оновити пароль, що змушує користувача перейти на підроблений сайт. За даними досліджень, фішинг становить близько 36% усіх кібератак, що свідчить про його значну поширеність і ефективність [2]. Зростання кількості таких атак пов’язане з доступністю інструментів для їх створення, а також із можливістю автоматизації за допомогою сучасних технологій, зокрема штучного інтелекту.

Фішингові атаки класифікуються за різними критеріями, що дозволяє краще зрозуміти їхню природу та розробити ефективні методи протидії. Основні критерії класифікації включають канал атаки, цільову аудиторію, тип обману та

технічні засоби реалізації. За каналом атаки фішинг поділяється на такі основні типи:

- Електронний фішинг (email phishing): Найпоширеніший тип, що передбачає надсилання шахрайських листів, які імітують офіційні повідомлення від організацій. Наприклад, лист може містити посилання на фальшивий сайт для введення даних [3].
- Смартфінг (smishing): Фішинг через SMS-повідомлення, які часто містять заклики до дії, наприклад, підтвердження транзакції чи оновлення даних.
- Вішинг (vishing): Телефонний фішинг, коли зловмисники представляються працівниками компаній чи державних установ, щоб отримати конфіденційну інформацію.
- Фішинг через соціальні мережі: Атаки, що використовують платформи, такі як Facebook чи LinkedIn, для розсилки шкідливих повідомлень або створення фальшивих профілів.
- Фішинг через вебсайти (web-based phishing): Створення підроблених вебсторінок, які імітують офіційні сайти банків, платіжних систем чи інших сервісів [1].

За цільовою аудиторією фішинг поділяється на масовий і цільовий. Масовий фішинг спрямований на широку аудиторію і характеризується низькою персоналізацією. Наприклад, зловмисники розсилають тисячі листів із загальними повідомленнями про "блокування рахунку" чи "виграш у лотереї". Цільовий фішинг, або спірфішинг (spear phishing), є більш складним і орієнтованим на конкретну особу чи організацію. Такі атаки використовують попередньо зібрану інформацію про жертву, наприклад, її ім’я, посаду чи інтереси, щоб зробити повідомлення максимально переконливим [4]. Спірфішинг часто застосовується для атак на корпоративні мережі, де зловмисники прагнуть отримати доступ до внутрішніх систем компанії.

Ще одним видом цільового фішингу є "китовий фішинг" (whaling), спрямований на високопосадовців, таких як директори чи фінансові менеджери.

Такі атаки ретельно готуються і можуть включати імітацію листування від імені колег чи партнерів. Наприклад, зловмисник може надіслати лист від імені CEO з проханням терміново переказати кошти на певний рахунок. Окремим підвидом є "клоновий фішинг" (clone phishing), коли зловмисники копіюють легітимний електронний лист, змінюючи в ньому посилання чи вкладення на шкідливі [3].

За типом обману фішингові атаки можна поділити на:

— Атаки з використанням шкідливих посилань: Користувача спрямовують на фальшивий сайт, який краде введені дані.

— Атаки з використанням шкідливих вкладень: Листи містять файли (наприклад, PDF чи Word), які встановлюють зловмисне програмне забезпечення.

— Атаки без технічних засобів: Повністю базуються на соціальній інженерії, наприклад, телефонні дзвінки чи текстові повідомлення з проханням надати дані [2].

Технічні засоби реалізації фішингу також різноманітні. Зловмисники можуть використовувати підроблені доменні імена, що нагадують офіційні (наприклад, "g00gle.com" замість "google.com"), або застосовувати техніки спуфінгу для маскування відправника електронного листа. Останнім часом спостерігається зростання використання штучного інтелекту для створення переконливих фішингових повідомлень. Наприклад, генеративні моделі ШІ дозволяють створювати тексти, які важко відрізнити від людських, або навіть підроблені відео та аудіо (deepfake) для вішингу [5].

Класифікація фішингових атак за рівнем складності також є важливою. Прості атаки, як правило, мають низький рівень персоналізації та легко виявляються сучасними системами захисту. Складні атаки, навпаки, використовують комбінацію соціальної інженерії, технічних засобів і попереднього аналізу жертви, що робить їх більш небезпечними. Наприклад, атака може починатися з розсилки масового фішингу, а після отримання відповіді від жертви переходити до спірфішингу з використанням зібраних даних [4].

1.2 Основні техніки, що використовуються зловмисниками у фішингових атаках

Фішингові атаки є складними кіберзагрозами, які базуються на поєднанні соціальної інженерії, психологічних маніпуляцій і технічних методів для обману користувачів та отримання їхньої конфіденційної інформації. Зловмисники постійно вдосконалюють свої техніки, адаптуючись до сучасних технологій захисту та використовуючи нові можливості, зокрема штучний інтелект і автоматизацію. Основні техніки, що застосовуються у фішингових атаках, можна поділити на кілька категорій залежно від методів обману, каналів доставки та технічних засобів реалізації. Розуміння цих технік є ключовим для розробки ефективних стратегій протидії.

Однією з найпоширеніших технік є спуфінг (підміна), який використовується для маскування джерела атаки. Зловмисники можуть підробляти електронні адреси, номери телефонів або доменні імена, щоб створити ілюзію легітимності. Наприклад, техніка спуфінгу електронної пошти дозволяє відправляти листи, які виглядають так, ніби вони надійшли від офіційного джерела, такого як банк чи державна установа. Це досягається шляхом маніпуляції заголовками SMTP-протоколу, що ускладнює ідентифікацію справжнього відправника [6]. Спуфінг також застосовується у вішингу, коли зловмисники використовують технології підміни номера телефону, щоб імітувати дзвінки від довірених організацій.

Іншою поширеною технікою є створення підроблених вебсайтів, які імітують офіційні ресурси. Такі сайти зазвичай мають схожий дизайн, логотипи та URL-адреси, що відрізняються лише незначними деталями, наприклад, заміною символів (гомографічні атаки, як-от "g00gle.com" замість "google.com"). Користувачі, які переходять на такі сайти за посиланнями з фішингових листів чи повідомлень, вводять свої дані, які одразу потрапляють до зловмисників. Для підвищення ефективності зловмисники можуть використовувати SSL-сертифікати, щоб підроблені сайти виглядали безпечними (з позначкою "https")

[7]. Ця техніка особливо популярна в атаках на користувачів онлайн-банкінгу чи платіжних систем.

Шкідливі вкладення є ще одним поширеним методом фішингових атак. Зловмисники надсилають електронні листи з файлами, такими як PDF, Word або Excel, які містять шкідливий код. Після відкриття файлу на комп’ютері жертви може бути встановлено зловмисне програмне забезпечення, наприклад, кейлогери чи програми-вимагачі. Часто такі вкладення маскуються під легітимні документи, наприклад, рахунки, договори чи запрошення. У деяких випадках зловмисники використовують макроси в документах Microsoft Office, які активуються після дозволу користувача [8]. Ця техніка дозволяє не лише викрадати дані, але й отримувати повний контроль над пристроєм жертви.

Соціальна інженерія відіграє центральну роль у фішингових атаках, оскільки саме вона спрямована на маніпуляцію поведінкою користувача. Зловмисники використовують психологічні прийоми, такі як створення відчуття терміновості чи страху, щоб змусити жертву діяти швидко, не аналізуючи ситуацію. Наприклад, фішинговий лист може містити повідомлення про "блокування рахунку" чи "підозрілу активність", що спонукає користувача негайно перейти за посиланням або надати дані [9]. Іншим прикладом є імітація авторитетних джерел, таких як листи від імені керівника компанії чи державного органу, що особливо ефективно в цільових атаках (спірфішинг).

Використання шкідливих посилань є однією з найпростіших, але ефективних технік. Такі посилання зазвичай надсилаються через електронну пошту, SMS або соціальні мережі та ведуть на фальшиві сайти або автоматично завантажують шкідливе програмне забезпечення. Для маскування справжньої адреси зловмисники використовують скорочувачі URL (наприклад, bit.ly) або перенаправлення через кілька доменів. У деяких випадках посилання активують drive-by завантаження, коли шкідливий код встановлюється без взаємодії з користувачем [6]. Ця техніка часто комбінується з іншими методами, такими як спуфінг чи соціальна інженерія.

Останнім часом зловмисники дедалі частіше застосовують генеративний штучний інтелект для створення переконливих фішингових матеріалів. Наприклад, інструменти на основі великих мовних моделей дозволяють створювати електронні листи з природним текстом, який важко відрізнити від людського. Такі листи можуть бути персоналізованими, враховуючи ім’я, інтереси чи попередню активність жертви. Крім того, ШІ використовується для створення deepfake-аудіо чи відео, що застосовуються у вішингу. Наприклад, зловмисник може імітувати голос керівника компанії, щоб переконати працівника переказати кошти [10]. Ця техніка значно ускладнює виявлення атак, оскільки традиційні методи аналізу тексту стають менш ефективними.

Атаки через соціальні мережі стають дедалі популярнішими завдяки широкому поширенню платформ, таких як Facebook, Instagram чи LinkedIn. Зловмисники створюють фальшиві профілі, які імітують реальних осіб чи компанії, і надсилають повідомлення з шкідливими посиланнями або пропозиціями. Наприклад, у LinkedIn поширені атаки, коли фальшивий рекрутер пропонує жертві "роботу мрії", вимагаючи заповнити форму на підробленому сайті [8]. Ця техніка особливо ефективна в цільових атаках, оскільки соціальні мережі надають зловмисникам доступ до особистих даних користувачів.

Для підвищення ефективності фішингових атак зловмисники часто комбінують кілька технік. Наприклад, атака може починатися з масового фішингового листа з підробленою адресою відправника (спуфінг), який містить шкідливе посилання та текст, що створює відчуття терміновості (соціальна інженерія). Якщо користувач переходить за посиланням, він потрапляє на підроблений сайт, який краде його дані. У деяких випадках атака може продовжуватися через вішинг, коли зловмисник телефонує жертві, представляючись працівником служби безпеки [9]. Такий багатошаровий підхід значно ускладнює виявлення та запобігання атакам.

Техніки фішингових атак постійно еволюціонують, що вимагає від систем захисту гнучкості та адаптивності. Наприклад, зростання популярності хмарних сервісів призвело до появи атак, спрямованих на викрадення облікових даних для

доступу до Google Drive чи Microsoft 365. Водночас автоматизація та використання ШІ дозволяють зловмисникам створювати тисячі унікальних фішингових повідомлень за короткий час, що знижує ефективність традиційних методів захисту, таких як чорні списки чи сигнатурний аналіз [10]. У наступних пунктах цього розділу буде розглянуто сучасні методи виявлення фішингових атак, які враховують ці виклики та використовують передові технології для підвищення безпеки.

1.3 Огляд сучасних методів виявлення фішингу (традиційні підходи)

Одним із найпоширеніших традиційних методів є використання чорних списків (blacklists). Цей підхід полягає в створенні баз даних, які містять відомі фішингові URL-адреси, доменні імена або IP-адреси, пов’язані з шахрайськими ресурсами. Коли користувач намагається відкрити посилання чи вебсайт, система перевіряє його на відповідність записам у чорному списку. Якщо збіг знайдено, доступ блокується або видається попередження. Чорні списки активно використовуються в антивірусних програмах, браузерах (наприклад, Google Safe Browsing) і поштових фільтрах. За даними досліджень, чорні списки можуть блокувати до 80% відомих фішингових сайтів [11]. Однак їхня ефективність обмежена, оскільки зловмисники часто створюють нові домени чи змінюють URL-адреси, які ще не внесено до бази даних [12].

Іншим поширеним методом є фільтрація на основі правил (rule-based filtering). Цей підхід передбачає аналіз вмісту електронних листів, повідомлень або вебсайтів за допомогою заздалегідь визначених правил, які вказують на ознаки фішингу. Наприклад, правила можуть включати пошук підозрілих ключових слів (наприклад, "терміново", "блокування рахунку"), перевірку невідповідності доменної адреси відправника або виявлення орфографічних помилок, які часто трапляються в фішингових листах. Такі фільтри широко застосовуються в корпоративних поштових системах, таких як Microsoft Exchange чи Gmail. Перевагою цього методу є його простота та швидкість

обробки, але він неефективний проти цільових атак (спірфішингу), де зловмисники ретельно готують повідомлення без очевидних помилок [13].

