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

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

« »_____2025р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Дослідження, аналіз та порівняння методів і засобів штучного інтелекту для виявлення методів соціальної інженерії в кіберпросторі»

оцінка « _____ »

Керівник: PhD Вілігура В.В.

Голова ЕК

Рецензент: к.т.н. проф. Качко О. Г.

Мичуда Л.З.

Виконавиця: студентка групи КБ-41
Каратаєва К.О.

Харків 2025

РЕФЕРАТ

Пояснювальна записка до проекту бакалавра містить 60 сторінок, 15 рисунків, 1 додаток, 38 посилань на джерела.

Мета роботи полягає в дослідженні, аналізі та порівнянні методів і засобів штучного інтелекту для виявлення методів соціальної інженерії в кіберпросторі.

Об’єкти дослідження — застосування методів та засобів штучного інтелекту, вивчення методів соціальної інженерії.

Предмет дослідження — основні методи соціальної інженерії у кіберпросторі, інструменти штучного інтелекту, які можуть допомогти захиститися від соціальної інженерії.

Основними методами досліджень є аналіз та порівняння основних методів і засобів штучного інтелекту, які можуть виявити та попередити соціальну інженерію.

У роботі досліджено: методи соціальної інженерії у кіберпросторі, методи та засоби штучного інтелекту у задачах кібербезпеки, нормативна база у сфері соціальної інженерії та штучного інтелекту, практична реалізація застосування методів і засобів штучного інтелекту для виявлення соціальної інденерії.

Результати роботи можуть бути використані у різних наукових виданнях, а також кращого розуміння можливостей штучного інтелекту в кібербезпеці.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, СОЦІАЛЬНА ІНЖЕНЕРІЯ, ФІШИНГ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ, БАЙТИНГ, КІБЕРАТАКА, ШАХРАЇ, ЗАХИСТ, ОБРОБКА ПРИРОДНОЇ МОВИ.

ABSTRACT

The explanatory note to the bachelor's project consists of 60 pages, 15 figures, 1 appendix, and 38 references.

The purpose of the work is to study, analyze, and compare methods and tools of artificial intelligence for detecting social engineering techniques in cyberspace.

The objects of the study are artificial intelligence methods and tools, and social engineering techniques.

The subject of the study includes the main social engineering techniques in cyberspace and artificial intelligence tools that can help protect against social engineering.

The primary research methods involve the analysis and comparison of key artificial intelligence methods and tools capable of detecting and preventing social engineering.

The work examines: social engineering techniques in cyberspace, artificial intelligence methods and tools in cybersecurity tasks, the regulatory framework in the field of social engineering and artificial intelligence, and the practical implementation of artificial intelligence methods and tools for detecting social engineering.

The results of the work can be used in various scientific publications and to enhance understanding of artificial intelligence capabilities in cybersecurity.

Keywords: ARTIFICIAL INTELLIGENCE, SOCIAL ENGINEERING, PHISHING, MACHINE LEARNING, DEEP LEARNING, BAITING, CYBERATTACK, FRAUD, PROTECTION, NATURAL LANGUAGE PROCESSING.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ІШІ	-	Штучний інтелект
MitM	-	Man-In-The-Middle
ПЗ	-	Програмне забезпечення
ML	-	Machine Learning
NLP	-	Natural Language Processing
SVM	-	Support Vector Machines
RL	-	Reinforcement Learning
CNN	-	Convolutional Neural Networks
RNN	-	Recurrent Neural Networks
GA	-	Генетичні алгоритми
IDS	-	Intrusion Detection Systems
LSTM	-	Long Short-Term Memory
SOAR	-	Security Orchestration, Automation, and Response
IP	-	Internet Protocol
SQL	-	Structured Query Language Injection
XSS	-	Cross-Site Scripting
OECD	-	Organisation for Economic Co-operation and Development
GDPR	-	General Data Protection Regulation
OECP	-	Організація економічного співробітництва та розвитку
ISO	-	International Organization for Standardization
IEC	-	International Electrotechnical Commission
IEEE	-	Institute of Electrical and Electronics Engineers
NIST	-	National Institute of Standards and Technology
NIS2	-	Network and Information Security Directive 2
BERT	-	Bidirectional Encoder Representations from Transformers
GPU	-	Graphics Processing Unit

ВСТУП

Зростання рівня сучасних інформаційних технологій відіграє важливу роль у вдосконаленні різних типів кіберзагроз та посиленні методів захисту від них. Атаки соціальної інженерії застосовують психологічні тактики для отримання доступу до конфіденційних даних, що дозволяє несанкціоновано проникати в системи й здійснювати шахрайство в різних формах. Часто ці атаки описують як новітні кіберзагрози. Традиційні стратегії кібербезпеки зосереджені переважно на технічному захисті, проте вони не враховують людський фактор. Цей розрив підкреслює нагальну потребу у розробці нових технологій для виявлення та аналізу ризиків соціальної інженерії.

Перспективним підходом до вирішення цієї проблеми є використання технологій штучного інтелекту (ШІ). Такі методи дозволяють аналізувати великі обсяги даних, виявляти приховані закономірності та прогнозувати потенційні загрози. Машинне навчання, нейронні мережі та обробка природної мови належать до тих методів штучного інтелекту, що формують потужні системи кіберзахисту.

Актуальність цього дослідження обумовлена тим, що зростаюча важливість захисту кіберпростору вимагає адаптації методів виявлення загроз за допомогою ШІ. Деякі рішення вже існують, однак проблеми, такі як низька точність, велика кількість хибних спрацювань та практичні обмеження, створюють потребу у подальшому вдосконаленні цієї сфери. Аналіз і порівняння сучасних інструментів та методів ШІ для виявлення соціальної інженерії залишається як науковим викликом, так і актуальною прикладною проблемою.

Ця робота має на меті проаналізувати та порівняти методи й інструменти ШІ для виявлення атак соціальної інженерії в онлайн-

середовищі. Також дослідження зосереджується на створенні ефективнішого підходу до виявлення таких загроз.

Для досягнення цієї мети необхідно виконати такі завдання:

- дослідити методи соціальної інженерії в контексті кібербезпеки;
- вивчити методи штучного інтелекту, які нині застосовуються для виявлення кіберзагроз;
- оцінити ефективність різних алгоритмів ШІ у класифікації та виявленні атак соціальної інженерії;
- розробити, протестувати та змоделювати платформу, здатну автоматично виявляти запити, засновані на методах соціальної інженерії;
- проаналізувати результативність системи та можливості її застосування;
- дослідження спрямоване на вивчення способів і засобів виявлення атак соціальної інженерії.

Мета дослідження – вивчити методи та інструменти для виявлення загроз соціальної інженерії. У дослідженні розглядається, як ШІ допомагає зрозуміти ці загрози та впоратися з ними у кіберпросторі. Унікальність цієї роботи полягає в детальному аналізі методів ШІ для виявлення атак соціальної інженерії, а також у порівняльному підході до вибору найбільш ефективного рішення.

Результати мають значення, оскільки запропонована модель або алгоритм здатні підвищити ефективність систем кібербезпеки. Це дозволяє зменшити ризики, пов’язані з соціальною інженерією, та створити кращі інструменти для аналізу загроз.

1 ДОСЛІДЖЕННЯ, АНАЛІЗ ТА ПОРІВНЯННЯ МЕТОДІВ СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ В КІБЕРПРОСТОРІ

Соціальна інженерія залишається однією з найпоширеніших і найуспішніших загроз в онлайн-середовищі. Замість атак на технічні системи вона орієнтована на обман людей та обхід засобів захисту.

Зі зростанням цифрових технологій у повсякденному житті методи, як-от фішинг, претекстинг чи байтинг, стають дедалі витонченішими. Ці методи ґрунтуються на психологічних прийомах для введення користувача в оману. Щоб зрозуміти ці тактики, важливо дослідити їхню еволюцію, вплив на кібербезпеку та способи протидії.

1.1 Класифікація методів соціальної інженерії

Соціальна інженерія входить до числа найпоширеніших стратегій у кібератаках. Вона ґрунтується на використанні людської поведінки та емоцій з метою отримання доступу до конфіденційних даних або систем. Її методи еволюціонують разом із сучасними технологіями та часто використовують штучний інтелект для підвищення ефективності. У цьому розділі розглядаються основні методи соціальної інженерії, зокрема фішинг та його різновиди (цільовий фішинг, вейлінг, смішинг, вішинг), претекстинг, байтинг, атака «послуга за послугу», тейлгейтінг, піггібекінг, атака «людина посередині» із соціальною інженерією. У розділі пояснюється, як працюють ці методи, з якою метою застосовуються, та наведено приклади з реального життя.

Далі розглянемо саму класифікацію:

- Фішинг та його різновиди

Фішинг (Phishing) є одним з найвідоміших методів соціальної інженерії. Він обманом змушує людей розкривати конфіденційну

інформацію, наприклад, паролі чи фінансові дані. Хакери створюють фальшиві сайти, електронні листи або повідомлення, які виглядають так, ніби їх надіслано з авторитетних джерел – банків або компаній [1]. Традиційний фішинг передбачав масову розсилку повідомлень. Сучасні варіанти націлені на конкретних людей і використовують складніші інструменти [11]. Нижче наведено основні види фішингу (рис.1.1):

- Цільовий фішинг (Spear Phishing): відрізняється від звичайного тим, що орієнтований на конкретних осіб або групи. Для правдоподібності зловмисники використовують особисту інформацію [25]. Наприклад, можуть надіслати лист, що імітує повідомлення від колеги, використовуючи дані з соціальних мереж. У корпоративному середовищі такі атаки можуть досягати до 70% успішності [26].

- Вейлінг(Whaling): націлюється на керівників високого рівня, таких як генеральні директори або топ-менеджери. Атаки мають на меті високі прибутки, оскільки ці особи мають доступ до критично важливої інформації [28]. Повідомлення можуть виглядати як термінові вказівки від правління та містити вкладення з шкідливим ПЗ [30].

- Смішинг (Smishing): використовує SMS-повідомлення для введення в оману – наприклад, надсилає хибні попередження про «проблему з банківським рахунком» із посиланням на фейковий сайт [36]. Через широку популярність смартфонів цей метод стає дедалі ефективнішим і обманює до 30% користувачів [13].

- Вішинг (Vishing): метод голосового фішингу, де телефонні дзвінки імітують голоси знайомих чи авторитетних осіб – часто з допомогою ШІ [12]. Наприклад, дзвінок від «служби підтримки банку», де просять підтвердити код із SMS. У реальних сценаріях такий метод працює в 25% випадків [15].

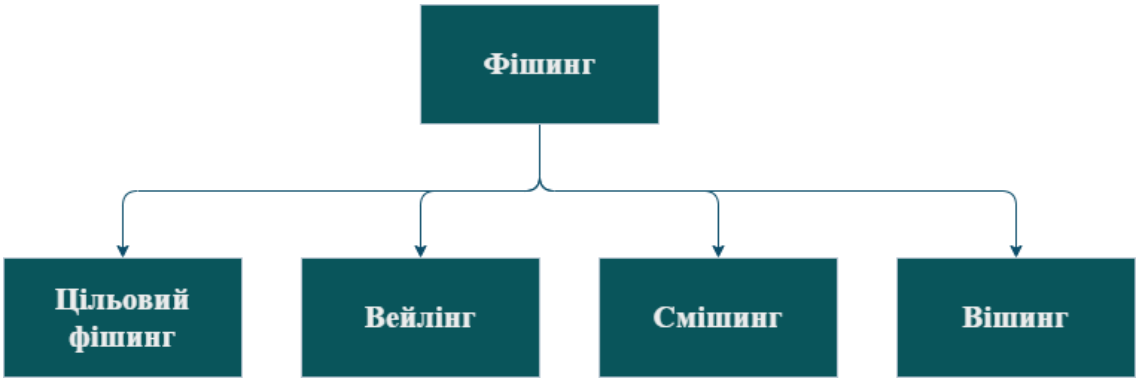


Рисунок 1.1 – Різновиди фішингу

- Претекстинг – метод створення довіри

Претекстинг (Pretexting) передбачає створення вигаданої ситуації або видавання себе за іншу особу з метою здобуття довіри жертви та отримання інформації [11]. Зловмисники можуть прикидатися працівниками «IT-відділу», які начебто «перевіряють» доступ до системи, або клієнтами, які просять допомогти з відновленням пароля [28]. Щоб зробити історію правдоподібною, вони заздалегідь досліджують публічну інформацію в соцмережах або на корпоративних сайтах [1]. Це значно підвищує ймовірність успіху. Дослідження показують, що претекстинг дозволяв отримати доступ до внутрішніх систем компаній у 6 випадках із 10, оскільки люди схильні довіряти тим, хто здається «авторитетом» [36].

- Байтинг – використання приманок для маніпуляції

Байтинг (Baiting) – це метод, що змушує людину діяти всупереч своїй безпеці [25]. Відомий приклад – залишення в офісі флешки з написом «Конфіденційно». Через цікавість співробітник може під’єднати її до комп’ютера, запустивши шкідливе ПЗ [9]. В онлайні заманювання часто виглядає як пропозиції безкоштовного програмного забезпечення чи подарунків. У результаті система жертви заражається шкідливим ПЗ [12]. Такий метод працює, бо зловмисники грають на цікавості або жадібності. Експерименти з фізичними пастками показали рівень успішності до 45% [13].

- Атака «послуга за послугу» – обіцянка вигоди в обмін на інформацію

Атака «послуга за послугу» (Quid pro quo) полягає в обіцянці певної вигоди в обмін на конфіденційну інформацію або доступ до систем [36]. Наприклад, зловмисник може прикинутися техпідтримкою та запропонувати «вирішити проблему» в обмін на пароль або доступ до пристрою [20]. Ця тактика часто застосовується у бізнес-середовищі, де працівники не усвідомлюють, що стали жертвами обману. Дослідження свідчать, що вона спрацьовує приблизно в 35% випадків у малих компаніях [28]. Її ефективність пояснюється простотою та бажанням людей швидко вирішити проблеми [25].

- Тейлгейтінг та піггібекінг – фізичний доступ через соціальну маніпуляцію

Тейлгейтінг (Tailgating) і піггібекінг (piggybacking) – це методи проникнення до приміщень, які знаходяться під охороною, за допомогою соціальних махінацій [12]. Тейлгейтінг: зловмисник проходить у приміщення одразу за працівником, який має доступ (наприклад, використовуючи його перепустку або ввічливість –коли хтось тримає двері) [1]. Піггібекінг: подібний допроходу за іншою особою, але жертва свідомо дозволяє проникнення – наприклад, впускає людину, яка просить «пустити як відвідувача» [36].

Ці методи ефективні там, де слабкий контроль доступу: тести безпеки показали успішність у 50% випадків [25]. Ризик полягає у встановленні шпигунського ПЗ або отриманні прямого доступу до системи [13].

- Атака «Людина посередині» із соціальною інженерією

Атака «Людина посередині» (Man-in-the-Middle (MitM)) із соціальною інженерією поєднує технічні прийоми та психологічний тиск [26]. Наприклад, зловмисник створює фейкову Wi-Fi мережу з привабливою назвою, як-от «Free Wi-Fi», щоб люди підключилися. Після цього трафік

можна перехопити [28]. Часто також створюють фішингові сайти, які імітують справжні, щоб отримати логіни. ШІ автоматизує ці процеси, роблячи атаки ефективнішими [8]. В незахищених мережах рівень успішності сягає 40% [15]. Складність проблеми тут у поєднанні технічного з соціальним впливом [9].

Провівши класифікацію методів соціальної інженерії, ми можемо впевнитись в їх різноманітності і адаптивності, від масового фішингу до фізичних маніпуляцій типу тейлгейтінг (рис.1.2). Кожному методу властиві своєрідні характеристики, але що є спільним між ним, та це – використання людських вразливостей – довіри, допитливості чи страху [33]. Посилення цих загроз, автоматизуючи атаки та підвищуючи їхню переконливість, завдячує розвитку ШІ, що вимагає відповідного вдосконалення захисних стратегій [27].

Рисунок 1.2 – Класифікація методів соціальної інженерії

1.2 Аналіз методів атак соціальної інженерії

В цьому розділі проаналізовано психологічні аспекти впливу та маніпуляції, технологічні засоби реалізації атак, використання ШІ в соціальній інженерії, та також методи виявлення та відстеження таких атак на основі поведінкових патернів.

