МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ
«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КІБЕРБЕЗПЕКИ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри кібербезпеки

_____Анна ІЛЬЄНКО
"_____"_____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ "МАГІСТР"

Тема: Метод виявлення фішингових атак з використанням штучного інтелекту.

Виконавець:		Дмитро ТАНЦЮРА
Керівник:	д.т.н., с.н.с.	Олександр ЛАПТЄВ
Нормоконтролер:	к.т.н., доцент	Андрій ПЕТРЕНКО

Київ 2024

ДЕРЖАВНЕ НЕКОМЕРЦІЙНЕ ПІДПРИЄМСТВО
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ
«КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Факультет комп'ютерних наук та технологій
Кафедра кібербезпеки
Освітній ступінь магістр
Спеціальність 125 «Кібербезпека та захист інформації»
Освітньо-професійна програма «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки

_____Анна ІЛЬЄНКО
«30»___08_____2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
Танцюри Дмитра Юрійовича

1. Тема кваліфікаційної роботи: Метод виявлення фішингових атак з використанням штучного інтелекту.
затверджена наказом ректора від 30.08.2024 р. №1695/ст.
2. Термін виконання роботи: з 30.08.2024 по 15.12.2024
3. Вихідні дані до роботи: Види кібератак, дослідження методів виявлення фішингових атак. Застосування та використання штучного інтелекту для вирішення завдань кібербезпеки.
4. Зміст пояснювальної записки: аналіз методів виявлення фішингових атак, розробка методу виявлення фішингових атак з використанням штучного інтелекту, розробка методичних рекомендацій щодо виявлення фішингових атак з використанням штучного інтелекту.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: презентація.

6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Провести аналіз існуючих методів виявлення фішингових листів	30.08.2024 – 05.09.2024	*Виконано*
2.	Провести дослідження щодо ефективності використання Штучного Інтелекту (ШІ) в існуючих системах інформаційної безпеки	06.09.2024 – 15.09.2024	*Виконано*
3.	Проаналізувати існуючи системи інформаційної безпеки виявлення фішингу з використанням ШІ	16.09.2024 – 25.09.2024	*Виконано*
4.	Порівняти ефективність використання різних моделей виявлення фішингових листів	26.09.2024 – 02.10.2024	*Виконано*
5.	Розробити рекомендації щодо використання релевантних моделей виявлення фішингових листів з використанням ШІ, відповідно до потреб різних організацій.	03.10.2024 – 25.10.2024	*Виконано*

7. Дата видачі завдання: «30» __08__ 2024 р.

Керівник кваліфікаційної роботи: _____ Олександр ЛАПТЄВ
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання: _____ Дмитро ТАНЦЮРА
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Метод виявлення фішингових атак з використанням штучного інтелекту.»: 137 с., 69 літературних джерела, 5 рисунків та 2 таблиці.

Об’єкт дослідження: процес виявлення фішингових атак з використанням штучного інтелекту.

Предмет дослідження: методи виявлення фішингових атак з використанням штучного інтелекту.

Мета кваліфікаційної роботи: розробка методу та видача рекомендацій щодо виявлення фішингових атак з використанням штучного інтелекту.

Методи дослідження: методи теорії множин, методи експертного оцінювання, методи логіки.

Практична цінність: розроблена методика виявлення фішингових атак з використанням штучного інтелекту

Наукова новизна: набув подальшого розвідку метод виявлення фішингових атак з використанням штучного інтелекту

Ключові слова: кібербезпека, штучний інтелект, фішинг, система виялення, захищеність, інтеграція, машинне навчання, оцінка ефективності.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ШІ	–	Штучний інтелект
AI	–	Artificial Intelligence
ISO	–	International Organization for Standardization
IEC	–	International Electrotechnical Commission
NIST	–	National Institute of Standards and Technology
SPF	–	Sender Policy Framework
DKIM	–	DomainKeys Identified Mail
DMARC	–	Domain-based Message Authentication, Reporting & Conformance
NLP	–	Natural Language Processing
DOM	–	Document Object Model
RNN	–	Recurrent Neural Network
OCR	–	Optical Character Recognition
SVM	–	Support Vector Machine
DL	–	Deep Learning
CNN	–	Convolutional Neural Network
GAN	–	Generative Adversarial Network
IPS	–	Intrusion Prevention System
NGFW	–	Next-Generation Firewall
SIEM	–	Security Information and Event Management
DLP	–	Data Loss Prevention
MFA	–	Multi-Factor Authentication
SOAR	–	Security Orchestration, Automation and Response
EDR	–	Endpoint Detection and Response
TIP	–	Threat Intelligence Platform

IoC	–	Indicator of Compromise
LSTM	–	Long Short-Term Memory
UBA	–	User Behavior Analytics
DNN	–	Deep Neural Network
LIME	–	Local Interpretable Model-agnostic Explanations
AUC	–	Area Under the Curve
GPU	–	Graphics Processing Unit
TPU	–	Tensor Processing Unit
FPR	–	False Positive Rate
PCA	–	Principal Component Analysis
RFE	–	Recursive Feature Elimination
СУІБ	–	Система управління інформаційною безпекою
CSF	–	Critical Success Factor
ENISA	–	European Union Agency for Cybersecurity
GDPR	–	General Data Protection Regulation
AWS	–	Amazon Web Services

ВСТУП

Фішинг – це одна з найбільш поширених кіберзагроз, яка за останні десятиліття стала серйозною проблемою для користувачів і організацій у всьому світі. У своїй суті, фішинг являє собою метод соціальної інженерії, що спрямований на обман і маніпуляцію людьми з метою отримання доступу до конфіденційних даних. Водночас розвиток фішингу як загрози пов'язаний з еволюцією технологій та популяризацією цифрових каналів комунікації. Зі збільшенням кількості користувачів Інтернету та підвищенням активності онлайн-сервісів цей тип атак став доступним і масовим засобом кіберзлочинності.

Фішинг найчастіше здійснюється за допомогою електронної пошти, текстових повідомлень, телефонних дзвінків або навіть соціальних мереж. У більшості випадків зловмисники створюють фальшиві повідомлення, які виглядають як офіційні звернення від банків, державних установ, соціальних мереж чи інших відомих організацій, з якими користувачі взаємодіють регулярно. Завдяки використанню психологічних прийомів – таких як страх втратити доступ до рахунку, повідомлення про несподівану перемогу чи вигоду, загроза штрафу або інші нагальні причини – жертви піддаються на обман і нерідко передають свої дані. Такі техніки зловмисники використовують у всіх видах фішингових атак, варіюючи лише канали та персоналізацію повідомлень.

Доволі небезпечним аспектом фішингу є його вплив на користувачів, які стають жертвами через необачність або брак знань про кібербезпеку. Втрати, пов'язані з фішингом, можуть бути значними: від фінансових збитків до компрометації особистих даних, що спричиняє серйозні проблеми для індивідуальних користувачів. Це призводить до серйозних фінансових труднощів для користувача і потребує значних зусиль для відновлення втрачених коштів і захисту своєї фінансової інформації. Окрім цього, крадіжка особистих даних може

призвести до оформлення незаконних угод, відкриття нових кредитних ліній, які залишаються непомітними для жертви до моменту, поки не виникнуть серйозні фінансові наслідки.

Однією з найважливіших цілей фішингових атак є персональні дані користувачів, такі як імена, адреси, дати народження, номери телефонів та інші види ідентифікаційної інформації. Ці дані можуть бути використані для створення фальшивих профілів або оформлення кредитів та позик на ім’я жертви. Отримані зловмисниками персональні дані дають їм можливість здійснювати різні види шахрайства, такі як подання заявки на кредит або покупка товарів у розстрочку. Інколи персональні дані стають цінним ресурсом для здійснення так званих цільових атак, коли зловмисники обирають жертву та формують персоналізовані фішингові повідомлення, які мають вищу ймовірність успіху. Завдяки персоналізованим даним, фішингові атаки можуть виглядати більш переконливо, особливо коли жертва отримує повідомлення, що містить її особисту інформацію.

Фішинг також є загрозою для організацій. У корпоративному середовищі цей вид атак часто спрямований на викрадення конфіденційних даних компанії, таких як дані клієнтів, фінансові звіти, внутрішні документи або навіть інтелектуальна власність. Компанії часто стикаються з фішинговими атаками, спрямованими на працівників, які мають доступ до чутливої інформації або корпоративних облікових записів. Якщо зловмисники отримують доступ до таких даних, це може призвести до значних фінансових втрат, репутаційних втрат та навіть правових наслідків. Організації можуть втратити довіру клієнтів, з якими вони працювали роками, що у свою чергу знижує їхню конкурентоспроможність на ринку. У разі витоку конфіденційних даних клієнтів, компанії також можуть бути зобов'язані відшкодовувати збитки постраждалим, витрачаючи значні ресурси на правові аспекти, що ще більше погіршує їхню фінансову ситуацію.

Крім того, облікові дані є однією з основних цілей фішингових атак. Під обліковими даними розуміються логіни, паролі, а також дані для доступу до

різноманітних онлайн-акаунтів, таких як облікові записи електронної пошти, соціальних мереж, банківських та інших фінансових сервісів. Зловмисники, які отримали доступ до облікових записів, можуть використовувати ці акаунти для подальших фішингових атак, розсилаючи шкідливі посилання або повідомлення до контактів жертви. Наприклад, у випадку доступу до електронної пошти зловмисник може змінити паролі до інших пов'язаних сервісів, використовуючи процедуру «відновлення доступу», що дозволяє їм легко проникнути в усі цифрові активи жертви. Викрадення облікових даних також може становити ризик для бізнесу, адже зловмисники можуть отримати доступ до корпоративних мереж, де зберігаються чутливі дані про клієнтів, контракти та інша конфіденційна інформація.

Серед інших цілей фішингових атак виділяється доступ до корпоративної інформації, яка може включати комерційну таємницю, фінансові звіти, бізнес-плани, списки клієнтів та іншу важливу інформацію для діяльності компанії. Викрадення такої інформації може мати серйозні наслідки для організації, адже конкуренти або кіберзлочинці можуть використовувати її для отримання переваг або навіть для шантажу. Іноді зловмисники, які отримали доступ до корпоративних даних, вимагають викуп за невикриття конфіденційної інформації. У випадку відмови, вони можуть публікувати ці дані в Інтернеті або продавати конкурентам, що може завдати організації не тільки фінансових збитків, але й шкоди репутації.

Особливо небезпечним є те, що фішингові атаки часто не обмежуються одним каналом комунікації. Зловмисники можуть використовувати комбінацію різних методів, щоб збільшити ймовірність успіху атаки. Наприклад, вони можуть надіслати фішинговий електронний лист із посиланням на підроблений сайт, а після цього зателефонувати жертві, видаючи себе за представників служби підтримки, щоб переконати користувача в необхідності введення своїх даних. Такі складні атаки можуть бути важко розпізнати, і навіть досвідчені користувачі можуть

піддатися обману, особливо коли атака виглядає переконливо та містить персоналізовані дані, які викликають довіру.

Окрім цього, фішинг як кіберзагроза постійно еволюціонує. Зловмисники активно використовують сучасні технології для підвищення ефективності своїх атак, наприклад, за допомогою штучного інтелекту, автоматизації або аналізу великих обсягів даних. Завдяки цьому фішингові атаки стають більш персоналізованими, таргетованими та складними для виявлення.

Фішингові атаки становлять серйозну загрозу як для індивідуальних користувачів, так і для організацій, і їх виявлення на ранніх етапах має вирішальне значення для запобігання великим втратам та мінімізації негативних наслідків. Важливість раннього виявлення фішингових атак пояснюється тим, що своєчасне розпізнавання небезпеки дозволяє швидко перервати доступ зловмисників до конфіденційної інформації або критично важливих ресурсів. У випадку з фішингом головна загроза полягає в тому, що жертва не відразу усвідомлює, що її особисті дані вже були скомпрометовані, що дає зловмисникам час використовувати цю інформацію в своїх інтересах. Ранні заходи дозволяють мінімізувати ці ризики, обмежуючи можливість подальшого несанкціонованого використання отриманих даних.

Значення раннього виявлення фішингових атак також проявляється у захисті фінансових ресурсів як окремих осіб, так і організацій. Фінансові втрати від фішингових атак можуть бути величезними, особливо коли зловмисники отримують доступ до банківських рахунків або кредитних карток жертв. У багатьох випадках, коли атака не виявлена на ранніх етапах, жертви навіть не підозрюють про несанкціоноване використання їхніх коштів до моменту, коли фінансовий збиток вже завдано. Наприклад, фішингові повідомлення, які містять посилання на підроблені банківські сайти або служби онлайн-оплати, можуть успішно обманути користувачів і призвести до того, що їхні рахунки будуть зламані. Однак, якщо фішинг виявляється на етапі, коли користувач лише починає взаємодію з підозрілим

повідомленням, то він може швидко розпізнати небезпеку і вчасно запобігти втратам. Організації, у свою чергу, можуть використовувати спеціальні засоби для моніторингу підозрілої активності на корпоративних рахунках і мінімізувати фінансові ризики шляхом блокування підозрілих транзакцій до завершення розслідування.

Окрім фінансових аспектів, раннє виявлення фішингових атак сприяє захисту репутації компаній. Для бізнесу репутація є важливим активом, і компрометація даних клієнтів або співробітників внаслідок фішингових атак може призвести до серйозних репутаційних втрат. Клієнти, дізнавшись про факт компрометації, можуть втратити довіру до організації, що негативно позначається на її бізнесі та конкурентоспроможності. Важливість репутаційного ризику обумовлює необхідність інвестицій у системи раннього виявлення загроз, які дозволяють бізнесу своєчасно ідентифікувати фішингові спроби та запобігати доступу до чутливих даних. Зокрема, підприємства можуть встановлювати засоби для автоматичного розпізнавання та блокування фішингових повідомлень до їх доставки в електронні скриньки співробітників. Це дозволяє уникнути ризику компрометації інформації, що є важливим кроком для підтримки надійності бренду та довіри клієнтів.

Раннє виявлення фішингових атак також має важливе значення для запобігання подальшому поширенню загрози. Фішинг часто стає початковою точкою більш серйозних атак, таких як злам корпоративної мережі або розповсюдження шкідливих програм. Зловмисники, отримавши початковий доступ через успішний фішинг, можуть використовувати цей доступ для поширення програм-вимагачів, шпигунського програмного забезпечення або навіть ботнетів, які здатні завдати значних збитків мережевій інфраструктурі організації. Таким чином, раннє виявлення фішингових повідомлень може стати бар'єром для проникнення зловмисників у корпоративну мережу та захисту від більш складних атак. Це особливо важливо для великих організацій, де фішингові атаки можуть

призвести до збоїв у роботі всієї компанії, спричиняючи затримки у бізнес-процесах та значні фінансові втрати.

Варто зазначити, що раннє виявлення фішингових атак тісно пов’язане з підвищенням рівня обізнаності користувачів про кіберзагрози. Освітні програми з кібербезпеки, проведення регулярних тренінгів і навчання співробітників дозволяють виявляти підозрілі повідомлення ще до моменту взаємодії з ними. Люди, які розуміють, як розпізнавати ознаки фішингових атак, значно рідше стають жертвами шахраїв. Раннє виявлення залежить не лише від автоматичних систем захисту, але й від підготовленості користувачів, які здатні швидко реагувати на потенційні загрози.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК

1.1. Традиційні методи виявлення фішингових атак

1.1.1. Використання чорних списків

Метод чорних списків (blacklists) є одним із традиційних та широко використовуваних підходів до виявлення фішингових атак. Основна ідея цього методу полягає в складанні списку відомих небезпечних сайтів, адрес або доменів, які були ідентифіковані як джерела фішингових атак. Коли користувач намагається відкрити певну вебсторінку або отримати доступ до певного ресурсу, система перевіряє цей ресурс у базі чорних списків і, якщо він включений до списку, блокує його. Даний метод досить популярний завдяки своїй простоті, адже він не вимагає значних обчислювальних ресурсів і дозволяє миттєво виявляти загрози, якщо вони вже додані до списку. Чорні списки є ефективними для блокування сайтів і доменів, які давно існують і вже отримали репутацію шкідливих. Вони широко застосовуються в антивірусних програмах, системах контролю доступу до мережі, електронній пошті та браузерах для запобігання переходам на небезпечні ресурси.

Однак метод чорних списків має кілька суттєвих обмежень і недоліків. Одним із основних є його залежність від актуальності даних у списку. У реальності фішингові домени часто змінюються, і зловмисники постійно створюють нові домени або IP-адреси для обходу чорних списків. Як наслідок, у випадках, коли фішинговий ресурс ще не доданий до чорного списку, система не зможе його виявити, що залишає користувача вразливим перед новими атаками. Оскільки зловмисники постійно вдосконалюють свої методи, підтримка актуальності чорного списку потребує постійного оновлення бази даних, а це, в свою чергу,

вимагає значних людських та технічних ресурсів. Багато організацій використовують автоматизовані системи для збору даних про шкідливі домени, але навіть такі системи не завжди встигають вчасно відстежити всі нові загрози.

Чорні списки також обмежені у своїй здатності протистояти так званим цільовим або персоналізованим фішинговим атакам. У таких випадках зловмисники можуть створити унікальні URL-адреси або сторінки, які призначені для конкретної жертви і, ймовірно, не були раніше використані. Це означає, що такі домени не містяться у жодному чорному списку, що дозволяє атакуючим обійти систему захисту. Наприклад, у випадках атак на великі корпорації зловмисники можуть створювати фальшиві сторінки, що імітують внутрішні ресурси компанії, і користувачі не підозрюють небезпеки через відсутність попереджень від системи безпеки. Тому чорні списки добре підходять для блокування відомих масових атак, однак вони є менш ефективними проти нових і специфічних загроз, які потребують додаткових методів виявлення, таких як аналіз поведінки або штучний інтелект.

Серед переваг методу чорних списків є його простота і швидкість роботи. Перевірка URL-адреси на наявність у чорному списку є дуже швидкою операцією, яка не потребує складного аналізу. Це дозволяє зменшити навантаження на систему і забезпечує миттєву реакцію на відомі загрози. Крім того, чорні списки легко інтегруються в більшість сучасних систем безпеки, що робить їх доступним засобом захисту для широкого кола користувачів. Більшість браузерів і поштових клієнтів вже мають вбудовану функціональність для блокування відомих шкідливих сайтів, і користувачам не потрібно додатково встановлювати будь-яке програмне забезпечення. Це спрощує процес захисту кінцевого користувача і підвищує загальну кібербезпеку в Інтернеті.

Проте обмеження методу чорних списків стають дедалі очевиднішими на тлі сучасних складних фішингових атак. Зловмисники активно використовують техніки обфускації URL-адрес, застосовують перенаправлення через кілька доменів, а також використовують послуги скорочення посилань, що ускладнює

роботу чорних списків. Усе це дозволяє обходити захист, оскільки система не завжди може правильно визначити, що кінцевий ресурс є фішинговим. Також, багато зловмисників використовують динамічні IP-адреси або підміняють адреси в реальному часі, що ускладнює точне відстеження шкідливих ресурсів і швидке внесення їх у чорні списки. Тому для повноцінного захисту чорні списки повинні використовуватися разом з іншими методами, зокрема, такими як машинне навчання, яке аналізує характерні ознаки підозрілих сайтів і здатне виявляти нові загрози [1].

1.1.2. Перевірка репутації доменів та URL-адрес

Водночас, перевірка репутації доменів і URL-адрес є однією з найефективніших технологій у сучасних системах для виявлення підозрілих сайтів. Цей метод базується на зборі, аналізі та оцінці історичних даних про домени та URL-адреси для визначення рівня їхньої надійності. У світі, де фішингова діяльність стає дедалі складнішою, а зловмисники вдаються до витончених технік маскування, перевірка репутації є ефективним інструментом, оскільки вона не просто фіксує відомі загрози, а й аналізує потенціал ризику нових або маловідомих доменів. Системи, що використовують цей підхід, перевіряють репутацію вебсайтів шляхом аналізу багатьох факторів, таких як вік домену, IP-адреса, реєстраційні дані, історія змін на сайті та наявність раніше зафіксованих інцидентів, пов’язаних із загрозами безпеки.

Репутація домену є показником його надійності і зазвичай базується на сукупності показників, що зменшують або підвищують ризики взаємодії з ним. Наприклад, нові або недавно зареєстровані домени часто викликають підозру, оскільки зловмисники нерідко використовують свіжі доменні імена для фішингових кампаній, які складніше відстежити та заблокувати відразу після реєстрації. Домени, які мають погану репутацію, можуть блокуватися автоматично, що є

значною перевагою у запобіганні доступу користувачів до підозрілих ресурсів. Зокрема, аналітики кібербезпеки відстежують активність доменів на предмет використання їх у масових фішингових розсилках, підробках легітимних ресурсів або поширенні шкідливих програм. Репутаційні системи дозволяють автоматично ідентифікувати такі домени та блокувати їх, навіть якщо конкретна URL-адреса раніше не фігурувала в чорних списках.

Завдяки аналізу репутації домену можна також відстежувати підозрілі зміни в поведінці вебсайту, які можуть вказувати на його компрометацію. Наприклад, сайт із позитивною репутацією може раптом почати переспрямовувати користувачів на фішингові сторінки через зламані URL-адреси. У таких випадках система перевірки репутації здатна виявити різкі зміни в активності домену, що часто є індикатором ризику. До таких показників належить, наприклад, зміна IP-адреси на ту, що належить відомому зловмиснику, раптове збільшення обсягу трафіку, а також аномальна поведінка домену, що свідчить про його можливе використання у злочинних схемах. Завдяки цьому метод перевірки репутації дозволяє відслідковувати підозрілу активність у режимі реального часу, попереджаючи користувачів про потенційні загрози ще до того, як вони почнуть взаємодію із сайтом.

Однією з переваг репутаційного підходу є його універсальність і здатність адаптуватися до різних умов. Метод перевірки репутації доменів і URL-адрес добре працює як для індивідуальних користувачів, так і для великих компаній, де існують високі вимоги до кібербезпеки. Організації можуть використовувати репутаційні бази даних для контролю доступу до певних сайтів, блокування підозрілих ресурсів і мінімізації ризику витоку інформації. У разі, якщо ресурс або домен має добру репутацію, доступ до нього зазвичай дозволяється без додаткових перевірок, що економить час і ресурси. З іншого боку, погана репутація сайту або домену може призвести до його автоматичного блокування в мережах компанії, що є ефективним способом зниження ризиків без необхідності постійного моніторингу.

Попри численні переваги, метод перевірки репутації також має свої обмеження. Зловмисники можуть адаптуватися до цієї технології, створюючи домени, які імітують надійні сайти, або використовувати домени з гарною репутацією для тимчасових фішингових атак, що можуть пройти непоміченими системою. Наприклад, часто використовуються компрометовані сайти із гарною репутацією, які спершу не викликають підозр, але можуть бути тимчасово використані для шкідливої діяльності, поки власники не встигнуть усунути загрозу. Це потребує від систем перевірки репутації здатності вчасно реагувати на такі динамічні загрози шляхом оперативного оновлення репутаційних баз та використання додаткових показників для оцінки потенційного ризику [3].

1.1.3. Аналіз метаданих електронної пошти

Аналіз метаданих електронної пошти, а саме заголовків, також є дуже важливим методом виявлення фішингових атак, який забезпечує як попереджувальне, так і ретроспективне виявлення підозрілих повідомлень. Заголовки електронних листів містять багатий набір технічної інформації про те, звідки надійшло повідомлення, через які сервери воно пройшло, а також про різні ідентифікатори автентифікації, які дозволяють відстежити надійність джерела. Фішингові атаки часто маскуються під легітимні повідомлення, тому саме за допомогою аналізу метаданих стає можливим виявлення таких підробок.

Ключовий елемент заголовка — це інформація про сервери, через які пройшло повідомлення, яка міститься в заголовках "Received". Цей компонент заголовка показує, через які вузли проходив лист до моменту доставки у поштову скриньку. У фішингових повідомленнях часто спостерігаються нетипові маршрути передачі, зокрема через сервери, розташовані в країнах з підвищеним ризиком кіберзлочинності або на IP-адресах, які раніше фігурували в базах даних зловмисних активностей. Наприклад, якщо лист, що начебто надійшов від відомої

компанії, містить вказівку на сервери, що не мають відношення до цієї компанії, це може бути вагомою ознакою підробки. Перевірка послідовності записів "Received" допомагає виявити відхилення від стандартного маршруту, а також аномальні затримки на певних вузлах, що також можуть свідчити про фішингову атаку.

Аналіз автентифікаційних записів, зокрема SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail) і DMARC (Domain-based Message Authentication, Reporting & Conformance), також є надзвичайно важливим для виявлення підроблених електронних листів. SPF дозволяє перевірити, чи надходить повідомлення з авторизованого сервера домену, що знижує ризик "спуфінгу" — підробки адреси відправника. DKIM додає цифровий підпис до заголовків листа, що підтверджує його цілісність і дозволяє отримувачу перевірити, чи не було змінено текст повідомлення після його відправлення. DMARC працює як політика безпеки, що об'єднує результати перевірок SPF і DKIM, дозволяючи організаціям визначати, як реагувати на невдалі спроби автентифікації. Завдяки DMARC компанії можуть заблокувати підроблені листи або відправити їх у спам, якщо повідомлення не відповідає стандартам автентифікації. У разі, коли фішинговий лист не проходить перевірку SPF чи DKIM, це сигналізує отримувачу, що повідомлення може бути зловмисним, оскільки зловмисник не має доступу до справжніх автентифікаційних записів домену, яким він маскується.