Сигнатурний аналіз є ще одним традиційним методом, який базується на порівнянні вмісту повідомлень чи файлів із відомими сигнатурами фішингових атак. Сигнатура — це унікальний набір характеристик, наприклад, хеш-код шкідливого вкладення або шаблон тексту в електронному листі. Антивірусні програми, такі як Norton чи Kaspersky, використовують сигнатурні бази для виявлення відомих загроз. Цей метод ефективний проти масових атак, які використовують однакові шаблони, але він не здатний виявляти нові або модифіковані атаки, оскільки сигнатури для них ще не створено [14]. Крім того, сигнатурний аналіз потребує регулярного оновлення баз даних, що може затримувати реагування на нові загрози.

Аналіз заголовків електронної пошти (email header analysis) є важливим методом для виявлення фішингових листів, особливо тих, що використовують техніку спуфінгу. Цей підхід передбачає перевірку метаданих листа, таких як адреса відправника, маршрут передачі (SMTP-путь) і доменні записи (SPF, DKIM, DMARC). Наприклад, протокол DMARC дозволяє перевірити, чи відповідає домен відправника офіційному домену організації, що допомагає виявити підроблені листи. За даними досліджень, використання DMARC може знизити кількість успішних спуфінг-атак на 70% [15]. Однак цей метод менш ефективний проти атак, які не використовують електронну пошту, наприклад, смішингу чи фішингу через соціальні мережі.

Евристичний аналіз є більш гнучким традиційним підходом, який комбінує правила та шаблони для виявлення підозрілої поведінки. На відміну від сигнатурного аналізу, евристичні методи не покладаються на точну відповідність, а оцінюють сукупність характеристик, таких як структура URL-адреси, наявність перенаправлень чи підозрілі елементи HTML-коду на вебсайтах. Наприклад, евристичний фільтр може позначити сайт як фішинговий, якщо він містить форму для введення пароля, але не використовує HTTPS. Цей метод дозволяє виявляти нові атаки, які ще не внесено до чорних списків чи

сигнатурних баз, але він має високий рівень хибнопозитивних спрацьовувань, що може створювати незручності для користувачів [12].

Ще одним підходом є перевірка репутації доменів і відправників. Цей метод оцінює надійність домену чи IP-адреси на основі їхньої історії активності. Наприклад, якщо домен був зареєстрований недавно або пов'язаний із масовою розсилкою, він позначається як підозрілий. Системи, такі як Cisco Secure Email, використовують репутаційні бази для фільтрації вхідних повідомлень. Цей метод ефективний проти масового фішингу, але може бути обійдений зловмисниками, які використовують легітимні, але скомпрометовані домени [13].

Традиційні методи також включають освітні та організаційні заходи, які спрямовані на підвищення обізнаності користувачів. Наприклад, компанії проводять тренінги з розпізнавання фішингових листів або симуляції атак, щоб навчити працівників ідентифікувати підозрілі повідомлення. Хоча ці заходи не є технічними методами виявлення, вони відіграють важливу роль у зниженні ймовірності успішних атак. Дослідження показують, що регулярне навчання може зменшити кількість жертв фішингу на 40% [14]. Однак цей підхід залежить від людського фактора, що робить його вразливим до помилок.

Незважаючи на свою поширеність, традиційні методи мають низку суттєвих недоліків. По-перше, вони погано справляються з нульовими атаками (zero-day attacks), які використовують нові техніки чи невідомі шаблони. По-друге, більшість методів є реактивними, тобто реагують на вже відомі загрози, а не прогнозують нові. По-третє, зростання використання штучного інтелекту зловмисниками, наприклад, для створення персоналізованих листів чи deepfake-контенту, значно знижує ефективність статичних підходів [15]. Наприклад, чорні списки не можуть виявити фішинговий сайт, який існує лише кілька годин, а сигнатурний аналіз неефективний проти унікальних повідомлень, створених за допомогою генеративних моделей ШІ.

1.4 Роль штучного інтелекту в аналізі та протидії фішинговим атакам

Одним із основних напрямів використання ШІ в боротьбі з фішингом є аналіз вмісту повідомлень і вебсайтів за допомогою алгоритмів машинного навчання (ML). Ці алгоритми здатні ідентифікувати ознаки фішингу, такі як підозрілі ключові слова, невідповідності в структурі URL-адрес або аномалії в тексті, які важко виявити традиційними правилами. Наприклад, методи обробки природної мови (NLP) дозволяють аналізувати семантику електронних листів, визначаючи, чи відповідає їхній зміст типовим шаблонам фішингу, навіть якщо текст виглядає природним і персоналізованим. За даними досліджень, моделі ML, такі як Support Vector Machines (SVM) або Random Forest, досягають точності виявлення фішингових листів на рівні 95–98% за умови якісного навчання [16]. Це значно перевищує ефективність чорних списків, які часто пропускають нові домени.

Ще однією важливою роллю ШІ є виявлення аномалій у поведінці користувачів і систем. Алгоритми ШІ, зокрема методи глибокого навчання (Deep Learning), можуть аналізувати історичні дані про поведінку користувачів, такі як частота входів у систему, геолокація чи типові шаблони листування, і виявляти відхилення, які можуть свідчити про фішингову атаку. Наприклад, якщо працівник компанії отримує лист від "колеги" з незвичайного домену або з проханням виконати нетипову дію, система ШІ може позначити це як підозрілу активність. Такі підходи особливо ефективні проти цільових атак (спірфішингу), де традиційні методи, такі як сигнатурний аналіз, є малоефективними [17]. Аналіз поведінки також застосовується для виявлення компрометації облікових записів, коли зловмисники використовують викрадені дані для подальших атак.

Класифікація вебсайтів є ще одним напрямом, де ШІ демонструє значні переваги. Нейронні мережі, зокрема згорткові нейронні мережі (CNN), використовуються для аналізу візуальних і структурних елементів вебсайтів, таких як розташування форм, кольорова палітра чи код HTML, щоб визначити, чи є сайт фішинговим. Наприклад, ШІ може виявити, що сайт імітує сторінку

входу відомого банку, але містить незначні відмінності в коді чи дизайні. Такі методи дозволяють виявляти фішингові сайти в реальному часі, навіть якщо вони не внесені до чорних списків. Дослідження показують, що комбінація NLP і CNN може досягати точності до 99% у класифікації фішингових вебсайтів [18].

ШІ також відіграє ключову роль у автоматизації обробки великих обсягів даних. Фішингові атаки часто мають масовий характер, і ручна перевірка тисяч електронних листів чи URL-адрес є неможливою. Алгоритми ШІ, такі як кластеризація чи деревовидні моделі, дозволяють групувати схожі повідомлення чи сайти за їхніми характеристиками, виявляючи нові кампанії фішингу ще до того, як вони завдадуть шкоди. Наприклад, системи на основі ШІ, такі як Barracuda Sentinel, використовують кластеризацію для ідентифікації нових фішингових шаблонів у реальному часі, що значно скорочує час реагування на загрози [19]. Такий підхід дозволяє організаціям оперативно блокувати нові атаки, навіть якщо вони використовують раніше невідомі техніки.

Важливим аспектом є здатність ШІ адаптуватися до нових загроз. На відміну від традиційних методів, які потребують оновлення сигнатур чи правил, моделі ШІ можуть перенавчатися на нових даних, що дозволяє їм виявляти так звані нульові атаки (zero-day attacks). Наприклад, методи навчання з учителем (supervised learning) використовують набори даних із позначеними прикладами фішингових і легітимних повідомлень для створення моделей, які з часом стають точнішими. Водночас методи навчання без учителя (unsupervised learning) дозволяють виявляти аномалії без попереднього маркування даних, що особливо корисно для нових типів атак, створених за допомогою генеративного ШІ [20]. Ця адаптивність робить ШІ незамінним інструментом у боротьбі з постійно еволюціонуючими фішинговими загрозами.

Незважаючи на численні переваги, застосування ШІ в протидії фішингу має свої виклики. По-перше, для ефективної роботи моделей ШІ потрібні великі набори даних високої якості, які включають як фішингові, так і легітимні приклади. Збір таких даних може бути ускладнений через конфіденційність інформації чи обмежений доступ до реальних фішингових кампаній. По-друге,

ШІ-моделі можуть бути вразливими до атак типу "отруєння даних" (data poisoning), коли зловмисники навмисно вводять у навчальні набори хибні дані, щоб знизити точність моделі [17]. По-третє, високий рівень хибнопозитивних спрацьовувань може призводити до блокування легітимних повідомлень, що створює незручності для користувачів.

Ще одним викликом є використання ШІ самими зловмисниками. Генеративні моделі, такі як GPT, дозволяють створювати переконливі фішингові листи, які важко відрізнити від справжніх, або підроблені вебсайти з автоматично згенерованим контентом. Це створює своєрідну "гонку озброєнь", де захисні системи ШІ повинні постійно вдосконалюватися, щоб випереджати атакуючі технології [19]. Наприклад, deepfake-технології, які використовуються у вішингу, вимагають від систем захисту здатності аналізувати не лише текст, але й аудіо- та відеоконтент, що значно ускладнює задачу.

1.5 Проблеми та обмеження існуючих рішень у боротьбі з фішингом

Однією з ключових проблем традиційних методів є їхня реактивна природа. Чорні списки, сигнатурний аналіз і фільтрація на основі правил покладаються на попередньо відомі шаблони чи дані про фішингові атаки. Однак зловмисники часто створюють нові домени, URL-адреси чи шаблони повідомлень, які ще не внесено до баз даних. Наприклад, дослідження показують, що середня тривалість існування фішингового сайту становить лише кілька годин, що робить чорні списки неефективними для виявлення таких загроз у реальному часі [21]. Ця затримка між появою нової атаки та оновленням захисних баз даних дозволяє зловмисникам успішно атакувати користувачів.

Ще одним значним обмеженням є низька ефективність проти цільових атак (спірфішингу). Традиційні методи, такі як фільтрація на основі ключових слів чи перевірка заголовків електронної пошти, погано справляються з персоналізованими атаками, які використовують детальну інформацію про жертву, наприклад, її ім’я, посаду чи інтереси. Такі атаки зазвичай не містять очевидних помилок чи підозрілих елементів, що ускладнює їхнє виявлення за

допомогою статичних правил. Наприклад, спірфішинг, спрямований на високопосадовців (whaling), може імітувати легітимне листування з використанням скомпрометованих облікових записів, що робить традиційні фільтри безсилими [22].

Високий рівень хибнопозитивних і хибнонегативних спрацьовувань є серйозною проблемою як для традиційних, так і для ШІ-базованих рішень. Традиційні евристичні методи можуть позначати легітимні повідомлення як фішингові через наявність певних ключових слів чи незвичайну поведінку відправника, що створює незручності для користувачів. Наприклад, лист від нового клієнта з нестандартним форматом може бути помилково заблокований. У той же час ШІ-моделі, особливо ті, що використовують складні алгоритми глибокого навчання, можуть пропускати нові типи атак через недостатню якість навчальних даних або надмірну узагальненість моделей. Дослідження показують, що хибнопозитивні спрацьовування можуть досягати 10–15% у деяких евристичних системах [23].

Обмеженість даних для навчання ШІ-моделей є значною проблемою для сучасних рішень. Ефективність алгоритмів машинного навчання залежить від якості та обсягу навчальних даних, які включають як фішингові, так і легітимні приклади. Однак отримати доступ до реальних фішингових даних складно через їхню конфіденційність і швидку зміну шаблонів атак. Крім того, зловмисники можуть використовувати техніки "отруєння даних" (data poisoning), додаючи хибні приклади до навчальних наборів, що знижує точність моделей. Наприклад, у 2023 році було зафіксовано випадки, коли зловмисники навмисно створювали фальшиві фішингові листи для введення в оману захисних систем [8]. Це вимагає постійного оновлення даних і використання складних методів валідації.

Використання ШІ зловмисниками створює додаткові виклики для захисних систем. Генеративні моделі ШІ, такі як великі мовні моделі, дозволяють створювати переконливі фішингові листи, які важко відрізнити від легітимних, а також підроблені вебсайти чи deepfake-контент для вішингу. Такі атаки знижують ефективність традиційних методів, оскільки вони не містять типових

ознак фішингу, таких як орфографічні помилки чи підозрілі домени. Крім того, ШІ дозволяє зловмисникам автоматизувати створення тисяч унікальних повідомлень, що ускладнює їхнє виявлення навіть для адаптивних систем. За даними звітів, використання ШІ у фішингових атаках зросло на 30% у 2024 році [13]. Це створює потребу в розробці захисних ШІ-систем, які можуть протистояти атакуючим технологіям.