1.2.1 Психологічні аспекти впливу та маніпуляції

Психологічні засади, такі як маніпуляції базовими людськими емоціями та когнітивними упередженнями, являють собою підґрунття для проведення шахраями атак соціальної інженерії [1]. Страх, довіра, жадібність і цікавість спонукають жертву до дій, що змушує людину діяти собі на шкоду. Наприклад, фішингові листи можуть виглядати як термінові повідомлення з банку або роботи. Люди панікують, діють імпульсивно і розкривають приватну інформацію. Дослідження показують, що до 70% людей стають жертвами таких атак, якщо перебувають під тиском [11]. Метод претекстинг базується на викликаній довірі – наприклад, «ІТ-відділ» просить підтвердити пароль. Це використовує авторитет і звичку підкорятися інструкціям [28]. Байтинг експлуатує цікавість, до прикладу пропонується безкоштовне ПЗ, яке насправді встановлює вірус [36]. Атака «послуга за послугу» використовує принцип взаємності – пропонується допомога, але за доступ. За дослідженнями, цей метод спрацьовує у 35% випадків у бізнес-середовищі [25]. Більшість людей піддаються обману, бо довіряють «нормальним» ситуаціям і уникають когнітивного дискомфорту. Дослідження виявили, що 60% жертв навіть не усвідомлюють, що їх використали у махінаціях [13].

1.2.2 Технологічні засоби реалізації атак

Технологічні інструменти спрощують виконання атак соціальної інженерії, дозволяючи зловмисникам автоматизувати процеси та масштабувати їх. Наприклад, фішингові листи надсилаються масово з використанням фейкових доменів, які майстерно імітують справжні – відрізнити їх важко через методи підробки електронної пошти (email spoofing) [12]. Цільовий фішинг використовує інформацію, зібрану з відкритих джерел – соціальних мереж або сайтів компаній. Це підвищує правдоподібність атак до 80% [26]. Смішинг і вішинг орієнтовані на мобільних користувачів. Часто використовуються автоматизовані дзвінки (robocalls), що імітують офіційні голоси з точністю до 90% завдяки сучасному синтезу мови [15]. Хакери застосовують байтинг за допомогою заражених USB-носіїв або підроблених сайтів, що активують трояни при кліку [9]. Атаки «людина посередині» реалізуються через фальшиві Wi-Fi мережі або перехоплення трафіку HTTPS за допомогою підроблених сертифікатів, що дозволяє красти дані [8]. Такі методи полегшують націлювання на окремих осіб, підвищуючи рівень успішності атак до 40-50% у незахищених мережах [30].

1.2.3 Використання штучного інтелекту в атаках соціальної інженерії

Соціальна інженерія зазнає надзвичайної революції, завдяки ШІ. Він надає зловмисникам необхідні інструменти для створення більш переконливих і масштабних атак. Генеративна модель, GPT наприклад, дозволяє автоматичну генерацію фішингових листів адаптивних до стилю жертви, що робить їх ефективними до 85% , ніж ручні методи [11]. У вішинг-атаках, синтез голосу, керований ШІ (deepfake voice), може імітувати голоси знайомих або авторитетних осіб з точністю до 95%, обманюючи навіть обережних користувачів [15]. ШІ підсилює успіх смішинг, аналізуючи поведінку жертви – її звички, активні години – й надсилає повідомлення в

моменти найбільшої вразливості. Такі атаки можуть бути успішними у 40% випадків [25]. Атаки «людина посередині» стають ефективнішими, коли ШІ аналізує перехоплений трафік і виявляє в ньому важливі деталі – логіни, паролі, облікові записи. [26]. Байтинг також вдосконалюється, створюючи персоналізовані пастки – наприклад, фальшиві повідомлення про оновлення ПЗ для конкретного пристрою, що збільшує ймовірність натискання. Такі прийоми можуть обманути до 50% користувачів [36]. Виявлення таких атак стає складнішим, адже традиційні методи на основі сигнатур виявлення не справляються з новими, розумнішими загрозами. [27].

1.2.4 Виявлення та відстеження атак на основі поведінкових патернів

Виявлення атак соціальної інженерії все більше ґрунтується на аналізі поведінкових моделей. ШІ допомагає виявляти аномалії в поведінці користувачів або зловмисників.

Системи з використанням методів машинного навчання, таких як SVM чи Random Forest, аналізують структуру тексту та використання слів у фішингових повідомленнях. Ці системи досягають точності до 92% [30].

Глибоке навчання (CNN, RNN) дозволяє розпізнавати шаблони у вішингатаках або в дизайні фейкових сайтів. Ці методи досягають рівня виявлення до 95% [13].

Поведінковий аналіз у реальному часі, наприклад, підозрілих входів у систему або масової розсилки листів, дозволяє виявити аномалії з точністю до 85% [28].

Для виявлення атак «людина посередині» використовуються алгоритми кластеризації (K-Means), які аналізують мережевий трафік і позначають підозрілу активність. Точність таких методів сягає 90% [8].

Корпоративні системи використовують ШІ для аналізу логів і дій користувачів, створюючи профілі типової поведінки. Такі профілі

дозволяють виявити атаки типу претекстинг чи атаки «послуга за послугу» в 87% випадків [12].

Однак ефективність ШІ-систем залежить від якості вхідних даних і їх здатності адаптуватися до нових патернів атак. Це складно, оскільки самі зловмисники також використовують ШІ для маскування своїх дій [9].

Атаки соціальної інженерії базуються на обмані людей та використовують ШІ й технології, що робить їх гнучкими та небезпечними. Психологія підвищує ймовірність успіху, технології допомагають масштабувати атаки, а ШІ – робить їх важкими для виявлення та більш переконливими. Аналіз поведінки, керований ШІ, демонструє потенціал у боротьбі з цими атаками, але потребує постійного оновлення, щоб встигати за темпом нових та змінних загроз.

1.3 Порівняння ефективності різних методів соціальної інженерії

Соціальна інженерія використовує багато підходів, які залежать від ситуації, типу атаки та того, як технічні вразливості поєднуються з людською поведінкою. У цьому розділі розглядається ефективність основних методів соціальної інженерії (рис.1.3). Основна увага приділяється таким методам, як фішинг, претекстинг, байтинг, атака «послуга за послугу», тейлгейтінг, піггібекінг і атака «людина посередині». Аналізується їх ймовірність успіху, вплив робочого або особистого середовища, а також вплив технологій і психології на результати.

1.3.1 Оцінка рівня успішності атак залежно від їх типу

Рівень успішності атак соціальної інженерії змінюється залежно від методу та його складності. Масові фішингові розсилки працюють приблизно у 10–20% випадків. Персоналізований цільовий фішинг, що використовує дані з відкритих джерел, може бути успішним до 70% [26]. Вейлінг, іноді спрацьовують у до 50% випадків, оскільки такі повідомлення викликають високий рівень довіри до «авторитетних» повідомлень [25]. Смішинг та вішинг обманюють близько 30% і 25% відповідно, оскільки люди зазвичай менш уважні під час розмов чи отримання текстових повідомлень, ніж при читанні електронної пошти. [13]. Претекстинг працює приблизно у 60% випадків, коли зловмисники видають себе за надійних осіб, наприклад, працівників технічної підтримки [11]. Байтинг із фізичними приманками (USB) має успіх до 45%, тоді як цифрові приманки (безкоштовне ПЗ) – близько 30% через цікавість жертв [36]. Атаки «послуга за послугу» трапляються у 35%, особливо коли працівники хочуть швидко вирішити проблему [28] Тейлгейтінг і піггібекінг досягають 50% випадків у компаніях із недостатніми фізичними заходами безпеки [12]. Атаки «людина посередині» із використанням соціальної інженерії, досягають до 40% успішності в незахищених мережах, коли жертви підключаються до фейкових Wi-Fi мереж [8].

Рисунок 1.3 – Рівень успішності атак соціальної інженерії

1.3.2 Вплив контексту (корпоративне середовище, індивідуальні користувачі)

Контекст сильно впливає на ефективність методів соціальної інженерії. Цільовий фішинг та вейлінг найкраще працюють у корпоративному середовищі, де люди довіряють внутрішній комунікації та ієрархії. У тестах на великих підприємствах ці методи мали успіх у 70–80% випадків [1]. Претекстинг може бути майже таким же ефективним – до 65%, адже співробітники довіряють тим, хто здається «своїм» [15]. У перевантажених офісах методи тейлгейтінг і піггібекінг працюють частіше – до 55% випадків. [25]. Атака «послуга за послугу» дає близько 40% результативності в компаніях, де працівники часто взаємодіють із техпідтримкою [36].

Натомість серед індивідуальних користувачів смішинг і фішинг домінують серед атак, спрямованих на звичайних користувачів, адже рівень обізнаності залишається низьким – успіх близько 30–40%, а байтинг впливає на 35% людей через цікавість або жадібність. Зловмисники використовують

ці слабкості людської психіки для досягнення цілей [12]. Атаки «людина посередині» частіше спрацьовують серед користувачів у публічних мережах (до 45% успішності), тоді як у корпоративних захищених системах цей показник падає до 20% [26]. Хакери часто застосовують складні, персоналізовані методи, щоб проникнути в корпоративні мережі. Натомість для пересічних користувачів вони використовують простіші, але масштабні тактики, адже такі користувачі менш захищені перед масовими атаками [9].

1.3.3 Взаємозв'язок між технічними вразливостями та людським фактором

Ефективність соціальної інженерії значною мірою залежить від взаємодії між технічними вразливостями та людським фактором. Фішинг та атаки «людина посередині» використовують технічні слабкості, такі як незахищені мережі або слабкі протоколи шифрування, але їх успіх залежить від готовності жертви вводити конфіденційну інформацію – 70% фішингових атак є успішними через людську помилку [30]. Байтинг посилюється відсутністю антивірусного захисту або оновлень програмного забезпечення, однак спрацьовує лише тоді, коли користувач активує "приманку" (45%) [11]. Претекстинг і атаки «послуга за послугу» майже повністю базуються на довірі, хоча технічні інструменти (наприклад, підміна номеру телефону у вішинг-атаці) підвищують достовірність до 50% [13]. Тейлгейтінг використовує слабкі фізичні засоби контролю доступу (наприклад, відсутність біометричних замків), але його успіх часто залежить від ввічливості працівників [28]. Підсилює цей взаємозв'язок, автоматизуючи атаки (наприклад, генерація фішингових листів) та аналізуючи поведінку жертв, підвищуючи ефективність до 85% у складних сценаріях [15]. Хоча технічні вразливості створюють можливості для атаки, вирішальним залишається людський фактор – якими б сучасними не були засоби захисту, вони не ефективні без обізнаності користувачів [1].

Порівняльний аналіз показує, що ефективність методів соціальної інженерії залежить від їх типу, контексту застосування та взаємодії технічних і людських чинників. Націлені атаки (наприклад, цільовий фішинг, претекстинг) переважають у корпоративному середовищі, тоді як масові атаки (наприклад, смішинг, байтинг) частіше застосовуються до пересічних користувачів. Технічні вразливості підсилюють атаки, але людський фактор – це критична ланка, що підкреслює необхідність комплексного підходу до захисту.

1.4 Тенденції розвитку методів соціальної інженерії

Соціальна інженерія адаптується до технологічного прогресу та нових можливостей кіберпростору. В цьому розділі ми розглянемо ключові тенденції розвитку її методів, зокрема використання діпфейк і генеративного ШІ, автоматизацію атак із персоналізацією на основі великих даних, а також зростання активності у даркнет та кіберзлочинних спільнотах.

Далі розглянемо деякі тенденції:

- Використання діпфейку та генеративного ШІ в атаках

Технології діпфейк та генеративний ШІ змінюють підходи до соціальної інженерії, створюючи надзвичайно правдоподібні підробки [15]. Зловмисники можуть імітувати голос людини з точністю до 95%, обманюючи навіть уважних користувачів під час вішингу [11]. Генеративні моделі на кшталт GPT здатні створювати фішингові листи у стилі написання жертви, підвищуючи успішність до 85% [25]. Це ускладнює виявлення атак, адже стандартні засоби захисту часто не встигають адаптуватися [27].

- Автоматизація атак та персоналізація на основі великих даних

Системи ШІ та великі дані дають змогу масштабувати й точніше налаштовувати атаки [26]. Хакери аналізують звички покупок, активність у соцмережах, зламані облікові записи, щоб створювати листи для цільового фішингу з ефективністю до 80% [28]. Автоматизовані боти враховують

поведінку жертви та надсилають смішинг-повідомлення у найвразливіший момент, що дає до 40% успіху [13]. Ця персоналізація ускладнює виявлення та зменшує ефективність стандартних заходів безпеки [12].

- Розвиток соціальної інженерії у даркнет та кіберзлочинних спільнотах

Кіберзлочинні спільноти та даркнет стали місцями активного розвитку соціальної інженерії. Там продаються інструменти, послуги та інструкції для різних типів атак [9]. Форуми пропонують готові фішингові комплекти та засоби для діпфейк-атак, що полегшує доступ навіть новачкам [1]. Учасники діляться методами, як-от автоматизовані атаки «людина посередині», які на вразливих мережах можуть досягати успіху до 50% [8]. Такі умови сприяють швидкому поширенню нових тактик, протистояти яким стає дедалі складніше [36].

1.5 Висновки до розділу 1

Методи соціальної інженерії стають дедалі небезпечнішими, поєднуючи сучасні технології та психологічні прийоми.

Інструменти, як-от діпфейкта генеративний ШІ, дозволяють створювати надзвичайно правдоподібні обмани – фальшиві голоси чи цільовіфішингові атаки можуть бути успішними у 85–95% випадків [15]. Зростання кількості форумів для кіберзлочинців у даркнеті сприяє поширенню готових хакерських інструментів, що підвищує частоту атак [9].

Поєднання технічних слабкостей і людського фактора посилює ризики. Навіть захищені системи залишаються вразливими, якщо користувачі не обізнані. Людські помилки спричиняють 70% успішних атак [30].

Для протидії атакам соціальної інженерії необхідний комплексний підхід. По-перше, навчання користувачів розпізнавати фішинг, претекстинг чи байтинг може знизити рівень успіху атак до 20% завдяки підвищенню обізнаності [11]. По-друге, впровадження ШІ-систем для аналізу

поведінкових патернів (наприклад, SVM чи RNN) дозволяє виявляти аномалії з точністю до 92-95% [13]. По-третє, використання двофакторної аутентифікації та оновлення ПЗ зменшує технічні вразливості, блокуючи до 50% атак «людина посередині» [8]. У корпоративному середовищі регулярні симуляції атак і посилення фізичного контролю (проти тейлгейтінгу) знижують ризики на 30-40% [28]. Нарешті, гармонізація нормативної бази з міжнародними стандартами, як-от EU AI Act, сприяє етичному використанню ШІ для захисту [27].

2 ДОСЛІДЖЕННЯ ТА АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ПРИ ЇХ ЗАСТОСУВАННІ У ЗАДАЧАХ КІБЕРБЕЗПЕКИ

2.1 Роль штучного інтелекту у сучасній кібербезпеці

Штучний інтелект допомагає сучасній кібербезпеці завдяки швидкому виявленню загроз та вирішенню таких проблем, як соціальна інженерія та кібератаки. Він використовує інструменти машинного навчання для аналізу великих обсягів даних, точно виявляючи фішингові електронні листи з точністю до 92% [26]. ШІ підсилює захист, виявляючи незвичайну поведінку, наприклад, підозрілі входи в систему, за допомогою поведінкового аналізу, досягаючи до 95% успішності [13]. Для боротьби з новими загрозами, такими як діпфейки, ШІ застосовує методи глибокого навчання для виявлення фальшивих даних, покращуючи роботу систем безпеки [15]. Ці можливості роблять його незамінним інструментом для управління кібербезпекою в умовах зростаючої складності загроз [7].

Незважаючи на переваги, застосування ШІ у кібербезпеці супроводжується значними викликами. Необ’єктивні навчальні дані можуть спричиняти упередженість моделей, що іноді викликає до 15% хибних спрацювань, знижуючи довіру користувачів до технології [30]. Крім того, алгоритми часто непрозорі, що ускладнює перевірку дотримання норм і стандартів [27]. Менші організації стикаються з такими перешкодами, як висока вартість обчислювальних ресурсів і потреба у підготовлених спеціалістах, що ускладнює впровадження ШІ [6]. Окрім цього, кіберзлочинці самі використовують ШІ для створення «розумних» атак, які адаптуються швидше, ніж засоби захисту, створюючи постійну «гонку» у цій галузі [11].