Ще один аспект аналізу метаданих електронної пошти стосується дослідження полів "From" і "Reply-To". Поле "From" відображає адресу відправника, але його значення легко змінити, і зловмисники часто використовують його для спуфінгу. Поле "Reply-To" може вказувати на іншу адресу для відповіді, що також часто застосовується у фішингових атаках, коли зловмисник хоче перенаправити відповіді на свої сервери або інші підконтрольні йому акаунти. Наприклад, фішинговий лист, надісланий начебто від банку, може містити адресу для відповідей, яка не належить цьому банку, що є черговим індикатором підробки.

Аналіз цих полів, а також пошук невідповідностей між ними, може дати цінну інформацію про справжнє джерело повідомлення.

Також, мета аналізу метаданих - виявлення так званих "шаблонів фішингу". Фішингові атаки часто мають схожі структури і шаблони, які зловмисники використовують для створення масових розсилок. Аналітики можуть автоматично ідентифікувати ці шаблони за допомогою алгоритмів машинного навчання, які обробляють метадані великих обсягів електронної пошти, що дозволяє попередити фішингові атаки на ранніх етапах. Виявлення шаблонів у метаданих, таких як однакові IP-адреси, повторювані заголовки або однакові "Reply-To" адреси, дозволяє ідентифікувати кампанії масових розсилок і знижувати їхній вплив на користувачів.

Ще варто відзначити роль спеціалізованого аналізу клієнтського ПЗ, з якого було відправлено повідомлення. Заголовок "User-Agent" містить інформацію про програмне забезпечення, що використовується для відправлення листа. У випадку, якщо лист надійшов нібито від офіційної організації, але в заголовку вказано нестандартний або застарілий поштовий клієнт, це може викликати підозру.

Розширений аналіз метаданих електронної пошти також охоплює інформацію про часові характеристики повідомлень. Зокрема, часовий штамп у заголовках електронної пошти може свідчити про аномалії, які нерідко трапляються у фішингових атаках. Наприклад, електронні листи від великих компаній зазвичай відправляються в межах робочого дня, відповідно до часового поясу головного офісу. Якщо лист, нібито від такої компанії, було надіслано пізно вночі або в нетиповий час, це може бути сигналом про можливу фальсифікацію. Аналіз часових характеристик також дозволяє виявляти спам-кампанії, які часто здійснюються в певні години доби, щоб максимізувати охоплення, або фішингові атаки, організовані на основі часу з низькою активністю перевірок безпеки.

Метадані електронної пошти можуть також містити інформацію про маршрутизацію, яка відображає конкретні IP-адреси, з яких було здійснено

передачу повідомлення. Це дозволяє аналітикам використовувати інструменти геолокації для визначення фізичного розташування серверів, які залучені до відправлення листа. Зіставлення даних IP-адрес із доменними іменами та географічними положеннями може вказати на невідповідність, що є явним індикатором фішингу. Наприклад, якщо лист, нібито від великої американської компанії, пройшов через сервери в країнах з високим рівнем кіберзлочинності, це вже викликає підозру. Сучасні інструменти автоматизації, які обробляють такі метадані, дозволяють виявляти підозрілі маршрути передачі й автоматично сигналізувати про них адміністраторам систем.

Окрім того, важливо звертати увагу на заголовки, пов’язані з типом контенту в повідомленні, наприклад, "Content-Type" та "MIME-Version". Ці заголовки вказують на формат повідомлення та тип вкладених файлів. Фішингові листи часто використовують нестандартні або застарілі типи вкладень або кодування, що дозволяє приховувати шкідливий вміст. Наприклад, атака може бути організована через вкладений документ у форматі, який має вразливості для експлойтів. Аналіз таких характеристик у метаданих допомагає ідентифікувати потенційно небезпечні вкладення або файли, які можуть містити шкідливий код або інструменти для викрадення даних.

Ще одним важливим елементом аналізу є вивчення інформації про сервери, які обробляли повідомлення, що часто фіксується в заголовках "Received-SPF" і "Authentication-Results". Ці заголовки показують результати перевірки повідомлень на відповідність стандартам SPF і DKIM на проміжних етапах. Якщо лист пройшов через сервери, які не підтримують автентифікаційні протоколи, або якщо вони не були належним чином налаштовані, це може стати індикатором того, що повідомлення було змінено. У багатьох випадках фішингові атаки включають передачу листа через низку підозрілих серверів, що не мають належного рівня безпеки, а це підвищує ризик підробки вмісту або заголовків.

Зрештою, методи штучного інтелекту та машинного навчання значно розширюють можливості аналізу метаданих електронної пошти. Наприклад, нейронні мережі можуть використовуватися для автоматичного виявлення аномалій у великих обсягах метаданих і заголовків електронної пошти. Ці алгоритми здатні враховувати комбінації численних параметрів, таких як IP-адреса відправника, час відправлення, маршрути передачі та інші показники, і виявляти складні шаблони, які характерні для фішингових атак. Використання таких систем дозволяє виявляти нові типи атак, які можуть бути невідомими традиційним системам аналізу, і забезпечує вищий рівень безпеки для користувачів та організацій, що протистоять кіберзагрозам [4].

1.1.4. Ключові недоліки традиційних методів

Традиційні методи виявлення фішингових атак, хоча й досі є відносно ефективними у боротьбі з відомими загрозами, мають певні недоліки, які обмежують їх здатність протистояти новим, складнішим типам атак. Одним із основних недоліків є їхня залежність від попередніх знань про атаки або від детермінованих правил, що регулюють поведінку системи. Такі методи часто базуються на чорних списках, сигнатурах або фільтрах, які розробляються на основі аналізу минулих інцидентів. У результаті вони стають неефективними проти нових атак, особливо тих, які використовують передові методи маскування і обходу захисту. Наприклад, якщо фішинговий сайт використовує новий, раніше невідомий домен або унікальні шаблони електронних листів, які не відповідають раніше визначеним характеристикам, традиційні методи можуть не ідентифікувати його як загрозу, оскільки він не включений до чорного списку або не відповідає наявним шаблонам.

Крім того, фішинг еволюціонує завдяки застосуванню методів соціальної інженерії та поведінкового маскування, що значно ускладнює його виявлення.

Традиційні методи часто фокусуються на технічних аспектах, таких як аналіз IP-адреси чи домену, але не можуть ефективно розпізнати психологічні тактики, які використовуються зловмисниками для маніпуляції користувачами. Наприклад, фішингові атаки, спрямовані на конкретних осіб, можуть використовувати персоналізовані повідомлення, які виглядають дуже правдоподібно. Традиційні системи часто не можуть врахувати цей психологічний контекст і мають низький рівень ефективності при зіткненні з індивідуально націленими атаками, такими як spear-phishing.

Ще одним обмеженням традиційних методів є їхня недостатня гнучкість та адаптивність. Традиційні системи захисту часто розробляються з фіксованими правилами та алгоритмами, які важко або довго оновлювати відповідно до змін у кіберпросторі. Через це вони можуть бути повільними в адаптації до нових видів загроз. Зловмисники можуть легко змінювати свої методи та уникати виявлення, використовуючи техніки, які були розроблені спеціально для обходу таких статичних систем. Наприклад, фішингові сайти можуть швидко змінювати свої URL-адреси, дизайн чи тексти для уникнення блокувальних фільтрів. Такі швидкі зміни ставлять традиційні методи в невигідне становище, адже виявлення загроз потребує часу на оновлення баз даних або модифікацію правил.

Традиційні методи також мають обмеження у виявленні фішингових атак, які використовують новітні технології, такі як підробка автентифікаційних записів або захоплення сесій через фішингові сайти. Наприклад, сьогодні фішингові сайти часто використовують сертифікати SSL/TLS, що робить їх вигляд більш легітимним. Користувачі можуть помилково вважати сайт безпечним через наявність значка безпечного з’єднання, тоді як традиційні методи не завжди здатні розпізнати цей вид маскування. Більше того, фішингові атаки часто використовують так звані "переміщення" або "редиректи", щоб уникнути виявлення. Вони перенаправляють користувача через кілька доменів перед

досягненням кінцевого шкідливого сайту, що дозволяє уникати фільтрів і традиційних методів виявлення, які працюють з фіксованими чорними списками.

Також, важливо зазначити обмеження в обробці великих обсягів даних, які створюють сучасні системи електронної пошти. Традиційні методи часто мають проблеми з масштабованістю, оскільки вони створені для обробки обмеженої кількості перевірок і порівнянь. Коли обсяги даних збільшуються, наприклад, у великій організації, традиційні методи стають менш ефективними. Це призводить до затримок у виявленні або, навіть гірше, до збільшення числа помилкових спрацьовувань, коли легітимні повідомлення помилково вважаються фішинговими.

Традиційні методи виявлення фішингових атак також страждають від високої кількості хибнопозитивних спрацьовувань, що може мати серйозні наслідки для користувацького досвіду та ефективності роботи організацій. Коли система фільтрації виявляє потенційно шкідливі листи, але помилково блокує легітимні повідомлення, це створює перешкоди для співробітників, які можуть не отримати важливу інформацію або обмінюватися повідомленнями зі своїми колегами чи клієнтами. Така ситуація змушує користувачів ігнорувати попередження або навіть вимикати фільтри, що знижує загальну безпеку системи. Крім того, надмірна кількість хибнопозитивних спрацьовувань призводить до зниження довіри до таких систем і може потребувати залучення додаткових ресурсів для перегляду та перевірки кожного потенційного випадку.

Ще одним важливим обмеженням традиційних методів є їхня нездатність ефективно обробляти мультимедійний контент, що стає все більш поширеним у сучасних фішингових атаках. Традиційні фільтри, які зосереджені на текстовому вмісті електронних листів, можуть не розпізнати загрозу, якщо фішингове повідомлення містить зображення або відео замість тексту. Зловмисники використовують цей метод, щоб уникнути текстових фільтрів, вбудовуючи важливу інформацію, як-от підроблені логотипи або посилання, у форматах, які важче піддаються автоматизованому аналізу. Таким чином, традиційні методи

виявляються безсилими перед мультимедійними фішинговими атаками, які можуть успішно обманювати користувачів.

Іншим аспектом є обмеження у здатності традиційних методів протидіяти атакам, які використовують змішані техніки для посилення ефективності фішингових кампаній. Наприклад, деякі зловмисники комбінують фішинг із техніками спуфінгу (підробки) електронних адрес або із зараженням шкідливим програмним забезпеченням, яке може запускатись після переходу на підроблений сайт. Такі складні атаки потребують глибшого рівня аналізу, який виходить за рамки стандартних перевірок на наявність сумнівних IP-адрес або нетипових заголовків у метаданих. Складність і комбіновані техніки роблять ці атаки важковловимими для традиційних методів, що призводить до значної вразливості систем безпеки.

З часом традиційні методи виявлення фішингових атак починають показувати свою обмеженість у забезпеченні масштабного захисту, особливо в умовах динамічно змінюваних загроз. Одним із ключових викликів є швидкість появи нових методів обходу захисту, які створюють реальну загрозу для компаній та користувачів. Зловмисники здатні швидко адаптуватися, впроваджуючи нові техніки для подолання застарілих систем захисту, тоді як традиційні методи вимагають часу на модифікацію і часто не встигають вчасно оновлюватися. Сучасні фішингові кампанії є складними, різноманітними, і для ефективної протидії їм потрібно застосовувати нові підходи, які враховують поведінкові і контекстуальні особливості загроз, що виходять за межі можливостей традиційних методів.

1.2. Методи, засновані на аналізі контенту

1.2.1. Лексичний аналіз контенту

Лексичний аналіз контенту також є доволі ефективним методом, що дозволяє виявляти фішингові атаки шляхом аналізу текстової частини повідомлень. Цей метод зосереджується на дослідженні ключових слів, фраз та патернів, які найчастіше зустрічаються у фішингових листах. Зловмисники часто використовують певний набір термінів та мовних конструкцій, які мають на меті викликати у користувачів почуття тривоги, страху або терміновості. Наприклад, такі слова, як “терміново”, “обмежений доступ”, “ваш обліковий запис заблоковано” є типовими для фішингових повідомлень, оскільки вони змушують користувача реагувати негайно та без роздумів. Мета такого мовного стилю – створити ілюзію невідкладності, що заважає людині критично оцінити ситуацію та сприяє поспішним діям, таким як натискання на небезпечне посилання або введення особистих даних.

Для ефективного застосування лексичного аналізу контенту створюються бази ключових слів, що мають підозрілий або ризикований характер. Алгоритми обробки тексту здатні порівнювати нові повідомлення з цими базами, визначаючи рівень загрози на основі частоти використання певних слів або фраз. До ключових категорій, які зазвичай використовуються для лексичного аналізу фішингових листів, належать слова і фрази, пов’язані з фінансовими транзакціями, безпекою акаунта, та вимогами негайної дії. Також звертається увага на специфічні формулювання, що намагаються переконати отримувача у легітимності повідомлення, наприклад, посилання на “офіційні політики”, згадки відомих банків або популярних онлайн-сервісів. Фішингові атаки часто містять слова, які можуть звучати правдоподібно та створюють враження, що лист надійшов від організації, з якою користувач має справи.

Ще один важливий аспект лексичного аналізу – це використання різних синтаксичних і граматичних особливостей, характерних для фішингових листів. Фішингові повідомлення часто мають граматичні та орфографічні помилки, неправильну побудову речень або використовують надмірно формальний або, навпаки, надмірно дружній стиль. Такі ознаки можуть бути непрямими індикаторами фішингу, оскільки офіційні організації зазвичай уникають цих недоліків. Лексичний аналіз дозволяє виявляти такі аномалії, що можуть

Лексичний аналіз контенту також охоплює аналіз структури тексту фішингових повідомлень, що допомагає виявити специфічні стилістичні прийоми та мовні шаблони, характерні для фішингу. Одним з таких прийомів є використання формулювань, які створюють ілюзію авторитету або офіційності. Наприклад, фішингові листи часто містять звернення до отримувача з використанням слів на кшталт “шанований клієнте” або “високошановний користувачу”, а також намагаються включити “офіційні” терміни, які підкреслюють серйозність звернення. Такі фрази та звернення використовуються, щоб надати повідомленню ваги і викликати враження довіри у жертви. Зловмисники використовують подібні мовні конструкції з метою вплинути на отримувача, створити відчуття важливості та змусити діяти відповідно до їхніх інструкцій.

Крім того, фішингові повідомлення часто насичені закликами до дії та директивними висловами, які спонукають користувача негайно виконати певні дії. Це можуть бути фрази на кшталт “оновіть пароль”, “підтвердьте свою особистість”, “натисніть тут для входу”, що підкреслюють терміновість і створюють враження, що зволікання може призвести до негативних наслідків. Ці заклики зазвичай супроводжуються попередженнями про потенційні ризики або втрату доступу, які активно тиснуть на отримувача емоційно. Наприклад, повідомлення може включати інформацію про можливу загрозу для фінансових ресурсів користувача, що знову ж таки стимулює до швидкої реакції, не залишаючи часу на аналіз ситуації.

Сучасні системи на основі лексичного аналізу контенту здатні ідентифікувати й аналізувати ці мовні патерни, виділяючи слова й вирази, що можуть бути індикаторами шахрайства. Такі системи використовують алгоритми обробки природної мови (Natural Language Processing, NLP), які дають змогу аналізувати текстові дані та виявляти аномалії, властиві фішинговим повідомленням. Використовуючи машинне навчання, ці алгоритми можуть навчатися на великих обсягах даних і поступово покращувати здатність розпізнавати нові мовні конструкції та ключові слова, які з’являються у фішингових повідомленнях.

Цікаво, що лексичний аналіз дозволяє не тільки ідентифікувати фішингові повідомлення, але й допомагає розрізняти їхні різновиди, такі як класичний фішинг, spear-phishing (цілеспрямований фішинг), або бізнес-фішинг, орієнтований на організації. Кожен з цих типів атак має свої мовні особливості та патерни, що може бути використано для більш точного виявлення та класифікації. Наприклад, у spear-phishing часто використовуються більш персоналізовані повідомлення, де враховуються особисті дані користувача або специфічні професійні терміни, що можуть бути частиною професійної діяльності цільової особи. З іншого боку, загальні фішингові листи мають універсальнішу структуру, зосереджену на тому, щоб привернути увагу широкого кола користувачів, незалежно від їхнього профілю чи посади.

Лексичний аналіз також активно використовується у поєднанні з іншими методами для побудови комплексних систем захисту. Наприклад, лексичний аналіз може бути частиною багатошарової системи, яка включає перевірку репутації відправника, аналіз метаданих, а також поведінковий аналіз активності користувача. Така інтеграція дозволяє збільшити точність виявлення фішингових загроз, зменшуючи ймовірність помилкових спрацьовувань і хибнонегативних результатів. Лексичний аналіз контенту є важливим компонентом таких систем, оскільки забезпечує ефективне виявлення навіть у випадках, коли інші методи, наприклад, чорні списки, можуть не спрацювати.

Однак, варто зазначити, що лексичний аналіз контенту має певні обмеження. Оскільки зловмисники постійно удосконалюють свої методи і адаптують свої повідомлення, вони можуть свідомо уникати очевидних ключових слів або використовувати більш витончені мовні конструкції, які важче розпізнати за допомогою лексичного аналізу. Так, зловмисники можуть використовувати техніки обфускації, наприклад, замінювати літери схожими символами або використовувати зображення з текстом замість простого тексту, що значно ускладнює автоматичне виявлення загроз за допомогою лише лексичного аналізу.

1.2.2. Аналіз шаблонів та структури вебсторінок

Аналіз шаблонів та структури вебсторінок є ще одним важливим методом для виявлення фішингових атак, адже зловмисники часто використовують різноманітні техніки для імітації дизайну легітимних вебсайтів, що створює ілюзію довіри у користувачів. Фішингові сторінки часто намагаються точно копіювати інтерфейс популярних платформ, таких як банківські сайти, онлайн-магазини, соціальні мережі або системи електронної пошти, щоб обманом змусити користувачів вводити особисті дані. Основними елементами, які підлягають аналізу у цьому методі, є візуальна структура сторінки, шрифти, кольори, логотипи та загальна організація інтерфейсу, що допомагають створити ілюзію легітимності. Імітація таких елементів підвищує ризик того, що користувач не зможе відрізнити фішингову сторінку від справжньої, особливо якщо зовнішні ознаки копії майже ідентичні.

Методи аналізу шаблонів та структури вебсторінок ґрунтуються на порівнянні структури сторінки з базами даних, що містять шаблони офіційних сайтів. Вебсторінки легітимних ресурсів зазвичай мають чітко визначені шаблони, що складаються з певної послідовності елементів, таких як форми для входу, панелі навігації, кнопки дій та інші компоненти. Зловмисники намагаються точно

відтворити цю послідовність, але часто допускають помилки, які можуть бути використані як індикатори фішингової активності. Наприклад, вони можуть створити подібний до оригіналу інтерфейс, але деякі елементи можуть бути розміщені не так, як на офіційному сайті, або мати відмінності у стилі шрифтів, відтінках кольорів чи використовуваних іконках. Алгоритми для аналізу структури сторінок здатні порівнювати ці елементи з оригінальними версіями та виявляти можливі відмінності, які сигналізують про шахрайський сайт.

Один із методів аналізу структури сторінок базується на аналізі HTML-коду, а також структури DOM (Document Object Model), що дозволяє отримати докладну інформацію про компоненти та їхні позиції на сторінці. Легітимні сайти, як правило, мають стабільний і відшліфований код, який регулярно оновлюється, тоді як фішингові сторінки можуть містити неякісний код або використовувати спрощені версії структур. Аналізуючи HTML та DOM-функції, системи захисту можуть виявляти певні аномалії, такі як відсутність потрібних скриптів або неповний функціонал, які можуть свідчити про шахрайський сайт. Наприклад, легітимні сторінки часто використовують складніші алгоритми захисту та перевірки користувача, такі як CAPTCHA або багатофакторна автентифікація, тоді як фішингові сторінки часто ігнорують ці аспекти через їхню складність у відтворенні.

Ще один підхід полягає в аналізі зовнішніх ресурсів та залежностей, на які посилається вебсторінка. Легітимні сайти часто використовують певні бібліотеки, сервіси для завантаження медіафайлів та інші залежності, які легко відстежити. Фішингові сайти, натомість, можуть використовувати сторонні або дешевші аналоги, що знижують вартість розробки сторінки. Зловмисники часто застосовують сервіси обфускації для приховування джерел завантаження або навіть повністю копіюють необхідні ресурси на свої сервери, що дозволяє уникнути залежності від зовнішніх джерел. Системи захисту здатні ідентифікувати такі відмінності, перевіряючи відповідність ресурсів, на які посилається сайт, базам

даних відомих легітимних сервісів, та виявляти нестандартні залежності, які можуть сигналізувати про фішингову активність.

Окрему увагу в аналізі шаблонів та структури сторінок приділяють також поведінковим аспектам, що можуть бути не властивими легітимним ресурсам. Фішингові сторінки часто спрощені і функціонують лише на поверхневому рівні: наприклад, вони можуть дозволяти лише один варіант дії, такий як введення пароля, тоді як справжній сайт пропонує різні функції та можливості для користувача.

Фішингові сторінки часто не мають повного функціоналу легітимних сайтів або виконують дії, які можуть здатися користувачеві підозрілими. Наприклад, при введенні даних фішингові сторінки можуть одразу перенаправляти користувача на іншу сторінку або відображати повідомлення про помилку, що вимагає повторного введення пароля. Такі неприродні дії можна аналізувати за допомогою спеціалізованих алгоритмів, що виявляють аномалії у взаємодії користувача з сайтом. Системи захисту використовують такі індикатори, щоб відстежувати й фільтрувати вебсторінки, які не відповідають стандартам функціональності та поведінки легітимних сайтів.

З метою маскування, зловмисники також використовують методи обфускації елементів вебсторінок, які є частиною дизайну справжніх сайтів. Обфускація дозволяє приховувати певні деталі, які можуть видати фішинговий характер сторінки. Наприклад, зловмисники можуть змінювати видимість певних HTML-елементів або використовувати стилі CSS для приховання окремих елементів від користувача, що може допомогти їм обійти систему виявлення. Проте такі методи є складними і можуть призводити до погіршення користувацького досвіду, що, у свою чергу, може бути виявлено спеціалізованими інструментами аналізу вебсторінок, які слідкують за тим, як відображаються та взаємодіють елементи на екрані.

Ще один важливий аспект аналізу структури — це перевірка сертифікатів безпеки, що використовуються на вебсторінках. Справжні сайти зазвичай

використовують сертифікати SSL/TLS від авторитетних центрів сертифікації, що гарантує надійне з'єднання та захист даних користувача. Фішингові сторінки, навпаки, часто не мають таких сертифікатів або використовують менш надійні альтернативи, які можуть бути відомі системам безпеки. Браузери та антивірусні системи можуть автоматично перевіряти наявність таких сертифікатів і повідомляти користувача про потенційні ризики. Проте в деяких випадках зловмисники можуть використовувати безкоштовні або фальшиві сертифікати, які створюють ілюзію захищеного з'єднання, тому комплексні алгоритми аналізу завжди перевіряють походження та легітимність сертифікатів.

1.2.3. Використання технік обробки природної мови (NLP)

Основний принцип NLP для виявлення фішингових атак полягає у використанні алгоритмів машинного навчання, які навчаються на великих масивах даних із текстами фішингових та легітимних повідомлень. Ці алгоритми можуть використовувати словники частих фішингових виразів, визначати аномалії у граматиці або синтаксисі та оцінювати загальний тон повідомлення. Наприклад, фішингові листи часто відрізняються від легітимних формулюванням та стилем; вони можуть містити помилки у написанні слів, грубі синтаксичні помилки чи інші відмінності, що є непритаманними для офіційної комунікації. Алгоритми NLP здатні автоматично виявляти такі деталі та сигналізувати про ризик шахрайства.

Однією з поширених технік NLP є аналіз настрою тексту (sentiment analysis), що дозволяє виявляти емоційний відтінок повідомлення. Багато фішингових атак використовують мову, яка створює атмосферу нагальності чи страху, наприклад, заявляючи про можливе блокування облікового запису чи необхідність негайної оплати. Аналіз настрою дозволяє системі визначати, коли текст спробує створити відчуття тривоги, що може бути маркером фішингу. Крім того, моделі NLP можуть

враховувати специфіку текстових сигналів, що притаманні різним типам фішингових атак, як-от підроблені повідомлення від банків чи страхових компаній, та використовувати цю інформацію для розпізнавання патернів, характерних для конкретних видів шахрайства.

Техніки NLP також дозволяють виявляти аномалії у структурі тексту, наприклад, нетипове поєднання слів або незвичний порядок фраз. Оскільки фішингові повідомлення часто створюються автоматизованими інструментами або перекладаються за допомогою машинних перекладачів, їх структура може виглядати незвично або нетипово. Застосування алгоритмів NLP для аналізу подібних особливостей дозволяє виділяти повідомлення, які не відповідають характерному стилю легітимних текстів. Для цього часто використовують моделі глибинного навчання, такі як рекурентні нейронні мережі (RNN) або трансформери, які можуть розпізнавати навіть складні та непрямі аномалії.