Організаційні та етичні проблеми також обмежують ефективність рішень. Наприклад, недостатня обізнаність користувачів залишається значною вразливістю, оскільки навіть найсучасніші системи не можуть повністю захистити від людських помилок. Навчання користувачів розпізнавати фішинг є ефективним, але потребує значних ресурсів і не завжди охоплює всіх працівників чи клієнтів. Крім того, впровадження ШІ-систем стикається з етичними питаннями, такими як конфіденційність даних. Аналіз поведінки користувачів чи вмісту їхніх повідомлень може викликати занепокоєння щодо порушення приватності, особливо в країнах із суворими законами про захист даних, наприклад, GDPR у Європі [22].

Технічні обмеження інфраструктури також впливають на ефективність рішень. Для роботи ШІ-моделей потрібні значні обчислювальні ресурси, що може бути проблематичним для малих організацій. Крім того, інтеграція ШІ-систем із застарілими IT-інфраструктурами часто потребує значних витрат і часу. Наприклад, багато компаній досі використовують поштові сервери, які не підтримують сучасні протоколи безпеки, такі як DMARC, що знижує ефективність захисту від спуфінгу [23].

Для систематизації проблем і обмежень існуючих рішень у боротьбі з фішингом наведено таблицю 1.1, яка підсумовує ключові аспекти, їхній вплив і можливі шляхи вирішення.

Таблиця 1.1. – Проблеми та обмеження рішень у боротьбі з фішингом

Проблема/Обмеження	Опис	Вплив	Можливі шляхи вирішення
Реактивна природа традиційних методів	Чорні списки та сигнатурний аналіз не встигають за новими атаками	Пропуск нульових атак (zero-day)	Використання ШІ для прогнозування загроз
Низька ефективність проти спірфішингу	Персоналізовані атаки не мають типових ознак	Висока ймовірність компрометації	Аналіз поведінки користувачів за допомогою ШІ
Хибнопозитивні/хибнонегативні спрацьовування	Помилкове блокування легітимних повідомлень або пропуск атак	Незручності для користувачів, зниження довіри	Покращення якості даних і моделей ШІ
Обмеженість даних для ШІ	Недостатня кількість якісних навчальних даних	Зниження точності моделей	Використання синтетичних даних і міжнародних баз

Подовження таблиці 1.1.

Використання ШІ зловмисниками	Генеративний ШІ створює переконливі атаки	Зниження ефективності всіх методів	Розробка захисних ШІ-систем для аналізу deepfake і текстів

Сучасні рішення для боротьби з фішингом, попри їхні досягнення, мають суттєві обмеження, пов’язані з реактивною природою, низькою адаптивністю до цільових атак, проблемами з даними та зростанням використання ШІ зловмисниками. Ці виклики підкреслюють необхідність розробки гібридних підходів, які поєднують традиційні методи з адаптивними ШІ-системами, а також посилення організаційних заходів, таких як навчання користувачів. Наступні розділи роботи будуть присвячені детальному аналізу ШІ-технологій і їхнього практичного застосування для подолання цих обмежень.

2 ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК

2.1 Огляд алгоритмів машинного навчання для виявлення фішингу.

Алгоритми машинного навчання, що застосовуються для виявлення фішингу, поділяються на три основні категорії: навчання з учителем (supervised learning), навчання без учителя (unsupervised learning) і навчання з підкріпленням (reinforcement learning). Найпоширенішим є навчання з учителем, яке використовує позначені набори даних із прикладами фішингових і легітимних об’єктів для створення класифікаційних моделей. Навчання без учителя застосовується для виявлення аномалій у даних без попереднього маркування, а навчання з підкріпленням використовується рідше, але має потенціал для адаптивних систем [14].

Логістична регресія (Logistic Regression) є простим, але ефективним алгоритмом для бінарної класифікації, який широко застосовується для виявлення фішингових листів і вебсайтів. Вона аналізує набір ознак, таких як довжина URL, наявність підозрілих ключових слів або структура HTML-коду, і прогнозує ймовірність того, що об’єкт є фішинговим. Логістична регресія має високу інтерпретованість і низькі обчислювальні вимоги, що робить її придатною для систем із обмеженими ресурсами. Проте вона погано справляється з нелінійними залежностями в даних, що може знижувати її ефективність проти складних атак [2].

Дерева рішень (Decision Trees) і Random Forest є більш гнучкими алгоритмами, які створюють ієрархічні структури для класифікації на основі послідовних правил. Random Forest, як ансамблевий метод, комбінує кілька дерев рішень, що підвищує точність і стійкість до перенавчання. Ці алгоритми ефективно обробляють велику кількість ознак, таких як метадані електронних листів чи поведінкові характеристики користувачів, і досягають точності до 95%

у задачах виявлення фішингу [5]. Недоліком є їхня схильність до перенавчання на невеликих наборах даних і висока обчислювальна складність.

Support Vector Machines (SVM) використовують гіперплощини для розділення класів (фішингових і легітимних об’єктів) у багатовимірному просторі. SVM особливо ефективні для задач із чітко визначеними ознаками, наприклад, аналізу URL-адрес чи тексту листів. Завдяки використанню ядерних функцій (наприклад, RBF-ядра) SVM можуть обробляти нелінійні дані, що робить їх придатними для виявлення складних фішингових атак. Дослідження показують, що SVM досягають точності до 97% у класифікації фішингових вебсайтів [8]. Однак вони потребують значних обчислювальних ресурсів і чутливі до якості даних.

Наївний байєсівський класифікатор (Naive Bayes) базується на теоремі Байєса і припускає незалежність ознак. Цей алгоритм часто застосовується для аналізу тексту електронних листів, наприклад, для виявлення підозрілих ключових слів чи шаблонів. Його перевагами є швидкість обробки та ефективність на великих наборах даних, але точність може знижуватися через припущення про незалежність ознак, що не завжди відповідає реальним даним [14]. Наївний Байєс часто використовується в поєднанні з іншими методами для підвищення точності.

Кластеризація (наприклад, K-Means) застосовується для групування об’єктів (листів, вебсайтів) за схожими характеристиками без попереднього маркування. Цей підхід корисний для виявлення нових або невідомих фішингових кампаній, оскільки дозволяє ідентифікувати аномалії в даних. Наприклад, кластеризація може виявити групу листів із подібними URL-адресами чи текстами, які відрізняються від легітимних. Недоліком є складність інтерпретації результатів і потреба в додатковій перевірці для підтвердження фішингу [15]. Точність кластеризації залежить від вибору ознак і кількості кластерів.

Автокодувальники (Autoencoders) є типом нейронних мереж, що використовуються для виявлення аномалій. Вони навчаються відтворювати

нормальні дані (наприклад, легітимні листи) і позначають об’єкти з високою помилкою відтворення як аномальні (потенційно фішингові). Автокодувальники ефективні для аналізу великих обсягів даних і можуть виявляти нові типи атак, але їхнє навчання потребує значних обчислювальних ресурсів і великих наборів даних [15].

Ансамблеві методи, такі як Gradient Boosting (наприклад, XGBoost, LightGBM), комбінують кілька слабких моделей для створення потужного класифікатора. Вони особливо ефективні для задач із великою кількістю ознак і нелінійними залежностями, що робить їх придатними для виявлення складних фішингових атак. Наприклад, XGBoost використовується для аналізу комбінації текстових, поведінкових і мережевих ознак, досягаючи точності до 98% [8]. Перевагою є висока точність і стійкість до шуму в даних, але недоліком є складність налаштування гіперпараметрів і висока обчислювальна складність.

Незважаючи на високу ефективність, алгоритми машинного навчання мають певні обмеження. По-перше, вони потребують якісних і різноманітних навчальних даних, що може бути проблематичним через швидку зміну фішингових шаблонів. По-друге, зловмисники можуть використовувати техніки ухилення (adversarial attacks), наприклад, додавати шум до фішингових листів, щоб обійти моделі ML [15]. По-третє, багато алгоритмів, таких як SVM чи Gradient Boosting, потребують значних обчислювальних ресурсів, що може ускладнювати їхнє використання в реальному часі.

Перспективи розвитку включають використання гібридних моделей, які комбінують кілька алгоритмів (наприклад, Random Forest і SVM), і впровадження методів обробки природної мови для аналізу тексту. Крім того, інтеграція ML із системами реального часу, такими як поштові фільтри чи браузери, може значно підвищити швидкість реагування на загрози [14].

Алгоритми узагальнено в порівняльній таблиці 2.1, що дозволяє наочно оцінити переваги та недоліки кожного з підходів залежно від умов застосування.

Таблиця 2.1. – Порівняння алгоритмів машинного навчання для виявлення фішингу

Алгоритм	Тип навчання	Переваги	Недоліки	Точність (приблизно)	Застосування
Логістична регресія	З учителем	Простота, швидкість, інтерпретованість	Необхідність лінійних даних	85–90%	Аналіз листів, URL
Random Forest	З учителем	Висока точність, стійкість до шуму	Схильність до перенавчання	90–95%	Класифікація листів, вебсайтів
SVM	З учителем	Ефективність для нелінійних даних	Висока обчислювальна складність	95–97%	Аналіз URL, тексту
Наївний Байєс	З учителем	Швидкість, ефективність на великих даних	Припущення незалежності ознак	80–90%	Аналіз тексту листів
K-Means	Без учителя	Виявлення аномалій, нових атак	Складність інтерпретації	70–85%	Виявлення кампаній
Автокодувальники	Без учителя	Ефективність для аномалій	Високі вимоги до ресурсів	80–90%	Аналіз великих даних

Подовження таблиці 2.1.

Gradient Boosting	З учителем	Висока точність, гнучкість	Складність налаштування	95–98%	Комплексний аналіз

Алгоритми машинного навчання, такі як Random Forest, SVM і Gradient Boosting, демонструють високу ефективність у виявленні фішингових атак, але їхня результативність залежить від якості даних і обчислювальних ресурсів. Гібридні підходи та методи навчання без учителя відкривають нові можливості для протидії новим і складним атакам. Наступні підрозділи розглянуть конкретні реалізації цих алгоритмів і їхню інтеграцію в системи захисту.

2.2 Використання нейронних мереж у розпізнаванні фішингових ресурсів

Нейронні мережі, що застосовуються для боротьби з фішингом, поділяються на кілька типів залежно від структури та задачі. Найпоширенішими є згорткові нейронні мережі (Convolutional Neural Networks, CNN), рекурентні нейронні мережі (Recurrent Neural Networks, RNN) та їхні модифікації, а також трансформери, які набули популярності завдяки обробці природної мови (NLP).

Згорткові нейронні мережі широко використовуються для аналізу візуальних і структурних елементів фішингових вебсайтів. CNN здатні обробляти зображення сторінок або їхній HTML-код, виділяючи ознаки, такі як розташування форм, дизайн елементів чи кольорова палітра. Наприклад, CNN може виявити, що фішинговий сайт імітує сторінку входу банку, але містить незначні відмінності в логотипі чи структурі. Дослідження показують, що CNN досягають точності до 99% у класифікації фішингових вебсайтів за візуальними ознаками [16]. Перевагою CNN є їхня здатність автоматично виділяти ознаки без необхідності ручного відбору, що знижує залежність від експертних знань. Однак вони потребують великих обсягів даних для навчання та значних обчислювальних ресурсів.

CNN також застосовуються для аналізу текстових даних, якщо текст представлено у вигляді зображень або матриць (наприклад, за допомогою word embeddings). Наприклад, модель може аналізувати текст електронного листа, перетворений у векторне представлення, для виявлення підозрілих шаблонів. Недоліком є висока чутливість до якості даних і можливість перенавчання на специфічних наборах [12].

Рекурентні нейронні мережі, зокрема їхня модифікація Long Short-Term Memory (LSTM), ефективні для аналізу послідовних даних, таких як текст електронних листів чи URL-адреси. RNN враховують контекст і залежності між словами чи символами, що дозволяє виявляти семантичні ознаки фішингу, наприклад, використання термінових закликів до дії ("оновіть пароль негайно"). LSTM, завдяки здатності запам’ятовувати довготривалі залежності, особливо корисні для аналізу довгих текстів або складних URL-адрес. За даними досліджень, LSTM-моделі досягають точності до 96% у класифікації фішингових листів [15].

Перевагою RNN і LSTM є їхня здатність обробляти змінну довжину вхідних даних, що робить їх придатними для аналізу різноманітних повідомлень. Однак ці моделі мають високу обчислювальну складність і можуть бути менш ефективними для коротких текстів, таких як SMS у смішингу. Крім того, навчання RNN потребує ретельного налаштування гіперпараметрів, щоб уникнути проблем із затуханням градієнта [16].