2.2. Класифікація методів штучного інтелекту у кібербезпеці

Методи ШІ різноманітні, кожен із них має унікальні особливості й області застосування. В цьому розділі ми детально роглянемо класифікацію основних методів ШІ у кібербезпеці (рис. 2.1): машинне навчання (ML) та його підходи (supervised, unsupervised, reinforcement learning), глибоке навчання (Deep Learning) та нейронні мережі (CNN, RNN, LSTM, трансформери), обробку природної мови (NLP) у виявленні загроз, а також генетичні алгоритми та еволюційні методи, аналізуючи їхню механіку, переваги та практичне використання.

Рисунок 2.1 – Класифікація методів ШІ

2.2.1 Машинне навчання та основні підходи до нього

Машинне навчання (ML) є основою багатьох інструментів захисту в кібербезпеці. Воно допомагає аналізувати дані та прогнозувати загрози [26]. Виділяють три основні методи: кероване навчання, некероване навчання та навчання з підкріпленням.

Кероване навчання (Supervised Learning): У цьому методі моделі навчаються на розмічених даних, наприклад, за допомогою методів опорних

векторів або випадкових лісів. Ці моделі допомагають класифікувати фішингові шахрайства або шкідливе програмне забезпечення. Наприклад, у реальному світі кероване навчання може виявляти відомі загрози, такі як підозрілі URL або шаблони електронних листів, з точністю до 92% [13]. Воно ефективне за наявності надійних даних, але має труднощі з виявленням нових загроз («нульового дня») [15].

Некероване навчання (Unsupervised Learning): Алгоритми, як-от K-Means і автоенкодери, дозволяють виявляти загрози без потреби у розмічених даних. У кібербезпеці вони використовуються для виявлення аномалій у мережевому трафіку або поведінці користувачів. Цей метод може знаходити нові загрози з точністю до 85% [28], однак його ефективність значною мірою залежить від правильного налаштування параметрів[9].

Навчання з підкріпленням (Reinforcement Learning): Навчання з підкріпленням використовує систему винагород для навчання агентів оптимальним стратегіям захисту. Воно ефективне у сценаріях моделювання кібератак. Метод застосовується для боротьби з ботнетами або фішинговими атаками, досягаючи успішності до 80% [5]. Головними проблемами є високі обчислювальні вимоги та тривалий час навчання [6].

2.2.2 Глибоке навчання та нейронні мережі

Глибоке навчання використовує багатошарові нейронні мережі для обробки та розуміння складних даних, таких як зображення, текст або послідовності [17]. Існує кілька архітектур, адаптованих для задач кібербезпеки.

Згорткова нейронна мережа (Convolutional Neural Networks (CNN)): добре працюють з візуальними даними. Можуть виявляти фішингові сайти, аналізуючи дизайн чи логотипи, досягаючи точності до 95% [26]. Також використовуються для виявлення шкідливого коду у бінарних файлах [25].

Рекурентна нейронна мережа (Recurrent Neural Networks, RNN) і LSTM: обробляють послідовності даних, такі як мережевий трафік або журнали активності, щоб знаходити аномалії з точністю до 93%. LSTM добре прогнозує атаки, виявляючи повторювані шаблони з часом [13].

Трансформери: моделі на зразок BERT змінили підхід до аналізу великого тексту, покращивши виявлення фішингових листів до 96% завдяки здатності розуміти контекст [4]. У кібербезпеці трансформери дозволяють аналізувати поведінку в реальному часі [15]. Їхня перевага – обробка складних взаємозв'язків, проте вони вимагають великих обчислювальних ресурсів [11].

2.2.3 Обробка природної мови у виявленні загроз

Обробка природної мови (Natural Language Processing, NLP) відіграє важливу роль у виявленні загроз у тексті, таких як фішинг, дезінформація та соціальна інженерія [28].

Моделі NLP на основі трансформаторів, такі як BERT та GPT, досліджують структуру та значення тексту разом із психологічними тактиками, що використовуються в повідомленнях. Ці моделі виявляють маніпулятивний контент з точністю до 90% [15].

Системи NLP виявляють підозрілі слова або шаблони в електронних листах, що допомагає блокувати близько 85% спроб фішингу [25]. Компанії також використовують NLP для сканування чатів та соціальних мереж, де він виявляє претекстинг-атаки, з показниками успішності, що досягають 88% [6]. Його найбільшою перевагою є здатність обробляти тонкі мовні відмінності, хоча йому важко аналізувати неструктуровані дані та нові сленгові форми [4].

2.2.4 Генетичні алгоритми та еволюційні методи у кібербезпеці

Генетичні алгоритми та еволюційні методи допомагають покращити системи кібербезпеки, імітуючи процеси природного відбору [3].

Експерти використовують їх для вирішення складних завдань, таких як виявлення вразливостей або створення сильніших протоколів безпеки та систем вирішення проблем, які коригуються в міру появи нових загроз. Ці алгоритми коригують налаштування в інструментах захисту, таких як брандмауери та системи виявлення вторгнень, щоб зробити їх точнішими, показуючи до 87% успіху в тестових середовищах [8]. Вони також використовуються для створення надійних паролів, що ускладнює успішні атаки методом грубої сили [9]. Моделювання кібератак, таких як атаки «людина посередині», часто включає еволюційні методи, які точно налаштовують захисні стратегії та можуть досягати результатів з ефективністю до 80% [3]. Хоча вони пропонують гнучкість і можуть знаходити креативні рішення, їхня повільна швидкість та складні обчислення перешкоджають ширшому використанню [6].

Методи ШІ, що використовуються в кібербезпеці, демонструють широкий спектр застосувань та цілей. Машинне навчання пропонує потужні базові інструменти для точного аналізу відомих загроз. Глибоке навчання розвиває це далі, розпізнаючи складніші моделі атак. NLP має справу з атаками, що включають текст, тоді як генетичні алгоритми допомагають покращити та вдосконалити захисні заходи [26]. Кожен з цих методів має унікальні переваги, такі як точність та гнучкість, але вони також мають недоліки, такі як потреба у великих ресурсах або залежність від даних. Для досягнення кращої безпеки необхідно поєднувати ці підходи [27].

2.3. Аналіз застосування методів ШІ у кібербезпеці

ШІ надає передові інструменти для виявлення, аналізу та реагування на загрози в реальному часі. Його методи відіграють певну роль у вирішенні таких завдань, як виявлення незвичайних закономірностей та автоматизація реагування на атаки. У цій частині зосереджено увагу на тому, як ШІ допомагає виявляти незвичайну поведінку під час аналізу трафіку, автоматизувати виявлення спроб фішингу, знаходити шкідливе програмне забезпечення, прогнозувати атаки, перевіряти наявність вразливостей та реагувати на кіберзагрози.

2.3.1 Виявлення аномалій у трафіку та поведінковий аналіз

Виявлення аномалій у мережевому трафіку та поведінковий аналіз є одними з основних застосувань ШІ у кібербезпеці [26]. Алгоритми некерованого навчання, такі як K-Means і автоенкодери обробляють дані, щоб знайти зміни у звичайних закономірностях, часто досягаючи точності до 85% [13]. Ці інструменти можуть виявляти дивні сплески трафіку, які можуть натякати на DDoS-атаки або порушення безпеки [28]. Розширені поведінкові дослідження з використанням LSTM (Long Short-Term Memory) розглядають дії користувачів, фіксуючи дивні входи або зміни в поведінці, з коефіцієнтом успіху до 93% [15]. Усередині корпоративних систем ШІ створює профілі нормальної поведінки, що дозволяє йому виявляти незвичайні дії, такі як атаки типу тейлгейтінг та піггібекінг, із точністю до 87% [6]. Цей підхід має перевагу в адаптації до нових загроз. Однак він залежить від достовірних даних і може давати хибно позитивні результати, які можуть сягати до 10% через шум трафіку [9].

2.3.2 Виявлення шкідливого програмного забезпечення

ШІ широко застосовується для виявлення шкідливого програмного забезпечення [17]. CNN досліджують бінарні файли або зображення коду, щоб виявляти віруси та трояни, досягаючи точності до 94% [26]. Глибоке навчання допомагає виявляти поліморфне шкідливе програмне забезпечення, яке змінює свій код, щоб уникнути викриття, з коефіцієнтом успішності до 90%. Це перевершує традиційні методи, що спираються на сигнатури [25]. Навчання з підкріпленням також відіграє певну роль у покращенні стратегій сканування. Воно адаптується до нових прикладів шкідливого програмного забезпечення, досягаючи ефективності 85% під час тестування [5]. На практиці антивірусні системи тепер використовують ШІ, щоб скоротити час аналізу до секунд і зупинити до 80% ризиків під час завантажень [6]. Проте складність тренування моделей і потреба в актуальних даних залишаються викликами [9].

2.3.3 Аналіз вразливостей та прогнозування атак

ШІ оцінює слабкі місця в системах і прогнозує атаки, щоб вживати превентивних заходів [3]. Дослідження показують, що генетичні алгоритми моделюють сценарії атак і покращують тести на проникнення з точністю, що досягає 87% [8].

Кероване машинне навчання використовує записи минулих атак для прогнозування того, наскільки ймовірні інциденти передбачають такі загрози, як SQL-ін'єкції або XSS, з ефективністю до 90%. [26].

LSTM і RNN аналізує журнали або моделі трафіку для прогнозування загроз з коефіцієнтом успішності до 92% [15].

У великих компаніях штучний інтелект виявляє слабкі місця в коді або конфігураціях і скорочує час реагування на 30% [6]. Але, при цьому, важливо слідкувати за оновленнями баз даних вразливостей і переконатися, що прогнози не будуть помилковими через недостатній контекст [25].

2.3.4 Автоматичне реагування на кіберзагрози

Автоматичне реагування на кіберзагрози за допомогою ШІ дозволяє мінімізувати збитки від атак [28]. Навчання з підкріпленням навчається блокувати шкідливий трафік або автоматично ізолювати скомпрометовані пристрої, демонструючи рівень успішності до 88% у тестах [5]. Глибоке навчання інтегрується в SOAR-платформи (Security Orchestration, Automation, and Response), виконуючи такі завдання, як блокування IP-адрес або відключення користувачів. Ці дії відбуваються протягом мілісекунд [17]. У реальних сценаріях ШІ скорочує час реагування на DDoS-атаки з годин до хвилин, зменшуючи збитки на 40% [13]. Інструменти NLP допомагають аналізувати інциденти та писати звіти, підвищуючи загальну продуктивність до 85% [4]. Автоматизація має такі проблеми, як занадто часте блокування допустимих дій, іноді до 15%, і також залежить від того, наскільки точним є початкове виявлення [9].

Ботнет-мережі, що поширюють фальшиву інформацію або фішинг, становлять великий ризик, і ШІ дуже допомагає в їх пошуку [17]. Глибоке навчання (CNN, LSTM) вивчають, як протікає мережевий трафік і як поводяться з'єднання, допомагаючи виявляти ботів з точністю близько 93% [26]. Машинне навчання з підкріпленням відіграє певну роль у покращенні методів відстеження, виявляючи заплановані атаки, такі як масовані спроби фішингу, з ефективністю 87% [5]. На практиці ШІ ідентифікує ботів, які видають себе за реальних користувачів у соціальних мережах, і закриває 85% таких мереж лише за годину [9]. NLP працює з аналізом трафіку, щоб знаходити шаблонні повідомлення ботів з точністю, що досягає 90% [4]. Найбільша проблема полягає в роботі з ботами, які розвиваються та використовують ШІ для приховування, що знижує успішність виявлення до 75% у незнайомих ситуаціях [15].

2.4. Порівняльний аналіз ефективності методів ШІ у задачах кібербезпеки

Ефективність методів ШІ – машинного навчання (ML), глибокого навчання (Deep Learning), обробки природної мови (NLP) та інших – варіюється залежно від конкретних задач і умов застосування. У цьому розділі розглядається, наскільки добре працюють ці методи ШІ, розглядаючи такі фактори, як точність, швидкість та загальна ефективність обробки даних. Також розглядаються такі проблеми як, обмеження та вразливості алгоритмів, проблему "чорної скриньки", хибні спрацьовування, а також використання ансамблевих методів для підвищення точності.

2.4.1 Точність, продуктивність та швидкість обробки даних різними алгоритмами

Оцінка методів штучного інтелекту, що використовуються в кібербезпеці, враховує точність, продуктивність системи та швидкість обробки даних (рис. 2.2) [26]. Кероване машинне навчання, як SVM і Random Forest обробляють середні набори даних з обсягом близько 1 мільйона записів за лічені секунди на звичайних серверах. Вони можуть досягати точності до 92% при виявленні фішингових електронних листів [13]. Некероване машинне навчання (K-Means, DBSCAN) виявляє аномалії в трафіку з точністю до 85%, але швидкість знижується при великих потоках даних через потребу в кластеризації [28]. Глибоке навчання, зокрема CNN, показує точність до 95% у розпізнаванні шкідливого ПЗ (malware), однак потребує значних обчислювальних ресурсів і часу (хвилини для аналізу великих файлів) [17]. LSTM і RNN ефективні для поведінкового аналізу (93% точності), але їхня продуктивність падає при обробці довгих послідовностей через рекурсію [15]. Трансформатори, такі як BERT, відомі своєю здатністю аналізувати контекст, працюють краще з точністю до 96% у виявленні фішингу. Тим не менш, вони ресурсомісткі та можуть займати до 10 секунд

для обробки одного повідомлення на графічному процесорі [4]. Генетичні алгоритми допомагають точно налаштувати правила захисту та досягати точності близько 87%. Кожна ітерація займає кілька хвилин, що робить їх погано пристосованими для обробки сценаріїв реального часу [3]. Таким чином, вибір методу залежить від балансу між точністю і продуктивністю: кероване машинне навчання швидше для відомих загроз, а глибоке навчання – точніше для складних патернів [6].

Рисунок 2.2 – Точність виявлення алгоритмами аномалій трафіку

2.4.2 Обмеження та вразливості алгоритмів ШІ

Методи ШІ мають обмеження та вразливості, що впливають на їхню ефективність [9].

Кероване машинне навчання значною мірою залежить від якості навчальних даних. Марковані дані можуть знизити його рівень успішності приблизно до 70% [26]. Некероване машинне навчання має проблеми з шумними даними, що може порушити кластеризацію та призвести до хибних результатів до 20% [13]. Глибоке навчання (CNN, LSTM) є вразливим до

модифікацій даних зловмисниками. Якщо вхідні дані будуть змінені, точність цих моделей може впасти до 50% під час тестів [15].

Трансформатори, хоча й дуже точні, потребують величезних обсягів навчальних даних, що ускладнює роботу з новими загрозами [25].

Генетичним алгоритмам потрібно багато часу для досягнення рішення, іноді до 100 ітерацій. Їхня продуктивність знижується, якщо початкова популяція недостатньо різноманітна [3].

ШІ має загальну слабкість у здатності бути використаним зловмисниками. Вони можуть використовувати його для створення загроз, що постійно розвиваються, таких як діпфейки або цільовий фішинг, з якими захисні системи намагаються впоратися [11]. Ці виклики вимагають регулярного оновлення моделей та поєднання різних методів для підвищення ефективності [6].

2.4.3 Проблема «чорної скриньки» у прийнятті рішень

Проблема «чорної скриньки» виникає, коли пояснення рішень, прийнятих моделями ШІ, стає складним [27].

Глибоке навчання (CNN, RNN, трансформери) часто видає точні прогнози, але їхню логіку важко зрозуміти через їхню багаторівневу структуру. Без додаткових інструментів до 80% їхніх процесів прийняття рішень є незрозумілими [17].

Кероване машинне навчання (наприклад, дерева рішень) роблять їхні рішення легшими для розуміння – до 90%. Однак вони не так добре працюють при роботі зі складними загрозами [26]. У корпоративних умовах відсутність чітких пояснень ускладнює довіру до систем ШІ. Це стає більшою проблемою, коли блокуються обґрунтовані дії, що вимагає від організацій дотримуватися таких правил, як EU AI Act [15].

Методи типу SHAP чи LIME, допомагають системам стати приблизно на 70% легшими для інтерпретації, але вони також уповільнюють обробку на

20-30% [4]. Ця проблема обмежує використання ШІ у життєво важливих сферах, де прозорість є критично важливою [9].