Ще однією важливою сферою застосування NLP для виявлення фішингових атак є визначення наявності підозрілих запитів на передачу конфіденційної інформації. Алгоритми NLP можуть знаходити типові фрази, пов’язані з проханнями надати особисті дані, фінансову інформацію чи паролі, що є ключовими сигналами фішингових атак. Наприклад, зловмисники часто використовують такі вирази, як "увійдіть у свій акаунт", "перевірте інформацію", "підтвердіть вашу особу", що змушують користувача виконати дії, які можуть призвести до крадіжки даних. Нейронні мережі та інші техніки машинного навчання здатні навчитися розпізнавати такі шаблони, що дозволяє підвищити ефективність виявлення фішингу.

Також, важливий аспект використання технік NLP для виявлення фішингових текстів полягає у здатності систем обробляти багатомовний контент. Оскільки фішингові атаки можуть бути спрямовані на користувачів з різних країн, зловмисники часто готують повідомлення на кількох мовах. Використання NLP для багатомовного аналізу дозволяє виявляти шахрайські тексти, незалежно від мови,

на якій вони написані. Сучасні моделі машинного навчання, як-от трансформери, можуть одночасно обробляти тексти на різних мовах і знаходити фішингові патерни у широкому контексті. Це робить NLP надзвичайно потужним інструментом, здатним аналізувати велику кількість текстових повідомлень з високим рівнем точності, незалежно від мови та культурних особливостей тексту.

Щоб реалізувати багатомовний аналіз, алгоритми часто проходять попереднє навчання на великому обсязі текстів з різних мов. Наприклад, моделі на зразок BERT або GPT, які пройшли навчання на мільярдах текстових фрагментів, здатні виявляти навіть тонкі лінгвістичні маркери, які можуть свідчити про шахрайську природу повідомлення. Цей підхід є особливо ефективним у багатомовних країнах, де зловмисники можуть цілеспрямовано використовувати місцеві діалекти або специфічні мовні варіанти для введення в оману користувачів. Використання NLP дозволяє не тільки знайти ключові фрази, але й розуміти контекст, у якому вони використовуються, що значно підвищує точність виявлення фішингу.

Крім того, технології NLP здатні проводити семантичний аналіз тексту, що дозволяє моделювати зв’язок між словами та фразами в тексті. Це означає, що системи можуть ідентифікувати фішингові повідомлення навіть тоді, коли зловмисники використовують схожі, але не зовсім ті ж самі слова та вирази. Наприклад, замість використання слова "пароль", фішингові листи можуть містити синонімічні варіанти або опосередковані натяки на необхідність надати конфіденційну інформацію. Семантичний аналіз дає змогу системам безпеки «розуміти» справжнє значення тексту, що дозволяє їм розпізнавати навіть ті повідомлення, які замасковані або сформульовані непрямо.

Ще одна техніка NLP, що ефективно застосовується у виявленні фішингових текстів, це аналіз частоти та повторюваності ключових слів. Фішингові повідомлення часто переповнені певними словами або фразами, які зловмисники вважають найбільш ефективними для залучення уваги користувача. Використовуючи методи статистичного аналізу, NLP-системи можуть визначати

аномально високу частоту використання певних слів у тексті, що може бути сигналом фішингової атаки. Системи можуть налаштовуватися так, щоб враховувати не тільки конкретні слова, але й шаблони, що містять ці слова, наприклад «увійдіть», «оновіть дані», «підтвердження акаунту» та інші. Це дозволяє ефективно ідентифікувати фішинг і за рахунок виявлення тонких нюансів, які важко виявити вручну.

Інтеграція NLP в системи кібербезпеки також дає можливість автоматизувати процес перевірки текстів, що суттєво прискорює виявлення фішингових атак і дозволяє реагувати на них в реальному часі. Оскільки NLP-системи працюють в автоматизованому режимі, вони можуть миттєво перевіряти великий обсяг текстів, таких як вхідні електронні листи, повідомлення в соціальних мережах або навіть текстовий контент на підозрілих вебсайтах.

1.2.4. Обмеження методів контент-аналізу

Методи контент-аналізу, незважаючи на їхню популярність і важливість у виявленні фішингових атак, мають низку обмежень, які ускладнюють ефективне виявлення сучасних та особливо адаптивних фішингових методів. Головним чином, ці обмеження пов’язані зі здатністю зловмисників удосконалювати свої тактики, що дозволяє обходити традиційні алгоритми аналізу контенту, побудовані на основі ключових слів, структурних шаблонів та лексичного аналізу. Однією з ключових причин неефективності методів контент-аналізу є гнучкість фішингових атак, які все частіше використовують нові підходи до побудови повідомлень, створюючи контент, що виглядає максимально природно та відповідно до специфіки легітимних текстів.

Сучасні фішингові атаки можуть використовувати методи соціальної інженерії, що дозволяють максимально наблизити повідомлення до справжнього, і навіть за допомогою машинного навчання змінювати текст залежно від отриманих

раніше реакцій користувачів. Наприклад, деякі зловмисники можуть автоматично коригувати структуру та стиль повідомлень, адаптуючи їх до мови користувача або з урахуванням конкретних формулювань, які є ефективнішими в певному контексті. Таким чином, якщо алгоритми контент-аналізу ґрунтуються на попередньо вивчених шаблонах фішингових повідомлень, то будь-які зміни у стилі та структурі можуть дозволити фішинговим атакам пройти повз систему виявлення.

Також, обмеження контент-аналізу полягає в залежності від списків заборонених слів та фраз, що часто використовуються у фішингових повідомленнях. Однак зловмисники навчилися обходити такі алгоритми, використовуючи синоніми або навіть навмисні помилки, що робить традиційний підхід до виявлення менш надійним. Крім того, зростає частка фішингових повідомлень, у яких мінімізується кількість таких характерних слів, натомість вони побудовані на загальних фразах або нейтральних формулюваннях, що робить їх складними для розпізнавання з допомогою контент-аналізу. Наприклад, замість прямої вказівки надати особисті дані фішингові повідомлення можуть містити завуальовані натяки або заклики, що не здаються підозрілими, особливо якщо система виявлення не розпізнає контекст, у якому вживаються ті чи інші слова.

Ще одне значне обмеження методів контент-аналізу стосується складнощів у виявленні багатомовного або локалізованого контенту. Зловмисники можуть застосовувати фішингові атаки, написані на рідкісних або малодоступних мовах, що знижує ефективність контент-аналізу, оскільки більшість систем спрямовані на аналіз основних мов та діалектів. Багато фішингових атак орієнтуються на специфічні регіони або використовують локалізовані версії мов, що створює додаткові труднощі для автоматизованих систем аналізу контенту, оскільки локальні ідіоми або специфічні формулювання не завжди включаються у словники підозрілих слів і фраз. Таким чином, навіть якщо система обробки природної мови працює ефективно на англомовному тексті, вона може пропустити загрози в

повідомленнях іншими мовами, що ускладнює створення універсальних систем виявлення.

Іншим викликом для методів контент-аналізу є використання мультимедійного контенту або нестандартних форматів тексту у фішингових атаках. Сучасні фішингові атаки можуть містити зображення тексту замість звичайного текстового контенту, що перешкоджає алгоритмам контент-аналізу коректно обробляти та аналізувати повідомлення. Текст, представлений у вигляді зображення, може бути розпізнаний лише за допомогою додаткових технологій, таких як оптичне розпізнавання символів (OCR), проте навіть у такому випадку точність може залишатися низькою, особливо якщо зображення мають низьку якість або текст прихований у складній графіці. Це значно знижує ефективність традиційних алгоритмів контент-аналізу, які розраховані на обробку лише текстового контенту.

Також важливо враховувати, що сучасні фішингові атаки активно використовують персоналізацію, що ускладнює виявлення шаблонів. Зловмисники можуть збирати інформацію про цільових користувачів із відкритих джерел або попередніх витоків даних, щоб створити повідомлення, які виглядають як особисті або адресовані конкретній людині. Такі повідомлення часто виглядають дуже природно і не містять жодних явних ознак фішингу, що робить їх важкими для виявлення традиційними методами контент-аналізу.

1.3. Методи, засновані на машинному навчанні

1.3.1. Підхід машинного навчання до фішингу

Методи машинного навчання (ML) стали однією з провідних стратегій у сучасній кібербезпеці, включаючи виявлення фішингових атак. Основна перевага ML у боротьбі з фішингом полягає в його здатності виявляти нові, раніше не

виявлені загрози, адаптуватися до змін у поведінці зловмисників і розпізнавати нетипові шаблони в різноманітному контенті. На відміну від традиційних методів, які покладаються на жорстко задані правила або пошук ключових слів, алгоритми машинного навчання здатні самостійно навчатися на основі великих обсягів даних і виявляти навіть найдрібніші відмінності між легітимними та фішинговими повідомленнями.

Одним з основних підходів, що використовуються в ML для виявлення фішингу, є класифікація. Класифікатори, такі як дерева рішень, логістична регресія, наївний баєсівський класифікатор та методи опорних векторів (SVM), аналізують певний набір характеристик електронного листа чи вебсторінки (таких як заголовки, контент, структура URL) і класифікують їх як "фішингові" або "легітимні". Кожна з цих моделей має свої переваги та недоліки, а також певні вимоги до обробки даних. Наприклад, наївний баєсівський класифікатор може бути ефективним для роботи з текстовим контентом, де існує залежність між словами, проте він менш точний для виявлення складних структур URL або поведінкових патернів. У свою чергу, SVM часто демонструє високу точність у складних завданнях класифікації, але вимагає значних обчислювальних ресурсів [6].

Інший підхід – це методи кластеризації, які використовуються для виявлення груп подібних фішингових атак. На відміну від класифікації, де кожен об’єкт відноситься до певної категорії, у кластеризації дані об’єднуються у групи на основі схожості без попередньо заданих міток. Кластеризація є особливо корисною у випадках, коли зловмисники використовують схожі шаблони, проте модифікують деталі, щоб уникнути виявлення. Завдяки цьому методи кластеризації дозволяють виявляти неочевидні шаблони та зв'язки між елементами, що допомагає у виявленні нових або модифікованих фішингових атак. Такий підхід дозволяє групувати схожі URL, що використовуються у фішингових атаках, виявляючи зміни у структурі чи підходах фішингових сайтів, які зазвичай важко помітити вручну.

Методи глибокого навчання (DL), які є підкласом машинного навчання, також знайшли широке застосування в боротьбі з фішингом. Такі методи, як нейронні мережі, обробка природної мови (NLP), рекурентні та згорткові нейронні мережі, можуть обробляти великі обсяги інформації, аналізуючи складні шаблони у контенті. Наприклад, рекурентні нейронні мережі (RNN) використовуються для аналізу текстового контенту електронних листів, особливо коли необхідно врахувати послідовність слів та структуру речень. Згорткові нейронні мережі (CNN) застосовуються для аналізу зображень та графічного контенту, що є корисним у випадках, коли фішингові атаки використовують нестандартні шрифти або зображення замість тексту для приховування контенту від традиційних методів аналізу [9].

Однією з ключових переваг ML є можливість створення моделей на основі великих датасетів, що включають приклади легітимного та фішингового контенту. Машинне навчання може аналізувати безліч ознак, таких як частота та розташування ключових слів, структура URL, IP-адреси відправників, поведінкові патерни тощо. Більше того, на відміну від традиційних методів, моделі машинного навчання можуть адаптуватися до нових загроз, постійно навчатися на основі нових даних і вдосконалювати свої алгоритми для підвищення точності та зменшення помилок.

Втім, найбільшою проблемою при застосуванні машинного навчання для виявлення фішингових атак є загроза помилкових спрацювань (false positives). Навіть найкращі моделі можуть іноді хибно класифікувати легітимний контент як фішинговий, особливо у випадках, коли користувачі отримують повідомлення з нетиповими формулюваннями або незвичними структурами посилань. Високий рівень помилкових спрацювань може знизити довіру до системи виявлення та підвищити навантаження на служби кібербезпеки через необхідність вручну перевіряти підозрілі випадки. Цю проблему можна частково вирішити шляхом налаштування порогів класифікації або впровадження багаторівневих підходів, які

використовують кілька моделей ML для прийняття остаточного рішення. Наприклад, первинний алгоритм може оцінювати ризик, а додатковий алгоритм — повторно перевіряти лише ті випадки, де ймовірність фішингової атаки висока.

Інтеграція методів обробки природної мови (NLP) у машинне навчання стала важливим кроком у розвитку сучасних алгоритмів для виявлення фішингу, як було згадано раніше. NLP дозволяє моделям глибше розуміти контент повідомлень, ідентифікуючи підозрілі мовні конструкції, фрази, стилістичні особливості, які характерні для фішингових атак. Наприклад, алгоритми можуть виявляти такі показники, як надмірне використання закликів до дії ("Urgent!", "Immediate Action Required") або структурування повідомлення, схоже на відомі шаблони шахрайських листів. За допомогою NLP моделі можуть навчитися розрізняти тон, стиль, словниковий запас та навіть виявляти емоційний вплив на отримувача. Це особливо корисно для виявлення тих повідомлень, де зловмисники намагаються впливати на емоції, щоб прискорити реакцію користувача.

Методи машинного навчання також ефективно використовуються для виявлення фішингу на основі URL-адрес та інших технічних аспектів повідомлень, про що вже згадувалось раніше. Зокрема, моделі можуть аналізувати доменні імена, піддомени, структуру URL, використання чисел або спеціальних символів, які часто характерні для підроблених вебсайтів. Використовуючи таку інформацію, алгоритми можуть визначити ймовірність того, що адреса є частиною фішингової атаки, навіть якщо домен раніше не був відомий як шкідливий. Деякі з новітніх моделей також включають аналіз IP-адрес, географічного розташування серверів, часу реєстрації домену тощо, щоб створити повну картину ризиків для кожного окремого випадку.

Одним із ключових інноваційних підходів у машинному навчанні для боротьби з фішингом є використання ансамблевих методів. Ансамблеві методи комбінують кілька різних алгоритмів або моделей для підвищення точності виявлення. Наприклад, рішення, яке комбінує класифікацію на основі дерев рішень,

SVM та глибокого навчання, може забезпечити значно точніше виявлення фішингових атак, ніж використання однієї моделі. Такий підхід дозволяє використовувати переваги кожного з алгоритмів, компенсуючи слабкі сторони іншими моделями, що підвищує адаптивність системи і робить її більш стійкою до нових видів атак.

Машинне навчання також дає можливість здійснювати проактивний моніторинг та передбачення. Деякі алгоритми здатні виявляти нові фішингові загрози ще до їхнього активного поширення. Це стає можливим завдяки аналізу змін у шаблонах поведінки зловмисників і виявленню нових ознак у реальному часі. Машинне навчання може виявляти так звані "ранні сигнали" нових загроз, що дозволяє кіберзахисту організацій завчасно підготуватися до можливих атак.

Ще одним важливим аспектом є можливість використання розподіленого навчання (Federated Learning), що дозволяє навчати моделі на основі даних з різних джерел, не передаючи самі дані між організаціями. Це корисно в умовах, коли важливо забезпечити конфіденційність і уникнути обміну чутливою інформацією між різними компаніями або навіть підрозділами. Такий підхід дозволяє створювати більш точні та загальні моделі для виявлення фішингових загроз, одночасно захищаючи конфіденційність даних [11].

1.3.2. Класифікація методів машинного навчання

Методи машинного навчання, які використовуються для виявлення фішингових атак, можуть бути класифіковані на різні категорії залежно від алгоритмів та підходів, що застосовуються для аналізу даних та прийняття рішень. Одним із найбільш важливих аспектів класифікації є розподіл методів на основі типів класифікаторів, які використовуються для навчання моделей. Серед популярних класифікаторів — нейронні мережі, метод опорних векторів (SVM), випадкові ліси та інші алгоритми, кожен із яких має свої унікальні властивості та

підходи до обробки інформації. Вибір відповідного класифікатора залежить від конкретних вимог до точності, швидкості обробки та ресурсів для навчання, а також від типу даних, з якими модель буде працювати.

Нейронні мережі є одним із найбільш потужних та гнучких інструментів для вирішення завдань машинного навчання, зокрема для виявлення фішингових атак. Нейронні мережі зазвичай складаються з багатьох шарів, які обробляють вхідні дані на різних рівнях абстракції. Це дозволяє нейронним мережам ідентифікувати складні шаблони та зв'язки у великих наборах даних, таких як тексти повідомлень, URL-адреси, метадані листів та інші елементи, які можуть містити ознаки фішингу. Зокрема, рекурентні нейронні мережі (RNN) та згорткові нейронні мережі (CNN) широко використовуються у сфері кібербезпеки. RNN особливо ефективні для обробки послідовних даних, таких як текст, оскільки вони здатні зберігати та використовувати контекст попередніх слів у послідовності, що може допомогти виявити лексичні ознаки фішингу. CNN, зі свого боку, часто застосовуються для аналізу структурованих даних або зображень, і вони можуть бути корисні для аналізу візуальної схожості фішингових сайтів з легітимними [12].

Метод опорних векторів (SVM) є ще одним потужним інструментом для класифікації, особливо корисним у випадках, коли потрібно розрізнити два класи даних — наприклад, фішингові та нефішингові повідомлення. SVM працює за принципом визначення гіперплощини, яка максимально відокремлює ці два класи. Цей метод дуже ефективний для обробки високорозмірних даних і має високу точність навіть при обмежених обсягах навчальних даних, що робить його корисним для виявлення фішингових атак у невеликих вибірках. Крім того, SVM може використовуватися у комбінації з іншими методами, щоб поліпшити загальну точність виявлення, особливо у складних сценаріях, де класичні методи аналізу можуть давати недостатні результати.

Інший популярний метод — випадковий ліс (Random Forest), який є ансамблевим методом машинного навчання. Випадковий ліс складається з

множини дерев рішень, кожне з яких дає своє рішення щодо того, чи є повідомлення фішинговим. Остаточне рішення базується на голосуванні всіх дерев, що дозволяє знизити ризик помилок, оскільки ансамбль компенсує неточності окремих дерев. Випадкові ліси особливо корисні для виявлення фішингових атак, оскільки вони можуть аналізувати складні взаємозв’язки між різними елементами повідомлень або вебсторінок, як-от ключові слова, посилання та IP-адреси. Висока ефективність цього методу виявляється в його здатності обробляти великі обсяги даних і адаптуватися до нових шаблонів атак завдяки можливості додавати нові дерева, які враховують нову інформацію [13].

На додаток до випадкових лісів, інші ансамблеві методи, такі як градієнтний бустинг (Gradient Boosting) та адаптивний бустинг (AdaBoost), також застосовуються для виявлення фішингу. Вони працюють шляхом послідовного покращення якості моделі на основі помилок попередніх класифікаторів. У випадку градієнтного бустингу кожне нове дерево додається таким чином, щоб мінімізувати помилки, зроблені попередніми деревами, що дозволяє досягати дуже високої точності. Такі методи є надзвичайно корисними для виявлення фішингових атак, оскільки вони можуть адаптуватися до постійно змінюваних тактик зловмисників і забезпечувати високу чутливість навіть у випадках нових або малопоширених атак.

Баєсівські класифікатори, зокрема Наївний баєсівський класифікатор (Naive Bayes), є широко використовуваним підходом для виявлення фішингових атак через свою здатність швидко аналізувати текст і обчислювати ймовірності на основі історичних даних. Цей метод вважається "наївним", оскільки припускає, що всі ознаки (наприклад, ключові слова, домени, IP-адреси) є незалежними, хоча насправді вони можуть мати взаємозв'язки. Незважаючи на цю умовну незалежність, баєсівські класифікатори демонструють високі результати у сфері виявлення фішингу, особливо коли фішинговий контент має чіткі патерни, такі як певні ключові слова чи фрази. Завдяки швидкій обробці даних і низьким обчислювальним витратам, Наївний баєсівський метод є ефективним рішенням для

систем фільтрації електронної пошти, де необхідно швидко обробляти велику кількість повідомлень.

Серед інших класифікаторів також можна відзначити метод k-найближчих сусідів (k-Nearest Neighbors, або k-NN), який, хоча і менш поширений у виявленні фішингу, може бути корисним для певних типів аналізу, особливо коли важливо враховувати подібність з уже відомими фішинговими повідомленнями. Метод k-NN працює, знаходячи k найближчих сусідів для нового зразка даних і визначаючи його категорію на основі категорій сусідів. Якщо більшість сусідів класифікується як фішингові, то нове повідомлення також позначається як потенційно шкідливе. Основним недоліком k-NN є висока обчислювальна витратність для великих наборів даних, що обмежує його ефективність у реальному часі, але він може бути корисним у комбінованих моделях, де різні класифікатори працюють разом для покращення загальної точності виявлення.

Інші сучасні методи, такі як дерева рішень, також знаходять своє місце в системах виявлення фішингу, оскільки вони можуть працювати з численними параметрами та умовами. Дерева рішень створюють модель на основі різних характеристик даних, яка поступово розподіляє повідомлення по категоріях за допомогою послідовності рішень. Вони особливо корисні для побудови інтуїтивно зрозумілих моделей, що полегшує їхнє використання для пояснення результатів та інтеграції в системи кібербезпеки. Хоча дерева рішень можуть бути менш точними в порівнянні з ансамблевими методами, вони дозволяють більш чітко виявляти логічні правила, які можуть допомогти відокремити фішингові повідомлення від легітимних.

Ансамблеві методи класифікації, такі як ансамблеві нейронні мережі або комбінації алгоритмів, часто виявляються ефективними, оскільки поєднують кілька підходів для досягнення більш точного результату. Такі ансамблі можуть поєднувати, наприклад, нейронні мережі з SVM або з баєсівськими класифікаторами, що дозволяє компенсувати слабкі сторони одного методу

сильними сторонами іншого. Ансамблевий підхід також дозволяє системам адаптуватися до нових шаблонів атак, що постійно змінюються, оскільки різні методи можуть враховувати зміни у фішингових тактиках, що робить систему більш гнучкою та адаптивною.

Логістична регресія є також одним з основних алгоритмів машинного навчання, який часто використовується для класифікаційних задач, таких як виявлення фішингових атак. Цей метод ґрунтується на статистичних принципах і використовує логістичну функцію для оцінки ймовірності того, що новий зразок даних належить до певного класу (наприклад, "фішинг" чи "нормальний"). Логістична регресія добре працює для бінарних класифікаційних задач, що робить її ідеальною для випадків, коли потрібно визначити, чи є певний електронний лист фішинговим. Перевагою логістичної регресії є її інтерпретованість і простота в налаштуванні та впровадженні. Завдяки лінійності, цей метод забезпечує швидку обробку даних та відносно низькі вимоги до обчислювальних ресурсів, що робить його привабливим варіантом для інтеграції в існуючі системи кібербезпеки.

1.3.4. Виклики та обмеження використання машинного навчання

Використання машинного навчання для виявлення фішингових атак значно підвищило ефективність сучасних систем кібербезпеки, однак такі методи мають свої виклики та обмеження. Одним із основних бар’єрів є потреба в великій кількості якісних даних для навчання моделей. Машинне навчання потребує значних обсягів даних, які повинні відображати широкий спектр фішингових і нефішингових атак, щоб модель могла навчитися розрізняти підозрілі ознаки. Ця проблема стає ще гострішою у випадках, коли дані можуть бути обмежені або недостатньо різноманітні, що призводить до зниження точності моделей.

Дані для тренування моделей машинного навчання повинні відповідати вимогам якості та повноти, оскільки на основі цих даних алгоритм "вчиться"

визначати, що є нормою, а що – відхиленням. У випадку фішингових атак, важливою вимогою є наявність широкого спектра зразків, які можуть включати різні типи електронних листів, посилань, URL-адрес та іншого контенту. Проблема ускладнюється тим, що фішингові атаки швидко розвиваються, і нові типи атак можуть бути дуже відмінними від попередніх. Наприклад, якщо модель навчена на старих даних, вона може не виявляти нові види атак, які використовують нові методи соціальної інженерії чи приховування шкідливих URL-адрес [15].

Ще однією серйозною проблемою є узагальнення моделей машинного навчання. Узагальнення — це здатність моделі ефективно працювати з новими даними, які вона раніше не бачила. Це є ключовим моментом, оскільки реальні дані можуть суттєво відрізнятися від даних, на яких модель була навчена. Занадто тісне прив’язування моделі до навчальних даних може призвести до перенавчання (overfitting), коли алгоритм добре працює з тренувальними даними, але демонструє низьку ефективність з новими. Фішингові атаки, зокрема, відзначаються високою варіативністю, оскільки зловмисники постійно змінюють тактики, а також використовують різні комбінації соціальної інженерії та технічних прийомів. Це створює складнощі для моделей, які можуть бути надмірно залежними від конкретних патернів, а не здатними розпізнавати нові, раніше не бачені підходи.

Крім того, складнощі виникають з точки зору вибору параметрів моделі та налаштування алгоритму, адже деякі методи машинного навчання вимагають тривалої та складної оптимізації. Процес налаштування моделі може бути трудомістким і потребує глибоких знань у галузі, щоб уникнути помилок, таких як перенавчання або недонавчання (underfitting).

Ще одним викликом у використанні машинного навчання для виявлення фішингових атак є постійна потреба у нових, актуальних даних. Оскільки фішингові атаки швидко еволюціонують, зловмисники адаптують свої стратегії, щойно старі методи перестають працювати. Це призводить до того, що моделі, які демонстрували хороші результати з певним набором даних, поступово втрачають

свою ефективність. Щоб зберігати високий рівень точності, ці моделі мають регулярно оновлюватися та "перенавчатися" на нових даних, які відображають сучасні патерни фішингових атак. Це вимагає великих ресурсів для збору та обробки інформації, а також витрат часу на перебудову та тестування моделей.