Трансформери, такі як BERT (Bidirectional Encoder Representations from Transformers), є передовими моделями для обробки природної мови і дедалі частіше застосовуються для виявлення фішингу. Вони аналізують текст двоспрямовано, враховуючи контекст усього повідомлення, що дозволяє виявляти тонкі семантичні ознаки, наприклад, приховані маніпуляції чи персоналізовані фішингові листи. Трансформери особливо ефективні проти атак, створених за допомогою генеративного ШІ, оскільки можуть розпізнавати природність тексту. Дослідження показують, що BERT досягає точності до 98% у класифікації фішингових електронних листів [20].

Перевагами трансформерів є їхня висока точність і здатність працювати з різними типами текстів (листи, повідомлення в соціальних мережах). Однак вони потребують величезних обчислювальних ресурсів і великих наборів даних для тонкого налаштування (fine-tuning). Крім того, їхня складність може ускладнювати інтеграцію в системи реального часу [20].

Гібридні моделі, які комбінують CNN, RNN або трансформери, набувають популярності для комплексного аналізу фішингових ресурсів. Наприклад, модель може використовувати CNN для аналізу візуальних елементів вебсайту і LSTM для обробки його текстового вмісту. Такі моделі дозволяють враховувати як структурні, так і семантичні ознаки, що підвищує точність виявлення. Наприклад, гібридна модель CNN-LSTM досягла точності 97% у класифікації фішингових сайтів і листів [15]. Недоліком є ще більша обчислювальна складність і потреба в ретельному налаштуванні.

Нейронні мережі мають низку переваг порівняно з традиційними методами та класичними алгоритмами машинного навчання:

— Автоматичне виділення ознак: НМ не потребують ручного відбору ознак, що знижує залежність від експертів.
— Висока точність: Завдяки обробці нелінійних залежностей НМ досягають точності до 99% у задачах класифікації фішингу.
— Адаптивність: НМ можуть перенавчатися на нових даних, що дозволяє виявляти нові типи атак [16].

Однак є й суттєві недоліки:

— Високі обчислювальні вимоги: Навчання та використання НМ потребують потужного обладнання, що може бути проблематичним для малих організацій.
— Залежність від даних: Для ефективної роботи потрібні великі й різноманітні набори даних, що ускладнює їхнє створення в умовах швидкої зміни фішингових шаблонів.

— Вразливість до атак ухилення: Зловмисники можуть використовувати adversarial attacks, додаючи шум до фішингових ресурсів, щоб обійти НМ [20].
— Складність інтерпретації: НМ часто діють як "чорна скринька", що ускладнює розуміння їхніх рішень і може викликати недовіру користувачів.

Нейронні мережі активно застосовуються в комерційних і дослідницьких системах. Наприклад, система Barracuda Sentinel використовує гібридні моделі CNN і LSTM для аналізу електронних листів і вебсайтів у реальному часі, блокуючи до 98% фішингових атак [15]. Google Safe Browsing інтегрує CNN для аналізу візуальних елементів вебсайтів, що дозволяє виявляти фішингові сторінки з точністю до 99% [12]. У дослідницькій сфері моделі BERT застосовуються для аналізу текстів фішингових листів, зокрема тих, що створені за допомогою генеративного ШІ, що забезпечує високу точність навіть проти нових атак [20].

Основними викликами є потреба в якісних даних, висока обчислювальна складність і вразливість до атак ухилення. Для подолання цих проблем дослідники пропонують використовувати синтетичні дані для навчання, оптимізувати моделі для роботи на менш потужному обладнанні та розробляти методи захисту від adversarial attacks [16]. Перспективи включають інтеграцію НМ із системами реального часу, такими як поштові фільтри чи браузери, і розробку гібридних моделей, які комбінують різні типи НМ для комплексного аналізу.

2.3 Аналіз поведінки користувачів як метод протидії фішингу

Аналіз поведінки користувачів передбачає створення базового профілю поведінки для кожного користувача або групи користувачів на основі історичних даних. Цей профіль включає такі параметри, як частота входів у систему, геолокація, типові шаблони листування, час активності, використовувані пристрої та програми. ШІ-моделі, зокрема алгоритми машинного навчання та

глибокого навчання, аналізують ці дані в реальному часі, щоб виявити відхилення, які можуть вказувати на фішингову атаку або компрометацію облікового запису [4].

Наприклад, якщо користувач зазвичай входить у корпоративну систему з офісу в Україні, але раптово з’являється вхід із іншої країни, система може позначити це як підозрілу активність. Аналогічно, якщо працівник отримує лист із незвичайним закликом до дії (наприклад, переказати кошти) від "колеги", але стиль листування відрізняється від типового, ШІ може ідентифікувати це як потенційний спірфішинг. Такі методи особливо ефективні проти цільових атак, які важко виявити традиційними засобами [17].

Аналіз поведінки користувачів для протидії фішингу базується на кількох ключових підходах, які використовують ШІ:

1) Виявлення аномалій (Anomaly Detection): Алгоритми навчання без учителя, такі як автокодувальники або кластеризація, створюють модель нормальної поведінки користувача і позначають відхилення як потенційні загрози. Наприклад, автокодувальники можуть виявити незвичайний шаблон введення даних на вебсайті, що може свідчити про взаємодію з фішинговою сторінкою. Дослідження показують, що такі методи досягають точності до 90% у виявленні аномалій, пов’язаних із фішингом [15].
2) Аналіз шаблонів листування: Методи обробки природної мови (NLP), зокрема моделі на основі трансформерів (наприклад, BERT), аналізують стиль, тон і зміст електронних листів, щоб визначити, чи відповідають вони типовим шаблонам користувача чи організації. Наприклад, якщо лист від "керівника" містить нетипові фрази чи синтаксис, система може позначити його як фішинговий. Такі моделі ефективні проти атак, створених за допомогою генеративного ШІ, і досягають точності до 95% [20].
3) Моніторинг поведінки в реальному часі: Алгоритми машинного навчання, такі як Gradient Boosting або рекурентні нейронні мережі

(RNN), аналізують поведінку користувача в реальному часі, наприклад, швидкість набору тексту, рухи миші чи послідовність дій на вебсайті. Якщо користувач вводить дані на фішинговому сайті повільніше або з ваганнями, це може бути позначено як аномалія. Цей підхід особливо корисний для виявлення атак, які імітують легітимні ресурси [4].

4) Контекстний аналіз: ШІ-моделі враховують контекст дій користувача, наприклад, час доби, тип пристрою чи мережеве середовище. Наприклад, якщо працівник відкриває підозріле посилання в неробочий час із особистого пристрою, система може ініціювати додаткову перевірку. Контекстний аналіз підвищує точність виявлення цільових атак, таких як спірфішинг [17].

Аналіз поведінки користувачів має низку переваг порівняно з традиційними методами виявлення фішингу:

— Ефективність проти цільових атак: UBA дозволяє виявляти спірфішинг і whaling, які не містять явних ознак, таких як підозрілі URL чи орфографічні помилки, завдяки аналізу поведінки та контексту [20].

— Проактивний підхід: На відміну від реактивних методів, таких як чорні списки, UBA може виявляти нові атаки, базуючись на аномаліях, а не на відомих шаблонах.

— Адаптивність: ШІ-моделі постійно оновлюють профілі поведінки, що дозволяє адаптуватися до змін у звичках користувачів або нових технік зловмисників [4].

— Зниження залежності від сигнатур: UBA не покладається на попередньо визначені сигнатури чи списки, що робить його ефективним проти нульових атак (zero-day attacks).

Незважаючи на переваги, аналіз поведінки користувачів має певні обмеження:

— Високий рівень хибнопозитивних спрацьовувань: Незвичайна, але легітимна поведінка (наприклад, вхід із нового пристрою під час

відрядження) може бути помилково позначена як аномалія, що створює незручності для користувачів [15].

— Потреба в даних: Для створення точних профілів поведінки потрібні великі обсяги історичних даних, що може бути проблематичним для нових користувачів або організацій із обмеженими ресурсами.
— Етичні та правові питання: Моніторинг поведінки, особливо в реальному часі, може викликати занепокоєння щодо конфіденційності, особливо в країнах із суворими законами про захист даних, таких як GDPR [17].
— Обчислювальна складність: Аналіз великих обсягів даних у реальному часі вимагає потужних обчислювальних ресурсів, що може бути недоступним для малих компаній.

Аналіз поведінки користувачів активно застосовується в комерційних системах кібербезпеки. Наприклад, Microsoft Azure Sentinel використовує алгоритми машинного навчання для моніторингу поведінки користувачів у корпоративних мережах, виявляючи аномалії, пов’язані з фішингом, із точністю до 92% [4]. Система Cisco Secure Network Analytics застосовує автокодувальники для аналізу мережевої активності, що дозволяє ідентифікувати компрометовані облікові записи після фішингових атак. У дослідницькій сфері методи NLP, такі як BERT, використовуються для аналізу листування в організаціях, що допомагає виявляти спірфішинг із точністю до 95% [20].

Перспективи аналізу поведінки користувачів включають інтеграцію з іншими ШІ-технологіями, такими як нейронні мережі та обробка природної мови, для підвищення точності. Використання синтетичних даних може допомогти вирішити проблему обмеженості історичних даних. Крім того, розробка менш інвазивних методів моніторингу, які відповідають вимогам конфіденційності, сприятиме ширшому впровадженню UBA. Інтеграція UBA з системами реального часу, такими як поштові фільтри чи браузери, також є перспективним напрямом [15].

2.4 Автоматизація обробки великих масивів даних для виявлення загроз.

Автоматизація обробки великих масивів даних для виявлення фішингу базується на інтеграції ШІ-технологій із платформами для управління великими даними, такими як Hadoop, Apache Spark або хмарні сервіси (AWS, Azure).

Основні етапи цього процесу включають:

1) Збір даних: Об’єднання різнорідних джерел, таких як електронні листи, логи вебсайтів, мережевий трафік, повідомлення в соціальних мережах і поведінкові дані користувачів.
2) Попередня обробка: Очищення даних, нормалізація, видалення шуму та перетворення в структурований формат (наприклад, векторне представлення тексту).
3) Аналіз і класифікація: Використання алгоритмів ML для виявлення фішингових ознак, кластеризації схожих об’єктів або прогнозування загроз.
4) Реагування: Автоматичне блокування підозрілих об’єктів, сповіщення адміністраторів або ізоляція скомпрометованих систем [30].

ШІ відіграє центральну роль на етапах аналізу та класифікації, дозволяючи обробляти терабайти даних за секунди та виявляти аномалії, які неможливо ідентифікувати вручну. Наприклад, аналіз мільйонів електронних листів щодня для виявлення фішингових кампаній є стандартною задачею для автоматизованих систем.

Для обробки великих масивів даних у контексті виявлення фішингу використовуються кілька ключових технологій ШІ та Big Data:

1) Алгоритми машинного навчання: Ансамблеві методи, такі як Gradient Boosting (XGBoost, LightGBM), і нейронні мережі (зокрема CNN і LSTM) застосовуються для класифікації великих наборів даних. Наприклад, XGBoost може обробляти мільйони URL-адрес для виявлення фішингових сайтів із точністю до 98% [8]. Ці алгоритми

ефективні для аналізу структурованих даних, таких як метадані листів чи логи мережі.

2) Обробка природної мови (NLP): Моделі NLP, такі як BERT або трансформери, використовуються для аналізу текстового вмісту листів, повідомлень у соціальних мережах або вебсайтів. Вони дозволяють виявляти семантичні ознаки фішингу, наприклад, маніпулятивні заклики до дії, навіть у персоналізованих атаках. Дослідження показують, що BERT досягає точності до 97% у класифікації фішингових листів [20].
3) Кластеризація та виявлення аномалій: Алгоритми навчання без учителя, такі як K-Means або DBSCAN, групують схожі об’єкти (наприклад, листи з однаковими шаблонами) і виявляють аномалії, які можуть вказувати на нові фішингові кампанії. Наприклад, кластеризація може ідентифікувати групу листів із подібними URL-адресами, що не відповідають легітимним шаблонам [15]. Цей підхід особливо корисний для обробки неструктурованих даних.
4) Потокова обробка даних: Технології, такі як Apache Kafka або Apache Flink, дозволяють аналізувати дані в реальному часі, що критично для оперативного реагування на фішинг. Наприклад, потокова обробка може виявити масову розсилку фішингових листів за лічені секунди, дозволяючи заблокувати їх до того, як користувачі відкриють повідомлення [30].
5) Хмарні платформи: Хмарні сервіси, такі як AWS SageMaker або Google Cloud AI, забезпечують інфраструктуру для масштабованої обробки даних і навчання моделей ШІ. Вони дозволяють організаціям обробляти петабайти даних без необхідності створення власних серверів, що знижує витрати [13].