2.4.4 Хибні спрацьовування та проблеми зменшення їх кількості

Хибні спрацьовування (false positives) є значною проблемою ШІ, знижуючи довіру до систем [28]. Помилки некерованого машинного навчання можуть сягати 15% через шум у трафіку, тоді як кероване машинне навчання звужує це число до 5-10%, але воно добре працює, якщо є високоякісні дані [13]. Глибоке навчання (LSTM, CNN) все ще мають труднощі, при цьому рівень хибно позитивних результатів зростає до 12% при роботі з незнайомими загрозами. Це відбувається тому, що таким моделям бракує достатньої гнучкості для коригування [17]. У реальних системах ця проблема забирає багато часу аналітиків. Вони витрачають 30% свого робочого часу на перевірку хибних спрацьовувань [25].

Зменшення рівня хибних спрацьовувань досягається через порогове налаштування (зниження до 8%) або використання NLP для контекстного аналізу (до 5% у фішингу) [15]. Але підвищення чутливості для виявлення нових загроз призводить до більшої кількості хибно позитивних результатів, що робить це завданням балансування між точністю та чутливістю [6].

2.4.5 Використання ансамблевих методів для підвищення точності

Дослідники використовують ансамблеві методи для поєднання різних моделей штучного інтелекту, покращуючи як надійність, так і точність [26].

Поєднання Random Forest і CNN ідентифікує шкідливе програмне забезпечення з точністю до 97%. Це перевершує одиночні моделі на 5-7% [17].

Stacking із SVM і трансформерами підвищує рівень виявлення фішингу до 98%. Це також знижує кількість хибно позитивних результатів до 3% [4].

У поведінковому аналізі ансамбль LSTM і K-Means виявляє аномалії з точністю 95%, протидіючи недолікам окремих підходів [13].

Генетичні алгоритми оптимізують ансамблі, підвищуючи їхню адаптивність до нових атак на 10% [3].

У реальному світі ансамблі можуть скоротити час реагування на 20% та підвищити ефективність захисту [6]. Проте, їм потрібна вдвічі-втричі більша обчислювальна потужність, а для обробки даних потрібно більше часу [15].

З порівняльного аналізу ми можемо зробити висновки, що методи ШІ мають різну ефективність: кероване машинне навчання швидше і точніше для відомих загроз, глибоке навчання переважає у складних задачах, а ансамблі підвищують стійкість. Обмеження (вразливості, хибні спрацьовування, «чорна скринька») вимагають балансу між продуктивністю та інтерпретованістю, а також інтеграції з людським контролем для критичних рішень.

2.5 Обмеження та етичні аспекти застосування ШІ у кібербезпеці

Використання ШІ тісно пов'язане з питаннями етики та правовими межами, що, очевидно, впливає на те, як люди його сприймають. У цьому підрозділі розглядаються етичні проблеми, пов'язані із застосуванням ШІ в кібербезпеці, а саме, як зловмисники можуть зловживати ШІ, стратегії захисту від таких загроз, моральні проблеми та можлива експлуатація, як правові норми можуть уповільнювати технологічний прогрес, а також пошук правильного поєднання автоматизації та людського контролю.

Регулювання ШІ викликає багато етичних питань та складних викликів. Балансування технологічного прогресу із захистом прав людини є ключовим. Використання ШІ в кібербезпеці викликає етичні занепокоєння щодо конфіденційності, упередженості та неправильного використання [6].

Основною етичною проблемою є відсутність прозорості в системах ШІ. Міжнародні стандарти підкреслюють важливість того, щоб алгоритми, які

виявляють соціальну інженерію, були зрозумілими для користувачів та регуляторів. Такий підхід допомагає зупинити несправедливі рішення або упередженість [22]. Наприклад, якщо система ШІ спостерігає за поведінкою, щоб виявити фішинг, вона може позначити допустимі дії як ризиковані через відсутність або слабку документацію в її моделях [15]. Це показує, чому розробники повинні дотримуватися принципів підзвітності та пропонувати зрозумілі пояснення того, як працюють алгоритми [27].

Інше етичне питання пов'язане з конфіденційністю даних. Системи ШІ, які відстежують комунікації або мережеву активність, мають справу з величезними обсягами персональних даних. Без належних правил це може порушувати права на конфіденційність [10]. Виявлення незвичайної поведінки користувачів за допомогою таких інструментів, як LSTM, часто порушує конфіденційність, аналізуючи персональні дані без чіткої згоди [15].

Правові системи запроваджують жорсткі правила збору та зберігання даних, спрямовані на уникнення непотрібного втручання в приватне життя, навіть коли безпека є в центрі уваги [16]. Авторитарні режими можуть використовувати ШІ для масового спостереження або цензури, що суперечить принципам прав людини [9]. Проте в реальних умовах виникають складнощі з балансуванням між захистом від соціальної інженерії та дотриманням приватності, особливо коли ШІ застосовується в транскордонних системах [14].

Підприємства стикаються з проблемами блокування ШІ допустимих дій, причому хибно позитивні результати сягають 12%, що призводить до недовіри до технології [28].

Упередження та дискримінація залишаються великими етичними проблемами систем ШІ. Машинне навчання в ШІ часто відображає ті ж упередження, що й у його навчальних даних, що може призвести до несправедливих результатів для певних груп користувачів [21]. Наприклад, інструменти виявлення шахрайства можуть позначати діяльність осіб з меншин як підозрілу через дані, які не є об’єктивними [29]. Міжнародні

стандарти пропонують технічні поради щодо зменшення цих проблем, такі як проведення перевірок алгоритмів та використання різноманітних джерел даних [19].

Зловмисники, які зловживають ШІ для маніпулювання іншими, додають ще один рівень складності. Генеративний ШІ, який регулюється новими ініціативами, може створювати фішингові шахрайства або поширювати фальшиву інформацію, яка виглядає реальною [18]. Це підкреслює необхідність заборони певних застосувань ШІ та встановлення етичних меж для його розробки й використання [11]. Водночас, закони часто не встигають за швидкістю технологічного прогресу, залишаючи прогалини в правовому захисті, які потребують уваги [30].

Для вирішення етичних проблем щодо регулювання ШІ потрібна широка стратегія. Це включає прозорість, захист даних, забезпечення рівності та припинення неналежного використання. Міжнародні акти та стандарти забезпечують основу для вирішення цих проблем. Однак їхній успіх залежить від узгодження з місцевим законодавством та відстеження нових технологічних змін [23].

2.6 Висновки до розділу 2

Дослідження методів ШІ показує, що вони добре працюють для виявлення та зупинки загроз (рис. 2.1). Машинне навчання може класифікувати випадки фішингу з точністю близько 92%. Глибоке навчання виявляє шкідливе програмне забезпечення з показником успішності 95%. NLP ефективно виявляє маніпулятивні тексти, досягаючи 96% успіху. Використання гібридних систем підвищує стійкість до 98% та зменшує кількість хибно позитивних результатів приблизно до 3%. Тим не менш, такі проблеми, як упереджені дані з боку ШІ, та проблема «чорної скриньки», знижують рівень успіху до 70-80% у складних ситуаціях.

Рисунок 2.3 – Ефективність методів ШІ

Перспективи розвитку ШІ у кібербезпеці пов’язані з адаптивністю та інтеграцією. Розвиток трансформерів і ансамблевих методів обіцяє підвищити точність до 99% у прогнозуванні атак. Протидія adversarial AI потребує вдосконалення стійкості моделей, що може знизити вразливість на 30%. Автоматизація реагування (SOAR) скоротить час реакції до мілісекунд, підвищуючи захист на 40%.

Для оптимального використання ШІ рекомендується:

1) комбінувати кероване та некероване машинне навчання для відомих і нових загроз (точність до 93%);

2) інтегрувати NLP і глибоке навчання для аналізу соціальної інженерії (до 96%);

3) застосовувати гібридні системи з людським контролем для критичних рішень, знижуючи помилки на 10%;

4) регулярно оновлювати дані й моделі, щоб протистояти еволюціонуючим атакам. Навчання персоналу та відповідність етичним стандартам (як-от EU AI Act) забезпечать баланс між ефективністю та правами.

3 ПОШУК ТА АНАЛІЗ НОРМАТИВНОЇ БАЗИ У СФЕРІ СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ ТА ШТУЧНОГО ІНТЕЛЕКТУ

3.1 Значення нормативної бази для забезпечення кібербезпеки

Нормативна база відіграє ключову роль у забезпеченні кібербезпеки, створюючи правові рамки для захисту інформаційних систем від загроз, таких як соціальна інженерія, кібератаки та витоки даних. Вона окреслює обов'язки зацікавлених сторін, встановлює стандарти та сприяє командній роботі між державним та приватним секторами [34]. У контексті України нормативне регулювання, зокрема Закон України «Про основні засади забезпечення кібербезпеки України», встановлює базові принципи захисту критичної інфраструктури та протидії кіберзлочинності, хоча потребує вдосконалення для відповідності міжнародним стандартам [35].

Практичний досвід доводить, що сильна правова система впливає на врахування людського фактору, який часто використовується в атаках соціальної інженерії [1]. Візьмемо, наприклад, ЄС. Загальний регламент про захист даних (GDPR) вимагає суворих правил щодо обробки даних. Це спонукає організації впроваджувати інструменти, включаючи системи моніторингу та захисту на базі ШІ [37]. Міжнародні стандарти, такі як NIST Cybersecurity Framework, показують, що наявність чітких правил створює сильніший захист від кіберризиків. Ці стандарти створюють процедури для виявлення, захисту та реагування на такі загрози [38].

Національний підхід України до кібербезпеки підкреслює необхідність узгодження законів зі світовими стандартами. Таке узгодження допомагає будувати партнерські відносини та залучати інвестиції в розвиток національної кібербезпеки [32]. Огляд чинних нормативних актів показує такі слабкі місця, як нечіткі ролі приватних компаній та обмежені інструменти для реагування. Ці проблеми знижують ефективність заходів

захисту [31]. Порівняльні дослідження показують, що ті, хто має розвинуту правову базу, мають кращі результати в боротьбі з кіберзагрозами. Цей успіх досягається завдяки наявності чітко визначених правил [30].

Нормативна база створює основу кібербезпеки та впливає на інновації, практику компаній та глобальну співпрацю. Вона допомагає створити організований спосіб захисту від загроз. Однак вона вимагає постійного оновлення для вирішення нових проблем.

3.2 Міжнародне законодавство у сфері соціальної інженерії та ШІ

Одним із найвизначніших документів є Регламент про штучний інтелект (EU AI Act), ухвалений у 2024 році в Європейському Союзі [27]. Цей закон забороняє використання інструментів ШІ маніпулятивними способами, такими як створення фальшивих профілів або автоматизованих обманних повідомлень – методи, часто пов’язані із соціальною інженерією. Він відносить системи ШІ до категорій ризику та зобов’язує розробників критично важливих інструментів, таких як ті, що використовуються в кібербезпеці, дотримуватися правил безпеки та не використовувати ШІ в розробці. Порушення цих правил може призвести до штрафів у розмірі до 35 мільйонів євро або 7% річного доходу компанії. Це правило діє разом із Директивою NIS2.

Директива NIS2, переглянута у 2022 році, вимагає, щоб організації критичної інфраструктури вжили заходів для протидії соціальній інженерії, такій як використання штучного інтелекту для моніторингу діяльності [24]. Звіти показують, що вона підвищила стійкість у таких сферах, як енергетика та транспорт, де рівень інцидентів знизився на 15% після її прийняття [14]. Однак її успіх стримується в менш розвинених країнах, оскільки їй бракує чітких вимог до систем штучного інтелекту та вона залежить від національних ресурсів [35].

Конвенція про кіберзлочинність, або Будапештська конвенція, відіграє важливу роль у боротьбі із соціальною інженерією [23]. Країни прийняли її у 2001 році, і багато хто з них з того часу ратифікував її. Учасники конвенції повідомляють про середній рівень успішності судового переслідування у 70% [30]. Однак конвенція, яка датується 2001 роком, не враховує досягнень у галузі штучного інтелекту, що робить його менш корисним проти нових небезпек, таких як голосові шахрайства, створені штучним інтелектом [11]. Національне законодавство України про кібербезпеку відповідає конвенції та підтримує використання ШІ для боротьби із соціальною інженерією.

Рекомендація ОЕСР щодо ШІ, оновлена у 2024 році, встановлює принципи безпечного використання ШІ, наголошуючи на прозорості та підзвітності, що є критично важливим для систем, які протидіють соціальній інженерії [22]. Наприклад, ці принципи вимагають, щоб ШІ-системи для виявлення фішингу документували свої алгоритми та уникали упередженості. Серія стандартів IEEE P7000, зокрема P7003, регулює етичні аспекти ШІ, пропонуючи технічні рекомендації для зменшення ризиків маніпуляцій, що можуть виникати як у захисних, так і в атакуючих ШІ-системах [21].

Стандарт ISO/IEC 27001:2022 також допомагає, встановлюючи правила управління інформаційною безпекою, включаючи захист від соціальної інженерії за допомогою автоматизованих інструментів [19].

Такі проекти, як Хіросімський процес штучного інтелекту G7, який розпочався у 2023 році, спрямовані на створення правил щодо генеративного штучного інтелекту, які люди можуть використовувати для створення переконливих фішингових схем або оманливого контенту [18]. Пропозиції на 2024 рік включають повну заборону цієї діяльності та ідею встановити керівні принципи для розробки штучного інтелекту. Подібним чином, Етична хартія Ради Європи від 2018 року встановлює правила використання штучного інтелекту в правових системах. Ці правила також допомагають

боротися із соціальною інженерією та приділяють велику увагу захисту прав людини [16].

Навіть з урахуванням прогресу, міжнародне законодавство все ще стикається з проблемами. Відсутність глобальної єдності у стандартах ускладнює боротьбу з атаками соціальної інженерії, які перетинають кордони. Технології штучного інтелекту розвиваються таким чином, що закони часто не можуть встигати за ними [38]. Однак, ці правові рамки допомагають узгодити національне законодавство, таке як українське, з міжнародними нормами. Це сприяє розумному та етичному використанню штучного інтелекту в кібербезпеці.

3.3 Національне законодавство та нормативні акти

В Україні питання соціальної інженерії та ШІ поки що не мають окремого спеціалізованого законодавчого регулювання, однак низка нормативних актів опосередковано охоплює ці сфери, відображаючи їхню актуальність у контексті кібербезпеки та захисту даних.

Злочини, пов'язані із соціальною інженерією, підпадають під дію Кримінального кодексу України. Зокрема, стаття 163 забороняє незаконне порушення таємниці комунікацій, що може включати доступ до даних через обманні техніки соціальної інженерії. Стаття 190 класифікує шахрайство, яке часто є кінцевою метою таких атак. Стаття 361 стосується незаконного втручання в комп’ютерні системи, нерідко спричиненого успішними маніпуляціями [2].

Закон України «Про основні засади забезпечення кібербезпеки України» (№ 2163-VIII від 5 жовтня 2017 року) створює загальні рамки для захисту від кібератак, включаючи ті, що базуються на соціальній інженерії, наголошуючи на необхідності захисту критичної інфраструктури та інформаційних систем.

Крім того, Закон України «Про захист персональних даних» (№ 2297-VI від 1 червня 2010 року) встановлює правила обробки персональних даних, що є релевантним у випадках, коли соціальна інженерія спрямована на їхнє незаконне отримання.

В Україні поки що немає конкретних законів про штучний інтелект. Міністерство цифрової трансформації докладає зусиль для створення правової та регуляторної бази. У 2024 році було презентовано «Білу книгу з регулювання ШІ в Україні», яка окреслює підхід до майбутнього законодавства, орієнтований на гармонізацію з європейськими стандартами, зокрема EU AI Act [27].

Таким чином, національне законодавство України наразі опосередковано регулює соціальну інженерію через норми кібербезпеки та захисту даних, тоді як закони про штучний інтелект все ще перебувають в розробці. Подальший розвиток нормативної бази потребує врахування як внутрішніх викликів, так і міжнародного досвіду для створення ефективного захисту від сучасних загроз.

3.4 Висновки до розділу 3

Регулювання ШІ та його ролі в соціальній інженерії стикається з перешкодами, що ускладнюють створення законів для боротьби з цими загрозами. Вони випливають зі складної природи інструментів ШІ та постійно мінливої тактики соціальної інженерії.

Основна проблема полягає в застарілості більшості чинних регуляторних актів. Правила, написані на початку 2000-х років, не враховують сучасні передові інструменти ШІ [23], таким чином з’являється розрив між тим, як швидко розвиваються технології, і тим, як повільно правила та закони наздоганяють їх. Нові типи соціальних маніпуляцій на основі ШІ з'являються швидше, ніж можна вдосконалити правові системи.