Проблема даних також пов'язана з явищем дисбалансу класів. Часто в наборах даних фішингових і нефішингових повідомлень спостерігається дисбаланс, тобто значно менше зразків фішингових повідомлень, ніж легітимних. Це особливо характерно для великих організацій, де кількість легітимних транзакцій чи комунікацій значно перевищує кількість спроб фішингу. У випадку дисбалансу класів модель може навчитися ігнорувати фішингові повідомлення або надавати їм низьку ймовірність виявлення, що значно знижує її ефективність. Для вирішення цієї проблеми використовуються різні техніки, як-от ресемплінг (перевзяття даних) та використання специфічних метрик, які допомагають оцінити ефективність моделі на менш представлених класах [17].

Іншим важливим викликом є забезпечення захисту від атак на самі моделі машинного навчання. Атаки на алгоритми, відомі як атаки на базі протидії (adversarial attacks), можуть використовуватися зловмисниками для маніпуляції або обману моделей, змушуючи їх приймати фальшиві рішення. Наприклад, фішингове повідомлення може бути спеціально змінене так, щоб уникнути виявлення моделлю. Це можна зробити шляхом додавання або заміни певних символів, використання синонімів чи інших засобів, що дозволяють замаскувати підозрілі риси. У результаті модель не здатна адекватно розпізнати шкідливий контент, що знижує її надійність.

Крім того, одним із обмежень машинного навчання є його обчислювальні вимоги. Деякі методи, такі як нейронні мережі або методи ансамблю, вимагають значних обчислювальних ресурсів і часу для тренування, особливо на великих наборах даних. Це стає важливою перешкодою для невеликих організацій або

підприємств, які можуть не мати доступу до потужного обладнання або фінансових можливостей для підтримки таких систем.

1.4. Сучасні тренди та інновації в методах виявлення фішингових атак

1.4.1. Використання штучного інтелекту (ШІ) для виявлення фішингових атак

В останні роки технології штучного інтелекту (ШІ) відіграють надзвичайно важливу роль у боротьбі з фішинговими атаками. ШІ здатний покращити традиційні методи виявлення шкідливих дій завдяки використанню потужних алгоритмів, які можуть швидко обробляти великі обсяги даних і знаходити приховані шаблони, що сигналізують про фішингову активність. Це дозволяє значно підвищити точність і швидкість виявлення, особливо в умовах, коли фішингові атаки постійно змінюються і розвиваються, ставлячи перед спеціалістами з кібербезпеки нові виклики. Однією з головних переваг ШІ є його здатність до навчання і самовдосконалення: моделі штучного інтелекту можуть адаптуватися до нових патернів атак, що з’являються, на відміну від традиційних методів, які часто потребують ручного оновлення та модифікації.

ШІ може застосовуватися для виявлення фішингових атак на кількох рівнях. Перший рівень — це аналіз вхідних даних, таких як електронні листи, вебсайти та повідомлення у месенджерах. На цьому етапі технології ШІ можуть допомогти виявити потенційно шкідливі ознаки, як-от підозрілі слова, структури повідомлень, а також особливості доменних імен, які часто використовуються зловмисниками. Наприклад, використовуючи обробку природної мови (NLP), алгоритми ШІ здатні визначати "мову фішингу", включаючи специфічні слова та фрази, такі як «терміново», «обов’язково», «натисніть тут», що можуть свідчити про спробу ввести користувача в оману. ШІ також здатен виявляти граматичні помилки або

некоректне форматування, які часто зустрічаються в фішингових повідомленнях і є однією з характерних ознак атак такого типу.

Другий рівень, на якому ШІ є ефективним, полягає у виявленні аномалій у поведінці користувачів або системи. Алгоритми можуть аналізувати потоки даних у реальному часі, відстежуючи, чи відповідають дії певних користувачів очікуваним нормам поведінки, чи ні. Наприклад, якщо обліковий запис починає здійснювати незвичні операції, як-от масове розсилання електронних листів чи доступ до конфіденційної інформації в позаштатний час, система може автоматично позначити таку поведінку як підозрілу. Це особливо актуально для захисту корпоративних мереж, де важливо виявляти фішингові спроби в режимі реального часу. ШІ здатен враховувати різні змінні, включаючи час доби, географічне розташування, тип операцій та історію поведінки, що допомагає запобігти фальшивим позитивам та зосередитися на дійсно небезпечних інцидентах.

Важливим напрямом використання ШІ в боротьбі з фішингом є розробка спеціалізованих моделей глибокого навчання, які аналізують як вміст повідомлень, так і структурні особливості вебсайтів. Ці моделі можуть виявляти відмінності між легітимними і фішинговими сайтами на основі багатьох факторів: візуальний стиль, розташування елементів інтерфейсу, використання певних кольорів, кнопок, форм та інших елементів дизайну. Глибоке навчання може використовуватися для створення інструментів, які автоматично перевіряють цільові сторінки і попереджають користувачів про можливі ризики, якщо сайт має підозрілу структуру, схожу на відому фішингову модель.

Штучний інтелект також використовується для аналізу зразків фішингових атак та адаптації до нових патернів. Сучасні зловмисники часто намагаються змінювати свої методи, щоб обійти системи захисту, впроваджуючи незначні зміни в структурі фішингових електронних листів або вебсайтів. Моделі ШІ можуть виявляти такі варіації, навчаючись на величезних обсягах даних. Вони здатні розрізняти навіть незначні відхилення від нормального, типового вмісту, що робить

ШІ надзвичайно ефективним у виявленні «змінюваних» фішингових кампаній. За допомогою технік, як-от трансферне навчання, моделі можуть перенавчатися на нових даних швидше, ніж традиційні системи. Це забезпечує здатність ШІ бути гнучким і швидко адаптуватися до нових тактик зловмисників.

Крім того, штучний інтелект допомагає значно знижувати кількість фальшивих спрацювань, що є серйозною проблемою для традиційних методів виявлення фішингу. Алгоритми ШІ можуть точніше диференціювати між фішинговими та звичайними повідомленнями, в результаті чого користувачі рідше отримують сповіщення про нешкідливий вміст. Це особливо важливо для корпоративного середовища, де надмірна кількість фальшивих спрацювань може призвести до втрати довіри до системи кіберзахисту та зниження її ефективності. Завдяки глибокому аналізу та вдосконаленим алгоритмам ШІ, система може забезпечувати більш високу точність і мінімізувати хибні спрацювання.

Одним із перспективних напрямів є використання комбінованих моделей ШІ, що поєднують різні методи машинного навчання, як-от дерева рішень, логістичну регресію, нейронні мережі та методи ансамблю. Такий підхід дозволяє забезпечити глибший аналіз та інтеграцію кількох критеріїв для виявлення фішингових атак. Наприклад, нейронні мережі можуть ефективно обробляти текстові та зображувальні дані, а інші алгоритми — аналізувати метадані, як-от частота згадуваних слів, час отримання повідомлень, IP-адреси тощо. Завдяки поєднанню цих технік створюються складні системи, що можуть з великою точністю визначати, чи є контент підозрілим, враховуючи широкий спектр показників.

Інша важлива область дослідження пов’язана з використанням генеративних моделей ШІ, таких як генеративно-змагальні мережі (GAN), які допомагають розробникам симулювати нові фішингові атаки і таким чином навчати системи захисту на більш різноманітних прикладах. GAN дозволяють генерувати «штучні» фішингові повідомлення та сайти, які використовуються для навчання моделей, що допомагає системам розширювати свої можливості в ідентифікації загроз. Це також

дає змогу швидше реагувати на нові типи фішингу, оскільки системи вже мають у своїх наборах даних схожі варіанти.

Штучний інтелект також сприяє покращенню кіберзахисту в реальному часі. Системи, засновані на ШІ, можуть оперативно реагувати на спроби фішингу, виявляючи їх буквально в момент потрапляння до користувача. Це дозволяє миттєво ізолювати потенційно небезпечний контент або блокувати його на серверному рівні ще до того, як він досягне кінцевого отримувача. Інструменти з використанням ШІ працюють, як своєрідний захисний бар'єр, що забезпечує багаторівневий захист, особливо в умовах постійного збільшення обсягів фішингових атак. Це дозволяє зменшити ризик шкідливого впливу на організацію або окремих користувачів.

1.4.2. Інтеграція з іншими системами безпеки

Інтеграція систем виявлення фішингових атак з іншими засобами кібербезпеки стає дедалі важливішою для ефективного протистояння сучасним загрозам. З розвитком технологій і збільшенням кількості кібератак поєднання різних підходів дає змогу створити багаторівневий захист, який враховує як зовнішні, так і внутрішні загрози. Такі інтегровані підходи включають використання системи запобігання вторгнень (IPS), фаєрволів нового покоління (NGFW), систем управління подіями та інформацією безпеки (SIEM), а також захисту від втрати даних (DLP) для комплексного моніторингу та реагування на спроби фішингу.

Однією з найважливіших складових інтегрованої системи безпеки є SIEM-системи, які об'єднують всі події та загрози, що виникають у мережі, та надають уніфіковану картину стану безпеки організації. Поєднання методів виявлення фішингових атак з SIEM дозволяє аналізувати події з різних джерел, таких як електронна пошта, веб-трафік, системи автентифікації тощо, що підвищує

ймовірність своєчасного виявлення фішингу. Завдяки застосуванню ШІ та алгоритмів машинного навчання, SIEM-системи можуть також прогнозувати можливі загрози на основі історичних даних і поведінкових шаблонів, виявляючи підозрілі зміни у користувацькій активності або аномальні спроби доступу до ресурсів.

Інтеграція з технологіями DLP також є важливим аспектом у боротьбі з фішингом. Системи DLP контролюють передачу конфіденційних даних, що особливо важливо в умовах, коли фішингова атака спрямована на викрадення чутливої інформації. Підключення DLP до системи виявлення фішингу дозволяє автоматично блокувати передачу даних, якщо система виявляє підозрілий запит або активність, що може бути результатом успішної фішингової атаки. Наприклад, якщо система виявляє, що користувач намагається відправити файли з конфіденційною інформацією на підозрілий зовнішній сервер, вона може заблокувати це діяння або попередити адміністратора.

Інтеграція з фаєрволами нового покоління (NGFW) також є важливою частиною сучасного підходу до кібербезпеки. NGFW забезпечують захист на мережевому рівні, аналізуючи не тільки IP-адреси, але й контент переданих даних. Сучасні NGFW можуть взаємодіяти з системами виявлення фішингу, ідентифікуючи підозрілі домени та блокуючи доступ до них ще до того, як фішингове повідомлення досягне користувача. Це дозволяє запобігти потенційним атакам, зменшивши кількість підозрілих повідомлень, що потрапляють до кінцевого користувача, і знижуючи ймовірність фішингових інцидентів.

Фільтрація електронної пошти є ще одним важливим компонентом інтегрованої системи виявлення фішингу. З використанням сучасних технологій, як-от ШІ, системи для фільтрації пошти можуть аналізувати заголовки, контент і вкладення листів, щоб ідентифікувати ознаки фішингових атак. Інтеграція з іншими системами кібербезпеки, такими як SIEM або DLP, дозволяє фільтрам електронної пошти отримувати дані про потенційні загрози з різних джерел і автоматично

навчатися на нових прикладах, що покращує їхню здатність виявляти атаки. Завдяки цьому організації можуть забезпечити безпеку свого поштового трафіку та зменшити ризик фішингових атак.

Інший напрямок інтеграції — використання багатофакторної автентифікації (MFA) у комбінації з системами виявлення фішингу. MFA додає ще один рівень захисту, що робить фішингові атаки менш ефективними навіть у разі успішного викрадення облікових даних користувача. Якщо фішингова атака спрямована на отримання пароля, то без одноразового коду, що генерується MFA, зловмисник не зможе отримати доступ до системи. Інтеграція між системами виявлення фішингу і MFA дозволяє блокувати підозрілі запити на автентифікацію або виводити попередження користувачам, якщо їхні облікові записи стають мішенню для потенційної атаки.

Також, з огляду на сучасні виклики, важливо інтегрувати методи виявлення фішингу з системами автоматизованого реагування на інциденти (SOAR). SOAR-системи дозволяють автоматично виконувати сценарії реагування на загрози, значно прискорюючи обробку потенційних інцидентів. Якщо система виявлення фішингу ідентифікує підозрілий лист або URL, SOAR може автоматично заблокувати його, створити інцидент у системі управління подіями та сповістити відповідальних фахівців. Це дає можливість швидко реагувати на загрози без ручного втручання, що значно підвищує ефективність кіберзахисту.

Інтеграція сучасних систем виявлення фішингових атак з іншими засобами кібербезпеки також забезпечує більш комплексний захист і дозволяє створювати взаємодіючі механізми реагування на складні атаки. Наприклад, інтеграція з технологіями поведінкового аналізу стає дедалі актуальнішою, оскільки вона дає змогу визначати відхилення в поведінці користувачів або пристроїв. Такі системи аналізують активність користувачів, спираючись на певні шаблони і поведінкові ознаки. У випадку фішингу, коли зловмисник отримує доступ до облікового запису, його поведінка може відрізнятися від звичайних дій користувача. Інтеграція методів

виявлення фішингових атак із системами поведінкового аналізу може дозволити вчасно виявити подібні аномалії та автоматично вжити заходів — від обмеження доступу до додаткової автентифікації.

Крім цього, великі компанії дедалі частіше інтегрують системи виявлення фішингу з технологіями хмарної безпеки, особливо в умовах зростання дистанційної роботи та перенесення багатьох сервісів у хмарні середовища. Завдяки цьому організації можуть здійснювати централізоване виявлення фішингу і блокувати підозрілі активності незалежно від фізичного місцезнаходження користувачів. Системи хмарної безпеки, які підтримують фішинговий моніторинг, забезпечують захист від атак, що націлені на дані, які зберігаються в хмарних сховищах. Вони можуть аналізувати трафік і дані, передані через хмарні служби, і виявляти підозрілі операції, що допомагає виявляти фішингові атаки навіть за межами локальної мережі організації.

Ще один важливий напрямок інтеграції — це поєднання систем виявлення фішингу з технологіями кіберзахисту кінцевих точок, як-от системи EDR (Endpoint Detection and Response). Виявлення фішингових атак на рівні кінцевих точок дозволяє запобігати поширенню шкідливих дій у разі успішної атаки. EDR-системи можуть реагувати на підозрілу активність на комп’ютері користувача, наприклад, блокувати виконання скриптів або доступ до шкідливих сайтів, які можуть бути частиною фішингової атаки. Інтеграція методів фішингового моніторингу з EDR дозволяє виявляти та блокувати фішингові атаки на стадії взаємодії з кінцевою точкою, знижуючи ризики витоку даних.

Нарешті, важливим є інтеграція з системами аналітики загроз (Threat Intelligence Platforms, TIP). TIP-системи дозволяють організаціям отримувати актуальну інформацію про нові загрози та індикатори компрометації (IoC), що використовуються у фішингових атаках. Використання такої інформації дозволяє системам виявлення фішингу швидше адаптуватися до нових загроз, ідентифікувати підозрілі елементи в реальному часі, що підвищує загальний рівень

захисту організації. Інтеграція систем виявлення фішингових атак з TIP дозволяє об’єднувати дані з різних джерел, таких як IoC, дані про домени та IP-адреси, підозрілі файли та URL, які використовуються у фішингових кампаніях. Це дає можливість оперативно блокувати небезпечний трафік та захищати кінцевих користувачів від нових загроз, знижуючи ризики і підвищуючи ефективність захисту від фішингу.

1.4.3. Автоматизовані рішення на основі поведінкового аналізу

Автоматизовані рішення на основі поведінкового аналізу стають дедалі популярнішими в контексті виявлення фішингових атак, адже вони дозволяють не тільки реагувати на відомі загрози, а й ідентифікувати нові, раніше невідомі види атак. Використовуючи поведінкові патерни, такі системи можуть помітити відхилення від нормальної поведінки користувача або пристрою, що є одним із ключових індикаторів потенційного фішингу. Такі рішення часто включають аналіз як діяльності користувача (наприклад, місце доступу, час роботи, незвичайні операції), так і активності з пристроями, на яких користувач здійснює взаємодію з мережевими ресурсами. Це робить їх універсальним інструментом, що може працювати у фоновому режимі та постійно відстежувати можливі аномалії.

Однією з переваг поведінкового аналізу є можливість створення профілю для кожного користувача або пристрою, що дозволяє виявляти підозрілі дії на основі минулого досвіду. Наприклад, якщо співробітник зазвичай працює з певного місця та у певний час, раптовий доступ до системи з іншої локації або в незвичний час може викликати підозру. Крім того, автоматизовані системи можуть звертати увагу на специфіку дій користувача — наприклад, спроби відкрити або завантажити незвичні файли, виконувати нестандартні фінансові операції, змінювати конфігурацію облікових записів. Усі ці поведінкові патерни дозволяють створити

комплексну систему безпеки, яка здатна самостійно виявляти потенційно шкідливі дії без необхідності постійного втручання людини.

У сфері поведінкового аналізу ключову роль відіграють алгоритми машинного навчання, які здатні адаптуватися до нових патернів поведінки та покращувати точність виявлення з часом. Використання таких алгоритмів дозволяє системі не лише відстежувати аномалії в режимі реального часу, але й швидко ідентифікувати їх як частину потенційної фішингової атаки. Одним із прикладів є застосування методів кластеризації, що дозволяють відокремити підозрілі патерни від звичайних, або використання алгоритмів класифікації, які здатні визначити, чи відповідає конкретна дія користувача потенційній фішинговій активності. Наприклад, якщо користувач несподівано взаємодіє з підозрілим доменом або вводить особисту інформацію на незвичному ресурсі, система може зреагувати миттєво і попередити можливе порушення.

Ще одним важливим аспектом автоматизованих рішень є можливість їхньої інтеграції з іншими системами безпеки, що дозволяє створювати комплексні рішення. Наприклад, поєднання поведінкового аналізу з системами обробки природної мови (NLP) може забезпечити додатковий рівень захисту, адже NLP-технології дозволяють виявляти потенційно небезпечні повідомлення або зміст на основі мови, якою вони написані. Це особливо корисно при обробці електронної пошти або чат-повідомлень, які можуть містити ознаки фішингу.

Окрім аналізу окремих поведінкових патернів, такі системи також можуть здійснювати кореляцію між різними подіями для виявлення складних атак, які можуть містити кілька фаз або етапів. Наприклад, багато фішингових атак не обмежуються лише збором інформації, а й здійснюють кілька етапів, таких як фаза розвідки, фаза доступу та використання зібраних даних. Автоматизовані системи можуть аналізувати ці події у контексті і відстежувати підозрілі зв'язки між ними, що дозволяє значно покращити здатність до виявлення комплексних фішингових кампаній.

Також важливо зазначити, що поведінковий аналіз є ефективним інструментом для виявлення не тільки звичайного фішингу, а й так званого «спірфішингу» (spear-phishing), який спрямований на конкретних осіб або організації. Спірфішинг часто містить персоналізовані повідомлення та розрахований на певний рівень довіри жертви, що робить його особливо небезпечним і складним для виявлення. За допомогою поведінкового аналізу система може відстежувати навіть незначні відхилення в поведінці користувача або взаємодії з потенційними зловмисниками, що значно збільшує шанси на успішне виявлення таких загроз.

Автоматизовані рішення, засновані на поведінковому аналізі, також мають потенціал для вдосконалення на основі навчання від попередніх інцидентів. Це означає, що система здатна адаптуватися та постійно оновлювати свої патерни, що забезпечує ефективний захист навіть проти нових, невідомих загроз. У міру того як система отримує більше даних про поведінку користувачів та інциденти безпеки, вона може постійно вдосконалювати свою здатність до виявлення фішингових атак, не обмежуючись лише раніше відомими патернами або сигнатурами. Такий підхід до виявлення фішингу є особливо цінним у великих організаціях, де кількість користувачів і пристроїв створює складне середовище для захисту.

Сучасні автоматизовані системи, засновані на поведінковому аналізі, також отримують додатковий рівень надійності завдяки можливості об'єднувати дані з різних джерел. Це дозволяє створювати загальну картину поведінки як окремих користувачів, так і цілих груп у мережі організації. Такий підхід дозволяє не тільки ідентифікувати підозрілу поведінку конкретного користувача, а й визначити групові патерни, які можуть вказувати на цілеспрямовані атаки. Наприклад, якщо зловмисники здійснюють фішингову атаку, спрямовану на конкретну компанію, вони часто діють через низку послідовних операцій, імітуючи нормальні дії співробітників. Використання поведінкових патернів дає змогу швидше виявити аномалії, які вказують на присутність загрози.

Іншим важливим аспектом є можливість поведінкового аналізу швидко адаптуватися до нових методів фішингових атак, які зловмисники постійно вдосконалюють. Оскільки фішинг активно використовує соціальну інженерію, яка орієнтована на взаємодію з кінцевим користувачем, система поведінкового аналізу може ідентифікувати нові загрози на підставі спроб зловмисників маніпулювати поведінкою співробітників. Наприклад, якщо користувач здійснює нехарактерні дії, такі як надання додаткової інформації, натискання на зовнішні посилання або введення конфіденційних даних на нетипових для нього ресурсах, система може попередити його про можливу загрозу.

Поведінковий аналіз також дозволяє організаціям використовувати адаптивні методи захисту, що базуються на динамічному формуванні ризикових профілів. Кожен користувач має певний профіль ризику, який оновлюється залежно від його дій і взаємодії з системою. Це дає змогу більш точно визначити, коли слід вживати додаткових заходів безпеки, таких як двофакторна аутентифікація, запит на підтвердження дій тощо. Наприклад, користувач, який зазвичай працює з корпоративною поштою і не відкриває зовнішніх файлів, раптом починає отримувати електронні листи з невідомих джерел і здійснює дії, пов'язані з доступом до конфіденційних даних. У такому випадку система може автоматично вимагати додаткової аутентифікації або обмежити доступ до певних ресурсів, поки користувач не підтвердить свої дії.

Ще однією важливою складовою поведінкового аналізу є здатність до виявлення «тихих» фішингових атак, які можуть залишатися непоміченими впродовж тривалого часу. Це стосується, наприклад, атак, де зловмисники поступово отримують доступ до інформації без яскравих ознак вторгнення. Використовуючи накопичення поведінкових даних, такі атаки можна виявити за рахунок малопомітних змін у діях користувача. Система може фіксувати навіть незначні відхилення, наприклад, нетипові локації доступу, тривалість сеансів, зміну в стилі взаємодії з певними додатками тощо. Усі ці дані допомагають виявляти

приховані загрози та вчасно реагувати на них, мінімізуючи можливі втрати для організації.

Крім того, використання поведінкових рішень для боротьби з фішингом також дозволяє побудувати більш гнучкі системи реагування на загрози. Замість того, щоб просто блокувати підозрілі дії, автоматизовані рішення можуть формувати більш «м'які» відповіді, такі як зміна рівня доступу, відправка попереджень користувачу або додаткова перевірка дій. Це робить систему безпеки менш обтяжливою для звичайних користувачів, дозволяючи їм продовжувати роботу, не відчуваючи сильного впливу з боку засобів безпеки. У той же час, такі заходи дозволяють вчасно запобігти можливим атакам, не обмежуючи основну роботу організації.

Таким чином, автоматизовані рішення на основі поведінкового аналізу становлять потужний інструмент у боротьбі з фішинговими атаками, забезпечуючи гнучкість, адаптивність і здатність до виявлення нових загроз. Вони дозволяють реагувати на аномалії в режимі реального часу, інтегруються з іншими технологіями кібербезпеки і можуть значно знижувати ризики від фішингових атак завдяки використанню передових алгоритмів машинного навчання та поведінкових патернів.

1.4.4. Перспективи розвитку методів виявлення фішингових атак

Одним із перспективних напрямів розвитку методів виявлення фішингових атак є інтеграція з технологіями штучного інтелекту та машинного навчання, яка дозволяє значно покращити ефективність і швидкість реагування на загрози. Штучний інтелект здатен здійснювати глибокий аналіз величезних обсягів даних у реальному часі, що дає можливість швидко ідентифікувати нові фішингові схеми та розпізнавати аномалії, які можуть свідчити про фішингову атаку. Прогнозується, що у майбутньому такі системи зможуть не лише виявляти фішингові атаки, а й

передбачати їх на основі аналізу історичних даних та поведінкових моделей користувачів, що дозволить підприємствам більш проактивно реагувати на кіберзагрози.

Також важливим трендом є розвиток гібридних методів виявлення фішингу, що поєднують різні підходи, зокрема аналіз контенту, поведінковий аналіз та машинне навчання. Гібридні моделі дозволяють підвищити точність та знизити кількість помилкових спрацьовувань, об’єднуючи переваги кількох підходів та компенсуючи їхні недоліки. Наприклад, аналіз контенту може швидко виявляти фішингові повідомлення на основі ключових слів або певних шаблонів, тоді як поведінковий аналіз дозволяє знайти фішинг, навіть якщо зміст повідомлення не має явних ознак підробки. Машинне навчання, у свою чергу, може навчитися розпізнавати нові патерни, які не були відомі раніше, і таким чином допомагає гібридним методам залишатися ефективними навіть у змінюваних умовах.

Іншим перспективним напрямом є розробка та впровадження інструментів для розподіленої кібербезпеки, зокрема із застосуванням блокчейн-технологій. Розподілені бази даних можуть використовуватися для зберігання чорних списків фішингових сайтів, репутаційних рейтингів та іншої інформації, необхідної для боротьби з фішингом. Завдяки блокчейну ця інформація може бути доступною для перевірки у реальному часі різними системами, що ускладнює для зловмисників маскування своїх дій. Кожен вузол у мережі може отримувати і підтверджувати інформацію від інших учасників, що робить систему менш залежною від централізованих структур та підвищує її надійність.