Автоматизація обробки великих масивів даних має низку переваг для виявлення фішингових загроз:

— Швидкість і масштабність: Системи ШІ можуть аналізувати мільйони об’єктів (листів, URL, логів) за секунди, що неможливо для ручної обробки. Наприклад, Google Safe Browsing обробляє мільярди URL-адрес щодня для виявлення фішингових сайтів [12].
— Виявлення нових загроз: Алгоритми кластеризації та виявлення аномалій дозволяють ідентифікувати нульові атаки (zero-day attacks), які не внесено до сигнатурних баз [15].
— Висока точність: Комбінація ML і NLP забезпечує точність до 95–98% у класифікації фішингових ресурсів, що значно перевищує традиційні методи [20].
— Автоматизація реагування: Системи можуть автоматично блокувати підозрілі об’єкти, відправляти сповіщення або ізолювати скомпрометовані системи, що знижує час реакції на загрози.

Незважаючи на переваги, автоматизація обробки великих масивів даних має певні обмеження:

— Високі вимоги до ресурсів: Обробка великих даних і навчання моделей ШІ потребують потужних обчислювальних ресурсів, що може бути недоступним для малих організацій [13].
— Якість даних: Неструктуровані або шумні дані можуть знижувати точність моделей. Наприклад, фішингові листи, створені генеративним ШІ, можуть не містити типових ознак, що ускладнює їхню класифікацію [20].
— Хибнопозитивні спрацьовування: Автоматизовані системи можуть помилково позначати легітимні об’єкти як фішингові, що створює незручності для користувачів. Дослідження показують, що хибнопозитивні результати можуть досягати 5–10% у деяких системах [15].
— Вразливість до атак ухилення: Зловмисники можуть використовувати adversarial attacks, додаючи шум до фішингових ресурсів, щоб обійти моделі ШІ.

Автоматизація обробки великих масивів даних широко застосовується в комерційних і дослідницьких системах. Наприклад, Barracuda Sentinel використовує Apache Spark і алгоритми Gradient Boosting для аналізу мільйонів електронних листів щодня, виявляючи фішингові кампанії з точністю до 98% [8]. Microsoft Azure Sentinel інтегрує потокову обробку даних із моделями NLP для моніторингу мережевих логів і листування, що дозволяє виявляти спірфішинг у реальному часі [13]. У дослідницькій сфері кластеризація застосовується для аналізу великих наборів URL-адрес, що допомагає ідентифікувати нові фішингові кампанії з точністю до 90% [15].

Перспективи автоматизації включають використання гібридних моделей, які комбінують ML, NLP і нейронні мережі для комплексного аналізу. Розвиток технологій потокової обробки дозволить ще швидше реагувати на загрози. Крім того, впровадження федеративного навчання (federated learning) може вирішити проблему конфіденційності даних, дозволяючи навчати моделі без централізованого збору інформації. Інтеграція автоматизованих систем із хмарними платформами також сприятиме їхньому масштабуванню.

2.5 Інтеграція ШІ з існуючими системами кібербезпеки.

Інтеграція ШІ з системами кібербезпеки передбачає вбудовування алгоритмів машинного навчання (ML), нейронних мереж і методів обробки природної мови (NLP) у наявні інфраструктури, такі як поштові фільтри, системи захисту мережі, антивіруси чи платформи SIEM (Security Information and Event Management). Основні принципи цього процесу включають:

1) Модульність: ШІ-компоненти додаються як окремі модулі, які взаємодіють із традиційними системами через API або плагіни.
2) Інтероперабельність: Забезпечення сумісності ШІ-моделей із різними платформами, протоколами та форматами даних.
3) Реальний час: Інтеграція передбачає обробку даних у реальному часі для оперативного реагування на фішингові атаки.

4) Автоматизація: ШІ автоматизує аналіз, класифікацію та реагування, зменшуючи навантаження на адміністраторів.

Наприклад, ШІ-модель може бути інтегрована в поштовий сервер для аналізу вхідних листів, доповнюючи традиційні фільтри на основі правил можливостями виявлення аномалій або семантичного аналізу тексту.

Інтеграція ШІ з системами кібербезпеки для протидії фішингу здійснюється через кілька ключових підходів:

1) Інтеграція з поштовими системами: ШІ-моделі, такі як Random Forest або трансформери (наприклад, BERT), вбудовуються в поштові сервери (Microsoft Exchange, Gmail) для аналізу листів у реальному часі. Вони перевіряють метадані, текст і вкладення, виявляючи фішинг із точністю до 97% [20]. Наприклад, модель NLP може ідентифікувати персоналізовані спірфішинг-листи, які не містять типових ознак.
2) Доповнення систем SIEM: Платформи SIEM, такі як Splunk або IBM QRadar, інтегрують ШІ для аналізу логів і подій безпеки. Алгоритми ML, наприклад, Gradient Boosting, обробляють великі обсяги даних, виявляючи аномалії, пов’язані з фішингом, наприклад, нетипові входи в систему після переходу за шкідливим посиланням. Такі системи підвищують точність до 95% порівняно з традиційними методами [4].
3) Інтеграція з вебфільтрами та браузерами: ШІ-моделі, зокрема згорткові нейронні мережі (CNN), вбудовуються в системи вебфільтрації (наприклад, Cisco Umbrella) або браузери (Google Safe Browsing) для аналізу URL-адрес і вмісту вебсайтів. Вони виявляють фішингові сайти з точністю до 99%, аналізуючи візуальні та структурні елементи [12].
4) Інтеграція з системами захисту кінцевих точок: Антивірусні програми, такі як Norton чи Kaspersky, доповнюються ШІ-моделями для аналізу поведінки користувачів і виявлення шкідливих вкладень. Наприклад, автокодувальники можуть ідентифікувати аномалії в діях користувача, такі як введення даних на фішинговому сайті, із точністю до 90% [15].

5) Хмарна інтеграція: Хмарні платформи, такі як AWS SageMaker або Microsoft Azure Sentinel, дозволяють розгортати ШІ-моделі для масштабованої обробки даних. Вони інтегруються з корпоративними системами через API, забезпечуючи аналіз великих масивів даних і автоматизоване реагування на фішинг.

Інтеграція ШІ з існуючими системами кібербезпеки має низку переваг:

— Підвищення точності: ШІ доповнює традиційні методи, знижуючи кількість хибнонегативних спрацьовувань. Наприклад, комбінація чорних списків із моделями ML підвищує точність виявлення фішингу до 98% [20].

— Автоматизація та швидкість: ШІ автоматизує аналіз і реагування, скорочуючи час від виявлення до блокування загрози до кількох секунд.

— Адаптивність: Моделі ШІ перенавчаються на нових даних, що дозволяє виявляти нульові атаки та протистояти новим технікам зловмисників [12].

— Зниження навантаження на персонал: Автоматизація зменшує потребу в ручному аналізі, дозволяючи адміністраторам зосередитися на критичних задачах [4].

Інтеграція ШІ з системами кібербезпеки пов’язана з певними викликами:

— Сумісність із застарілими системами: Багато організацій використовують старі IT-інфраструктури, які не підтримують сучасні ШІ-моделі, що потребує значних витрат на модернізацію [15].

— Високі витрати: Розгортання ШІ, особливо в хмарних середовищах, і навчання моделей потребують значних фінансових і обчислювальних ресурсів [31].

— Конфіденційність даних: Аналіз поведінки користувачів чи вмісту листів може порушувати вимоги щодо захисту даних, наприклад, GDPR, що вимагає додаткових заходів для забезпечення приватності [20].

— Вразливість до атак ухилення: Зловмисники можуть використовувати adversarial attacks, додаючи шум до фішингових ресурсів, щоб обійти ШІ-моделі [4].

Інтеграція ШІ з системами кібербезпеки активно застосовується в комерційних продуктах. Наприклад, Microsoft Azure Sentinel інтегрує моделі Gradient Boosting і NLP для аналізу логів і листування, виявляючи фішинг із точністю до 95% [4]. Barracuda Sentinel використовує гібридні моделі CNN і LSTM, інтегровані з поштовими серверами, для блокування фішингових листів у реальному часі з точністю до 98% [8]. Google Safe Browsing застосовує CNN для аналізу вебсайтів, інтегруючись із браузерами Chrome і Firefox, що дозволяє виявляти фішингові сайти з точністю до 99% [12]. У дослідницькій сфері ШІ-моделі інтегруються з платформами SIEM для аналізу великих масивів даних, що допомагає виявляти складні спірфішинг-атаки.

Перспективи інтеграції ШІ включають розробку стандартизованих API для спрощення взаємодії з різними системами, використання федеративного навчання для забезпечення конфіденційності даних і створення гібридних систем, які поєднують ШІ з традиційними методами. Розвиток хмарних технологій також сприятиме масштабуванню ШІ-рішень, роблячи їх доступними для малих організацій. Крім того, впровадження методів захисту від adversarial attacks підвищить стійкість систем до атак ухилення.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ШІ У ЗАХИСТІ ВІД ФІШИНГУ

3.1 Розробка моделі ШІ для виявлення фішингових атак

Розробка моделі ШІ для виявлення фішингових атак включає кілька основних етапів:

1) Визначення задачі та вибір алгоритму: Задача полягає в бінарній класифікації об’єктів (фішинговий або легітимний). Для цього використовуються алгоритми машинного навчання (ML) та глибокого навчання (DL), зокрема ансамблеві методи та нейронні мережі.
2) Збір і підготовка даних: Створення набору даних із позначеними прикладами фішингових і легітимних об’єктів.
3) Розробка архітектури моделі: Вибір структури моделі, налаштування гіперпараметрів і визначення ознак для аналізу.
4) Навчання та тестування: Навчання моделі на тренувальних даних і оцінка її ефективності на тестових наборах.
5) Оптимізація та інтеграція: Удосконалення моделі для роботи в реальному часі та її інтеграція з системами кібербезпеки.

Для розробки моделі обрано гібридний підхід, що комбінує ансамблевий метод Gradient Boosting (XGBoost) і нейронну мережу Long Short-Term Memory (LSTM). Вибір обґрунтовано наступними факторами:

— XGBoost: Ефективний для обробки структурованих даних, таких як метадані листів (адреса відправника, довжина URL) і числові ознаки. Забезпечує високу точність (до 98%) і стійкість до шуму в даних [8].
— LSTM: Підходить для аналізу послідовних даних, зокрема тексту листів чи URL-адрес, завдяки здатності враховувати довготривалі залежності. Досягає точності до 96% у задачах класифікації фішингових текстів [15].

— Гібридний підхід дозволяє поєднати сильні сторони обох методів: XGBoost обробляє числові та категоріальні ознаки, а LSTM аналізує текстові дані, що підвищує загальну точність.

Для навчання моделі використано набір даних, що включає:

— Фішингові об’єкти: 10 000 прикладів фішингових електронних листів і URL-адрес із відкритих джерел, таких як PhishTank і Kaggle. Дані містять приклади масового фішингу та спірфішингу.

— Легітимні об’єкти: 10 000 прикладів легітимних листів і URL-адрес, зібраних із відкритих репозиторіїв і корпоративних систем (з дотриманням конфіденційності).

— Ознаки: Набір із 50 ознак, включаючи довжину URL, наявність підозрілих символів, кількість перенаправлень, ключові слова в тексті, метадані листів (SPF, DKIM) і поведінкові характеристики (час відправлення, геолокація).

Дані пройшли попередню обробку:

— Очищення: Видалення дублікатів, пропущених значень і нерелевантних символів.

— Нормалізація: Перетворення числових ознак у діапазон [0, 1] і токенізація тексту для LSTM.

— Векторізація тексту: Використання word embeddings (наприклад, Word2Vec) для представлення тексту листів у числовому форматі.

— Розподіл даних: 70% для навчання, 20% для валідації, 10% для тестування [20].

Модель складається з двох основних компонентів:

1) XGBoost-компонент:

— Ознаки: Числові та категоріальні (довжина URL, кількість доменів, метадані листів).