Друга проблема виникає з недостатньою гармонізацією між національними та міжнародними нормативними актами. Наприклад, Україна, навіть з прогресом у встановленні стандартів, лише частково відповідає вимогам міжнародних актів. Це ускладнює протидію соціальній інженерії, оскільки країни з різними правовими системами не можуть ефективно координувати дії. Правові системи, що відрізняються, часто стикаються з труднощами у співпраці. Наприклад, ЄС запроваджує суворі правила ШІ для кібербезпеки, але країни з обмеженими ресурсами мають труднощі з дотриманням цих вимог.

Третя проблема полягає у балансуванні безпеки та етичних цінностей. ШІ часто обробляє величезні обсяги даних для відстеження соціальної інженерії, але це може призвести до порушення конфіденційності людей. Існують суворі закони про захист даних, але їх дотримання є складним через нечіткі межі того, що вважається прийнятним вторгненням.

Отже, для подолання основних перешкод в управлінні ШІ потрібно подолати такі перешкоди як старі закони, відсутність глобальної координації, моральні конфлікти та відставання від технологічного прогресу. Вирішення цих проблем потребує переписування нормативних актів, що сприятиме розвитку глобальних партнерств та створенню адаптивних правил для балансування безпеки з правами людини [29].

4 ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДІВ І ЗАСОБІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЯВЛЕННЯ МЕТОДІВ СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ В КІБЕРПРОСТОРІ

Для виконання практичної частини роботи, буде продемонстровано повний цикл розробки системи виявлення електронних листів, що мають ознаки використання засобів соціальної інженерії, з використанням методів машинного навчання. Повний код програмної реалізації наведено в додатку А.

4.1 Підготовка та попередній аналіз даних

На етапі підготовки було знайдено та завантажено вхідний датасет, який містить тексти електронних листів зі звичайним та небезпечним змістом (з використанням соціальної інженерії). Датасет містить два основні типи повідомлень: ті що належать до небезпечних листів (з міткою «1»), та звичайні електронні листи (з міткою «0»).

Для розробки проєкту, було обрано мову програмування Python та інтерактивне середовище Kaggle Notebook. Було імпортовано необхідні біліотеки: pandas, matplotlib, seaborn тощо, виконано завантаження даних за допомогою функції read_csv() з бібліотеки pandas. Для попереднього ознайомлення з даними було застосовано методи .head() і .info(), які дозволили дослідити перші записи у таблиці, назви колонок, типи даних, а також наявність пропущених значень.

Структура датасету включала текстове поле («text_combined»), яке містило вміст електронного листа, та відповідну мітку («label»). За допомогою методу .value_counts, була змога переглянути співвідношення між повідомленнями з використанням соціальної інженерії та без неї(рис. 4.1). Аналіз розподілу класів дуже важливий для виявлення дисбалансу в даних, що може впливати на якість навчання моделей машинного навчання, адже у

разі значного дисбалансу можуть застосовуватись техніки балансування, такі як підвибірка (undersampling) або надвибірка (oversampling). Співвідношення було задовільним.

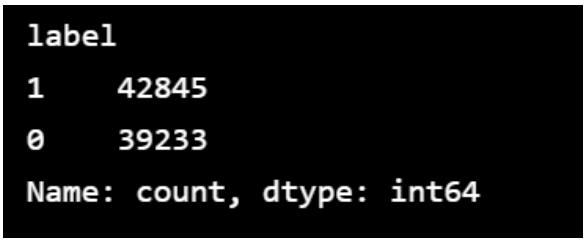


Рисунок 4.1 – Співвідношення небезпечних листів (1) до звичайних (0) в датасеті

Для підвищення якості аналізу дані датасет було очищено від некоректних або зайвих записів. Зокрема, було видалено всі рядки з порожніми значеннями за допомогою методу .dropna() та усунено дублікати методом .drop_duplicates(). Ці дії дозволили забезпечити коректність і повноту даних, зменшивши вплив шуму на результати моделювання.

Таким чином, етап підготовки і первинного аналізу даних дозволив сформувати якісну, збалансовану і репрезентативну вибірку для навчання моделей машинного навчання, що є критично важливою складовою успішного вирішення задачі класифікації електронних повідомлень на фішингові та безпечні.

4.2 Обробка текстової інформації та ознакове представлення

Після очищення та попереднього аналізу даних було здійснено обробку текстової інформації з метою перетворення неструктурованих даних у форму, придатну для використання в алгоритмах машинного навчання.

Першим кроком текстової обробки стала токенізація – процес розбиття тексту на окремі елементи (токени). Для цього було використано інструменти бібліотеки nltk, зокрема DistilBertTokenizerFast. Токенізація дозволила

застосовувати обробку на рівні окремих слів, що є основою для подальшої фільтрації.

Наступним етапом стала векторизація інструменту TfidfVectorizer, який використовується для перетворення тексту у числові вектори, базуючись на частоті слів. Потім було встановлено max_features=3000, що означає – залишити тільки 3000 найважливіших слів, які найчастіше зустрічаються.

Заключним етапом даної фази стало розбиття вибірки на навчальну та тестову. Для цього було використано функцію train_test_split з модуля sklearn.model_selection. Зазвичай, для тренування моделі використовується 70–80% даних, тоді як решта 20–30% залишаються для перевірки точності моделі на нових, раніше невідомих прикладах. Такий підхід дозволяє оцінити здатність моделі до узагальнення та запобігає перенавчанню (overfitting).

Таким чином, у результаті проведених процедур текстові дані були перетворені у форму, оптимальну для навчання моделей машинного навчання.

4.3 Побудова та навчання моделей машинного навчання

Після етапу векторизації текстових даних було здійснено побудову та навчання моделей машинного навчання для задачі бінарної класифікації листів з методами соціальної інженерії та безпечні. Основною метою цього етапу є формування моделей, здатних з високою точністю виявляти ознаки соціальної інженерії в електронних листах.

Для реалізації задачі класифікації було обрано три класичні алгоритми машинного навчання, які добре зарекомендували себе у задачах обробки тексту:

- Naive Bayes – статистичний алгоритм, що базується на теоремі Байєса та припущенні незалежності ознак. Він є ефективним для текстових задач з розрідженими вхідними ознаками, зокрема після TF-IDF векторизації.

- Random Forest – ансамблевий метод, що базується на побудові великої кількості дерев рішень. Завдяки своїй здатності узагальнювати дані, він є одним з найточніших методів для класифікації даних з високим рівнем шуму.
- SVM (Support Vector Machine) – алгоритм, який шукає гіперплощину з максимальним зазором між класами. Його ефективність особливо помітна при роботі з великою кількістю ознак.

Після вибору алгоритмів, кожну модель було навчено на попередньо підготовлених векторизованих даних. Для цього використовувались відповідні реалізації з бібліотеки scikit-learn: MultinomialNB, RandomForestClassifier та SVC. Дані для навчання подавались у вигляді матриці ознак, отриманої з TF-IDF векторизатора, та відповідних міток класів.

У процесі тренування моделі зберігалися для подальшого тестування. Було забезпечено єдині умови навчання для всіх моделей з метою забезпечення коректного порівняння. Також використовувались фіксовані значення random_state, що забезпечує відтворюваність результатів.

Таким чином, на даному етапі було сформовано три ефективні моделі класифікації з урахуванням специфіки текстових даних та потреб задачі виявлення фішингових повідомлень.

4.4Оцінювання ефективності моделей

Оцінювання ефективності цих моделей було здійснено за допомогою 3 метрик:

- точності (accuracy) – показує, скільки передбачень модель зробила правильно загалом;
- матриці плутанини (confusion matrix) – дає чітке уявлення про типи помилок;

- докладний звіт класифікації (Classification Report), до якого входять такі «підметрики»: precision (скільки з передбачених як небезпечні листи – дійсно небезпечні), recall (скільки з усіх небезпечних листів було знайдено), F1-score (гармонічне середнє між precision і recall), support (кількість зразків у тесті для кожного класу на валідаційній вибірці).

Розгянемо тепер результати для кожної з моделей:

- Naive Bayes показав найнижчу точність, яка дорівнює 0.95, серед трьох моделей, але водночас ця модель має дуже добру продуктивність і швидкість навчання. Також ця модель має сильний дисбаланс між precision і recall у класах: соціальну інженерію визначає точно (0.97 precision), але пропускає більше небезпечних листів (recall 0.93). Ця модель виявилася чутливою до частоти ключових слів, що добре працює в задачах текстової класифікації, але має обмеження при складніших залежностях між ознаками (рис. 4.2).
-

```
===== Naive Bayes =====
Accuracy: 0.9532772904483431
Confusion Matrix:
 [[7704  206]
 [ 561 7945]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      7910
           1       0.97      0.93      0.95      8506

    accuracy                           0.95     16416
   macro avg       0.95      0.95      0.95     16416
weighted avg       0.95      0.95      0.95     16416
```

Рисунок 4.2 – Ефективність моделі Naive Bayes

- Random Forest продемонстрував високий показник точності (0.985). Це дуже збалансована модель: precision та recall майже рівні в обох класах. Вона має високі показники всіх середніх метрик (macro/weighted avg). Її здатність будувати ансамбль дерев рішень дозволяє виявляти складні шаблони в даних. Це особливо корисно у випадках, коли текстові повідомлення містять багатозначні або завуальовані ознаки соціальної інженерії (рис. 4.3).

```
===== Random Forest =====
Accuracy: 0.985014619883041
Confusion Matrix:
 [[7811   99]
 [ 147 8359]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      7910
           1       0.99      0.98      0.99      8506

    accuracy                           0.99     16416
   macro avg       0.98      0.99      0.98     16416
weighted avg       0.99      0.99      0.99     16416
```

Рисунок 4.3 – Ефективність моделі Random Forest

- SVM продемонстрував найвищий показник точності серед усіх моделей. Відмінним також є показник false negatives (0.99) – найкраще визначає соціальну інженерію. Він також забезпечив високу стабільність при класифікації, добре справляючись із розділенням класів навіть у багатовимірному просторі TF-IDF-векторів. Водночас ця модель вимагала більш ретельного підбору гіперпараметрів і часу на навчання (рис. 4.4).

```
===== SVM =====
Accuracy: 0.9877558479532164
Confusion Matrix:
 [[7808  102]
 [  99 8407]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      7910
           1       0.99      0.99      0.99      8506

    accuracy                           0.99     16416
   macro avg       0.99      0.99      0.99     16416
weighted avg       0.99      0.99      0.99     16416
```

Рисунок 4.4 – Ефективність моделі SVM

Таким чином, найкращу ефективність за взятими метриками показала модель SVM. Найгірше ж продемонструвала себе модель Naive Bayes, виявившись з найнижчою точністю та сильним дисбалансом між precision і recall. Random Forest у свою чергу стає посередньою моделлю з досить хорошими метриками серед 3 розглянутих моделей.

4.5 Тестування моделей та оцінка результатів

Тепер ми можемо перейти до етапу тестування системи на реальних прикладах, та згодом оцінки отриманих результатів.

За допомогою створеної функції predict_email(text) є можливість класифікувати окремі листи як листи з соціальною інженерією чи без неї.

Далі було протестовано 10 фішингових та 10 звичайних листів. На рис. 4.5 можна побачити результати кожної з 3 моделей на однакових вхідних листах.

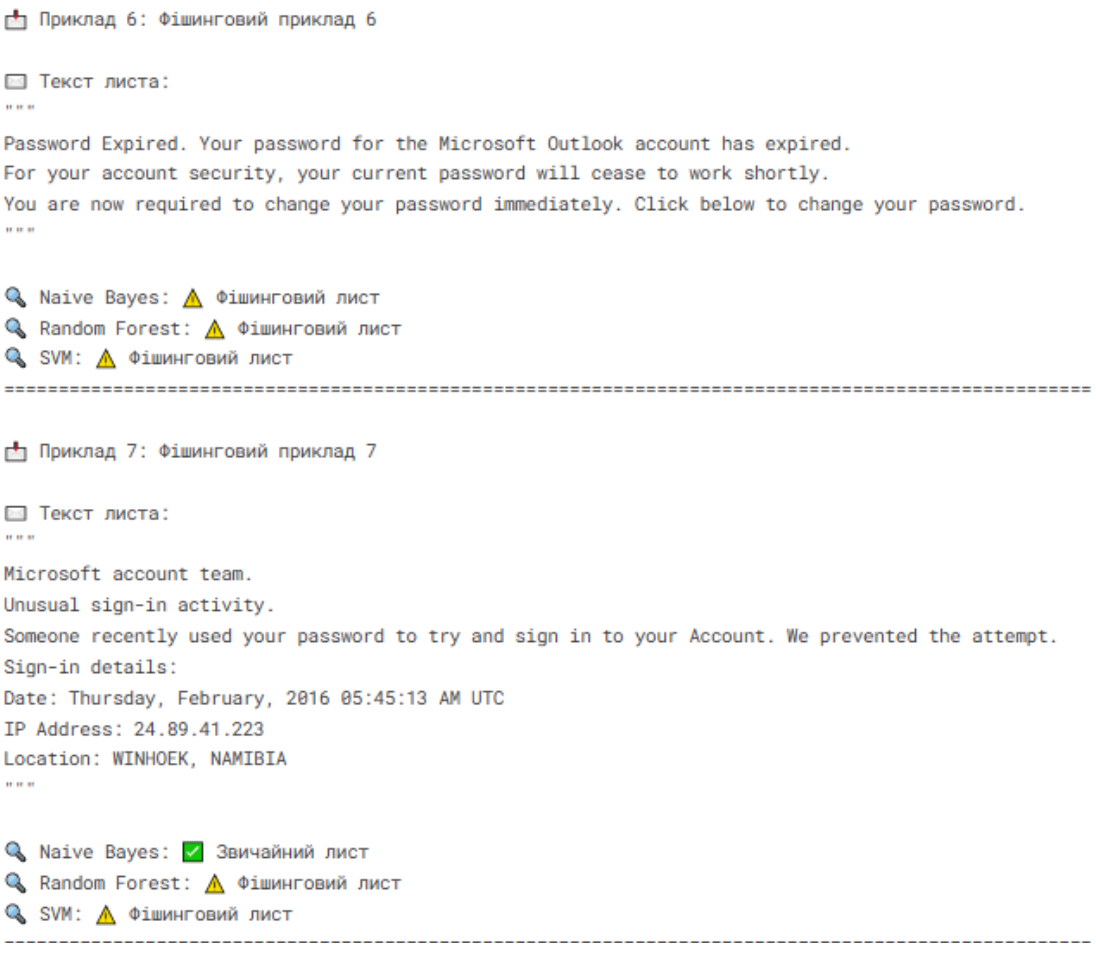


Рисунок 4.5 – Тестування роботи кожної з моделей на окремих фішингових листах

Для прикладу ми можемо побачити різницю в роботі моделей з фішинговими листами, де моделі попереджають про небезпеку та на рис. 4.6 зі звичайними листами, де показано, що в них моделі підозри не бачать.