Перспективним є також розвиток інтелектуальних агентів, здатних автономно взаємодіяти з користувачем та допомагати йому у розпізнаванні фішингових атак. Такі агенти можуть аналізувати вхідну електронну пошту, повідомлення у месенджерах та інші канали комунікації, перевіряючи наявність підозрілих елементів. Інтелектуальні агенти можуть працювати як на рівні окремих пристроїв, так і на рівні корпоративних мереж, забезпечуючи багаторівневий захист

від фішингових атак. У майбутньому такі агенти можуть стати невід'ємною частиною систем інформаційної безпеки, надаючи користувачам рекомендації або автоматично блокуючи доступ до потенційно небезпечних ресурсів.

Крім того, прогнози щодо майбутнього розвитку технологій виявлення фішингу вказують на необхідність активного впровадження методів адаптивної безпеки, які дозволяють налаштовувати захисні механізми в залежності від конкретної ситуації та ризикового профілю користувача. Адаптивні методи можуть застосовуватися як для індивідуального захисту користувачів, так і для загального захисту організаційних мереж. Наприклад, якщо система виявляє, що користувач працює з високоризиковими ресурсами або відвідує вебсайти, що є новими для його поведінкової моделі, вона може тимчасово обмежити доступ до певних функцій або підвищити рівень аутентифікації.

Інновації у сфері обробки природної мови (NLP) також відіграватимуть важливу роль у вдосконаленні методів боротьби з фішингом. Технології NLP дозволяють аналізувати текстове наповнення електронних листів, повідомлень і вебсайтів, ідентифікуючи маніпулятивні прийоми або риторичні стратегії, що часто використовуються у фішингових атаках. У майбутньому NLP-алгоритми можуть стати настільки досконалими, що зможуть ідентифікувати фішинг навіть у повідомленнях, написаних різними мовами або з застосуванням метафор, іронії чи інших складних мовних структур, що робить фішинг ще більш різноманітним і важким для автоматичного виявлення.

Також очікується посилення використання контекстуального аналізу для покращення ефективності антифішингових рішень. Завдяки зростанню обсягів зібраних даних та покращенню алгоритмів обробки інформації, системи зможуть більш глибоко розуміти контекст дій користувача та проводити аналіз на основі ширших історичних даних. Це дозволить не лише розпізнавати окремі аномальні дії, а й враховувати їхній зв'язок із загальним профілем поведінки користувача, що зробить виявлення фішингових атак більш точним.

Ще одним перспективним напрямом розвитку методів виявлення фішингових атак є впровадження технологій прогнозної аналітики, що дозволяє передбачати потенційні загрози на основі моделювання майбутніх дій користувачів та зловмисників. Прогнозна аналітика здатна використовувати дані з попередніх атак, соціально-демографічну інформацію та поведінкові моделі, щоб оцінити, які користувачі, платформи або ресурси можуть бути вразливими в майбутньому. Такий підхід не тільки зменшує час на реагування, але й створює можливості для запобігання атакам ще на етапі їх планування, забезпечуючи більш проактивну стратегію кіберзахисту.

Трендом, що набирає популярності, є використання індивідуалізованих моделей безпеки, які враховують унікальний профіль кожного користувача або організації. Ці моделі формуються на основі персоналізованих даних, таких як шаблони поведінки в мережі, переваги в способах зв'язку та робочі звички. Інтеграція індивідуалізованих моделей дозволяє створити динамічну систему безпеки, яка швидко адаптується до змін у поведінці користувача, і тим самим підвищує ефективність антифішингових заходів. Наприклад, якщо система виявляє, що певний користувач зазвичай не отримує повідомлень від сторонніх організацій, і раптом отримує підозрілий лист, вона може автоматично його маркувати як потенційно небезпечний.

Ще один важливий аспект перспективного розвитку – вдосконалення алгоритмів, що займаються так званим «глибоким фішингом», коли зловмисники використовують неочевидні підходи та витончені тактики соціальної інженерії для маніпулювання користувачем. Сучасні інструменти виявлення фішингу часто засновані на шаблонних методах аналізу, які можуть бути недостатньо ефективними проти динамічних і змінних патернів поведінки. У зв’язку з цим прогнозується розвиток технологій, які забезпечують більш глибокий аналіз людського фактору та допомагають виявляти навіть непрямі загрози, зважаючи на поведінкові закономірності та індивідуальні особливості користувачів. Це включає

аналіз послідовностей дій, що передують атаці, відстеження найдрібніших аномалій у взаємодії з поштовими скриньками або сайтами, а також побудову персональних профілів ризику для користувачів і організацій.

Також очікується зростання популярності хмарних рішень для боротьби з фішингом, особливо у великих організаціях, які працюють з великим обсягом даних та розподіленими командами. Хмарні рішення дозволяють обробляти велику кількість інформації в реальному часі, забезпечуючи доступність антифішингових інструментів з будь-якої точки світу. Це важливо для багатонаціональних компаній, де необхідно забезпечити безпеку для співробітників, що працюють віддалено, або на території з обмеженими можливостями інфраструктури кіберзахисту. Хмарні сервіси також забезпечують централізоване оновлення безпекових політик та можливість швидкого впровадження нових методів боротьби з фішингом.

Додатково прогнозується значне зростання застосування багатофакторного аналізу ризиків, який враховує не лише зміст або структуру повідомлень, але й сукупність додаткових факторів, що свідчать про можливу атаку. Це можуть бути, наприклад, час відправлення повідомлення, IP-адреса відправника, географічне розташування та часова зона. Багатофакторний підхід дозволяє системі формувати ширшу картину можливих загроз, зменшуючи ймовірність помилкових спрацьовувань і підвищуючи загальну точність виявлення.

Отже, у цьому розділі було проведено комплексний аналіз сучасних методів виявлення фішингових атак, який охоплює як традиційні підходи, так і інноваційні рішення на основі машинного навчання. Розглянуто переваги й недоліки найпоширеніших методів виявлення фішингових атак, які забезпечують базовий рівень захисту.

РОЗДІЛ 2
РОЗРОБКА МЕТОДУ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК З ВИКОРСИТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

2.1. Використання штучного інтелекту в кібербезпеці

2.1.1. Розвиток ШІ у світі

Штучний інтелект (ШІ) — це розгалужена галузь інформатики, яка займається створенням алгоритмів і систем, здатних виконувати завдання, що зазвичай вимагають людського інтелекту, такі як розпізнавання образів, обробка природної мови, ухвалення рішень, навчання та адаптація. Основою ШІ є спроба моделювання розумових процесів, а також створення систем, що можуть удосконалюватись із часом, навчаючись на нових даних. Сучасний ШІ базується на комбінації математичних моделей, алгоритмів, обчислювальних потужностей та великих обсягів даних, які дозволяють створювати гнучкі та адаптивні системи.

Історія розвитку штучного інтелекту (ШІ) охоплює багатовіковий шлях від філософських міркувань до наукових теорій і сучасних практичних досягнень у сфері комп’ютерних технологій. Ідея створення механізмів, здатних імітувати інтелектуальні процеси людини, має давнє коріння, яке починається з античної філософії та математики і досягає свого апогею лише в наш час, завдяки стрімкому розвитку обчислювальної техніки.

У давніх цивілізаціях, таких як Єгипет та Греція, вже існували міфи про автоматичні механізми. Наприклад, у грецькій міфології бог Гефест створював роботів-слуг, які виконували різноманітні завдання у його майстерні. Такі легенди показують, що вже тоді людина прагнула розробити засоби, здатні автоматично

діяти й допомагати в повсякденному житті. Далі, вже в середньовічній Європі, під час епохи Відродження, виникли численні наукові та технічні інновації, серед яких були і перші механічні автоматони. Одним із відомих інженерів того часу був Леонардо да Вінчі, який створював складні механізми, такі як роботизований лицар, що міг рухати кінцівками.

Інший важливий етап в історії розвитку штучного інтелекту починається в 17 столітті з появою логічних теорій та обчислень. Піонером у цій сфері був Рене Декарт, який вважав, що мислення можна поділити на окремі операції, які піддаються формалізації. На його думку, завдяки цьому мислення можна було б перекласти на мову алгоритмів і правил, ідею, яка пізніше стала основою для створення перших моделей штучного інтелекту. Розвиток логіки продовжувався у працях інших вчених, таких як Блез Паскаль і Ґотфрід Лейбніц, які мали ідеї щодо обчислювальних машин і робили спроби розробити апарати, що могли б здійснювати прості арифметичні операції.

Важливий зсув у бік практичного створення інтелектуальних механізмів стався на межі 19-20 століть завдяки праці Чарльза Беббіджа і Ади Лавлейс. Беббідж запропонував конструкцію «аналітичної машини» — пристрою, який міг би виконувати послідовність інструкцій, тобто бути програмованим. Ця машина стала першим прототипом сучасного комп’ютера. Ада Лавлейс, працюючи з Беббіджем, написала програму для його машини, яка дозволяла їй обчислювати числа з використанням спеціального алгоритму. Вона також передбачала, що такі машини могли б виконувати й інші, не лише арифметичні завдання, що зробило її першим програмістом та однією з перших, хто зрозумів концепцію універсального комп’ютера.

Наступним важливим етапом у розвитку ШІ став початок комп’ютерної ери у 1940-х роках. З винаходом першого електронного комп’ютера ENIAC (1946 рік) відкрилися нові можливості для дослідження процесів обчислення та автоматизації. Цей період також став часом розквіту кібернетики, нової наукової галузі, яка

вивчала зв’язки між автоматами та живими організмами. Норберт Вінер, один із засновників кібернетики, розробив теорію зворотного зв’язку, яка виявилася ключовою для створення машин, здатних адаптуватися до умов середовища і саморегулюватися.

1950-ті роки стали переломним моментом у розвитку штучного інтелекту, коли англійський математик Алан Тюрінг опублікував статтю «Обчислювальні машини та інтелект», у якій поставив питання: «Чи можуть машини мислити?». Тюрінг розробив тест, відомий як тест Тюрінга, для оцінки здатності машини імітувати людське мислення настільки, щоб обдурити людину. Це стало першим визначенням, яке окреслило межі того, що можна назвати «інтелектом» у машин. Незважаючи на те, що концепція тесту Тюрінга зазнала критики, вона стала символічною і важливою для розвитку штучного інтелекту.

У 1956 році в Дартмутському коледжі відбулася знаменита конференція, яка вважається датою народження науки про штучний інтелект як окремої дисципліни. На конференції були присутні відомі науковці того часу, такі як Джон Маккарті, Марвін Мінський, Клод Шеннон та Герберт Саймон. Саме на цій конференції Джон Маккарті вперше запропонував термін «штучний інтелект», визначивши його як науку та інженерію, яка займається створенням інтелектуальних машин. Це була перша спроба сформулювати основи і завдання нової науки, спрямованої на дослідження методів і принципів, які дозволили б створювати машини з властивостями інтелекту.

З середини XX століття почався швидкий розвиток обчислювальної техніки, що стимулювало дослідження в галузі ШІ. У цей час створюються перші програми, здатні вирішувати математичні завдання, грати в шахи та переводити тексти. Перші досягнення, такі як програма General Problem Solver (1957), були зосереджені на вирішенні логічних задач і математичних рівнянь, що демонструвало базову здатність машин імітувати мислення. Крім того, у цей час з’явилися перші системи на основі алгоритмів логічного виведення.

Розвиток ШІ на практичному рівні зіткнувся з численними труднощами, особливо через недостатню потужність обчислювальних машин того часу. Проте у 1970-х роках почався розвиток експертних систем, що стали першими практичними застосуваннями ШІ в реальних умовах. Експертні системи дозволяли автоматизувати роботу фахівців, аналізуючи бази знань у конкретних галузях, таких як медицина чи фінанси, і допомагали приймати рішення на основі заданих правил.

У 1980-х роках набувають популярності нейронні мережі, ідея яких була закладена ще в 1940-х роках. Спершу, через обмеження технічних можливостей, нейронні мережі були малоефективними, проте згодом, завдяки збільшенню обчислювальної потужності, їхні можливості значно розширилися. Нейронні мережі стали основою сучасних систем машинного навчання, що дало поштовх новій хвилі досліджень у галузі ШІ.

Розвиток штучного інтелекту у 1990-х роках зазнав значного піднесення завдяки збільшенню обчислювальних потужностей, покращенню алгоритмів і зростанню інтересу до застосувань ШІ у різних сферах. Саме у цей період відбувається становлення машинного навчання як окремого підрозділу ШІ, і відзначаються важливі досягнення, зокрема в обробці природної мови та комп'ютерному баченні. Машинне навчання, яке базується на здатності комп’ютерів навчатися на даних, без прямого програмування на кожне завдання, почало відігравати важливу роль у створенні інтелектуальних систем. Принципово новим стало використання алгоритмів, що дозволяють комп'ютеру адаптуватися до зміни умов і виконувати дедалі складніші завдання на основі набутого досвіду.

У 1997 році в історії ШІ відбувся один із найбільш значущих моментів: суперкомп’ютер Deep Blue, розроблений компанією IBM, переміг у матчі в шахи чемпіона світу Гаррі Каспарова. Це досягнення стало своєрідним символом можливостей ШІ, демонструючи, що інтелектуальні системи можуть перевершувати людський інтелект навіть у складних стратегічних іграх. Хоча

шахові алгоритми Deep Blue не були глибоко інтелектуальними, а скоріше використовували велику кількість обчислень і певні алгоритмічні евристики, ця подія стала поворотним моментом у сприйнятті штучного інтелекту в суспільстві.

Ще один важливий прорив відбувся на початку 2000-х років із появою алгоритмів для обробки великих обсягів даних. Розвиток інтернету призвів до накопичення значних масивів інформації, і це відкрило нові можливості для тренування ШІ-моделей. Науковці й інженери почали створювати системи, здатні розпізнавати закономірності в даних, класифікувати інформацію і навіть прогнозувати певні події. На цьому етапі ключову роль у розвитку ШІ починають відігравати корпорації, зокрема Google, Microsoft, IBM та Amazon, які інвестують значні ресурси в дослідження та комерційні застосування штучного інтелекту.

Серед ключових технологій, що змінили підхід до створення інтелектуальних систем, варто відзначити алгоритми глибокого навчання, які почали набувати популярності приблизно в середині 2000-х. Глибоке навчання базується на використанні багатошарових нейронних мереж і є частиною ширшого напряму машинного навчання. Ці мережі, натхненні біологічними нейронами, дозволяють моделювати складні процеси та розпізнавати навіть неочевидні закономірності в великих наборах даних. Важливим поштовхом для розвитку глибокого навчання стало поєднання з методами оптимізації та високопаралельних обчислень на графічних процесорах, що дозволило значно прискорити навчання моделей і розширити сфери їх застосування.

Паралельно з розвитком глибокого навчання, інтерес до нейронних мереж стимулював дослідження в напрямку комп'ютерного зору та обробки природної мови. Зокрема, великі прориви відбулися у розпізнаванні зображень, що дозволило застосовувати ШІ у таких сферах, як медична діагностика, контроль якості на виробництві та автономне керування транспортними засобами. Завдяки вдосконаленню алгоритмів обробки природної мови ШІ також почав використовуватися у створенні віртуальних помічників, чат-ботів, систем

автоматичного перекладу та інших технологій, що спростили взаємодію між людиною і машиною.

Ще одним етапом у розвитку ШІ стали досягнення у сфері підкріплювального навчання, де алгоритми навчаються шляхом взаємодії з середовищем і отримання винагороди за правильні дії. У 2016 році система AlphaGo, розроблена компанією DeepMind, перемогла чемпіона світу з гри в го Лі Седоля. Гра в го вважалася надзвичайно складною для комп'ютерного моделювання, адже вона потребує не лише обчислювальної потужності, але й стратегічного мислення. AlphaGo використовувала поєднання глибокого навчання та підкріплювального навчання, що дозволило їй навчатися на базі мільйонів партій у го і досягти рівня, що перевершував людські здібності. Цей момент ознаменував нову еру в історії штучного інтелекту, коли системи стали здатні досягати високих результатів у завданнях, що вимагають складного прогнозування й адаптивного навчання.

Розвиток ШІ у цей період також супроводжувався зростанням обізнаності про потенційні етичні та соціальні наслідки цієї технології. Учені та інженери почали активно обговорювати можливості контролю за розробкою штучного інтелекту, зокрема вплив автоматизації на ринок праці, питання приватності та безпеки, а також потенційні ризики використання ШІ у військових цілях. Наукове співтовариство та великі корпорації створюють рекомендації і принципи для забезпечення відповідального підходу до розробки інтелектуальних систем, спрямовані на те, щоб зробити ШІ корисним та безпечним для всього суспільства.

2.1.2. Огляд концепцій та підходів штучного інтелекту

У ШІ існують кілька основних концепцій, на яких базуються методи і підходи до його реалізації. Одна з ключових концепцій — це машинне навчання, яке передбачає навчання моделей на великих наборах даних для виконання завдань, таких як класифікація, регресія чи кластеризація. Машинне навчання можна

розділити на три основні підходи: контрольоване навчання, неконтрольоване навчання та навчання з підкріпленням. Контрольоване навчання використовує марковані набори даних, щоб навчити модель розпізнавати певні шаблони або здійснювати передбачення. Неконтрольоване навчання працює з немаркованими даними і дозволяє моделі знаходити схожі структури або кластери. Навчання з підкріпленням, навпаки, орієнтоване на процес, у якому агент взаємодіє з середовищем і отримує винагороду за правильні дії, що сприяє його навчальній адаптації.

Іншим важливим підходом є глибоке навчання, яке використовує багатошарові нейронні мережі для обробки великих масивів даних. Глибоке навчання здатне автоматично виділяти ознаки на основі навчальних даних, що робить його особливо корисним для обробки складних завдань, таких як обробка зображень, розпізнавання мовлення або виявлення аномалій у текстах. Однією з найбільш популярних архітектур глибокого навчання є згорткові нейронні мережі (CNN), які успішно застосовуються для аналізу зображень. Рекурентні нейронні мережі (RNN) використовуються для роботи з послідовними даними, такими як текст, ігри та час серійних сигналів, а трансформери — сучасна архітектура глибокого навчання — стали основою обробки природної мови, забезпечуючи високий рівень точності в таких задачах, як машинний переклад або відповіді на запити.

Серед інших підходів також виділяється концепція гібридних систем, що поєднують кілька методів ШІ для досягнення оптимального результату. Наприклад, гібридні системи можуть поєднувати нейронні мережі з класичними алгоритмами машинного навчання, такими як дерева рішень або методи опорних векторів, що дозволяє зменшити недоліки кожного окремого методу і підвищити загальну ефективність. Ще однією ключовою концепцією є пояснюваний ШІ, який зосереджений на створенні моделей, що забезпечують прозорість рішень. Це

особливо важливо в кібербезпеці, де важливо розуміти, чому система зробила певне передбачення або сигналізувала про загрозу.

Важливими концепціями ШІ є машинне навчання (ML), глибинне навчання (DL) і обробка природної мови (NLP), кожна з яких відповідає за специфічні підходи до аналізу великих обсягів інформації. Наприклад, алгоритми ML дозволяють створювати моделі, які можуть класифікувати чи прогнозувати на основі тренувальних даних, а глибинне навчання зосереджується на роботі з великими обсягами даних і складними архітектурами нейронних мереж.

У рамках машинного навчання є кілька підходів, включаючи навчання з учителем, без учителя та підкріплення. Навчання з учителем використовується тоді, коли доступні мітки даних, що дозволяє алгоритму навчатися на основі зразків. Цей підхід застосовується для класифікації текстів, зокрема для виявлення фішингових листів, де алгоритми навчаються розпізнавати певні патерни, властиві фішингу. Навчання без учителя є корисним, коли дані не мають міток, як у випадку з великими потоками мережевих даних, де мета – знаходити аномалії. Нарешті, методи підкріплення підходять для задач, де модель навчається через систему винагород і покарань, що важливо для реактивних кібербезпекових систем, які пристосовуються до нових загроз в режимі реального часу.

Глибинне навчання, підвид машинного навчання, зосереджується на створенні багатошарових нейронних мереж, що мають велику кількість параметрів і здатні самостійно видобувати найважливіші особливості даних. Зокрема, у боротьбі з фішингом DL дозволяє аналізувати велику кількість характеристик контенту, таких як структура повідомлень, використані ключові слова та навіть стилістичні елементи. Завдяки таким підходам, глибинне навчання здатне розпізнавати більш складні патерни, які важко виявити простими алгоритмами, і забезпечує більш високий рівень узагальнення, що допомагає виявляти нові види атак.

Одним із найперспективніших підходів до обробки тексту у ШІ є методи обробки природної мови (NLP), зокрема застосування трансформерів, таких як моделі BERT (Bidirectional Encoder Representations from Transformers) або GPT (Generative Pre-trained Transformer). Ці моделі відомі здатністю працювати з контекстом у тексті, що робить їх надзвичайно корисними для аналізу повідомлень у реальному часі, визначення фраз і шаблонів, характерних для фішингу. NLP дозволяє системам безпеки аналізувати ідентифікаційні слова та синтаксис у текстах фішингових листів, а також забезпечувати виявлення специфічних стилістичних елементів, характерних для соціальної інженерії.

Розглядаючи архітектури нейронних мереж, важливими є згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN). CNN зазвичай застосовуються для аналізу зображень, однак їх модифіковані версії можуть використовуватися для класифікації тексту, розпізнавання патернів у веб-сторінках, які мають фішингову природу. RNN, зокрема у варіаціях як LSTM (Long Short-Term Memory), дозволяють працювати із послідовностями даних, що є корисним для аналізу текстів, де важливим є порядок слів. LSTM зберігають контекст попередніх слів, що дозволяє моделі краще розуміти структуру і зміст повідомлень [21].

Інтеграція методів ШІ в системи виявлення фішингу також включає використання методів ансамблю, які об'єднують результати декількох моделей для покращення загальної точності. Наприклад, ансамбль може складатися з дерев рішень, логістичної регресії та нейронної мережі, які об'єднують свої передбачення для зниження ймовірності помилок. У фішингових системах ці методи дозволяють значно підвищити точність і стабільність, що є критичним при роботі з непередбачуваними загрозами та новими техніками атаки.

Одним з найбільш обговорюваних аспектів ШІ є адаптивність моделей та їх здатність до переносу навчання (transfer learning). Це дозволяє системам навчатися на основі певних типів атак і згодом адаптуватися до нових загроз без повного перенавчання. У випадку фішингових атак перенесення навчання може бути

особливо ефективним, оскільки різні кампанії можуть мати подібні патерни, які можна використовувати для швидкої адаптації системи до нових векторів атак.

2.1.3. Роль ШІ в сучасній кібербезпеці

Штучний інтелект відіграє важливу роль у сучасній кібербезпеці завдяки своїй здатності швидко та точно обробляти великі обсяги даних, аналізувати поведінкові патерни, виявляти аномалії та автоматизувати численні процеси. Використання ШІ виявляється особливо корисним у протидії динамічним та складним загрозам, з якими традиційні методи не завжди можуть справлятися.

Один з основних напрямків застосування ШІ в кібербезпеці — це виявлення аномалій, що полягає у виявленні відхилень від звичної поведінки системи або користувачів. Аномалії можуть свідчити про наявність потенційної загрози, як-от несанкціонованого доступу до системи або спроби фішингової атаки. Наприклад, якщо користувач раптом починає відправляти велику кількість електронних листів з підозрілими вкладеннями, система на базі ШІ може помітити цю аномалію та активувати механізм захисту. Для цього ШІ аналізує історичні дані та створює профіль "нормальної" поведінки для кожного користувача чи системи, а потім порівнює з ним поточну активність. Цей підхід є ефективним завдяки здатності машинного навчання до самонавчання та адаптації на основі нових даних [23].

Іншим важливим аспектом є автоматизація процесів аналізу загроз. Завдяки методам машинного навчання та глибинного навчання, ШІ може автоматично класифікувати різні типи загроз та попереджати користувачів або адміністраторів систем про потенційні небезпеки. Наприклад, у випадку фішингових атак системи можуть аналізувати заголовки електронних листів, перевіряти URL-адреси на наявність шкідливих посилань і навіть ідентифікувати спроби соціальної інженерії, які включають специфічні патерни поведінки у листуванні. Такі системи, як правило, можуть працювати в режимі реального часу, що значно підвищує рівень безпеки.

Прикладом застосування ШІ є системи аналізу поведінки користувачів (User Behavior Analytics, UBA), які спостерігають за діями користувачів і виявляють аномалії на основі попередньо встановлених моделей поведінки. Такі системи можуть сповіщати про підозрілу активність, як-от спроби входу в систему з незвичних місць або в нехарактерний для користувача час. UBA-системи також використовуються для виявлення загроз від інсайдерів, де поведінковий аналіз може виявити користувачів, які, ймовірно, становлять загрозу.

ШІ також застосовується для виявлення шкідливого програмного забезпечення. Антивірусні рішення на основі ШІ можуть аналізувати файли на предмет шкідливих ознак, і навіть якщо вони не знайдуть точної відповідності з уже відомим зразком шкідливого ПЗ, вони можуть розпізнати шкідливі елементи за схожими патернами. Наприклад, поведінковий аналіз програм дозволяє виявляти дії, характерні для шкідливого програмного забезпечення, такі як спроби несанкціонованого доступу до системних файлів або мережевих ресурсів.