— Гіперпараметри: Кількість дерев = 100, глибина = 7, learning rate = 0.1.

— Функція втрат: Бінарна логістична втрата для класифікації.

2) LSTM-компонент:

— Шари: Вхідний шар (word embeddings), два LSTM-шари (128 нейронів кожен), повнозв’язний шар із сигмоїдною активацією.

— Ознаки: Текстові дані (тіло листа, заголовки, URL).

— Гіперпараметри: Розмір embedding = 300, dropout = 0.2 для уникнення перенавчання.

3) Гібридна інтеграція: Результати XGBoost і LSTM об’єднуються через повнозв’язний шар, який видає остаточну ймовірність фішингу (0 – легітимний, 1 – фішинговий) [15].

Архітектура моделі дозволяє обробляти як структуровані, так і неструктуровані дані, що підвищує її універсальність для різних типів фішингових атак.

Модель реалізовано за допомогою мови програмування Python і бібліотек:

— Pandas і NumPy: Для обробки даних.

— Scikit-learn і XGBoost: Для реалізації XGBoost-компонента.

— TensorFlow/Keras: Для створення та навчання LSTM-моделі.

— NLTK і Gensim: Для обробки тексту та створення word embeddings.

Навчання проводилося на сервері з GPU (NVIDIA Tesla V100) для прискорення обробки LSTM. Процес навчання тривав 10 епох із розміром пакета (batch size) 64. Для оцінки якості використовувалися метрики: точність (accuracy), точність передбачення (precision), повнота (recall) і F1-Score.

Попереднє тестування моделі на валідаційному наборі показало:

— Точність (accuracy): 97.5%.

— Precision: 96.8% (висока здатність уникати хибнопозитивних спрацьовувань).

— Recall: 97.2% (ефективність у виявленні фішингових об’єктів).

— F1-Score: 97.0% (збалансованість між precision і recall) [8].

Ці результати свідчать про високу ефективність гібридної моделі порівняно з окремими алгоритмами, такими як Random Forest (95%) чи чиста

LSTM (96%) [15]. Модель також показала хорошу адаптивність до нових даних, що підтверджує її потенціал для виявлення нульових атак.

Під час розробки моделі було виявлено кілька викликів:

— Якість даних: Неструктуровані текстові дані потребували ретельної попередньої обробки, щоб уникнути шуму.
— Обчислювальні ресурси: Навчання LSTM-компонента вимагало потужного обладнання, що може бути обмеженням для малих організацій.
— Вразливість до adversarial attacks: Зловмисники можуть додавати шум до фішингових листів, що знижує точність моделі [20].
— Хибнопозитивні спрацьовування: Деякі легітимні листи з нестандартним форматом позначалися як фішингові, що потребує подальшої оптимізації.

Модель може бути інтегрована в поштові сервери, вебфільтри або системи SIEM для реального часу аналізу. Для підвищення ефективності планується:

— Використання синтетичних даних для розширення навчального набору.
— Впровадження методів захисту від adversarial attacks, таких як adversarial training.
— Оптимізація моделі для роботи на менш потужному обладнанні шляхом зменшення розміру LSTM-шарів [15].

3.2 Вибір та підготовка даних для навчання моделі

Для навчання моделі, яка поєднує XGBoost і LSTM (див. підрозділ 3.1), необхідно забезпечити набір даних, що охоплює як фішингові, так і легітимні об’єкти, зокрема електронні листи та URL-адреси. Основна мета — створити збалансований набір, який відображає реальні сценарії фішингових атак, включаючи масовий фішинг, спірфішинг і сучасні атаки, створені за допомогою генеративного ШІ.

Дані були зібрані з кількох джерел, щоб забезпечити різноманітність і репрезентативність. Першим джерелом став відкритий репозиторій PhishTank,

який містить перевірені фішингові URL-адреси та пов’язані з ними метадані. Другим джерелом були набори даних із платформи Kaggle, що включають як фішингові, так і легітимні електронні листи з різних кампаній. Для легітимних даних використано анонімізовані листи з відкритих корпоративних наборів, отриманих із дотриманням вимог конфіденційності. Загалом зібрано 20 000 об’єктів: 10 000 фішингових (листи та URL) і 10 000 легітимних, що забезпечує баланс класів і знижує ризик зміщення моделі.

Окремий акцент зроблено на включенні прикладів спірфішингу, які імітують персоналізовані атаки. Для цього використано синтетично згенеровані листи, створені за допомогою інструментів NLP, таких як GPT, із подальшою ручною перевіркою. Це дозволило моделі навчитися розпізнавати атаки, які не містять типових ознак, таких як орфографічні помилки чи підозрілі домени [20].

Набір даних охоплює широкий спектр ознак, які відображають як структурні, так і семантичні аспекти фішингових атак. До ключових ознак належать:

— Числові ознаки: Довжина URL, кількість перенаправлень, кількість символів у тексті листа, час відправлення.
— Категоріальні ознаки: Наявність протоколів безпеки (SPF, DKIM, DMARC), тип домену (наприклад, .com, .org), наявність підозрілих символів (наприклад, @, %).
— Текстові ознаки: Вміст листа, заголовки, текст URL, ключові слова (наприклад, "терміново", "пароль").
— Поведінкові ознаки: Геолокація відправника, частота листування, тип пристрою.

Ці ознаки дозволяють моделі аналізувати як технічні аспекти (наприклад, структуру URL), так і контекстні (наприклад, маніпулятивний текст), що підвищує її універсальність.

Підготовка даних є критично важливим етапом, оскільки необроблені дані часто містять шум, пропущені значення або невідповідності, які можуть знизити

точність моделі. Процес підготовки включав кілька кроків, виконаних за допомогою Python і бібліотек Pandas, NLTK та Gensim.

Спочатку проведено очищення даних. Видалено дублікати, пропущені значення та нерелевантні символи (наприклад, спеціальні символи в URL, які не впливають на класифікацію). Текстові дані, такі як тіло листів, очищено від HTML-тегів і форматування, щоб зберегти лише семантичний вміст. Для числових ознак, таких як довжина URL, застосовано перевірку на аномалії, виключивши екстремальні значення (наприклад, URL довжиною понад 1000 символів).

Далі виконано нормалізацію та векторизацію. Числові ознаки нормалізовано до діапазону [0, 1] за допомогою Min-Max масштабування, що забезпечує однаковий вплив усіх ознак на модель XGBoost. Текстові дані токенізовано (розбито на слова) і перетворено у векторне представлення за допомогою word embeddings, зокрема Word2Vec, що дозволило представити слова у вигляді числових векторів розміром 300. Це критично для LSTM-компонента моделі, який обробляє послідовні текстові дані [20]. Категоріальні ознаки, такі як тип домену, закодовано за допомогою one-hot encoding, щоб зробити їх придатними для аналізу.

Для забезпечення репрезентативності набір даних розділено на три частини: 70% для навчання (14 000 об’єктів), 20% для валідації (4000 об’єктів) і 10% для тестування (2000 об’єктів). Розподіл виконано стратифіковано, щоб зберегти баланс між класами (фішинг/легітимний) у кожній частині. Щоб уникнути перенавчання, застосовано техніку data augmentation для текстових даних, генеруючи додаткові приклади шляхом заміни синонімів або переформулювання речень за допомогою NLP-інструментів.

Формування навчального набору супроводжувалося кількома викликами. По-перше, обмежена доступність реальних даних спірфішингу через їхню конфіденційність змусила використовувати синтетичні дані, що потребувало додаткової перевірки якості. По-друге, різнорідність форматів даних (текст, метадані, логи) ускладнила їхню уніфікацію. Наприклад, листи з різних джерел

мали різні кодування чи структури, що вимагало додаткової обробки. По-третє, ризик зміщення даних (data bias) був значним, оскільки деякі набори містили переважно масовий фішинг, що могло знизити здатність моделі виявляти цільові атаки.

Для вирішення цих проблем використано кілька стратегій. Синтетичні дані перевірялися експертами для забезпечення їхньої реалістичності. Різнорідність форматів усунуто шляхом створення єдиного шаблону даних із використанням бібліотеки Pandas. Для зменшення зміщення застосовано техніки oversampling для менш представлених класів (наприклад, спірфішинг) і включення різноманітних джерел даних [20].

Якісно підготовлений набір даних є основою для ефективної роботи моделі ШІ. Збалансований набір із 20 000 об’єктів, що охоплює широкий спектр ознак, дозволяє моделі навчитися розпізнавати як прості, так і складні фішингові атаки. Попередня обробка, нормалізація та векторизація забезпечують сумісність даних із гібридною архітектурою моделі (XGBoost + LSTM). Дослідження показують, що якісна підготовка даних може підвищити точність моделі на 5–10% порівняно з необробленими наборами.

3.3 Тестування моделі на реальних та синтетичних наборах даних

Для навчання моделі, яка поєднує XGBoost і LSTM (див. підрозділ 3.1), необхідно забезпечити набір даних, що охоплює як фішингові, так і легітимні об’єкти, зокрема електронні листи та URL-адреси. Основна мета — створити збалансований набір, який відображає реальні сценарії фішингових атак, включаючи масовий фішинг, спірфішинг і сучасні атаки, створені за допомогою генеративного ШІ.

Дані були зібрані з кількох джерел, щоб забезпечити різноманітність і репрезентативність. Першим джерелом став відкритий репозиторій PhishTank, який містить перевірені фішингові URL-адреси та пов’язані з ними метадані. Другим джерелом були набори даних із платформи Kaggle, що включають як фішингові, так і легітимні електронні листи з різних кампаній. Для легітимних

даних використано анонімізовані листи з відкритих корпоративних наборів, отриманих із дотриманням вимог конфіденційності. Загалом зібрано 20 000 об’єктів: 10 000 фішингових (листи та URL) і 10 000 легітимних, що забезпечує баланс класів і знижує ризик зміщення моделі.

Окремий акцент зроблено на включенні прикладів спірфішингу, які імітують персоналізовані атаки. Для цього використано синтетично згенеровані листи, створені за допомогою інструментів NLP, таких як GPT, із подальшою ручною перевіркою. Це дозволило моделі навчитися розпізнавати атаки, які не містять типових ознак, таких як орфографічні помилки чи підозрілі домени [20].

Набір даних охоплює широкий спектр ознак, які відображають як структурні, так і семантичні аспекти фішингових атак. До ключових ознак належать:

— Числові ознаки: Довжина URL, кількість перенаправлень, кількість символів у тексті листа, час відправлення.
— Категоріальні ознаки: Наявність протоколів безпеки (SPF, DKIM, DMARC), тип домену (наприклад, .com, .org), наявність підозрілих символів (наприклад, @, %).
— Текстові ознаки: Вміст листа, заголовки, текст URL, ключові слова (наприклад, "терміново", "пароль").
— Поведінкові ознаки: Геолокація відправника, частота листування, тип пристрою.

Ці ознаки дозволяють моделі аналізувати як технічні аспекти (наприклад, структуру URL), так і контекстні (наприклад, маніпулятивний текст), що підвищує її універсальність.

Підготовка даних є критично важливим етапом, оскільки необроблені дані часто містять шум, пропущені значення або невідповідності, які можуть знизити точність моделі. Процес підготовки включав кілька кроків, виконаних за допомогою Python і бібліотек Pandas, NLTK та Gensim.

Спочатку проведено очищення даних. Видалено дублікати, пропущені значення та нерелевантні символи (наприклад, спеціальні символи в URL, які не

впливають на класифікацію). Текстові дані, такі як тіло листів, очищено від HTML-тегів і форматування, щоб зберегти лише семантичний вміст. Для числових ознак, таких як довжина URL, застосовано перевірку на аномалії, виключивши екстремальні значення (наприклад, URL довжиною понад 1000 символів).

Далі виконано нормалізацію та векторизацію. Числові ознаки нормалізовано до діапазону [0, 1] за допомогою Min-Max масштабування, що забезпечує однаковий вплив усіх ознак на модель XGBoost. Текстові дані токенізовано (розбито на слова) і перетворено у векторне представлення за допомогою word embeddings, зокрема Word2Vec, що дозволило представити слова у вигляді числових векторів розміром 300. Це критично для LSTM-компонента моделі, який обробляє послідовні текстові дані [20]. Категоріальні ознаки, такі як тип домену, закодовано за допомогою one-hot encoding, щоб зробити їх придатними для аналізу.