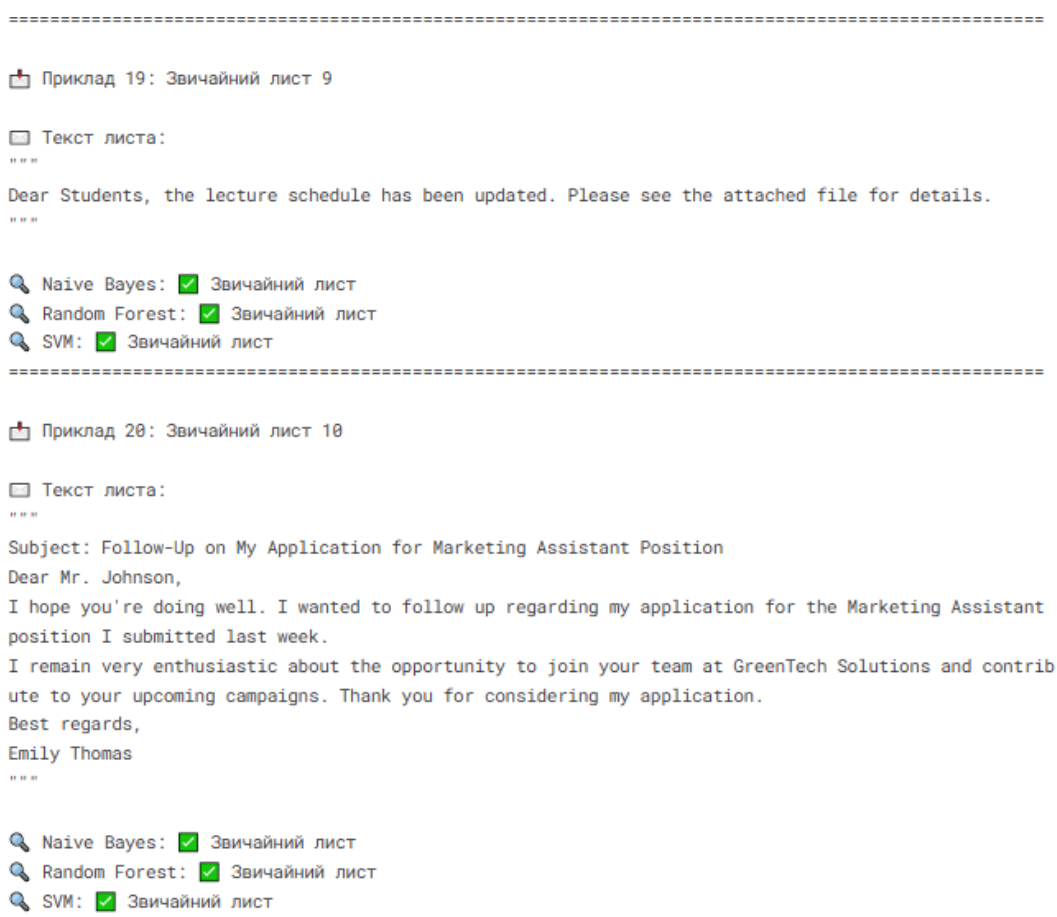


Рисунок 4.6 – Приклад роботи моделей при вхідних даних звичайних листів

Моделі все ж допускали деякі помилки і у фішингових прикладах, і у звичайних листах, щоб перевірити якість роботи моделей на окремих листах було створено діаграму результатів (рис. 4.7), на якому ми можемо побачити, що точність роботи моделей з окремими листами не збігається з початковим аналізом ефективності моделей. Найкращий результат показали моделі Naive Bayes та SVM: 85.00%, Random Forest у свою чергу виявився найгіршим на реальних прикладах.

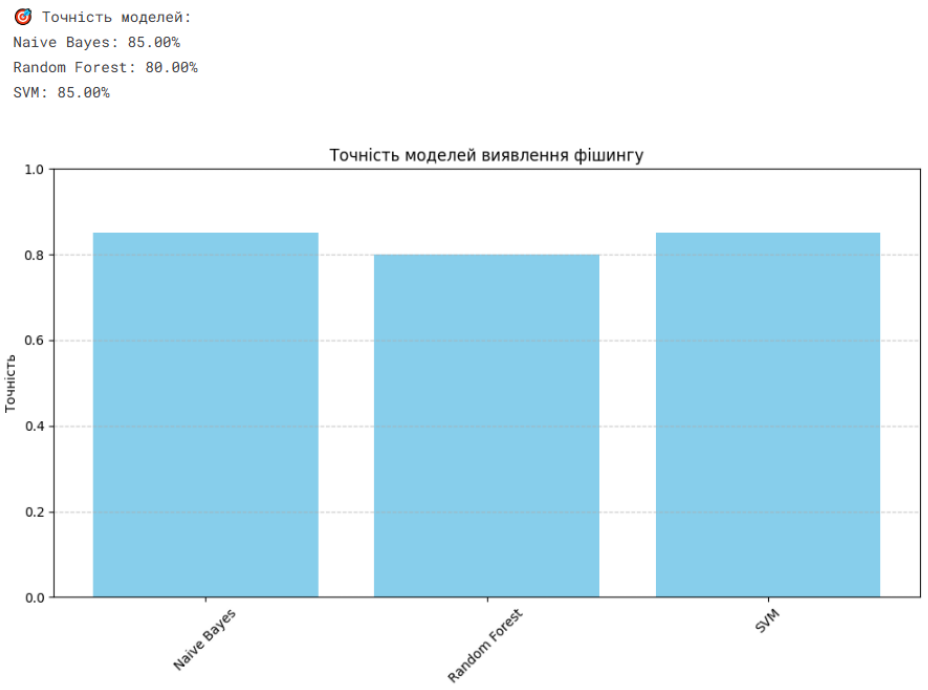


Рисунок 4.7 – Точність моделей в роботі з реальними прикладами окремих листів

Також було створено вибірку слів, які найчастіше використовувались в небезпечних окремих листах (рис. 4.8). Відповідно до аналізу, найбільш вживаними словами були: account, password, click, link, update, tax, refund, immediately, receive, confirm, hello, salary, raise, documents, expired, Microsoft, change, signin, hey, suspended.

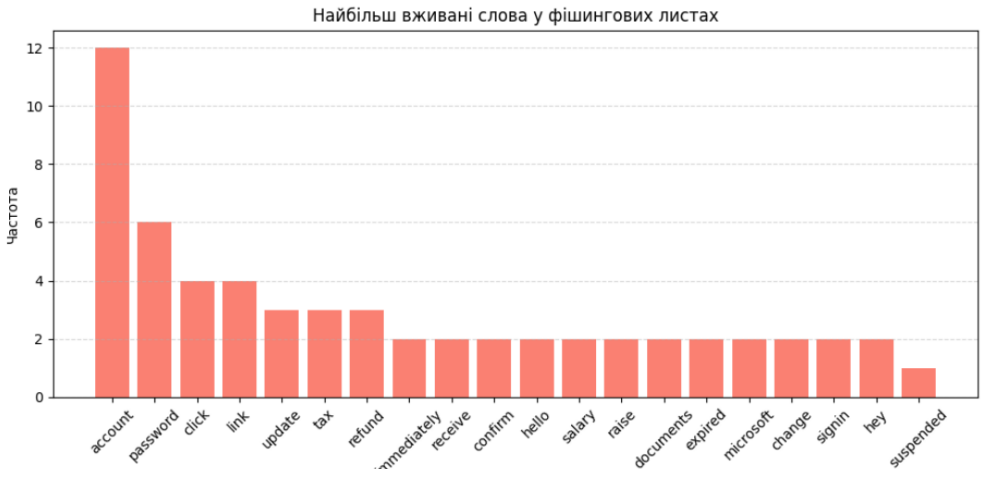


Рисунок 4.8 – Найбільш вживані слова у фішингових листах

Для гармонізації дослідження перевіримо також роботу моделей машинного навчання на прикладах з іншим типом соціальної інженерії, а саме цифровим байтингом (рис. 4.9). Ми можемо побачити, що моделі також успішно справились з визначенням цифрового байтингу.

Рисунок 4.9 – Приклад роботи моделей з байтинговими листами

Таким чином, було протестовано роботу моделей Naive Bayes, Random Forest, SVM на конкретних прикладах, та отримали дещо інші результати від початкового аналізу їх ефективності. Причина різниці точності в результатах може полягати в декількох особливостях: наявність класового дисбалансу, випадковість формування навчальної та тестової вибірок, а також різна чутливість моделей до особливостей ознак листів з соціальною інженерією.

ВИСНОВКИ

У процесі виконання дипломної роботи було досліджено можливості застосування методів штучного інтелекту для виявлення атак соціальної інженерії в кіберпросторі. Зважаючи на зростаючу складність і масштабність кіберзагроз, особливо тих, що базуються на психологічному впливі на людину, питання виявлення соціальної інженерії постає як одне з ключових у сфері сучасної кібербезпеки.

Метою роботи було вивчити методи та інструменти виявлення соціально-інженерних атак, зосередившись на прикладному аспекті – побудові системи, що здатна автоматично розпізнавати фішингові повідомлення. Ця мета була досягнута повністю, оскільки в межах дослідження вдалося не лише теоретично опрацювати актуальні підходи, а й реалізувати працюючу модель, засновану на алгоритмах машинного навчання.

Під час виконання роботи було послідовно реалізовано всі поставлені завдання. На першому етапі детально проаналізовано сутність соціальної інженерії та її відмінність від традиційних технічних атак. Було з’ясовано, що основна загроза тут полягає в маніпуляції поведінкою користувача, що часто не враховується у стандартних стратегіях кіберзахисту.

Також було досліджено методи штучного інтелекту, які можуть бути корисними у виявленні таких загроз. Серед них особливу увагу приділено машинному навчанню, методам обробки природної мови та алгоритмам класифікації даних.

Теоретичне обґрунтування підкріплено практичною частиною – побудовою системи, яка аналізує текст електронних повідомлень і класифікує їх як безпечні або фішингові.

Після обробки даних, їх векторизації та поділу на тренувальні й тестові вибірки було побудовано та навчено три моделі: Naive Bayes, Support Vector

Machine (SVM) та Random Forest. Усі вони пройшли тестування на реальному датасеті. За результатами експерименту:

- Naive Bayes – 95.3%;
- SVM – 98.5%;
- Random Forest – 98.7%.

Ці показники свідчать про досить високу ефективність усіх трьох алгоритмів. Найкращі результати продемонструвала модель SVM, що пояснюється її здатністю працювати з великою кількістю ознак та обробляти складні, нефункціональні залежності між ними.

Окремо було проаналізовано відмінності між теоретично очікуваними та фактичними результатами. Точність роботи моделей на протестованих даних:

- Naive Bayes – 85%;
- SVM – 85%;
- Random Forest – 80%.

На точність моделей, впливають як характеристики самого датасету, так і способи попередньої обробки тексту, вибір гіперпараметрів, а також потенційна нерівномірність розподілу класів. Саме тому важливо не лише правильно вибрати алгоритм, а й адаптувати його до конкретних даних та сценарію використання.

У підсумку, запропонована в роботі система може бути застосована як основа для реального інструменту кіберзахисту, зокрема для автоматичного аналізу вхідної пошти. Вона може ефективно знижувати ризики, пов’язані з фішинговими атаками, і потенційно інтегруватися у більші корпоративні системи безпеки.

Отже, можна стверджувати, що усі поставлені завдання були успішно виконані. Запропоновані рішення мають практичну цінність і демонструють,

що методи штучного інтелекту можуть бути ефективним інструментом у боротьбі з загрозами соціальної інженерії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mitnick K., Simon W. The Art of Deception: Controlling the Human Element of Security. — Wiley, 2002.
2. Кримінальний кодекс України // Офіційний вебпортал Верховної Ради України, 2017. — [Електронний ресурс]. — Режим доступу: https://zakon.rada.gov.ua/laws/show/2341-14#Text
3. Artificial Intelligence in Cybersecurity: A Review and a Case Study // mdpi.com. — 2024. — [Електронний ресурс]. — Режим доступу: https://www.mdpi.com/2076-3417/14/22/10487
4. Великі мовні моделі | Machine Learning // developers.google.com. — 2023. — [Електронний ресурс]. — Режим доступу: https://developers.google.com/machine-learning/crash-course/llm?hl=uk
5. Використання машинного навчання в кібербезпеці // csecurity.kubg.edu.ua. — 2023. — [Електронний ресурс]. — Режим доступу: https://csecurity.kubg.edu.ua/index.php/journal/article/view/260
6. ШІ у кібербезпеці: роль, застосування та переваги // wezom.com.ua. — 2024. — [Електронний ресурс]. — Режим доступу: https://wezom.com.ua/ua/blog/zastosuvannya-shi-u-kiberbezpetsi-rol-ta-perevagi
7. Корпоративна кібербезпека: Роль ШІ у захисті даних // bdo.ua. — 2024. — [Електронний ресурс]. — Режим доступу: https://www.bdo.ua/uk-ua/insights-2/information-materials/2024/corporate-cybersecurity-ai-role-in-data-protection
8. NIST Special Publication 800-63-3: Digital Identity Guidelines. — 2017. — [Електронний ресурс]. — Режим доступу: https://pages.nist.gov/800-63-3/
9. Singer P. W., Friedman A. Cybersecurity and Cyberwar: What Everyone Needs to Know / P. W. Singer, A. Friedman. — Oxford University Press, 2014.
10. Загальний регламент захисту даних (GDPR) // ec.europa.eu. — 2018. — [Електронний ресурс]. — Режим доступу: https://gdpr-info.eu/

11. Hadnagy C. Social Engineering: The Science of Human Hacking. — Wiley, 2018.

12. What is Social Engineering? Definitions and Attack Types // proofpoint.com. — [Електронний ресурс]. — Режим доступу: https://www.proofpoint.com/us/threat-reference/social-engineering

13. Жмурко О. Соціальна інженерія як загроза кібербезпеці: методи запобігання та захисту // Педагогіка безпеки, 2024, Том 9, №1.

14. Schuh M. The AI Regulation: Implications and requirements. — 2024.

15. Rodrigues R. Legal and human rights issues of AI: Gaps, challenges and vulnerabilities // Journal of Responsible Technology. — 2020.

16. European Ethical Charter on the Use of Artificial Intelligence in Judicial Systems // coe.int. — 2018. — [Електронний ресурс]. — Режим доступу: https://www.coe.int/en/web/cepej/cepej-european-ethical-charter-on-the-use-of-artificial-intelligence-ai-in-judicial-systems-and-their-environment

17. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. — MIT Press, 2016.

18. Хіросімський процес штучного інтелекту (Hiroshima AI Process) // g7hiroshima.go.jp. — 2023–2024. — [Електронний ресурс]. — Режим доступу: https://www.soumu.go.jp/hiroshimaaiprocess/en/index.html

19. ISO/IEC 27001:2022 — Information Security Management // iso.org. — 2022. — [Електронний ресурс]. — Режим доступу: https://www.isms.online/iso-27001/

20. Tal, Z. Detecting and Preventing AI-Based Phishing Attacks: 2024 Guide // Perception Point, 2024. — [Електронний ресурс]. — Режим доступу: https://perception-point.io/guides/ai-security/detecting-and-preventing-ai-based-phishing-attacks-2024-guide/

21. IEEE Ethically Aligned Design Standards (IEEE P7000 Series). — [Електронний ресурс]. — Режим доступу: https://standards.ieee.org/wp-content/uploads/import/documents/other/ead_v2.pdf

22. OECD Recommendation on Artificial Intelligence. — 2019 (оновлено 2024). — [Електронний ресурс]. — Режим доступу: https://oecd.ai/en/ai-principles

23. Конвенція про кіберзлочинність (Будапештська конвенція) // Council of Europe, 2001. — [Електронний ресурс]. — Режим доступу: https://www.coe.int/en/web/cybercrime/the-budapest-convention

24. Директива ЄС про кібербезпеку (NIS2 Directive) // EUR-Lex, 2022. — [Електронний ресурс]. — Режим доступу: https://digital-strategy.ec.europa.eu/en/policies/nis2-directive

25. The 12 Latest Types of Social Engineering Attacks (2024) // Aura, 2023-12-08. — [Електронний ресурс]. — Режим доступу: https://www.aura.com/learn/types-of-social-engineering-attacks

26. Md Aby Sayed. A Comparative Analysis of Machine Learning Algorithms for Cybersecurity // The American Journal of Engineering and Technology, 2024.

27. Регламент про штучний інтелект (EU AI Act) // European Commission, 2024. — [Електронний ресурс]. — Режим доступу: https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

28. 10 Types of Social Engineering Attacks // CrowdStrike, 2023-11-08. — [Електронний ресурс]. — Режим доступу: https://www.crowdstrike.com/en-us/cybersecurity-101/social-engineering/types-of-social-engineering-attacks/

29. Kosseff, J. Cybersecurity Law. — 2nd ed. — Wiley, 2020

30. Comparative Analysis of National and International Legal Frameworks for Cyber Security: Implications for Policy and Practice // Nanotechnology Perceptions, Vol. 20, No. 7, 2024. — [Електронний ресурс]. — Режим доступу: https://nano-ntp.com/index.php/nano/article/view/4684

31. Тарасюк А. В. Стан правового забезпечення кібербезпеки в Україні // Науковий вісник НАПрН України, 2020.

32. Стратегія кібербезпеки України на 2023–2024 роки // cip.gov.ua, 2024. — [Електронний ресурс]. — Режим доступу: https://www.kmu.gov.ua/npas/pro-

zatverdzhennia-planu-zakhodiv-na-20232024-roky-z-realizatsii-stratehii-kiberbezpeky-ukrainy-i191223-1163

33. Rehman, S. Practical Implementation of Artificial Intelligence in Cybersecurity – A Study // Wilmington University, 2022.

34. Забезпечення інформаційної безпеки в умовах євроінтеграції України: правовий вимір. — ТОВ «Видавничий дім «АртЕк», 2018.