ШІ також ефективно використовується у сфері захисту мереж, де він допомагає виявляти та блокувати шкідливу активність у мережевому трафіку. Використовуючи методи глибинного навчання, сучасні системи аналізу мереж можуть фільтрувати трафік, виявляючи потенційно небезпечні запити на основі історії та специфіки поведінки в мережі. Для цього системи аналізують велику кількість з'єднань і виявляють патерни, що свідчать про несанкціоновані доступи або спроби злому.

Окрім виявлення аномалій, штучний інтелект відіграє важливу роль у прогнозуванні кіберзагроз. Одним із методів прогнозування є використання предиктивної аналітики, коли алгоритми ШІ аналізують великі масиви історичних даних для виявлення закономірностей, які передували атакам у минулому. На основі цього аналізу система здатна передбачати, коли і де може відбутися наступна атака, а також які вектори та методи ймовірно будуть використані. Це дозволяє

організаціям вжити проактивних заходів для посилення захисту і підготуватися до потенційних загроз.

Ще одним прикладом використання ШІ в кібербезпеці є застосування обробки природної мови (NLP) для аналізу текстового контенту в повідомленнях, соціальних мережах та інших джерелах інформації. За допомогою NLP-систем можна автоматично визначати наявність підозрілих ключових слів або фраз, що часто використовуються у фішингових атаках чи шахрайських схемах. Алгоритми можуть аналізувати структуру та стиль повідомлень, відокремлюючи звичайні повідомлення від тих, що мають ознаки маніпуляцій або спроб соціальної інженерії. Такий аналіз є дуже ефективним, оскільки фішингові атаки часто базуються на психологічному впливі, і використання NLP допомагає автоматизовано відстежувати цей вплив у повідомленнях [25].

Крім того, завдяки технології машинного навчання можна побудувати складні нейронні мережі, здатні класифікувати кіберзагрози за різними категоріями, визначати їх рівень ризику та автоматично обирати найефективніші стратегії реагування. Наприклад, штучні нейронні мережі можуть застосовуватися для виявлення різних типів шкідливих програм, таких як віруси, трояни або програми-вимагачі, аналізуючи тисячі характеристик файлів або поведінкових патернів, що супроводжують їх дію. Це дозволяє швидко визначати потенційно небезпечні файли, навіть якщо раніше вони не зустрічалися в базах даних шкідливого ПЗ.

Ще одним значущим прикладом є адаптивні системи на базі ШІ, які використовуються для управління політиками доступу до інформаційних систем. Ці системи застосовують поведінковий аналіз для визначення нормальної поведінки кожного користувача в мережі. Якщо система виявляє, що поведінка користувача відрізняється від звичайної — наприклад, користувач спробує отримати доступ до даних, які він зазвичай не використовує, або відправляє великі обсяги інформації на зовнішні ресурси — адаптивна система може обмежити або

заблокувати доступ цього користувача, попереджаючи можливі порушення безпеки [26].

2.2. Вибір моделі штучного інтелекту для виявлення фішингових атак

2.2.1. Порівняння моделей ШІ

Порівняння різних моделей штучного інтелекту (ШІ) для виявлення фішингових атак показує, що кожна з них має унікальні характеристики, які можуть впливати на точність, ефективність та швидкість обробки. Однією з найпопулярніших моделей у цій сфері є нейронні мережі, зокрема глибинні нейронні мережі (Deep Neural Networks, DNN). Вони забезпечують можливість аналізувати складні, багатовимірні дані завдяки здатності автоматично виділяти важливі ознаки без втручання людини. Це досягається завдяки багатошаровій архітектурі, де кожен наступний шар навчається на виході попереднього, поступово відтворюючи більш детальні характеристики [27]. Перелік деяких з моделей зображено на рис. 2.1.

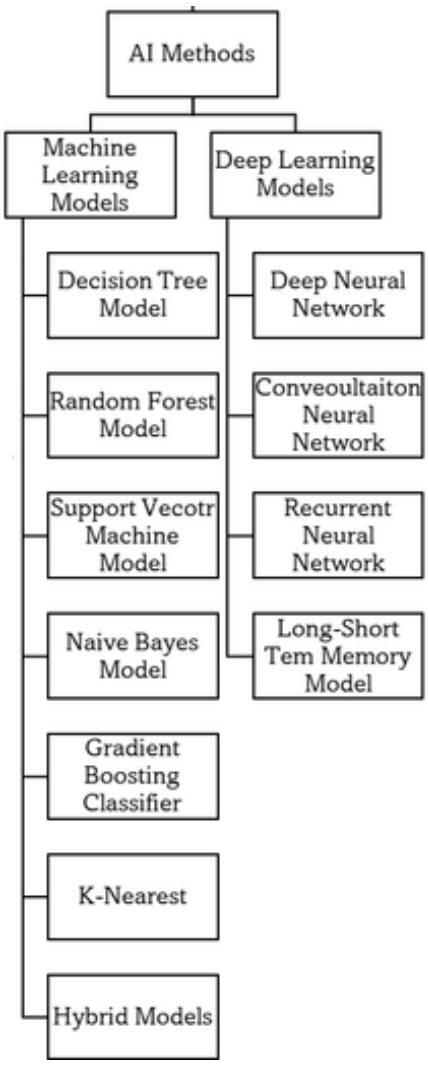

Рис. 2.1. Існуючі моделі машинного навчання та глибокого навчання

Наприклад, у контексті фішингових атак нейронні мережі можуть бути навчені розпізнавати різні патерни в тексті електронних листів, поведінкові особливості користувачів або структуру вебсторінок. Архітектура DNN дозволяє цим моделям обробляти великі обсяги даних, що робить їх особливо ефективними у виявленні складних фішингових схем, які можуть містити незвичні патерни або приховані елементи соціальної інженерії. Однак цей підхід має свої недоліки: нейронні мережі є ресурсомісткими, вимагають великої кількості даних для навчання і мають тенденцію до перевищення обчислювальних потужностей, особливо коли мова йде про обробку потоків даних у реальному часі.

З іншого боку, метод опорних векторів (Support Vector Machines, SVM) надає інший підхід до завдання виявлення фішингових атак. SVM є алгоритмом класифікації, який особливо добре підходить для задач з двома класами, таких як розрізнення фішингових та нефішингових повідомлень. SVM працює шляхом пошуку оптимальної гіперплощини, яка максимально розділяє дані на два класи. Однією з головних переваг цього методу є його здатність обробляти високовимірні простори і знаходити складні патерни в даних. Наприклад, у випадку виявлення фішингу SVM може аналізувати численні ознаки електронних листів або вебсторінок, такі як довжина тексту, використання специфічних слів або характеристик URL. При цьому метод є менш схильним до перенавчання, ніж нейронні мережі, і може давати точні результати навіть за наявності обмеженого набору навчальних даних. Однак для складніших і багатовимірних даних, таких як текстові та поведінкові патерни, ефективність SVM може знижуватись у порівнянні з DNN [29].

Методи ансамблю, такі як Random Forest або градієнтний бустинг (Gradient Boosting Machines, GBM), також відіграють важливу роль у виявленні фішингових атак. Ці моделі працюють шляхом комбінування декількох слабких моделей для створення однієї сильної. Наприклад, у випадку Random Forest алгоритм генерує безліч дерев рішень, кожне з яких робить свій прогноз, а остаточний результат визначається за принципом більшості голосів. Такий підхід дозволяє покращити стабільність і точність моделі та знижує ризик перенавчання. У випадку градієнтного бустингу, кожна наступна модель фокусується на виправленні помилок попередніх, що дозволяє досягти вищої точності, але за рахунок збільшення обчислювальної складності.

Методи ансамблю особливо корисні при роботі з гетерогенними даними, де можуть знадобитися різні алгоритми для аналізу різних аспектів фішингової атаки. Наприклад, вони можуть обробляти дані, які одночасно містять текстову інформацію, метадані і характеристики поведінки користувача, поєднуючи ці

різнорідні ознаки для створення цілісної моделі. Проте основним недоліком методів ансамблю є їх висока складність та велика кількість параметрів, що потребує значного часу на налаштування та обчислювальних ресурсів для тренування.

Ще однією з важливих переваг методів ансамблю є їх стійкість до варіацій у даних і зменшення ризику перенавчання. Наприклад, у випадку фішингових атак, дані часто можуть бути неповними або містити «шум» через наявність різних форматів повідомлень, змішаних мов чи варіантів вебсторінок. Вибір ансамблевих методів для обробки таких складних даних часто виправдовується їх здатністю підвищувати точність за рахунок поєднання результатів кількох моделей. Методи на кшталт Random Forest також мають вбудовану можливість оцінки важливості ознак, що дає змогу ідентифікувати, які атрибути найбільш впливають на класифікацію. Це може бути корисним у випадках, коли важливо знати, які саме характеристики (довжина URL, наявність специфічних ключових слів тощо) найбільше впливають на рішення системи [31].

Серед інших алгоритмів, які активно використовуються у виявленні фішингових атак, можна виділити логістичну регресію (Logistic Regression), особливо для задач, де потрібна інтерпретованість. Логістична регресія є лінійним класифікатором, що передбачає ймовірність належності до одного з класів. Незважаючи на те, що цей алгоритм є відносно простим у порівнянні з нейронними мережами чи методами ансамблю, він добре працює на структурованих наборах даних, які містять лінійні залежності між ознаками. У випадку фішингових атак логістична регресія дозволяє швидко обчислити ймовірність того, що повідомлення або вебсторінка є фішинговими на основі визначених критеріїв. Завдяки цьому цей метод є корисним для початкової фільтрації та визначення ймовірності ризику, що робить його ефективним доповненням до складніших моделей у багатоступеневих системах виявлення загроз.

На додаток, методи на основі наївного Баєсова класифікатора (Naive Bayes Classifier) також активно застосовуються в задачах фільтрації спаму і фішингу.

Баєсовий підхід особливо підходить для текстового аналізу, зокрема, коли йдеться про ключові слова або загальну частотність певних слів у повідомленні. Цей алгоритм будує свою класифікацію на основі ймовірності кожної ознаки, вважаючи їх незалежними, що спрощує процес обчислення і дозволяє швидко обробляти вхідні дані. Баєсові моделі часто використовують у комбінації з іншими алгоритмами, зокрема з методами ансамблю або SVM, що дозволяє підвищити їх ефективність і точність. Проте через припущення незалежності між ознаками, наївний Баєсовий підхід може бути менш точним у випадках складних залежностей, характерних для багатоаспектних фішингових атак [32].

Ще один цікавий напрям — використання рекурентних нейронних мереж (RNN), особливо в контексті аналізу поведінкових даних та тексту. RNN добре підходять для обробки послідовних даних, зокрема, історії відвідувань вебсторінок або динаміки поведінки користувача. Наприклад, рекурентні мережі можуть відслідковувати зміни в патернах поведінки користувача і сигналізувати про аномальні дії, характерні для фішингових атак. Зокрема, якщо користувач без прецеденту починає відвідувати сумнівні вебсторінки або вводить свої дані на незнайомих сайтах, модель може ідентифікувати таку поведінку як ризиковану. Проте, як і більшість нейронних мереж, RNN потребують великого обсягу даних для навчання і належної обробки, щоб знизити ймовірність хибнопозитивних результатів [34].

2.2.2. Порівняння нейронних мереж для виявлення фішингових атак

Нейронні мережі є одним із найбільш потужних інструментів у сфері виявлення фішингових атак завдяки їхній здатності автоматично навчатися з даних та виявляти складні патерни, які традиційні алгоритми не можуть виявити. Зокрема, вони особливо корисні в ситуаціях, коли фішингові атаки постійно змінюються та

адаптуються до нових методів захисту. Нейронні мережі здатні аналізувати великі обсяги даних, включаючи текст повідомлень, структури вебсторінок, графічні елементи та поведінкові патерни користувачів, що робить їх надзвичайно ефективними для задач кібербезпеки.

Однією з найбільш поширених архітектур, що використовується у виявленні фішингових атак, є глибокі нейронні мережі (DNN). Вони складаються з багатьох шарів штучних нейронів, які дозволяють їм виявляти складні нелінійні залежності між входами та виходами. У контексті фішингових атак такі мережі можуть аналізувати багатовимірні ознаки, такі як структуру URL-адреси, частоту появи певних ключових слів у тексті та навіть взаємодію між різними компонентами вебсторінки. Наприклад, DNN може виявити, що певне поєднання фраз і дизайну сторінки часто зустрічається на шахрайських сайтах, і сигналізувати про потенційну загрозу.

Конволюційні нейронні мережі (CNN) також знайшли широке застосування у виявленні фішингу, особливо при аналізі графічних і візуальних даних. Візуальна схожість є поширеним методом, який використовують зловмисники для введення користувачів в оману. CNN можуть ефективно порівнювати логотипи, дизайн вебсторінок та інші графічні елементи фішингових сайтів з легітимними. Наприклад, CNN може навчитися розпізнавати тонкі відмінності у логотипах компаній, які шахраї змінюють для створення фальшивих вебсторінок, таких як змінений колір або форма символів. Ця технологія також корисна для аналізу зображень капчі чи іконок, які використовуються на сайтах для введення в оману [35].

Рекурентні нейронні мережі (RNN), зокрема їхні варіанти на кшталт LSTM (Long Short-Term Memory), використовуються для аналізу послідовних даних, таких як текст повідомлень чи історії взаємодій користувачів із сайтом. RNN здатні враховувати контекст і послідовність подій, що є особливо важливим при виявленні фішингових атак через електронну пошту або месенджери. Наприклад, RNN може

ідентифікувати підозрілі шаблони в листуванні, такі як раптова поява термінових запитів на оплату або повідомлення з граматичними помилками, характерними для шахраїв. Крім того, аналіз поведінки користувача на вебсторінках може допомогти RNN визначити, чи дії є типовими для цього користувача, чи можуть свідчити про скомпрометований обліковий запис.

Генеративно-змагальні мережі (GAN) також мають перспективи у виявленні фішингових атак. GAN складаються з двох нейронних мереж: генератора, який створює нові зразки, і дискримінатора, який намагається відрізнити справжні зразки від фальшивих. У контексті фішингу GAN можуть бути використані для створення синтетичних фішингових прикладів, що дозволяє покращити навчання інших моделей, а також для аналізу методів, які зловмисники можуть використовувати у майбутньому. Це забезпечує можливість тестування систем виявлення на нових типах атак, які ще не з'явилися в реальних умовах [36].

Комбінування нейронних мереж з іншими методами машинного навчання також дає позитивні результати. Наприклад, гібридні моделі, що використовують CNN для аналізу зображень і RNN для тексту, дозволяють одночасно обробляти графічні та текстові дані з високою точністю. Це корисно у випадках, коли шахраї використовують комбінацію тексту та зображень у своїх атаках. Застосування таких моделей дозволяє системам виявлення фішингу бути більш універсальними і реагувати на ширший спектр загроз.

Одним із ключових факторів успішності нейронних мереж у виявленні фішингових атак є їх здатність до витягання прихованих характеристик даних, які важко визначити вручну. Ця властивість дозволяє моделі адаптуватися до нових варіантів фішингових атак, навіть якщо вони мають лише часткову схожість із попередніми прикладами. Наприклад, у текстах фішингових повідомлень часто використовуються нетипові словосполучення або стилі мовлення, які складно ідентифікувати традиційними алгоритмами. Нейронні мережі можуть автоматично навчитися таким особливостям, підвищуючи точність класифікації.

Додатковим кроком у покращенні ефективності є попереднє навчання нейронних мереж на великих наборах даних загального характеру з подальшим донавчанням на спеціалізованих наборах. Наприклад, використання моделей, попередньо натренованих на великих текстових корпусах, таких як BERT або GPT, дозволяє ефективно аналізувати фішингові повідомлення з урахуванням контексту. Ці моделі можуть бути додатково налаштовані на специфічні задачі, наприклад, розпізнавання фішингових електронних листів або аналіз повідомлень у соціальних мережах [38].

Ще одним важливим напрямом є поєднання нейронних мереж із методами обробки природної мови (NLP). За допомогою NLP можна витягати такі характеристики тексту, як синтаксична структура, семантичний зміст або частота використання певних слів. Нейронні мережі використовують ці характеристики для створення глибоких моделей класифікації, здатних розпізнавати навіть добре замасковані фішингові тексти. Наприклад, NLP-інструменти можуть аналізувати стиль написання тексту та виявляти невідповідності, які характерні для автоматично згенерованих повідомлень або текстів від осіб, що імітують офіційні організації.

У випадках, коли фішингові атаки включають аналіз складних структур, таких як багаторівневі вебсайти або багатокомпонентні шаблони електронних листів, застосування моделей на основі багатомодальних нейронних мереж стає надзвичайно корисним. Ці моделі об'єднують аналіз тексту, зображень і інших даних у єдину архітектуру. Наприклад, багатомодальна мережа може одночасно аналізувати зміст тексту на вебсторінці, візуальні елементи (логотипи, зображення) і структуру URL-адреси. Такий підхід дозволяє з високою точністю ідентифікувати складні фішингові сайти, які використовують різноманітні методи маскування.

Інтеграція нейронних мереж із системами реального часу є ще одним перспективним напрямом. Системи, які працюють на основі потокових даних, можуть використовувати рекурентні нейронні мережі (RNN) або трансформери для

безперервного аналізу інформації. Наприклад, трансформерна архітектура, як-от BERT або GPT, може бути застосована для моніторингу активності в реальному часі, автоматично аналізуючи вхідні повідомлення електронної пошти або поведінку користувачів у мережі. Це забезпечує оперативне виявлення загроз і зменшує час на реагування [40].

Удосконалення алгоритмів навчання також сприяє підвищенню ефективності нейронних мереж у кібербезпеці. Наприклад, техніки навчання з підкріпленням (reinforcement learning) дозволяють моделям адаптуватися до нових сценаріїв на основі зворотного зв’язку. Такі моделі можуть використовувати інформацію про успішність або невдачу попередніх класифікацій для вдосконалення своїх стратегій. Це особливо корисно в контексті фішингових атак, які часто швидко еволюціонують.

Важливу роль також відіграє використання технологій розподіленого навчання, таких як федеративне навчання. У цьому підході моделі нейронних мереж можуть навчатися на розподілених наборах даних без необхідності їх централізованого збору. Це забезпечує конфіденційність даних і знижує ризики витоку інформації, що є критично важливим для організацій, які обробляють чутливі дані клієнтів.

Попри численні переваги нейронних мереж, їх застосування у виявленні фішингових атак має певні виклики. Наприклад, навчання глибоких нейронних мереж вимагає значних обчислювальних ресурсів і великих обсягів якісних даних. Для досягнення високої точності модель повинна бути навчена на репрезентативному наборі фішингових і нефішингових прикладів, які враховують різноманіття атак. Це створює труднощі в збиранні, обробці та маркуванні даних, а також в оновленні моделей для врахування нових методів шахраїв.

Одним із викликів є стійкість нейронних мереж до атак на саму модель, таких як атакуючі приклади (adversarial examples). Шахраї можуть використовувати спеціально створені дані, які вводять модель в оману, змушуючи її класифікувати

фішинговий сайт як легітимний. Для вирішення цієї проблеми використовуються методи підвищення стійкості моделей, такі як навчання з включенням атакуючих прикладів (adversarial training) або регуляризація [41].

Іншим викликом є пояснюваність нейронних мереж. Багато моделей працюють як «чорні ящики», що ускладнює розуміння, які саме ознаки вони використовують для прийняття рішень. Це може створювати проблеми для кібербезпеки, оскільки неможливо завжди пояснити, чому конкретне повідомлення або сайт були позначені як фішингові. Для вирішення цього виклику активно розробляються техніки пояснення рішень моделей, такі як методи інтегрованих градієнтів і локальних пояснень (LIME) [43].

2.2.3. Критерії вибору моделі

Ефективність моделі є головним критерієм при виборі штучного інтелекту для задач виявлення фішингових атак. Вона визначає, наскільки точно модель здатна виконувати класифікацію та розрізняти фішингові атаки від легітимних дій. Ефективність вимірюється за допомогою низки метрик, кожна з яких важлива для специфічних сценаріїв. Ці метрики включають точність класифікації (accuracy), повноту (recall), прецизійність (precision), значення F1-score і Area Under Curve (AUC).

Метрики оцінки ефективності:

1. Точність (Accuracy):

   Точність показує, яку частку з усіх передбачень модель класифікувала правильно. Хоча це популярна метрика, вона може бути оманливою, якщо дані є незбалансованими. Наприклад, якщо більшість електронних листів у датасеті є легітимними, модель може просто класифікувати всі листи як легітимні та досягти високої точності, ігноруючи фішингові атаки.

2. Прецизійність (Precision):

   Прецизійність відображає частку реальних фішингових атак серед усіх, що були класифіковані як фішингові. Ця метрика важлива для зменшення кількості хибнопозитивних результатів, що можуть відволікати користувачів і адміністраторів безпеки. Висока прецизійність особливо корисна в системах, де кожна помилка має значні наслідки, наприклад, у корпоративних мережах.

3. Повнота (Recall):

   Повнота, або чутливість, демонструє здатність моделі виявляти всі реальні фішингові атаки. Високий рівень recall є критично важливим у контексті фішингу, адже навіть одна пропущена атака може призвести до серйозних наслідків, таких як викрадення облікових даних чи компрометація системи.

4. F1-score:

   Ця метрика є середнім значенням прецизійності та повноти, що дозволяє знайти баланс між хибнопозитивними та хибнонегативними результатами. Для задач фішингових атак F1-score забезпечує комплексну оцінку ефективності, оскільки одночасно враховує обидві ключові метрики.

5. AUC (Area Under Curve):

   AUC відображає здатність моделі правильно розрізняти класи фішингових та легітимних елементів на різних рівнях порогових значень. Високе значення AUC свідчить про універсальність моделі, що є важливим для змінюваних умов фішингових атак.

Здатність до узагальнення

Ефективність моделі також залежить від її здатності до узагальнення. У фішингових атаках часто використовуються нові техніки, які раніше не зустрічалися в тренувальних даних. Моделі, такі як глибокі нейронні мережі, демонструють високий рівень узагальнення завдяки використанню великих обсягів даних і складних архітектур. Наприклад, Convolutional Neural Networks (CNN)

можуть аналізувати структуру вебсторінок, тоді як Recurrent Neural Networks (RNN) та їх розширення, такі як LSTM, ефективні для аналізу текстів електронної пошти.

Адаптація до динамічних загроз

Ефективність моделі також залежить від її здатності оновлюватися у відповідь на нові загрози. Моделі, які використовують методи переносного навчання (transfer learning), дозволяють швидко адаптуватися до нових сценаріїв, переносячи знання з інших доменів. Наприклад, модель, навчена на даних фінансового фішингу, може бути переналаштована для роботи з фішингом у соціальних мережах.

Приклади ефективних моделей:

- Логістична регресія:

  Хоча ця модель проста, вона забезпечує базовий рівень ефективності для задач, де важливо швидко аналізувати невеликі набори характеристик, такі як частота використання ключових слів.
- Дерева рішень та методи ансамблю (Random Forest, Gradient Boosting):

  Ці моделі ефективні для аналізу табличних даних, таких як метрики поведінки користувачів або структури URL-адрес.
- Глибокі нейронні мережі (Deep Neural Networks):

  Завдяки своїй складності ці моделі можуть аналізувати широкий спектр характеристик, від текстів до зображень, забезпечуючи високий рівень ефективності навіть для складних атак.

2.2.4. Оцінка точності та ефективності обраної моделі

Час тренування моделі

Тренування моделі включає навчання на великому обсязі даних. Час тренування залежить від таких параметрів, як розмір і складність даних, архітектура моделі та оптимізаційні алгоритми. Прості алгоритми, такі як логістична регресія

чи дерева рішень, мають короткий час тренування, що дозволяє швидко створити базову модель. Однак у складніших методах, таких як глибокі нейронні мережі (Deep Neural Networks, DNN), час тренування значно більший через багатошарову структуру та велику кількість параметрів, які потребують оптимізації.

Для задач, що потребують частого оновлення моделей, наприклад, через появу нових типів фішингових атак, моделі з коротшим часом тренування мають перевагу. Наприклад, методи ансамблю, такі як Random Forest, тренуються довше, ніж прості дерева рішень, але вони можуть бути оптимізовані за допомогою паралельних обчислень. У випадку DNN, використання спеціалізованого апаратного забезпечення, такого як графічні процесори (GPU) або тензорні процесори (TPU), суттєво знижує час тренування.

Час прогнозування

Час прогнозування (inference time) — це час, необхідний для аналізу одного зразка та видачі результату. Цей параметр є ключовим у системах реального часу, таких як фільтри електронної пошти або веб-проксі-сервери, де обробка повинна виконуватися майже миттєво. Наприклад:

- Логістична регресія і дерева рішень мають дуже короткий час прогнозування завдяки простій структурі обчислень.
- Методи ансамблю, такі як Gradient Boosting Machines (GBM), забезпечують високу точність, але кожне прогнозування вимагає обчислення для кількох дерев, що збільшує час обробки.
- Глибокі нейронні мережі зазвичай мають довший час прогнозування, особливо для моделей з великою кількістю шарів, якщо не використовуються оптимізовані процесори.

Пропускна здатність

Пропускна здатність (throughput) визначає кількість зразків, які модель може обробити за певний час. Цей показник особливо важливий для корпоративних рішень, де аналізу підлягають тисячі повідомлень або вебзапитів одночасно. Моделі

з високою пропускною здатністю мають перевагу, дозволяючи масштабувати системи без зниження продуктивності.