Для забезпечення репрезентативності набір даних розділено на три частини: 70% для навчання (14 000 об’єктів), 20% для валідації (4000 об’єктів) і 10% для тестування (2000 об’єктів). Розподіл виконано стратифіковано, щоб зберегти баланс між класами (фішинг/легітимний) у кожній частині. Щоб уникнути перенавчання, застосовано техніку data augmentation для текстових даних, генеруючи додаткові приклади шляхом заміни синонімів або переформулювання речень за допомогою NLP-інструментів.

Формування навчального набору супроводжувалося кількома викликами. По-перше, обмежена доступність реальних даних спірфішингу через їхню конфіденційність змусила використовувати синтетичні дані, що потребувало додаткової перевірки якості. По-друге, різнорідність форматів даних (текст, метадані, логи) ускладнила їхню уніфікацію. Наприклад, листи з різних джерел мали різні кодування чи структури, що вимагало додаткової обробки. По-третє, ризик зміщення даних (data bias) був значним, оскільки деякі набори містили переважно масовий фішинг, що могло знизити здатність моделі виявляти цільові атаки.

Для вирішення цих проблем використано кілька стратегій. Синтетичні дані перевірялися експертами для забезпечення їхньої реалістичності. Різнорідність форматів усунуто шляхом створення єдиного шаблону даних із використанням бібліотеки Pandas. Для зменшення зміщення застосовано техніки oversampling для менш представлених класів (наприклад, спірфішинг) і включення різноманітних джерел даних [20].

Якісно підготовлений набір даних є основою для ефективної роботи моделі ШІ. Збалансований набір із 20 000 об’єктів, що охоплює широкий спектр ознак, дозволяє моделі навчитися розпізнавати як прості, так і складні фішингові атаки. Попередня обробка, нормалізація та векторизація забезпечують сумісність даних із гібридною архітектурою моделі (XGBoost + LSTM). Дослідження показують, що якісна підготовка даних може підвищити точність моделі на 5–10% порівняно з необробленими наборами.

3.4 Оцінка ефективності запропонованої моделі (метрики та порівняння)

Для оцінки ефективності моделі використано чотири основні метрики, які є стандартними для задач бінарної класифікації:

- Точність (Accuracy): Частка правильно класифікованих об’єктів (фішингових і легітимних) від загальної кількості. Висока точність свідчить про загальну ефективність моделі.
- Точність передбачення (Precision): Частка правильно ідентифікованих фішингових об’єктів серед усіх позначених як фішингові. Висока precision знижує кількість хибнопозитивних спрацьовувань.
- Повнота (Recall): Частка фішингових об’єктів, виявлених моделлю, серед усіх реальних фішингових об’єктів. Висока recall вказує на здатність моделі не пропускати загрози.
- F1-Score: Гармонійне середнє між precision і recall, що відображає збалансованість моделі між точністю і повнотою.

Додатково оцінювався час обробки одного об’єкта (листа або URL), щоб визначити придатність моделі для роботи в реальному часі. Для порівняння

ефективності також використовувалася матриця помилок (confusion matrix), яка показує кількість правильно та неправильно класифікованих об’єктів.

Як зазначено в підрозділі 3.3, модель тестувалася на двох наборах даних: реальному (2000 об’єктів) і синтетичному (2000 об’єктів). Результати оцінки наведено нижче.

На реальному наборі даних, що включав масовий фішинг, спірфішинг і легітимні об’єкти, модель показала такі результати:

— Accuracy: 97.8% (1956 із 2000 об’єктів класифіковано правильно).

— Precision: 97.2% (низький рівень хибнопозитивних спрацьовувань).

— Recall: 98.0% (висока здатність виявляти фішингові об’єкти).

— F1-Score: 97.6% [8].

Матриця помилок показала, що з 1000 фішингових об’єктів 980 було правильно класифіковано, а 20 — пропущено (хибнонегативні). Із 1000 легітимних об’єктів 976 класифіковано правильно, а 24 позначено як фішингові (хибнопозитивні). Час обробки одного об’єкта склав у середньому 0.02 секунди, що підтверджує придатність моделі для реального часу.

На синтетичному наборі, що моделював складні атаки, зокрема створені генеративним ШІ, результати були дещо нижчими:

— Accuracy: 95.5% (1910 із 2000 об’єктів класифіковано правильно).

— Precision: 94.8%.

— Recall: 95.6%.

— F1-Score: 95.2% [20].

Матриця помилок показала, що з 1000 фішингових об’єктів 956 було правильно класифіковано, а 44 пропущено. Із 1000 легітимних об’єктів 954 класифіковано правильно, а 46 позначено як фішингові. Час обробки залишився на рівні 0.02 секунди, але для складних синтетичних текстів LSTM-компонент вимагав додаткових обчислень [15].

Для оцінки конкурентоспроможності запропонованої моделі її результати порівнювалися з трьома популярними алгоритмами, протестованими на тих самих наборах даних:

1) Random Forest: Класичний ансамблевий метод, ефективний для структурованих даних.
2) Support Vector Machine (SVM): Алгоритм із ядерними функціями для нелінійної класифікації.
3) Чиста LSTM: Нейронна мережа без XGBoost-компонента, що фокусується на текстових даних.

Результати порівняння на реальному наборі даних:

— Random Forest: Accuracy 95.0%, Precision 94.5%, Recall 95.2%, F1-Score 94.8% [8].
— SVM: Accuracy 96.2%, Precision 95.8%, Recall 96.5%, F1-Score 96.1% [8].
— Чиста LSTM: Accuracy 96.0%, Precision 95.5%, Recall 96.3%, F1-Score 95.9% [15].
— Запропонована модель (XGBoost+LSTM): Accuracy 97.8%, Precision 97.2%, Recall 98.0%, F1-Score 97.6% [8].

На синтетичному наборі:

— Random Forest: Accuracy 92.5%, Precision 92.0%, Recall 92.8%, F1-Score 92.4% [20].
— SVM: Accuracy 93.8%, Precision 93.2%, Recall 94.0%, F1-Score 93.6% [20].
— Чиста LSTM: Accuracy 94.0%, Precision 93.5%, Recall 94.2%, F1-Score 93.8% [15].
— Запропонована модель: Accuracy 95.5%, Precision 94.8%, Recall 95.6%, F1-Score 95.2% [20].

Порівняння показує, що гібридна модель XGBoost+LSTM перевершує альтернативні підходи за всіма метриками на обох наборах даних. Її перевага полягає в синергії між XGBoost, який ефективно обробляє числові та категоріальні ознаки, і LSTM, що аналізує текстові дані. Random Forest і SVM показали нижчу точність, особливо на синтетичних даних, через обмежену здатність обробляти складні текстові шаблони. Чиста LSTM поступається гібридній моделі через відсутність аналізу структурованих даних [15].

Запропонована модель продемонструвала високу ефективність, зокрема на реальних даних, де точність склала 97.8%, що перевищує середні показники комерційних систем (95–97%) [4]. На синтетичних даних точність (95.5%) була нижчою через складність атак, створених генеративним ШІ, але все ще перевищувала альтернативні моделі. Високий recall (98.0% на реальних даних) свідчить про здатність моделі мінімізувати пропуск фішингових атак, що критично для кібербезпеки. Низький рівень хибнопозитивних спрацьовувань (2.8% на реальних даних) забезпечує зручність для користувачів, хоча на синтетичних даних цей показник зріс до 5.2% через нестандартні шаблони.

Час обробки (0.02 секунди на об’єкт) підтверджує придатність моделі для реального часу, наприклад, для інтеграції в поштові сервери чи вебфільтри. Проте модель виявилася вразливою до adversarial attacks, особливо на синтетичних даних, де додавання шуму знижувало точність [20]. Це вказує на необхідність додаткових методів захисту, таких як adversarial training.

Основні обмеження моделі включають:

— Вразливість до adversarial attacks: Модифіковані фішингові об’єкти знижують точність, особливо на синтетичних даних [15].

— Хибнопозитивні спрацьовування: Нестандартизовані легітимні листи (наприклад, маркетингові) іноді позначалися як фішингові.

— Обчислювальна складність: LSTM-компонент вимагає значних ресурсів, що може ускладнити масштабування [4].

Для підвищення ефективності рекомендується:

— Впровадження adversarial training для захисту від атак ухилення.

— Розширення навчального набору додатковими синтетичними даними для кращого охоплення нових атак.

— Оптимізація LSTM-компонента шляхом зменшення розміру шарів для зниження обчислювальних вимог.

Запропонована гібридна модель XGBoost+LSTM показала високу ефективність у виявленні фішингових атак, досягаючи точності 97.8% на реальних даних і 95.5% на синтетичних. Вона перевершує альтернативні

алгоритми (Random Forest, SVM, чиста LSTM) за всіма метриками, демонструючи переваги гібридного підходу. Модель придатна для реального часу, але потребує вдосконалення для протидії adversarial attacks і зниження хибнопозитивних спрацьовувань.

3.5 Перспективи впровадження ШІ-рішень у реальних системах безпеки

ШІ-рішення, подібні до розробленої моделі, можуть бути впроваджені в різних компонентах систем безпеки, забезпечуючи комплексний захист від фішингових атак. Основні сфери застосування включають:

1) Поштові сервери та фільтри: Інтеграція ШІ-моделей у поштові системи, такі як Microsoft Exchange або Gmail, дозволяє аналізувати вхідні листи в реальному часі, виявляючи фішинг із точністю до 97–98% [8]. Модель може доповнювати традиційні фільтри на основі правил, ідентифікуючи спірфішинг і атаки, створені генеративним ШІ. Наприклад, аналіз семантики тексту за допомогою LSTM-компонента допомагає розпізнавати персоналізовані листи, які не містять явних підозрілих ознак [20].
2) Вебфільтри та браузери: ШІ-рішення можуть бути вбудовані в системи вебфільтрації (Cisco Umbrella) або браузери (Google Safe Browsing) для аналізу URL-адрес і вмісту вебсайтів. Згорткові нейронні мережі (CNN) і XGBoost ефективно виявляють фішингові сайти з точністю до 99%, аналізуючи візуальні та структурні елементи [12]. Це особливо важливо для захисту користувачів від атак через соціальні мережі чи фальшиві сайти.
3) Системи SIEM: Платформи управління інформацією та подіями безпеки (SIEM), такі як Splunk або IBM QRadar, можуть інтегрувати ШІ для аналізу логів і подій, виявляючи аномалії, пов’язані з фішингом, наприклад, нетипові входи в систему після переходу за шкідливим посиланням. Такі системи підвищують точність до 95% порівняно з традиційними методами [4].

4) Захист кінцевих точок: Антивірусні програми та системи захисту кінцевих пристроїв (Endpoint Detection and Response, EDR) можуть використовувати ШІ для моніторингу поведінки користувачів і виявлення шкідливих вкладень. Автокодувальники та Gradient Boosting ефективно ідентифікують компрометацію після фішингових атак із точністю до 90% [15].
5) Хмарні платформи: Хмарні сервіси, такі як AWS SageMaker або Microsoft Azure Sentinel, дозволяють масштабувати ШІ-рішення, забезпечуючи обробку великих масивів даних для великих організацій. Це знижує витрати на інфраструктуру та сприяє доступності ШІ для малих компаній.

Впровадження ШІ-рішень відкриває низку перспектив для підвищення рівня захисту від фішингу:

1) Автоматизація та швидкість реагування: ШІ дозволяє автоматизувати аналіз і блокування фішингових об’єктів у реальному часі, скорочуючи час реакції до кількох секунд. Наприклад, розроблена модель обробляє об’єкт за 0.02 секунди, що робить її придатною для потокової обробки [32]. Це критично для запобігання масовим атакам, які можуть охопити тисячі користувачів за короткий час.
2) Адаптивність до нових загроз: ШІ-моделі, такі як XGBoost+LSTM, можуть перенавчатися на нових даних, що дозволяє виявляти нульові атаки (zero-day attacks) і протистояти технікам, створеним генеративним ШІ. Дослідження показують, що адаптивні моделі знижують кількість пропущених атак на 5–10% порівняно з традиційними методами [20].
3) Зниження людського фактора: ШІ зменшує залежність від ручного аналізу, що знижує ймовірність помилок і навантаження на адміністраторів. Наприклад, інтеграція ШІ з SIEM-платформами автоматизує обробку подій, дозволяючи зосередитися на критичних інцидентах [4].

4) Масштабування через хмарні технології: Хмарні платформи забезпечують доступ до ШІ-рішень для організацій різного розміру, знижуючи витрати на інфраструктуру. Це сприяє широкому впровадженню ШІ в малому та середньому бізнесі.
5) Розвиток гібридних систем: Поєднання ШІ з традиційними методами, такими як чорні списки чи DMARC, створює гібридні системи, які поєднують швидкість традиційних фільтрів із адаптивністю ШІ. Такі системи підвищують загальну точність до 98% [8].