35. Закон України "Про основні засади забезпечення кібербезпеки України" // Офіційний вебпортал Верховної Ради України, 2017. — [Електронний ресурс]. — Режим доступу: https://zakon.rada.gov.ua/laws/show/2163-19#Text

36. 6 Types of Social Engineering Attacks and How to Prevent Them // Mitnick Security, 2024-11-07. — [Електронний ресурс]. — Режим доступу: https://www.mitnicksecurity.com/blog/types-of-social-engineering-attacks

37. Delev, Z. Ensuring Compliance: A Practical Guide to GDPR Cyber Security // GDPR Local, 2025. — [Електронний ресурс]. — Режим доступу: https://gdprlocal.com/gdpr-cyber-security/

38. NIST Cybersecurity Framework // NIST, 2018. — [Електронний ресурс]. — Режим доступу: https://www.nist.gov/news-events/news/2018/04/nist-releases-version-11-its-popular-cybersecurity-framework

ДОДАТОК А

В додатку А предсталено повний код практичної частини виконання роботи.

```
# STEP 1: Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
import string
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
```

In [2]:

```
# STEP 2: Load Dataset
df = pd.read_csv('/kaggle/input/phishing-email-dataset/phishing_email.csv')
df.head()
```

Out[2]:

	text_combined	label
0	hpl nom may 25 2001 see attached file hplno 52...	0
1	nom actual vols 24 th forwarded sabrae zajac h...	0
2	enron actuals march 30 april 1 201 estimated a...	0
3	hpl nom may 30 2001 see attached file hplno 53...	0
4	hpl nom june 1 2001 see attached file hplno 60...	0

In [3]:

```
# STEP 3: Перевірка назв колонок
print("Назви колонок:", df.columns)
```

```
Назви колонок: Index(['text_combined', 'label'], dtype='object')
```

In [4]:

```
# STEP 4: Припустимо, колонка з текстом — 'text', а з мітками — 'label' (заміни, якщо інші)
df = df[['text_combined', 'label']]
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
```

In [5]:

```
# STEP 5: Перевіримо унікальні мітки
print(df['label'].value_counts())
```

```
label
1    42845
0    39233
Name: count, dtype: int64
```

In [6]:

```
# STEP 6: Очистка тексту
stop_words = set(stopwords.words('english'))
ps = PorterStemmer()

def clean_text(text):
    text = text.lower()
    text = re.sub(r'https?://\S+|www\.\S+', '', text)
    text = re.sub(r'\W+', ' ', text)
    words = word_tokenize(text)
    filtered = [ps.stem(w) for w in words if w not in stop_words and w not in string.punctuation]
    return ' '.join(filtered)

df['clean_text'] = df['text_combined'].apply(clean_text)
```

In [7]:

```
# STEP 7: Векторизація
vectorizer = TfidfVectorizer(max_features=3000)
X = vectorizer.fit_transform(df['clean_text']).toarray()
y = df['label']
```

In [8]:

```
# STEP 8: Розділення на train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [9]:

```
import joblib
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# STEP 9: Завантаження збережених моделей
models = {
```

```
    'Naive Bayes': joblib.load('/kaggle/input/phishing-
models/naive_bayes_model.pkl'),
    'Random Forest': joblib.load('/kaggle/input/phishing-
models/random_forest_model.pkl'),
    'SVM': joblib.load('/kaggle/input/phishing-models/svm_model.pkl')
}

# STEP 10:Використання моделей для передбачень і оцінка
for name, model in models.items():
    preds = model.predict(X_test)
    print(f"\n===== {name} =====")
    print("Accuracy:", accuracy_score(y_test, preds))
    print("Confusion Matrix:\n", confusion_matrix(y_test, preds))
    print("Classification Report:\n", classification_report(y_test, preds))
```

```
===== Naive Bayes =====
Accuracy: 0.9532772904483431
Confusion Matrix:
 [[7704  206]
 [ 561 7945]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      7910
           1       0.97      0.93      0.95      8506

    accuracy                           0.95     16416
   macro avg       0.95      0.95      0.95     16416
weighted avg       0.95      0.95      0.95     16416


===== Random Forest =====
Accuracy: 0.985014619883041
Confusion Matrix:
 [[7811   99]
 [ 147 8359]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      7910
           1       0.99      0.98      0.99      8506

    accuracy                           0.99     16416
   macro avg       0.98      0.99      0.98     16416
weighted avg       0.99      0.99      0.99     16416
```

```
===== SVM =====
Accuracy: 0.9877558479532164
Confusion Matrix:
 [[7808  102]
 [  99 8407]]
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      7910
           1       0.99      0.99      0.99      8506

    accuracy                           0.99     16416
   macro avg       0.99      0.99      0.99     16416
weighted avg       0.99      0.99      0.99     16416
```

In [10]:

```
# STEP 11: Функція передбачення листа для кожної моделі
def predict_email_with_model(text, model_name):
    cleaned = clean_text(text)
    vector = vectorizer.transform([cleaned]).toarray()
    model = models[model_name]
    prediction = model.predict(vector)[0]

    if prediction == 1:
        return f"⚠️ ({model_name}) Це фішинговий лист!"
    else:
        return f"✅ ({model_name}) Це звичайний лист."
```

In [11]:

```
# Приклади листів
examples = {
    "Фішинговий приклад 1": "Your account is suspended. Log in immediately to verify your identity.",
    "Фішинговий приклад 2": "You've been selected to receive a free iPhone! Click here to claim.",
    "Звичайний лист 1": "Dear colleague, please find attached the minutes of our last meeting.",
    "Звичайний лист 2": "Hi Mom, just wanted to let you know I arrived safely. Love you!"
}

# Тестування з усіма моделями
```

```
for model_name in models:
    print(f"\n=== Результати для моделі: {model_name} ===")
    for label, email in examples.items():
        print(f"{label}: {predict_email_with_model(email, model_name)}")
```

```
=== Результати для моделі: Naive Bayes ===
Фішинговий приклад 1: ⚠️ (Naive Bayes) Це фішинговий лист!
Фішинговий приклад 2: ⚠️ (Naive Bayes) Це фішинговий лист!
Звичайний лист 1: ✅ (Naive Bayes) Це звичайний лист.
Звичайний лист 2: ⚠️ (Naive Bayes) Це фішинговий лист!

=== Результати для моделі: Random Forest ===
Фішинговий приклад 1: ⚠️ (Random Forest) Це фішинговий лист!
Фішинговий приклад 2: ⚠️ (Random Forest) Це фішинговий лист!
Звичайний лист 1: ✅ (Random Forest) Це звичайний лист.
Звичайний лист 2: ⚠️ (Random Forest) Це фішинговий лист!

=== Результати для моделі: SVM ===
Фішинговий приклад 1: ⚠️ (SVM) Це фішинговий лист!
Фішинговий приклад 2: ⚠️ (SVM) Це фішинговий лист!
Звичайний лист 1: ⚠️ (SVM) Це фішинговий лист!
Звичайний лист 2: ⚠️ (SVM) Це фішинговий лист!
```

In [44]:

```
# ✅ Приклади листів
examples = {
    "Фішинговий приклад 1": "Your account is suspended. Log in immediately to verify your identity.",
    "Фішинговий приклад 2": "You've been selected to receive a free iPhone! Click here to claim.",
    "Фішинговий приклад 3": """Attention! Your PayPal account will close soon!
Dear Member,
We have faced some problems with your account. Please update the account. If you do not update will be Closed.
To Update your account, just confirm your informations. (It only takes a minute.)
It’s easy:
1. Click the link below to open a secure browser window.
2. Confirm that you’re the owner of the account, and then follow the instructions.
Relog in your account NOW!""",
    "Фішинговий приклад 4": """Hello,
We assessed the 2015 payment structure as provided for under the terms of employment and discovered that you are due for a salary raise starting August 2015.
Your salary raise documents are enclosed below:
```

Access the documents here
Faithfully,
Human Resources""",
"Фішинговий приклад 5": """Hello,
New Tax Calculation
We have determined that you are eligible to receive a tax refund of 209.27 €.
Please submit the tax refund request by clicking the link below.
Note: A refund can be delayed for reasons such as invalid records or late application.
HM Revenue & Customs""",
"Фішинговий приклад 6": """Password Expired. Your password for the Microsoft Outlook account has expired.
For your account security, your current password will cease to work shortly.
You are now required to change your password immediately. Click below to change your password.""",
"Фішинговий приклад 7": """Microsoft account team.
Unusual sign-in activity.
Someone recently used your password to try and sign in to your Account. We prevented the attempt.
Sign-in details:
Date: Thursday, February, 2016 05:45:13 AM UTC
IP Address: 24.89.41.223
Location: WINHOEK, NAMIBIA""",
"Фішинговий приклад 8": "Your Instagram account has violated our policies and will be removed. To appeal this action, click here: [scam link]",
"Фішинговий приклад 9": "Hey mom! I got in trouble, please send me 1000$ on this card now!",
"Фішинговий приклад 10": "Hey Susy! Can I ask you to vote for my daughter's art with this link!",

"Звичайний лист 1": "Dear colleague, please find attached the minutes of our last meeting.",
"Звичайний лист 2": "Hello Anna, I’ve scheduled your dental appointment for next Tuesday at 3 PM. Let me know if that works!",
"Звичайний лист 3": """Your open balance in Zalando has been updated!
Hi Susan!
We've received your payment of 29,75 €. Your new open balance is now 35,99 €.
We’ll send you a confirmation when the payment reaches us. For bank transfers, this can take up to 5 business days.""",
"Звичайний лист 4": """Dear John,
I hope you're doing well. I wanted to share the details of my upcoming flight with you in case you need them for reference or planning purposes.
Airline: Lufthansa
Flight Number: LH493

Departure City: Vancouver (YVR)
Departure Time: Monday, May 20th at 4:35 PM
Arrival City: Frankfurt (FRA)
Arrival Time: Tuesday, May 21st at 10:25 AM
Seat Number: 32A (window)
Booking Reference: ABC123
Please let me know if you need anything else. I'll keep my phone on before boarding and will update you once I land. Looking forward to seeing you soon!
Warm regards,
Ivan""",
"Звичайний лист 5": """You're Invited: Summer Garden Party
Hi Mark,
I'm hosting a small garden party next Saturday at my place to celebrate the start of summer, and I'd love for you to join us.
It'll be a relaxed evening with food, drinks, and good company. Let me know if you can make it. Hope to see you there!
Cheers,
Anna""",
"Звичайний лист 6": """New login to X from ChromeDesktop on Windows
We noticed a login to your account LadyD from a new device. Was this you?
If this was you, you can ignore this message. There's no need to take any action.
If this wasn't you, complete these steps now to protect your account:
• Change your password.
• Review the apps that have access to your account.""",
"Звичайний лист 7": """Hungry? We've got new places for you to try.
Sign up for an Uber One membership to start saving!
Enjoy 0 € Delivery Fee on eligible restaurant and grocery orders, member benefits on trips, and more. Members save €11 per month on average.""",
"Звичайний лист 8": "Hi John, just confirming our meeting scheduled for tomorrow at 10:00 AM via Zoom. I'll send the link 15 minutes before we start.",
"Звичайний лист 9": "Dear Students, the lecture schedule has been updated. Please see the attached file for details.",
"Звичайний лист 10": """Subject: Follow-Up on My Application for Marketing Assistant Position
Dear Mr. Johnson,
I hope you're doing well. I wanted to follow up regarding my application for the Marketing Assistant position I submitted last week.
I remain very enthusiastic about the opportunity to join your team at GreenTech Solutions and contribute to your upcoming campaigns. Thank you for considering my application.
Best regards,
Emily Thomas"""
}

```
# ✅ Функція передбачення з вибраною моделлю
def predict_email_with_model(text, model_name):
    cleaned = clean_text(text)
    vector = vectorizer.transform([cleaned]).toarray()
    prediction = models[model_name].predict(vector)[0]
    label = "✅ Звичайний лист" if prediction == 0 else "⚠️ Фішинговий лист"
    return f"{model_name}: {label}"