- Дерева рішень і логістична регресія демонструють високу пропускну здатність через лінійну природу обчислень.
- Алгоритми ансамблю мають нижчу пропускну здатність через потребу в обчисленнях для кожного компонента ансамблю.
- Глибокі нейронні мережі, хоча й споживають більше ресурсів, можуть бути оптимізовані для високої пропускної здатності за допомогою розподілених обчислень або використання прискорювачів, таких як GPU.

Фактори, що впливають на швидкість

1. Об'єм даних:

   Моделі, що аналізують великі обсяги даних, потребують ефективної обробки для забезпечення прийнятної швидкості. Для цього застосовуються методи попередньої обробки, такі як скорочення розмірності.
2. Тип алгоритму:

   Наприклад, нейронні мережі з рекурентними архітектурами (RNN) можуть бути повільнішими, ніж CNN, для завдань, де важлива послідовність даних.
3. Інфраструктура:

   Використання апаратного прискорення, таких як GPU або FPGA, може значно підвищити швидкість обчислень. Хмарні сервіси, такі як AWS Sagemaker або Google Cloud AI, пропонують готові рішення для масштабування.
4. Оптимізація:

   Техніки компресії моделі (model pruning), квантилізація та зменшення складності обчислень дозволяють збільшити швидкість без значних втрат у точності.

Приклади використання

1. Електронна пошта:

   Для обробки великих обсягів вхідних листів у реальному часі найкраще підходять легкі моделі, такі як дерева рішень або SVM, що забезпечують швидкий аналіз із мінімальними затримками.

2. Блокування фішингових вебсайтів:

   У цій сфері використовуються гібридні підходи. Легкі моделі можуть виконувати первинну фільтрацію, а складніші, такі як DNN, — детальний аналіз.

3. Мобільні пристрої:

   Для додатків із обмеженими обчислювальними ресурсами, таких як антивірусні програми для смартфонів, оптимізовані моделі, що мають швидкий час прогнозування, є незамінними.

Для оцінки складності враховують такі аспекти, як кількість параметрів, потреби в обчислювальних ресурсах, алгоритмічна гнучкість і доступність для інтеграції. Вибір моделі має забезпечувати баланс між продуктивністю та витратами на її реалізацію.

Кількість параметрів і розмір моделі

Моделі з великою кількістю параметрів, наприклад, глибокі нейронні мережі (DNN), зазвичай мають вищу складність. Кожен додатковий параметр збільшує обсяг обчислень, пам'яті та часу, необхідних для тренування й роботи моделі. З іншого боку, прості моделі, такі як логістична регресія чи дерева рішень, мають менше параметрів, що робить їх легшими у використанні.

Наприклад:

- Логістична регресія містить лінійну кількість параметрів, пропорційну кількості ознак, що робить її компактною та придатною для задач із обмеженими ресурсами.

- Глибокі нейронні мережі, такі як BERT або GPT, мають сотні мільйонів параметрів, що забезпечує високу продуктивність, але вимагає значних ресурсів.

У контексті фішингових атак, прості моделі можуть бути застосовані для початкового виявлення, тоді як складніші моделі використовуються для поглибленого аналізу, коли знайдено підозрілі патерни.

Обчислювальні ресурси

Складні моделі потребують значних обчислювальних потужностей для тренування та прогнозування. Наприклад:

- Легкі моделі, такі як SVM із лінійним ядром або дерева рішень, можуть працювати навіть на звичайному офісному комп'ютері, оскільки їх обчислювальні витрати низькі.
- Глибокі нейронні мережі вимагають GPU або кластерів серверів для обробки великих обсягів даних і складних архітектур.

Для систем із обмеженим бюджетом або тих, що працюють у реальному часі, пріоритетними є моделі, які можуть забезпечити високу швидкість і точність за мінімального споживання ресурсів.

Гнучкість і адаптивність

Складність також визначає, наскільки легко модель можна адаптувати до змінних умов. Сучасні фішингові атаки швидко еволюціонують, тому моделі мають бути гнучкими:

- Прості моделі, як-от логістична регресія, легко адаптуються до нових даних, але можуть не враховувати складних взаємозв'язків.
- Комплексні моделі, наприклад, методи ансамблю або DNN, здатні враховувати більш складні патерни, але їх адаптація потребує значного часу та ресурсів.

Для організацій, що стикаються зі швидкими змінами у методах атак, важливо враховувати цей компроміс між гнучкістю та складністю.

Інтеграція та підтримка

Складність моделі впливає на її інтеграцію з іншими системами та подальшу підтримку. Простота інтеграції залежить від доступності інструментів і бібліотек, які підтримують обрану модель:

- Прості моделі, наприклад, SVM або дерева рішень, мають багато реалізацій у популярних бібліотеках, таких як Scikit-learn, що спрощує їх інтеграцію.
- Глибокі моделі часто потребують спеціалізованих фреймворків, як-от TensorFlow або PyTorch, а їх налаштування вимагає глибоких технічних знань.

Крім того, складні моделі мають вищі витрати на підтримку через необхідність регулярного оновлення, оптимізації та забезпечення їх безперебійної роботи.

2.2.5. Оцінка точності та ефективності обраної моделі

Оцінка точності та ефективності моделей для виявлення фішингових атак є критичним етапом у процесі їх розробки та впровадження. Вона включає використання різних кількісних метрик, експериментальних підходів і верифікаційних методологій для забезпечення максимального рівня захисту від фішингових загроз. Основним завданням є вимірювання, наскільки добре модель справляється з розпізнаванням фішингових повідомлень у різних умовах, а також визначення її слабких місць.

Метрики точності та ефективності

Однією з основних метрик є точність (accuracy), яка визначає відсоток правильно класифікованих прикладів (як фішингових, так і легітимних) серед усіх протестованих. Проте точність може бути оманливою в умовах дисбалансу даних, коли кількість легітимних повідомлень значно перевищує кількість фішингових. У таких випадках більш важливими стають точність позитивного передбачення (precision) та повнота (recall).

- Precision показує, наскільки модель уникає помилкових спрацювань, тобто, яка частка повідомлень, класифікованих як фішингові, справді є такими.
- Recall вимірює здатність моделі виявляти всі фішингові повідомлення в наборі даних.

Для інтеграції цих двох показників використовується F1-міра, яка є гармонійним середнім між precision та recall. Вона особливо корисна, коли важливо досягти балансу між помилковими спрацюваннями та пропущеними загрозами.

Ще одна критично важлива метрика — False Positive Rate (FPR), яка визначає частку легітимних повідомлень, помилково класифікованих як фішингові. Низький FPR є ключовим для забезпечення зручності користувачів, оскільки високий рівень помилкових спрацювань може знизити довіру до системи.

Роль ROC-кривих та AUC

Для візуалізації ефективності класифікатора широко використовуються ROC-криві (Receiver Operating Characteristic), які показують залежність між чутливістю (sensitivity) та специфічністю (specificity) при різних порогах класифікації. Площа під кривою (AUC, Area Under Curve) є числовим вираженням загальної продуктивності моделі. Чим ближчий цей показник до 1, тим краща здатність моделі розрізняти класи між собою [44].

Експериментальна перевірка

Оцінка моделі здійснюється на основі розділення доступного набору даних на навчальну, валідаційну та тестову вибірки. Це дозволяє уникнути проблеми перенавчання (overfitting) і забезпечити об'єктивність оцінки. Одним із поширених підходів є перехресна перевірка (k-fold cross-validation), яка дозволяє оцінити стабільність моделі шляхом багаторазового тренування та тестування на різних підмножинах даних.

Для моделювання реальних умов використовується тестування на незнайомих даних, тобто таких, що не були представлені під час навчання моделі.

Це дозволяє оцінити здатність алгоритму узагальнювати знання і правильно працювати зі змішаними або новими наборами фішингових атак.

Порівняння ефективності

Для комплексної оцінки моделі її результати часто порівнюються з іншими алгоритмами або базовими підходами. Наприклад, модель на основі нейронних мереж може бути порівняна з моделями SVM, деревами рішень чи методами ансамблю за такими параметрами:

- Точність у виявленні фішингових атак.
- Час обробки одного повідомлення.
- Чутливість до змін у даних.

Такі порівняння дозволяють визначити найбільш підходящий підхід для певного сценарію використання.

Використання симуляційних платформ

Для тестування моделі можуть застосовуватися спеціалізовані симуляційні платформи, які імітують реальні атаки в контрольованому середовищі. Це дає змогу оцінити ефективність в умовах, максимально наближених до реальних. Наприклад, модель може бути перевірена на здатність розпізнавати фішингові повідомлення, що надходять через різні канали (електронна пошта, соціальні мережі, SMS).

Оцінка масштабованості

Крім точності, важливо оцінити, наскільки модель здатна працювати з великими обсягами даних у режимі реального часу. Це включає перевірку її продуктивності в умовах високого навантаження на сервери або при обробці потоку даних від тисяч користувачів одночасно. Ефективна модель повинна забезпечувати високу швидкість класифікації без втрати якості.

2.3. Підготовка та збір даних для навчання моделей

2.3.1. Джерела даних для навчання моделей

Для навчання моделей штучного інтелекту, які здатні ефективно виявляти фішингові атаки, використовуються різноманітні джерела даних. Це включає відкриті бази даних, API для інтеграції зі сторонніми сервісами, а також спеціалізовані сервіси, що надають актуальну інформацію про загрози. Такий широкий спектр джерел дозволяє отримати необхідні дані для тренування та тестування моделей [45].

Таблиця 1.

Перелік найпоширеніших баз даних для тренування фішингових моделей

Dataset Name	Update Year	Type	URL
Phishing-Tank	2024	Phisher	https://phishtank.org/developer_info.php
Alexa-dataset	2022	Legit	https://www.similarweb.com/website/alexa.com/
Wein-dataset	2021	Phisher	https://jeowein.net/
Crawl-dataset	2021	Legit	https://commoncrawl.org/
Open-Phish-dataset	2021	Phisher	https://openphish.com/
Phishing-army dataset	2021	Phisher	https://www.phishing.army/
Kaggle-phishing dataset	2021	Legit	https://www.kaggle.com/datasets/shashwatwork/phishing-/dataset-for-machine-learning
UCI-Dataset	2022	Phisher	https://data.world/uci/phishing-websites
Parsed-dataset	2022	Legit	https://doi.org/10.7910/dvn/omv
Yahoo-Phishing	2022	Legit	https://webscope.sandbox.yahoo.com/
Yandex-phishing	2022	Legit	https://Yandex.com/dev/xml/
Phished-dataset	2022	Phisher	https://www.medien.ifi.lmu.de/team

Відкриті бази даних фішингових сайтів

Одним із найважливіших джерел даних є відкриті бази, які підтримуються спільнотою дослідників кібербезпеки. Вони забезпечують доступ до тисяч зразків фішингових URL, доменів та інших метаданих.

- PhishTank: Це платформа, де користувачі можуть додавати URL-адреси, підозрювані у фішингу, після чого ці дані перевіряються спільнотою. PhishTank також надає API для автоматизованого доступу до їхньої бази, що спрощує інтеграцію із системами виявлення фішингу. База регулярно оновлюється, що дозволяє отримувати актуальні дані.
- OpenPhish: Ще одна популярна база даних, яка автоматично збирає фішингові URL, використовуючи сканери та аналітичні інструменти. OpenPhish має два рівні доступу: безкоштовний і платний. Безкоштовний рівень включає основні списки фішингових URL, тоді як платний забезпечує більше контексту та інформації про атаки.
- URLhaus: Ця база даних створена для ідентифікації шкідливих доменів та URL, зокрема фішингових сайтів. Вона дозволяє легко завантажувати списки URL, що підозрюються в зловмисній діяльності, та інтегрувати їх у програми для аналізу.

Ці бази забезпечують зручний спосіб отримання великої кількості мітокованих даних, необхідних для навчання моделей машинного навчання. Однак обмеження таких баз можуть включати нерівномірний розподіл зразків та можливу застарілість окремих записів.

API для доступу до даних

Інтеграція з API дозволяє отримувати дані в реальному часі, що є важливим для боротьби з динамічно змінюваними фішинговими атаками.

- VirusTotal API: Цей сервіс дозволяє отримувати детальну інформацію про URL, включаючи звіти від численних антивірусних рішень та сервісів із

перевірки репутації. Він використовується для збору даних про підозрілі сайти, які можуть бути фішинговими.

- Google Safe Browsing API: Google надає API для перевірки репутації URL. Цей інструмент може бути використаний для отримання даних про фішингові ресурси, а також для миттєвої перевірки URL у реальному часі.
- IPVoid та DNSlytics: Ці сервіси дозволяють отримувати дані про IP-адреси та домени, включаючи їхню репутацію. Використання таких API допомагає розширити контекст для аналізу фішингових загроз.

Завдяки цим API можна інтегрувати дані безпосередньо у процес тренування моделей, забезпечуючи їхню актуальність та релевантність.

Спеціалізовані сервіси

Окрім відкритих баз даних, існують комерційні сервіси, які надають високоякісні дані для навчання моделей.

- IBM X-Force Exchange: Ця платформа надає інформацію про загрози, включаючи фішингові атаки. Її можна використовувати для отримання розширених даних про домени, URL та IP-адреси, пов'язані з фішингом.
- Recorded Future: Цей сервіс використовує ШІ для аналізу загроз та надає деталізовану інформацію про фішингові кампанії. Recorded Future інтегрується з іншими інструментами кібербезпеки, що дозволяє оптимізувати аналіз даних.
- Threat Intelligence Platform: Платформа пропонує API для доступу до даних про шкідливі сайти, які можна використовувати для побудови та тестування моделей.

2.3.2. Передобробка даних

Передобробка даних є ключовим кроком у створенні моделей машинного навчання для виявлення фішингових атак. Цей процес включає очищення даних від

шуму, нормалізацію значень і трансформацію в форму, що максимально відповідає вимогам алгоритмів.

Очищення даних починається з видалення пропущених значень, що можуть порушувати узгодженість вибірки. Для цього застосовуються методи заповнення, як-от підставлення середніх значень, медіани або моди для числових і категоріальних змінних. У випадках, коли дані мають складну структуру, можуть бути використані алгоритми машинного навчання, наприклад, регресія або KNN, які здатні точно передбачити відсутні елементи на основі інших параметрів.

Далі проводиться видалення зайвого шуму, такого як аномалії або дубльовані записи. Аномалії виявляються через аналіз міжквартильних відхилень або застосування статистичних підходів для оцінки віддаленості точок. Для розпізнавання дубльованих даних використовуються інструменти хешування чи алгоритми порівняння тексту, як-от Levenshtein distance.

Нормалізація забезпечує приведення числових значень у вибірці до спільного масштабу, що допомагає уникнути домінування параметрів із великим діапазоном значень над менш значущими. Стандартизація за допомогою Z-score або масштабування в діапазон [0,1] є популярними методами, які використовуються для цього. Наприклад, кількість кліків по посиланню може бути зведена до відносного масштабу для забезпечення рівного впливу з такими параметрами, як довжина URL.

Наступним кроком є кодування категоріальних змінних. Для цього застосовуються такі техніки:

- One-hot encoding - перетворює категоріальні значення у двійкові вектори, створюючи окремий біт для кожної категорії. Наприклад, атрибути типу "легітимний", "фішинговий" переводяться у двійковий формат.
- Label encoding - присвоює унікальне числове значення кожній категорії. Цей підхід зручний для змінних із природним порядком значень, як-от рівень загрози.

Для текстових даних, які часто зустрічаються у фішингових URL, необхідно застосувати токенізацію. Цей процес розділяє текст на частини — токени (слова, фрази або символи), які далі аналізуються моделлю. Після токенізації текст представляється у векторній формі за допомогою технік, таких як TF-IDF або вбудовування слів (Word2Vec, GloVe) [46].

Крім цього, застосовується фільтрація небажаних символів, як-от пунктуація або стоп-слова. Ці елементи не додають цінності для аналізу, але збільшують обчислювальні витрати.

Також, передобробка включає розділення даних на навчальні, валідаційні та тестові набори. Це дозволяє оцінити модель на кожному етапі навчання і забезпечити її узагальнюваність на реальних даних. Розподіл зазвичай здійснюється у пропорції 70/20/10 або 80/10/10 залежно від обсягу наявних даних.

Для підготовки даних для подальшого навчання важливо зусереджуватися на зменшенні розмірності вибірки та виборі найбільш значущих характеристик. Один із способів досягти цього — застосування методів відбору характеристик, таких як аналіз головних компонент (PCA) або Recursive Feature Elimination (RFE). Ці підходи дозволяють зберегти лише ті параметри, які найбільш впливають на результат, виключаючи другорядні та нерелевантні дані. Наприклад, для аналізу URL-адрес можна зосередитися на довжині адреси, кількості параметрів, наявності підозрілих символів і доменів другого рівня, ігноруючи метадані, які не мають явної кореляції з фішингом [47].

Ще одним ключовим аспектом передобробки є створення нових змінних (feature engineering), які покращують якість даних. Наприклад, для фішингових атак корисно визначати змінні, такі як:

- співвідношення цифр до букв у URL;
- частка підозрілих слів у тексті повідомлення;
- кількість перенаправлень на інші сайти.

Такі змінні можуть бути розраховані на основі початкових даних і значно підвищити продуктивність моделі.

У випадках роботи з великими обсягами даних важливим є застосування методів обробки незбалансованих вибірок. У сфері виявлення фішингових атак легітимні зразки часто значно переважають у кількості над шкідливими, що призводить до перекосів у навчанні моделі. Для вирішення цієї проблеми застосовуються такі техніки, як:

1. Oversampling - збільшення кількості малопредставлених зразків (наприклад, через SMOTE).
2. Undersampling - зменшення кількості зразків із більш представленого класу.
3. Комбіновані методи - поєднання перших двох технік для досягнення балансу між класами.

Після обробки та нормалізації даних вони часто потребують шифрування для захисту від несанкціонованого доступу. У багатьох випадках дані можуть бути анонімізовані — наприклад, шляхом видалення ідентифікаційної інформації, що дозволяє дотримуватися вимог конфіденційності.

Також доцільно використовувати методи виявлення дубльованих зразків у навчальних даних, особливо коли база даних збирається з кількох джерел. Ця проблема є типовою у випадках роботи з відкритими базами, такими як PhishTank чи OpenPhish, які можуть містити повтори. Видалення таких зразків знижує ризик переобчислення й забезпечує стабільність моделі.

З аключним етапом передобробки є створення документів або візуалізацій, які детально описують отримані характеристики, трансформації та прийняті рішення. Це допомагає підтримувати прозорість процесу та забезпечує легшу інтеграцію моделей у практичні сценарії виявлення фішингових атак.

2.4. Розробка та оцінка ефективності методу

2.4.1. Тренування моделей на реальних даних

Тренування моделей штучного інтелекту (ШІ) на реальних даних вимагає багатостороннього підходу, щоб забезпечити надійність і високу продуктивність моделі у практичних умовах. Першочергове завдання — це правильне формування тренувальної вибірки. Реальні дані мають бути достатньо різноманітними, аби модель могла адекватно реагувати на різні сценарії, які можуть зустрітися у реальних умовах. Наприклад, для виявлення фішингових атак слід використовувати дані з різних джерел: реальні URL-адреси фішингових сайтів, аналізовані електронні листи, метадані, отримані через API сервісів, та різні мови й стилі текстів, які зустрічаються у фішингових повідомленнях [51]. На рис. 2.2 зображена загальна схема функціонування моделі, яка здатна розділяти легітимні листи від фішингових.

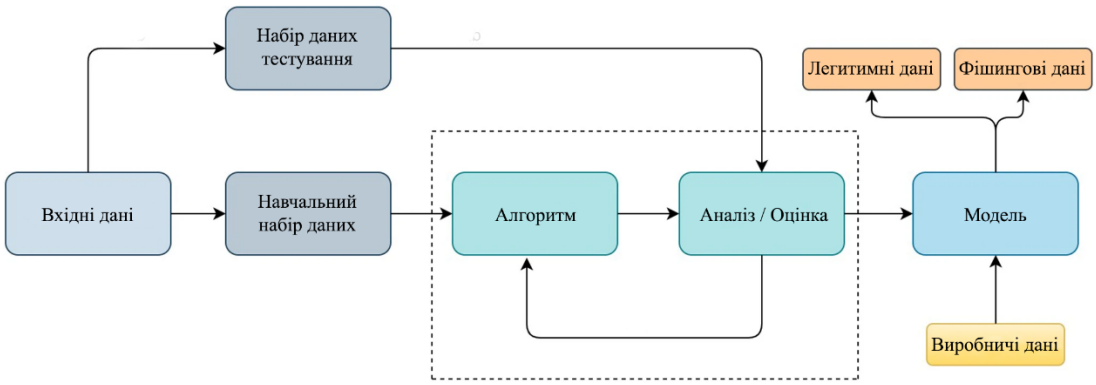

Рис. 2.2. Процес поділення фішингових та легітимних даних

Один із критичних аспектів — збалансування вибірки, оскільки реальні дані часто є нерівномірними. У випадку фішингу, наприклад, легітимних повідомлень буде значно більше, ніж шкідливих. Це може призводити до перекосів у навчанні

моделі, коли вона занадто фокусується на розпізнаванні легітимних даних, але ігнорує підозрілі. Щоб уникнути цього, можна застосовувати техніки балансування, такі як синтетичне додавання фішингових даних (oversampling), або видалення надлишкових зразків легітимних повідомлень (undersampling). Додатково використовуються комбіновані методи, наприклад, SMOTE (Synthetic Minority Oversampling Technique), які створюють синтетичні зразки на основі існуючих, зберігаючи при цьому їхню варіативність.

Реальні дані часто містять шум: неповні записи, дублювання або помилкові значення. Наприклад, база даних фішингових сайтів може включати URL-адреси, які вже неактивні або були помилково класифіковані. Використання методів очищення, таких як видалення дублікатів, заповнення пропущених значень або фільтрація за релевантними критеріями, сприяє покращенню якості даних для навчання.

Багато уваги слід приділяти адаптації моделі до змін у реальному середовищі. У випадку з фішинговими атаками загрози можуть швидко еволюціонувати. Наприклад, постійно розробляються нові техніки маскування, які не завжди виявляються у тренувальних вибірках. Це означає, що модель має регулярно перевірятися та оновлюватися на основі нових даних. Ідеальним варіантом є інтеграція автоматичних процесів оновлення, що дозволяє моделі постійно адаптуватися. Рис. 2.3. демонструє розроблену модель виявлення фішингових листів, яка використовує алгоритми Random Forest або Support Vector Machine.

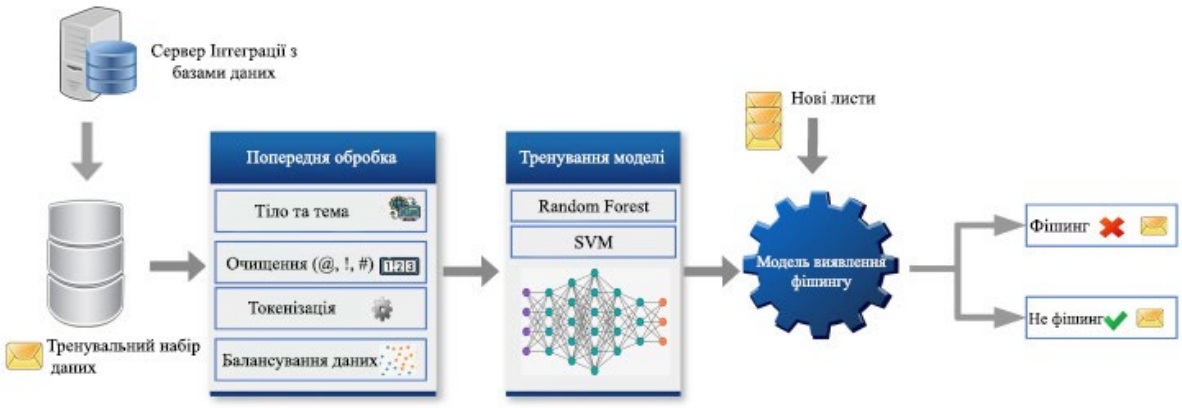

Рис. 2.3. Виявлення фішингових листів на моделі CNN

Важливою є також інтеграція механізмів оцінки якості моделі під час її навчання. Наприклад, використання технік крос-валідації дозволяє отримати більш точні оцінки продуктивності моделі на нових даних. Розподіл даних на тренувальний, тестовий та валідаційний набори допомагає уникнути проблеми перенавчання та забезпечити генералізацію. Для виявлення фішингу специфічними метриками оцінки можуть бути точність (precision), повнота (recall) та F1-міра, які враховують дисбаланс у класах.

Ще одним викликом є підвищення стійкості моделі до маніпуляцій із боку зловмисників, які можуть цілеспрямовано вводити викривлені дані, аби обійти систему. Цього можна досягти за допомогою навчання моделі на основі принципів захисного машинного навчання (adversarial training). Наприклад, у випадку фішингових атак можна створювати спеціальні зразки з навмисно модифікованими характеристиками, які все ще залишаються шкідливими, і використовувати їх для навчання. Це дозволяє моделі краще адаптуватися до нетипових загроз.

Щоб забезпечити надійність моделі, під час її тренування на реальних даних необхідно враховувати вимоги до масштабованості. Великі набори даних можуть вимагати використання розподілених обчислень або кластерних систем для тренування моделей. Використання платформ, таких як TensorFlow чи PyTorch, дозволяє ефективно працювати з великими обсягами даних, водночас забезпечуючи

паралельну обробку інформації. Інструменти для розподіленого зберігання даних, наприклад, Hadoop чи Apache Spark, допомагають інтегрувати великі бази даних і забезпечити доступ до них у реальному часі.