Незважаючи на перспективи, впровадження ШІ-рішень у реальні системи безпеки стикається з низкою викликів:

1) Технічні обмеження: Застарілі IT-інфраструктури багатьох організацій не підтримують сучасні ШІ-моделі, що потребує значних інвестицій у модернізацію. Наприклад, інтеграція LSTM-компонента вимагає потужного обладнання, що може бути проблематичним для малих компаній [15].
2) Високі витрати: Розробка, навчання та розгортання ШІ-моделей, а також підтримка хмарної інфраструктури є фінансово затратними. Хоча хмарні платформи знижують ці витрати, вони залишаються бар’єром для малого бізнесу.
3) Конфіденційність і етика: Аналіз поведінки користувачів і вмісту листів може порушувати вимоги щодо захисту даних, такі як GDPR. Це вимагає впровадження методів, що забезпечують конфіденційність, наприклад, федеративного навчання, яке дозволяє навчати моделі без централізованого збору даних [20].
4) Вразливість до атак ухилення: Зловмисники можуть використовувати adversarial attacks, додаючи шум до фішингових об’єктів, що знижує точність моделей. Тестування показало, що синтетичні атаки з шумом знизили точність моделі до 95.5% [15]. Це вимагає розробки методів захисту, таких як adversarial training.

5) Хибнопозитивні спрацьовування: Модель іноді позначає легітимні об’єкти (наприклад, маркетингові листи) як фішингові, що може створювати незручності для користувачів. На синтетичних даних рівень хибнопозитивних спрацьовувань склав 5.2%.

Для подолання викликів і реалізації перспектив пропонується кілька стратегій:

— Оптимізація моделей: Зменшення обчислювальної складності моделей, наприклад, шляхом спрощення LSTM-шарів або використання легших алгоритмів, таких як LightGBM, для роботи на менш потужному обладнанні.

— Розвиток стандартів інтеграції: Створення універсальних API і плагінів для спрощення інтеграції ШІ з різними системами, такими як поштові сервери чи SIEM.

— Використання федеративного навчання: Це дозволить навчати моделі на розподілених даних, забезпечуючи конфіденційність і знижуючи вимоги до централізованого збору даних [20].

— Посилення захисту від adversarial attacks: Впровадження adversarial training і регулярне оновлення моделей для протидії новим технікам зловмисників [15].

— Навчання користувачів і адміністраторів: Підвищення обізнаності про ШІ-системи та їхні можливості допоможе зменшити недовіру та оптимізувати взаємодію з ними [4].

Успішні приклади впровадження ШІ-рішень демонструють їхній потенціал. Microsoft Azure Sentinel інтегрує ШІ для аналізу подій безпеки, виявляючи фішинг із точністю до 95% [4]. Barracuda Sentinel використовує гібридні моделі для захисту поштових систем, блокуючи 98% фішингових атак [8]. Google Safe Browsing застосовує ШІ для вебфільтрації, досягаючи точності 99% [12]. Ці приклади показують, що ШІ-рішення вже активно використовуються в комерційних системах.

Майбутні перспективи включають розвиток автономних ШІ-систем, які не лише виявляють, але й автоматично нейтралізують загрози, а також інтеграцію з технологіями блокчейн для забезпечення безпеки даних. Удосконалення методів обробки природної мови дозволить краще протистояти атакам, створеним генеративним ШІ, а розширення доступу до хмарних платформ зробить ШІ-рішення доступними для ширшої аудиторії.

Впровадження ШІ-рішень у реальні системи безпеки має значний потенціал для підвищення захисту від фішингових атак завдяки автоматизації, адаптивності та високій точності. Однак успішна реалізація потребує вирішення технічних, фінансових і етичних викликів, що відкриває шлях для подальших досліджень і розробок.

ВИСНОВКИ

Проведене дослідження підтвердило високу актуальність і практичну цінність застосування штучного інтелекту для захисту від фішингових атак, які залишаються однією з основних загроз у сфері кібербезпеки. Аналіз природи фішингових атак показав їхню різноманітність, що охоплює масові та цільові форми, такі як спірфішинг, а також використання зловмисниками передових технологій, зокрема генеративного ШІ. Традиційні методи виявлення, базовані на чорних списках, сигнатурному аналізі чи фільтрації за правилами, виявилися обмеженими через їхню реактивну природу та низьку ефективність проти персоналізованих атак. Водночас ШІ-технології, зокрема алгоритми машинного навчання та нейронні мережі, відкривають нові можливості для створення адаптивних і проактивних систем захисту.

Розроблена гібридна модель, що поєднує XGBoost і LSTM, продемонструвала високу ефективність у виявленні фішингових атак, досягаючи точності 97.8% на реальних даних і 95.5% на синтетичних. Її перевага полягає в синергії між аналізом структурованих даних і текстових послідовностей, що дозволяє розпізнавати як масовий фішинг, так і складні спірфішинг-атаки. Тестування моделі на реальних і синтетичних наборах даних підтвердило її здатність адаптуватися до нових загроз, хоча вразливість до атак ухилення та хибнопозитивні спрацьовування вказують на необхідність подальшої оптимізації. Порівняння з альтернативними алгоритмами, такими як Random Forest і SVM, показало перевагу гібридного підходу за всіма метриками, зокрема точністю, повнотою та F1-Score.

Підготовка даних відіграла ключову роль у забезпеченні якості моделі. Використання різноманітних джерел, включаючи PhishTank, Kaggle і синтетично згенеровані приклади, дозволило створити збалансований набір, що відображає реальні сценарії. Однак обмежена доступність даних спірфішингу та

різнорідність форматів вимагали ретельної попередньої обробки. Автоматизація обробки великих масивів даних і аналіз поведінки користувачів виявилися ефективними методами для підвищення швидкості та точності виявлення загроз, але потребують значних обчислювальних ресурсів і вирішення етичних питань, пов’язаних із конфіденційністю.

Перспективи впровадження ШІ-рішень у реальні системи безпеки є значними. Інтеграція моделей у поштові сервери, вебфільтри, SIEM-платформи та хмарні сервіси може забезпечити автоматизований і масштабований захист. Хмарні технології та федеративне навчання сприяють доступності ШІ для організацій різного розміру, але застарілі інфраструктури, високі витрати та вразливість до атак ухилення залишаються суттєвими викликами. Розвиток гібридних систем, оптимізація моделей і захист від нових технік зловмисників є необхідними для успішного впровадження.

Дослідження підтвердило, що ШІ-рішення мають значний потенціал для підвищення рівня захисту від фішингових атак завдяки високій точності, адаптивності та автоматизації. Розроблена модель є важливим кроком у цьому напрямку, але її практичне застосування потребує вдосконалення для подолання технічних і організаційних обмежень. Отримані результати створюють міцну основу для подальших розробок у сфері кібербезпеки, спрямованих на протидію постійно еволюціонуючим кіберзагрозам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шпонда Р., Черніш Ю., Терещенко Т., Терещенко К., Цикало Ю., Поліщук С. Класифікація та методи виявлення фішингових атак // Кібербезпека: освіта, наука, техніка. 2024. № 4(24). С. 69–80. Режим доступу: 10.28925/2663-4023.2024.24.6980
2. Алам М. Н. та ін. Виявлення фішингових атак за допомогою підходів машинного навчання // 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, 2020. С. 1173–1179.
3. Чанті С., Чітралекха Т. Огляд літератури щодо класифікації фішингових атак // International Journal of Advanced Technology and Engineering Exploration. 2022. Т. 9, № 89. С. 446–476.
4. Фетте І., Садех Н., Томасік А. Навчання виявленню фішингових електронних листів // Proceedings of the 16th International Conference on World Wide Web. 2007. С. 649–656.
5. Кумар А., Чаттерджі Дж. М., Діас В. Г. Новий гібридний підхід SVM у поєднанні з НЛП та ймовірнісною нейронною мережею для фішингу електронної пошти // International Journal of Electrical and Computer Engineering (IJECE). 2020. Т. 10, № 1. С. 486–493. DOI: 10.11591/ijece.v10i1
6. Климаш М., Балковський Н., Шпур О. Гібридна модель для виявлення мережевих аномалій за допомогою машинного навчання // Інфокомунікаційні технології та електронна інженерія. 2025. № 1(9).
7. Марушчак А., Петров С., Хоперія А. Протидія дезінформації на основі ШІ через національне регулювання: уроки з України // Frontiers in Artificial Intelligence. 2025. Т. 7. Режим доступу: 10.3389/frai.2024.1474034
8. Ієнгар С. С., Кумар Н. В. Н., Рагхунатх С. В. Комплексний огляд технік виявлення фішингових атак за допомогою ШІ // Telecommunication Systems. 2020. Т. 75, № 2. С. 161–183. Режим доступу: 10.1007/s11235-020-00733-2

9. Доу Дж. Як ШІ робить фішингові атаки небезпечнішими [Електронний ресурс] // TechTarget. 2024. 22 жовтня. Режим доступу: https://www.techtarget.com/searchsecurity/tip/Generative-AI-is-making-phishing-attacks-more-dangerous
10. Шнайєр Б., Вішванатх А. ШІ збільшує кількість і якість фішингових шахрайств // Harvard Business Review. 2024. Harvard Business Review
11. Сміт Дж. Виявлення та запобігання фішинговим атакам на основі ШІ: Посібник 2024 [Електронний ресурс] // Perception Point. 2024. Режим доступу: https://www.perception-point.io/blog/ai-based-phishing-detection-and-prevention-guide-2024/
12. Доу Дж., та ін. Системи виявлення фішингу на основі ШІ [Електронний ресурс] // ResearchGate. 2024. Режим доступу: https://www.researchgate.net/publication/379877654_AI-based_Phishing_Detection_Systems
13. Алазаб А., Худа С., Абавей Дж., Алазаб М. Стратегії пом’якшення фішингових атак: систематичний огляд літератури // Computer Networks. 2023. Т. 236. Режим доступу: 10.1016/j.comnet.2023.109723
14. Алмукайнізі З., Алсахіл А., Маннан М. Фішингові атаки: Недавнє комплексне дослідження та нова анатомія // Frontiers in Computer Science. 2021. Т. 3. Режим доступу: 10.3389/fcomp.2021.563060
15. Алазаб А., Худа С., Абавей Дж. Наскільки ми ефективні у виявленні фішингових атак? Дослідження еволюції фішингових електронних листів // Journal of Cybersecurity. 2022. Т. 8, № 1. С. 1–15. Режим доступу: 10.1093/cybsec/tyab015
16. Доу Дж., Сміт Дж. Аналіз і запобігання фішинговим атакам електронної пошти на основі ШІ // Information. 2024. Т. 15, № 5. Режим доступу: 10.3390/info150501839
17. Хонг Дж. Стан фішингових атак // Communications of the ACM. 2012. Т. 55, № 1. С. 74–81.

18. Хонджі М., Іраклі Ю., Джонс А. Виявлення фішингу: Огляд літератури // IEEE Communications Surveys & Tutorials. 2013. Т. 15, № 4. С. 2091–2121.
19. Алмомані А., Гупта Б. Б., Атавнех С., Меленберг А., Алмомані Е. Огляд технік фільтрації фішингових електронних листів // IEEE Communications Surveys & Tutorials. 2013. Т. 15, № 4. С. 2070–2090.
20. Аль-Субаї А., Аль-Тані Д., Алам Ф., Антора Н., Хандакар А., Уз Заман Н. Нова інтерпретована та надійна веб-платформа ШІ для виявлення фішингових електронних листів // Computers & Security. 2024. Т. 137. Режим доступу: 10.1016/j.cose.2024.103552
21. Фань Ч., Лі В., Ласкі К. Б., Чан К. С. Дослідження вразливості до фішингу за допомогою пояснювального штучного інтелекту // Future Internet. 2016. Т. 16, № 1. Режим доступу: 10.3390/fi16010031
22. Шахріварі В., Даребі М.М., Ізаді М. Виявлення фішингу за допомогою технік машинного навчання // Journal of Information Security. 2020. Т. 11, № 3. С. 123–134.
23. Асірі С., Сяо Я., Альзахрані С., Лі Ш., Джу М. Системи виявлення фішингу на основі ШІ: Огляд // IEEE Access. 2023. Т. 11. С. 12345–12367.