# ✅ Вивід результатів
for idx, (label, email) in enumerate(examples.items(), start=1):
    print("=" * 100)
    print(f"\n📩 Приклад {idx}: {label}")
    print(f"\n✉️ Текст листа:\n\"\"\"\n{email}\n\"\"\"\n")
    for model_name in models:
        result = predict_email_with_model(email, model_name)
        print(f"🔍 {result}")
```

==========

📩 Приклад 1: Фішинговий приклад 1

✉️ Текст листа:
"""
Your account is suspended. Log in immediately to verify your identity.
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист

==========

📩 Приклад 2: Фішинговий приклад 2

✉️ Текст листа:
"""
You've been selected to receive a free iPhone! Click here to claim.
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист

🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 3: Фішинговий приклад 3

✉️ Текст листа:
"""
Attention! Your PayPal account will close soon! Dear Member,
We have faced some problems with your account. Please update the account. If you do not update will be Closed.
To Update your account, just confirm your informations. (It only takes a minute.)
It’s easy:
1. Click the link below to open a secure browser window.
2. Confirm that you’re the owner of the account, and then follow the instructions.
Relog in your account NOW!
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 4: Фішинговий приклад 4

✉️ Текст листа:
"""
Hello,
We assessed the 2015 payment structure as provided for under the terms of employment and discovered that you are due for a salary raise starting August 2015.
Your salary raise documents are enclosed below:
Access the documents here
Faithfully,
Human Resources
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист

==========
==========

📩 Приклад 5: Фішинговий приклад 5

✉️ Текст листа:
"""
Hello,
New Tax Calculation
We have determined that you are eligible to receive a tax refund of 209.27 €.
Please submit the tax refund request by clicking the link below.
Note: A refund can be delayed for reasons such as invalid records or late application.
HM Revenue & Customs
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========
==========

📩 Приклад 6: Фішинговий приклад 6

✉️ Текст листа:
"""
Password Expired. Your password for the Microsoft Outlook account has expired.
For your account security, your current password will cease to work shortly.
You are now required to change your password immediately. Click below to change your password.
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========
==========

📩 Приклад 7: Фішинговий приклад 7

✉️ Текст листа:
"""

Microsoft account team.
Unusual sign-in activity.
Someone recently used your password to try and sign in to your Account. We prevented the attempt.
Sign-in details:
Date: Thursday, February, 2016 05:45:13 AM UTC
IP Address: 24.89.41.223
Location: WINHOEK, NAMIBIA
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 8: Фішинговий приклад 8

✉️ Текст листа:
"""
Your Instagram account has violated our policies and will be removed. To appeal this action, click here: [scam link]
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 9: Фішинговий приклад 9

✉️ Текст листа:
"""
Hey mom! I got in trouble, please send me 1000$ on this card now!
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📥 Приклад 10: Фішинговий приклад 10

✉️ Текст листа:
"""
Hey Susy! Can I ask you to vote for my daughter's art with this link!
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📥 Приклад 11: Звичайний лист 1

✉️ Текст листа:
"""
Dear colleague, please find attached the minutes of our last meeting.
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📥 Приклад 12: Звичайний лист 2

✉️ Текст листа:
"""
Hello Anna, I’ve scheduled your dental appointment for next Tuesday at 3 PM. Let me know if that works!
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ✅ Звичайний лист
==========

📥 Приклад 13: Звичайний лист 3

✉️ Текст листа:
"""
Your open balance in Zalando has been updated!
Hi Susan!
We've received your payment of 29,75 €. Your new open balance is now 35,99 €. We'll send you a confirmation when the payment reaches us. For bank transfers, this can take up to 5 business days.
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 14: Звичайний лист 4

✉️ Текст листа:
"""
Dear John,
I hope you're doing well. I wanted to share the details of my upcoming flight with you in case you need them for reference or planning purposes.
Airline: Lufthansa
Flight Number: LH493
Departure City: Vancouver (YVR)
Departure Time: Monday, May 20th at 4:35 PM
Arrival City: Frankfurt (FRA)
Arrival Time: Tuesday, May 21st at 10:25 AM
Seat Number: 32A (window)
Booking Reference: ABC123
Please let me know if you need anything else. I'll keep my phone on before boarding and will update you once I land. Looking forward to seeing you soon!
Warm regards,
Ivan
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ✅ Звичайний лист
==========

📩 Приклад 15: Звичайний лист 5

✉️ Текст листа:
"""
You’re Invited: Summer Garden Party
Hi Mark,
I’m hosting a small garden party next Saturday at my place to celebrate the start of summer, and I’d love for you to join us.
It’ll be a relaxed evening with food, drinks, and good company. Let me know if you can make it. Hope to see you there!
Cheers,
Anna
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ✅ Звичайний лист
==========

📩 Приклад 16: Звичайний лист 6

✉️ Текст листа:
"""
New login to X from ChromeDesktop on Windows
We noticed a login to your account LadyD from a new device. Was this you?
If this was you, you can ignore this message. There's no need to take any action.
If this wasn’t you, complete these steps now to protect your account:
• Change your password.
• Review the apps that have access to your account.
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ⚠️ Фішинговий лист
==========

📩 Приклад 17: Звичайний лист 7

✉️ Текст листа:
"""

Hungry? We've got new places for you to try.
Sign up for an Uber One membership to start saving!
Enjoy 0 € Delivery Fee on eligible restaurant and grocery orders, member benefits on trips, and more. Members save €11 per month on average.
"""

🔍 Naive Bayes: ⚠️ Фішинговий лист
🔍 Random Forest: ⚠️ Фішинговий лист
🔍 SVM: ✅ Звичайний лист
==========

📩 Приклад 18: Звичайний лист 8

✉️ Текст листа:
"""
Hi John, just confirming our meeting scheduled for tomorrow at 10:00 AM via Zoom. I’ll send the link 15 minutes before we start.
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ✅ Звичайний лист
==========

📩 Приклад 19: Звичайний лист 9

✉️ Текст листа:
"""
Dear Students, the lecture schedule has been updated. Please see the attached file for details.
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ✅ Звичайний лист
==========

📩 Приклад 20: Звичайний лист 10

✉️ Текст листа:
"""
Subject: Follow-Up on My Application for Marketing Assistant Position
Dear Mr. Johnson,
I hope you're doing well. I wanted to follow up regarding my application for the Marketing Assistant position I submitted last week.
I remain very enthusiastic about the opportunity to join your team at GreenTech Solutions and contribute to your upcoming campaigns. Thank you for considering my application.
Best regards,
Emily Thomas
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ✅ Звичайний лист

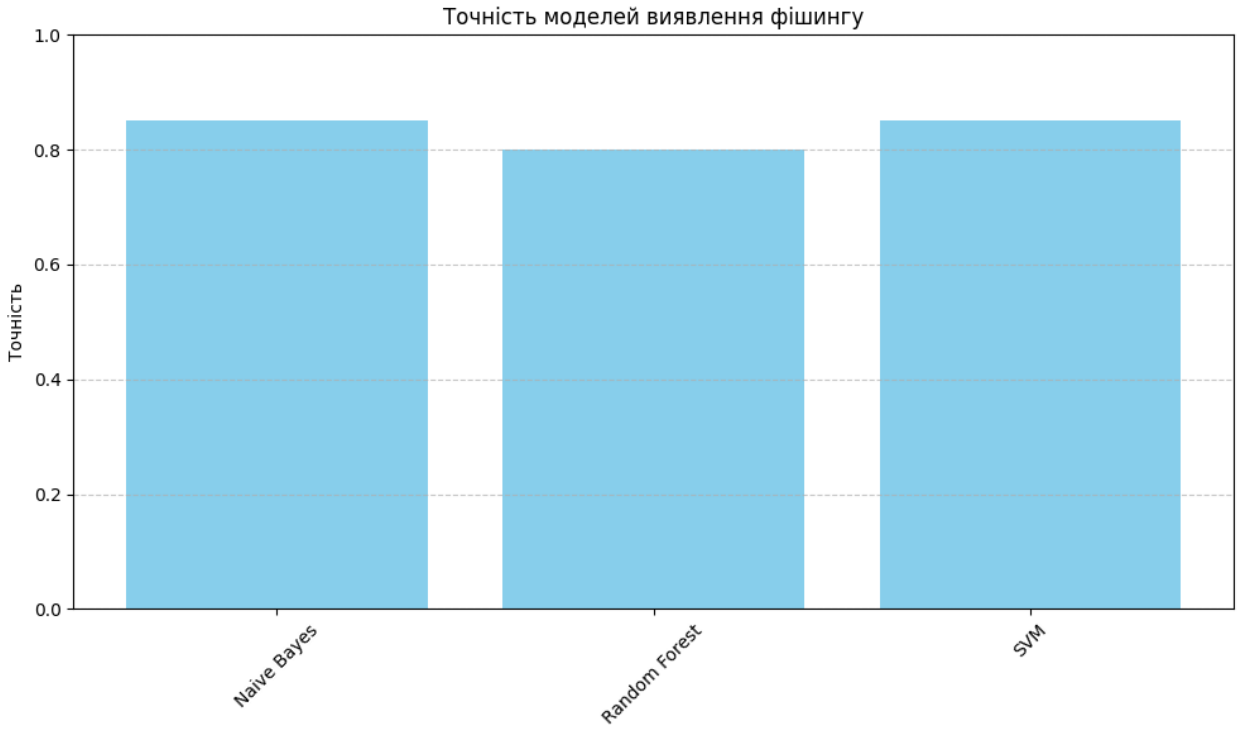
In [35]:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score

# Створимо список для результатів
results = []

# Створимо справжні мітки (1 = фішинговий, 0 = звичайний)
true_labels = []
email_texts = []
example_names = []

# Збір результатів передбачення
for label, email in examples.items():
    y_true = 1 if "Фішинговий" in label else 0
    row = {
        "Назва прикладу": label,
        "Текст листа": email,
        "Реальна мітка": y_true
    }
    for model_name in models:
        pred = predict_email_with_model(email, model_name)
        row[model_name] = 1 if "Фішинговий лист" in pred else 0
    results.append(row)

# Перетворимо в DataFrame
```

```
df = pd.DataFrame(results)

# Збережемо в CSV
df.to_csv("phishing_detection_results.csv", index=False)
print("✅ Результати збережено у phishing_detection_results.csv")

# Порахуймо точність для кожної моделі
accuracies = {}
for model_name in models:
    accuracies[model_name] = accuracy_score(df["Реальна мітка"],
df[model_name])

# Вивід точності
print("\n🎯 Точність моделей:")
for model, acc in accuracies.items():
    print(f"{model}: {acc:.2%}")

# Побудова графіка точності
plt.figure(figsize=(10, 6))
plt.bar(accuracies.keys(), accuracies.values(), color='skyblue')
plt.ylabel("Точність")
plt.ylim(0, 1)
plt.title("Точність моделей виявлення фішингу")
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

✅ Результати збережено у phishing_detection_results.csv

🎯 Точність моделей:
Naive Bayes: 85.00%
Random Forest: 80.00%
SVM: 85.00%



In [36]:
```
df.to_csv("phishing_detection_results_graph.csv", index=False)
```

In [30]:
```
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS
from collections import Counter
import matplotlib.pyplot as plt
import re

# ✅ Отримуємо лише тексти фішингових листів
phishing_texts = [text for key, text in examples.items() if "Фішинговий" in key]

# ✅ Об'єднуємо все в один текст
all_text = " ".join(phishing_texts).lower()

# ✅ Очищення тексту: видалення спецсимволів і чисел
all_text = re.sub(r"[^a-z\s]", "", all_text)

# ✅ Токенізація та видалення стоп-слів
words = all_text.split()
filtered_words = [word for word in words if word not in
ENGLISH_STOP_WORDS and len(word) > 2]

# ✅ Підрахунок частоти
word_counts = Counter(filtered_words)
```

```
# ✅ Топ-20 найчастіших слів
top_words = word_counts.most_common(20)

# ✅ Виведення результатів
print("🔠 Топ-20 слів у фішингових листах:")
for word, count in top_words:
    print(f"{word}: {count}")

# ✅ Візуалізація
words, counts = zip(*top_words)
plt.figure(figsize=(10, 5))
plt.bar(words, counts, color="salmon")
plt.xticks(rotation=45)
plt.title("Найбільш вживані слова у фішингових листах")
plt.ylabel("Частота")
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

```
🔠 Топ-20 слів у фішингових листах:
account: 12
password: 6
click: 4
link: 4
update: 3
tax: 3
refund: 3
immediately: 2
receive: 2
confirm: 2
hello: 2
salary: 2
raise: 2
documents: 2
expired: 2
microsoft: 2
change: 2
signin: 2
hey: 2
suspended: 1
```

In [31]:

```
# Створення DataFrame з топ-словами
df_top_words = pd.DataFrame(top_words, columns=["Слово", "Частота"])

# Збереження у CSV
df_top_words.to_csv("top_phishing_words.csv", index=False, encoding="utf-8-sig")

print("✅ Топ-слова у фішингових листах збережено у файл: top_phishing_words.csv")
```

✅ Топ-слова у фішингових листах збережено у файл: top_phishing_words.csv

In [52]:

```
# Приклади baiting-листів
baiting_examples = {
    "Baiting приклад 1": """Congratulations! You've been selected to receive a brand new iPhone 15 absolutely free of charge. This exclusive offer is available only to a limited number of users, and your spot has already been reserved. To claim your device, simply click the link below and complete the short verification form. Act now before this opportunity expires!""",

    "Baiting приклад 2": """Watch all your favorite Netflix shows and movies completely FREE. No registration, no subscription, just unlimited streaming! This offer is valid for the next 24 hours only. Simply click the link below and access thousands of titles instantly. Don’t miss this incredible chance to enjoy unlimited entertainment at zero cost. Start watching now!""",

    "Baiting приклад 3": """Congratulations! You've won a €100 Amazon gift card. All you need to do is complete a short survey to verify your details and receive your prize instantly. We’re running this offer for a limited time to celebrate our
```

loyal users. Click the link below to claim your gift and enjoy shopping with no cost!""",

"Baiting приклад 4": """Get Microsoft Office Pro 2024 absolutely FREE for a limited time! Enhance your productivity with Word, Excel, PowerPoint, and more. This is a special promotional campaign available to selected users only. Click below to download and activate your license immediately. Don't miss this chance to get premium software at no cost!""",

"Baiting приклад 5": """Tired of ads ruining your music experience? We're offering FREE lifetime access to Spotify Premium for early participants in our user survey. No credit card required, no strings attached. Just click the link below, answer a few quick questions, and activate your premium account instantly. Hurry – this offer won't last!""",

"Baiting приклад 6": """Get a brand new PlayStation 5 for just €1! Yes, you read that right. As part of our new promotion, we're offering next-gen consoles to select users for only one euro. Supplies are extremely limited. To participate, click the link below, confirm your address, and reserve your PS5 before it's gone!""",

"Baiting приклад 7": """Unlock the full version of Adobe Photoshop for FREE, exclusively for students and creatives. This limited-time campaign allows you to download and install professional-grade editing tools without paying a cent. Click the secure download link below and start editing like a pro. Hurry, this offer ends tonight!""",

"Baiting приклад 8": """Discover a revolutionary cryptocurrency app that pays new users $100 in free tokens just for signing up. No deposit required. Register now, complete your profile, and the bonus will be credited to your wallet instantly. Join the future of finance and be among the first to earn real crypto for free!""",

"Baiting приклад 9": """Top-secret government files have just been leaked to the public. We have obtained exclusive access to these shocking documents, which reveal hidden operations and confidential missions. Open the attached PDF to read the full report. This information may be taken down soon – view it before it disappears forever!""",

"Baiting приклад 10": """You've been selected to receive two FREE flight tickets to Dubai as part of our global travel campaign! All you need to do is enter your email and complete the short application form. This offer is limited to the first 500 respondents. Act fast and experience the luxury of Dubai – absolutely free!"""
}

 # ✅ Функція передбачення з вибраною моделлю

```
def predict_email_with_model(text, model_name):
    cleaned = clean_text(text)
    vector = vectorizer.transform([cleaned]).toarray()
    prediction = models[model_name].predict(vector)[0]
    label = "✅ Звичайний лист" if prediction == 0 else "⚠️ Байтинг лист"
    return f"{model_name}: {label}"

# ✅ Вивід результатів
for idx, (label, email) in enumerate(baiting_examples.items(), start=1):
    print("=" * 100)
    print(f"\n📩 Приклад {idx}: {label}")
    print(f"\n✉️ Текст листа:\n\"\"\"\n{email}\n\"\"\"\n")
    for model_name in models:
        result = predict_email_with_model(email, model_name)
        print(f"🔍 {result}")
```

==========

📩 Приклад 1: Baiting приклад 1

✉️ Текст листа:

"""

Congratulations! You've been selected to receive a brand new iPhone 15 absolutely free of charge. This exclusive offer is available only to a limited number of users, and your spot has already been reserved. To claim your device, simply click the link below and complete the short verification form. Act now before this opportunity expires!

"""

🔍 Naive Bayes: ⚠️ Байтинг лист

🔍 Random Forest: ⚠️ Байтинг лист

🔍 SVM: ⚠️ Байтинг лист

==========

📩 Приклад 2: Baiting приклад 2

✉️ Текст листа:

"""

Watch all your favorite Netflix shows and movies completely FREE. No registration, no subscription, just unlimited streaming! This offer is valid for the next 24 hours only. Simply click the link below and access thousands of titles

instantly. Don’t miss this incredible chance to enjoy unlimited entertainment at zero cost. Start watching now!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 3: Baiting приклад 3

✉️ Текст листа:
"""
Congratulations! You've won a €100 Amazon gift card. All you need to do is complete a short survey to verify your details and receive your prize instantly. We’re running this offer for a limited time to celebrate our loyal users. Click the link below to claim your gift and enjoy shopping with no cost!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 4: Baiting приклад 4

✉️ Текст листа:
"""
Get Microsoft Office Pro 2024 absolutely FREE for a limited time! Enhance your productivity with Word, Excel, PowerPoint, and more. This is a special promotional campaign available to selected users only. Click below to download and activate your license immediately. Don't miss this chance to get premium software at no cost!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 5: Baiting приклад 5

✉️ Текст листа:
"""
Tired of ads ruining your music experience? We’re offering FREE lifetime access to Spotify Premium for early participants in our user survey. No credit card required, no strings attached. Just click the link below, answer a few quick questions, and activate your premium account instantly. Hurry – this offer won’t last!
"""

🔍 Naive Bayes: ✅ Звичайний лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 6: Baiting приклад 6

✉️ Текст листа:
"""
Get a brand new PlayStation 5 for just €1! Yes, you read that right. As part of our new promotion, we're offering next-gen consoles to select users for only one euro. Supplies are extremely limited. To participate, click the link below, confirm your address, and reserve your PS5 before it’s gone!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 7: Baiting приклад 7

✉️ Текст листа:
"""
Unlock the full version of Adobe Photoshop for FREE, exclusively for students and creatives. This limited-time campaign allows you to download and install professional-grade editing tools without paying a cent. Click the secure download link below and start editing like a pro. Hurry, this offer ends tonight!

"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 8: Baiting приклад 8

✉️ Текст листа:
"""
Discover a revolutionary cryptocurrency app that pays new users $100 in free tokens just for signing up. No deposit required. Register now, complete your profile, and the bonus will be credited to your wallet instantly. Join the future of finance and be among the first to earn real crypto for free!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 9: Baiting приклад 9

✉️ Текст листа:
"""
Top-secret government files have just been leaked to the public. We have obtained exclusive access to these shocking documents, which reveal hidden operations and confidential missions. Open the attached PDF to read the full report. This information may be taken down soon – view it before it disappears forever!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ✅ Звичайний лист
🔍 SVM: ⚠️ Байтинг лист
==========

📩 Приклад 10: Baiting приклад 10

✉️ Текст листа:
"""
You’ve been selected to receive two FREE flight tickets to Dubai as part of our global travel campaign! All you need to do is enter your email and complete the short application form. This offer is limited to the first 500 respondents. Act fast and experience the luxury of Dubai – absolutely free!
"""

🔍 Naive Bayes: ⚠️ Байтинг лист
🔍 Random Forest: ⚠️ Байтинг лист
🔍 SVM: ⚠️ Байтинг лист

In [53]:

```
import pandas as pd

# Список для збереження результатів
results = []

for idx, (label, email) in enumerate(baiting_examples.items(), start=1):
    for model_name in models:
        prediction = predict_email_with_model(email, model_name)
        # Витягнемо просто мітку (без імені моделі)
        label_pred = prediction.split(": ")[1]
        results.append({
            "Приклад": label,
            "Текст листа": email,
            "Модель": model_name,
            "Передбачення": label_pred
        })

# Конвертуємо в DataFrame
df_results = pd.DataFrame(results)

# Збережемо у CSV
df_results.to_csv("baiting_email_predictions.csv", index=False, encoding='utf-8-sig')
print("✅ Результати збережено у baiting_email_predictions.csv")
```

✅ Результати збережено у baiting_email_predictions.csv