Додатково, слід враховувати питання етичності використання реальних даних. Наприклад, дані користувачів, які містяться у фішингових повідомленнях, можуть бути конфіденційними, і їх використання для навчання моделі потребує відповідності законам, таким як GDPR. Тому під час тренування необхідно проводити анонімізацію даних, видаляючи персональну інформацію, яка може порушувати конфіденційність.

Також варто забезпечити інтеграцію моделей із системами моніторингу продуктивності, які дозволяють аналізувати поведінку моделі під час її роботи в реальному часі. Це допомагає оперативно ідентифікувати зниження точності чи інші проблеми, що виникають через зміни у структурі даних [53].

Один з найважливіших етапів — управління розподілом характеристик даних, які використовуються для тренування. У випадку виявлення фішингу, частота використання ключових слів, структура URL, кількість перенаправлень, IP-адреси та поведінкові патерни можуть варіюватися залежно від типу атаки. Щоб уникнути навчання моделі на обмеженому наборі сценаріїв, доцільно застосовувати техніки аугментації даних. Це включає модифікацію існуючих даних шляхом невеликих змін, які зберігають сутність інформації. Наприклад, URL-адреси можна піддавати незначним змінам у структурі домену, а текстові дані — підміняти синонімами або перетворювати до різних форматів представлення.

Тренування моделей також вимагає тестування на реальних сценаріях. Наприклад, у контексті фішингових атак це може включати симуляцію роботи електронної пошти в середовищі підприємства з використанням різних політик захисту. Це дозволяє не лише покращити адаптивність моделі до нетипових ситуацій, але й перевірити її на витривалість до змін у поведінці атакуючих.

Роль обчислювальних ресурсів при тренуванні моделей на реальних даних також неможливо переоцінити. Великі обсяги даних потребують ефективного використання апаратних можливостей, таких як графічні процесори (GPU) або тензорні процесори (TPU). Наприклад, великі моделі, побудовані на основі глибокого навчання, такі як CNN або RNN, вимагають високої потужності для обробки багатовимірних даних. У той же час, у деяких випадках можна використовувати оптимізовані моделі, які потребують менше обчислювальних ресурсів, наприклад LightGBM або XGBoost, що особливо корисно в умовах обмежених систем.

Щоб забезпечити ефективність у реальних умовах, важливо використовувати метрики, які дозволяють оцінити надійність моделі. Наприклад, у випадку виявлення фішингу слід враховувати не лише точність та повноту, але й специфічність (specificity), час реагування (latency) і стійкість до хибно-позитивних спрацьовувань. У цьому контексті розробники часто використовують матриці помилок (confusion matrices) та аналіз ROC-кривих, які дають змогу оцінити, наскільки модель добре балансує між хибними тривогами та пропущеними загрозами.

Одним із перспективних підходів для підвищення надійності є використання змішаних даних, які об’єднують синтетичні й реальні джерела. Наприклад, синтетичні дані можна генерувати з використанням спеціальних фреймворків, таких як Faker або TextAttack, щоб доповнити нестачу специфічних прикладів фішингових атак. Це не лише покращує якість навчання, але й дозволяє уникнути проблеми конфіденційності, оскільки синтетичні дані не містять реальної інформації про користувачів.

Ще одним ключовим аспектом є розробка механізмів постійного оновлення моделі. Це може включати періодичне перенавчання на нових даних, зібраних за допомогою автоматизованих процесів. Наприклад, інтеграція із системами, які відслідковують нові фішингові сайти або створюють звіти про атаки, дозволяє

підтримувати актуальність моделі. У таких системах використовуються методики incremental learning, які додають нові зразки до існуючої моделі без необхідності повторного повного тренування.

Важливу роль відіграють також методики валідації моделі в реальному середовищі. Це може включати тестування в умовах реальної мережі з використанням поточних даних або інтеграцію моделі у систему захисту з метою оцінки її продуктивності у реальних сценаріях. Наприклад, моделі для виявлення фішингових атак можна перевіряти в умовах активного електронного листування компанії, аналізуючи, як вони справляються із щоденними загрозами.

Також варто враховувати специфіку реалізації моделі в розподіленому середовищі. Наприклад, при інтеграції моделі в хмарні платформи необхідно забезпечити її масштабованість та можливість обробки великого обсягу даних у реальному часі. Це особливо актуально для організацій, які використовують моделі для моніторингу мережі в масштабах підприємства.

2.4.2. Аналіз результатів тестування

Аналіз результатів тестування моделі для виявлення фішингових атак є фінальним етапом, який дозволяє оцінити її продуктивність, виявити слабкі місця й удосконалити її функціонування. Розглядаються такі аспекти, як точність визначення фішингових атак, помилки класифікації, здатність моделі працювати з різними наборами даних і загальна ефективність алгоритму.

У процесі тестування моделей головним індикатором є точність (accuracy) — частка правильно класифікованих зразків серед усіх. Однак точність часто не відображає реальної ефективності у випадках, коли розподіл між класами є нерівномірним. Наприклад, у наборах даних, де реальних фішингових сайтів значно менше, ніж легітимних, модель може показувати високу точність, просто класифікуючи більшість зразків як безпечні. У таких випадках важливішими

метриками є повнота (recall) та точність позитивного класу (precision). Повнота показує, який відсоток фішингових сайтів було виявлено серед усіх реальних загроз, а precision вказує, яка частка сайтів, позначених як фішингові, справді є небезпечними [54].

Таблиця 2.

Результати тестування алгоритмів на різних наборах даних

Назва Алгоритму	Використані набори даних	Точність виявлення
Bayes Net	Phishing Corpus and SpamAssassin	Acc: 92%
RF	Phishing Corpus and SpamAssassin	Acc: 97%
SVM	Phishery and 2007 TREC Corpus	Acc: 98.2%
RF	Enron	Acc: 96.18%
SVM	Phishing Corpus and SpamAssassin	Acc: 97.25%
SVM	Enron1	Acc: 99.38%
RF and NLP	Phishing webpages	Acc: 97.98%
RF	Spam Corpus	Acc: 99.91%

Детальний аналіз помилок класифікації допомагає зрозуміти, чому модель може давати некоректні результати. Помилки першого роду (false positives) виникають, коли легітимний сайт класифікується як фішинговий. Це може бути спричинено нестандартною структурою легітимного веб-ресурсу, схожою на шаблони, характерні для фішингових атак. Наприклад, сайти, які використовують скорочені URL або нестандартні доменні зони, можуть помилково вважатися загрозами. Помилки другого роду (false negatives) є серйознішою проблемою, оскільки вони означають, що фішинговий сайт залишився непоміченим. Такі помилки часто виникають, якщо атакуючі застосовують методи, які раніше не зустрічалися у навчальних даних моделі, наприклад, нові варіанти соціальної інженерії чи унікальні алгоритми шифрування URL.

Для поглибленого аналізу помилок класифікації застосовують методику візуалізації, наприклад, heatmaps для порівняння вхідних характеристик. Це

дозволяє визначити, які атрибути мали найбільший вплив на рішення моделі. Наприклад, якщо URL-адреси фішингових сайтів містять підозрілі ключові слова, але їх не було враховано під час навчання, необхідно доопрацювати відповідні фільтри.

Тестування моделі також охоплює аналіз її продуктивності на різних наборах даних. Наприклад, якщо модель була навчена на даних, зібраних у межах однієї регіональної зони або для конкретної мови, її ефективність може бути нижчою при застосуванні до сайтів з інших країн або мовних груп. У таких випадках оцінюють здатність моделі до узагальнення (generalization) — можливості коректно класифікувати нові зразки, які не були представлені під час навчання.

Особливу увагу приділяють оцінці часу обробки запитів (latency). У реальному часі, коли модель інтегрована у системи захисту, занадто довгий час відповіді може зробити її непрактичною. Вимірюють середній час класифікації одного зразка та визначають, чи відповідає він допустимим межам для поточного застосування.

Аналіз результатів також включає порівняння роботи моделі з іншими існуючими рішеннями. Це дозволяє зрозуміти, наскільки конкурентоспроможною є розробка. Використовують стандартизовані набори даних, наприклад, PhishTank або OpenPhish, і проводять бенчмаркінг, порівнюючи основні метрики: recall, precision, F1-score. У разі виявлення слабких місць можна внести корективи до архітектури моделі або змінити алгоритм передобробки даних.

У другому розділі було розроблено ефективний метод виявлення фішингових атак, заснований на використанні алгоритмів машинного та глибокого навчання. Сформовано підхід до збору, обробки та аналізу даних про фішингові атаки, що включає автоматизовану класифікацію URL, аналіз контенту електронних листів і поведінкові аспекти користувачів. Розроблений метод демонструє високу ефективність, забезпечуючи адаптивність до нових загроз завдяки постійному навчанню моделей на актуальних даних.

РОЗДІЛ 3
РОЗРОБКА МЕТОДИЧНИХ РЕКОМЕНДАЦІЙ ЩОДО ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

3.1. Аналіз існуючих методичних рекомендацій

3.1.1. Огляд стандартів і рекомендацій у сфері кібербезпеки

Стандарти ISO/IEC у кібербезпеці

Сімейство стандартів ISO/IEC 27000 пропонує комплексний підхід до управління інформаційною безпекою. Зокрема, ISO/IEC 27001 встановлює вимоги до систем управління інформаційною безпекою (СУІБ). Цей стандарт акцентує увагу на управлінні ризиками, включаючи ризики, пов’язані з соціальною інженерією та фішинговими атаками. ISO/IEC 27002 є допоміжним стандартом, що надає рекомендації щодо імплементації заходів контролю. У сфері фішингових атак акцент робиться на підвищенні обізнаності користувачів, контролі доступу та використанні технологічних заходів, таких як багатофакторна автентифікація [66].

Ще одним важливим документом є ISO/IEC 29147, який орієнтований на управління вразливостями. У контексті фішингу цей стандарт підкреслює важливість швидкого реагування на повідомлення про нові загрози, а також необхідність створення механізмів обміну інформацією між організаціями.

Рекомендації NIST

Національний інститут стандартів і технологій (NIST) пропонує комплекс рекомендацій для захисту від кіберзагроз. Документ NIST Special Publication 800-53 визначає заходи контролю для інформаційних систем, які можуть

застосовуватись для мінімізації впливу фішингових атак. У цьому контексті розглядаються методи захисту електронної пошти, включаючи використання протоколів DMARC, SPF та DKIM, які дозволяють виявляти підроблені домени [68].

Інший ключовий документ — NIST Cybersecurity Framework (CSF). Цей фреймворк містить п’ять основних функцій: ідентифікація, захист, виявлення, реагування та відновлення. У контексті фішингу особливу увагу приділено функціям виявлення (використання систем моніторингу для аналізу мережевого трафіку) та реагування (планування дій на випадок успішного фішингового нападу) [69].

Рекомендації ENISA

Агентство Європейського Союзу з кібербезпеки (ENISA) також публікує рекомендації, спрямовані на підвищення безпеки. Наприклад, ENISA Cybersecurity Guide для малого та середнього бізнесу підкреслює важливість навчання персоналу в питаннях розпізнавання фішингових повідомлень. ENISA також надає рекомендації щодо створення систем виявлення загроз у реальному часі та активного використання Threat Intelligence.

Додаткові міжнародні стандарти

Документ CIS Controls пропонує конкретні заходи для захисту організацій. Він включає рекомендації щодо сегментації мережі, постійного моніторингу активності, а також впровадження антифішингових технологій, таких як пісочниці для перевірки підозрілих вкладень.

Стандарт PCI DSS для організацій, які працюють з платіжними картками, вимагає суворого контролю доступу, шифрування даних та регулярного аудиту. Ці вимоги також застосовні до виявлення фішингових загроз у фінансових системах.

Роль стандартів у створенні методичних рекомендацій

Інтеграція міжнародних стандартів до методичних рекомендацій дозволяє забезпечити комплексний підхід до кібербезпеки. Наприклад:

- Використання ISO/IEC 27001 як основи для управління ризиками.
- Інтеграція NIST SP 800-53 для покращення технічного захисту.
- Впровадження CIS Controls для підвищення обізнаності користувачів.

3.1.2. Ідентифікація слабких місць у поточних методиках

Складність імплементації

Реалізація таких стандартів, як ISO/IEC 27001 або NIST CSF, потребує значних фінансових і часових ресурсів. Для малих і середніх підприємств (МСП) це стає серйозною проблемою. Наприклад, повна сертифікація ISO/IEC може зайняти від кількох місяців до року, що включає аудит, оцінку ризиків та впровадження заходів. Крім того, стандарти часто вимагають використання дорогих інструментів, які не завжди виправдані для малих організацій. ENISA у своїх рекомендаціях пропонує загальні принципи, однак вони інколи занадто високорівневі, що ускладнює їхню адаптацію до конкретних потреб.

Недостатня гнучкість

Багато стандартів є статичними документами, які оновлюються повільно, іноді раз на кілька років. Наприклад, навіть у NIST CSF оновлення відбуваються з урахуванням вже існуючих викликів, а не тих, що тільки виникають. Це призводить до затримок в адаптації до нових методів фішингу, які можуть з’являтися щомісяця. У випадку ENISA, рекомендації часто базуються на широкому аналізі, але їхнє застосування на практиці вимагає створення додаткових протоколів для конкретних випадків, наприклад, для захисту IoT.

Відсутність спеціалізації на соціальній інженерії

Традиційні стандарти здебільшого зосереджені на технічних аспектах безпеки, таких як шифрування, контроль доступу або управління ідентифікацією. Проте фішингові атаки, що використовують психологічні маніпуляції, вимагають іншого підходу. Стандарти, як правило, не включають детальних рекомендацій

щодо навчання персоналу або виявлення ознак соціальної інженерії. Наприклад, ISO/IEC 27002 включає пункти щодо підвищення обізнаності, але їхня деталізація залишає прогалини для самостійного розроблення конкретних програм.

Недостатня автоматизація

Методики, описані в стандартах, здебільшого передбачають використання ручного управління. У контексті сучасних реалій, коли кількість кіберзагроз зростає експоненційно, такі підходи стають неефективними. Наприклад, у NIST SP 800-61 щодо реагування на інциденти процес часто залежить від людського втручання, що створює затримки. Використання автоматизованих засобів для моніторингу, таких як системи на основі штучного інтелекту, не завжди згадується як стандартна рекомендація.

Недостатнє тестування на реальних даних

Більшість рекомендацій, особливо в стандартах ISO/IEC, побудовані на абстрактних сценаріях і передбачають використання даних, зібраних організацією самостійно. Однак реальні дані, наприклад, списки фішингових доменів, можуть значно змінюватися залежно від галузі або регіону. ENISA, хоча й надає аналітичні звіти про загрози, не пропонує універсальних рекомендацій щодо їхнього використання для тестування власних систем безпеки.

Орієнтація на захист інфраструктури, а не користувача

Традиційні стандарти зосереджуються на технологічних аспектах захисту, ігноруючи кінцевого користувача як ключову ланку в ланцюжку атак. Наприклад, рекомендації NIST SP 800-53 мають розділи про навчання персоналу, але вони не враховують специфіку атак, спрямованих безпосередньо на окремих співробітників, таких як CEO-фішинг.

Проблеми з інтеграцією нових технологій

Інтеграція сучасних технологій, таких як штучний інтелект або машинне навчання, у традиційні методики часто викликає труднощі через невідповідність стандартних процедур. Наприклад, ISO/IEC 27002 включає загальні вимоги до

управління ризиками, але не надає вказівок щодо адаптації систем ШІ для автоматизації процесів оцінки. Так само NIST SP 800-61 описує ручний підхід до аналізу інцидентів, не беручи до уваги переваги використання автоматизованих засобів аналізу журналів подій чи трафіку.

Недостатній аналіз специфічних кіберзагроз

Традиційні стандарти зосереджуються на загальних принципах управління ризиками, часто не приділяючи належної уваги специфіці певних кіберзагроз. Наприклад, у NIST CSF більше уваги приділено фізичній безпеці інфраструктури та захисту мережі, ніж специфічним методам фішингу. ENISA надає огляд тенденцій у кіберзагрозах, але не надає детальних технічних інструкцій для протидії фішинговим атакам у реальному часі.

Також, існуючі стандарти, такі як ISO/IEC, NIST, та рекомендації ENISA, демонструють певні обмеження у застосуванні до виявлення фішингових атак. Наприклад:

Обмежена актуальність стандартів щодо сучасних фішингових технік

Стандарти ISO/IEC 27001 та 27002 орієнтовані на загальні принципи інформаційної безпеки, такі як управління ризиками та контроль доступу. Однак, вони не надають конкретних інструментів для ідентифікації складних методів фішингу, як-от атаки, що використовують підроблені сторінки з адаптивними алгоритмами. NIST SP 800-61 сфокусований на реагуванні на інциденти, але не враховує проактивного виявлення загроз, які змінюються у реальному часі. Рекомендації ENISA також часто акцентують увагу на макрорівні загроз, що ускладнює їх застосування для локалізованих чи цільових атак.

Обмеження у використанні інтелектуальних систем

ISO/IEC та NIST надають мінімальну увагу інтеграції сучасних алгоритмів штучного інтелекту. Наприклад, їх рекомендації щодо виявлення фішингу часто базуються на сигнатурному аналізі та ручному розпізнаванні шаблонів. Проте сучасні фішингові кампанії використовують техніки, як-от нейронні мережі для

генерації текстів, що легко обходять традиційні методи виявлення. ENISA у своїх звітах визнає важливість використання машинного навчання, але не деталізує підходів до інтеграції таких рішень у стандартні методики безпеки.

Проблеми з масштабованістю та обробкою великих обсягів даних

Методики NIST, такі як аналіз мережевого трафіку, не розраховані на обробку великих обсягів даних, що є ключовим у сучасних системах кібербезпеки. Приклади фішингових кампаній із масовою розсилкою показують, що необхідно швидко аналізувати мільйони повідомлень або веб-сторінок. Традиційні підходи, рекомендовані ISO/IEC, обмежені у здатності працювати з великими потоками даних, що знижує їхню ефективність у реальному часі.

Відсутність детальних вказівок щодо динамічних фішингових загроз

ENISA та NIST часто розглядають фішингові атаки як статичні загрози, тоді як у реальності вони дедалі частіше включають динамічні аспекти, як-от одноразові URL-адреси або сайти, що автоматично адаптуються до користувацьких даних. Стандарти ISO/IEC орієнтуються на методи, які не враховують поведінкових змін у загрозах. Це створює прогалину, яку важко заповнити без використання сучасних динамічних систем виявлення.

Недостатня увага до людського фактора

Більшість стандартів і рекомендацій акцентують на технічних аспектах, залишаючи осторонь людський фактор, який відіграє критичну роль у фішингових атаках. ENISA рекомендує навчання користувачів, але не деталізує, як ефективно інтегрувати ці заходи з технічними засобами. ISO/IEC не враховують психологічні аспекти впливу фішингу, як-от соціальна інженерія.

3.2. Впровадження методики в організаціях та на підприємствах

3.2.1. Підготовка організації до впровадження ШІ-рішень для виявлення фішингу

Для впровадження штучного інтелекту (ШІ) в систему виявлення фішингових атак організація повинна забезпечити ряд підготовчих заходів, які охоплюють кілька ключових напрямків: технічну підготовку, формування команди, створення відповідних політик і залучення необхідних ресурсів. На рис. 3.1. зображені загальні етапи, завдяки яким можливе впровадження ШІ в систему виявлення фішингу.

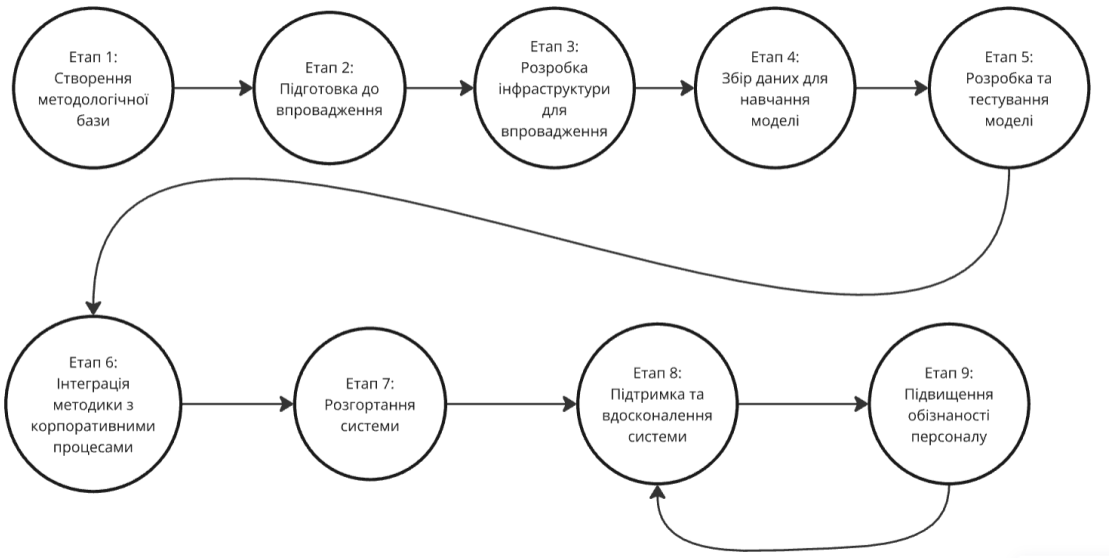

Рис. 3.1. Етапи впровадження ШІ в існуючі системи захисту інформації в організаціях

Аналіз технічної інфраструктури та вибір платформ

Організація повинна оцінити поточний стан своєї технічної інфраструктури. Впровадження ШІ-рішень потребує наявності апаратного забезпечення, яке здатне

підтримувати роботу інструментів машинного навчання. Для цього необхідні високопродуктивні сервери з потужними процесорами (CPU) або графічними процесорами (GPU), що забезпечують паралельну обробку великих обсягів даних. Крім того, важливо оцінити обсяг доступної пам’яті для зберігання даних. Для зручності можна використовувати хмарні сервіси, такі як Google Cloud, AWS або Microsoft Azure, які пропонують спеціалізовані середовища для розробки та навчання моделей ШІ.

Також слід забезпечити наявність програмного забезпечення, яке підтримує інтеграцію з існуючими інструментами кібербезпеки, наприклад SIEM-системами (Splunk, IBM QRadar), щоб автоматизувати виявлення та реагування на фішингові атаки. Слід визначити, чи існує можливість масштабування цієї інфраструктури для майбутніх потреб, оскільки обсяги даних та складність алгоритмів можуть зростати.

Створення команди

Ключовим елементом успішного впровадження ШІ є наявність компетентної команди. Організація повинна або найняти фахівців, або підвищити кваліфікацію наявного персоналу. У складі команди мають бути:

- Дослідники даних (Data Scientists): спеціалісти, які вміють аналізувати дані та будувати моделі машинного навчання.
- Фахівці з кібербезпеки: експерти, які розуміють специфіку фішингових атак і можуть налаштувати модель відповідно до реальних загроз.
- Інженери DevOps: для забезпечення надійної інтеграції ШІ-рішень у поточні процеси організації.
- Керівники проєкту: для планування впровадження та контролю результатів.

Для підготовки персоналу слід використовувати курси з питань ШІ, машинного навчання та обробки даних, наприклад, на платформах Coursera або Udemy. Окрім того, спеціалізовані тренінги від ENISA або NIST допоможуть зрозуміти, як впроваджувати ШІ у сфері кібербезпеки.

Збір та підготовка даних

Перед впровадженням системи важливо мати чітко визначені джерела даних. Для цього можна використовувати:

- Відкриті бази даних, такі як PhishTank, OpenPhish або APWG, які містять інформацію про реальні фішингові атаки.
- Власні історичні дані організації про інциденти безпеки.
- Дані, отримані через API-інтерфейси провідних сервісів з моніторингу кіберзагроз.

Зібрані дані повинні пройти процес передобробки, включаючи очищення, нормалізацію та видалення дублікатів, щоб підвищити якість навчання моделі. Крім того, слід забезпечити анонімізацію конфіденційної інформації відповідно до міжнародних стандартів (GDPR, ISO/IEC 27001).

Розробка політик і стратегій

Впровадження ШІ-рішень потребує адаптації внутрішніх політик організації. Слід оновити стратегію кібербезпеки, додавши вимоги щодо моніторингу, тестування та оновлення моделей ШІ. Також необхідно визначити критерії оцінки ефективності рішень, включаючи рівень виявлення фішингових атак, кількість помилкових спрацювань і час реагування на інциденти.

Організація повинна створити план дій на випадок технічних проблем або кібератак, спрямованих на інфраструктуру ШІ. Наприклад, передбачити резервні копії моделей, регулярне оновлення систем та проведення тестів на проникнення.

Фінансове планування та ресурси

Для впровадження ШІ потрібне значне фінансування. Необхідно врахувати витрати на:

- Закупівлю або оренду апаратного забезпечення та ліцензії на ПЗ.
- Зарплати для нових співробітників або оплату навчання.
- Абонентську плату за використання хмарних платформ і сервісів.

- Тестування та подальшу оптимізацію рішень.

Фінансове планування повинно включати оцінку довгострокових витрат, таких як оновлення обладнання, підтримка моделей і додаткові ресурси для масштабування.

Оцінка організаційної готовності

Перед початком роботи з впровадження ШІ важливо провести аудит поточних процесів та інфраструктури організації. Це включає аналіз готовності існуючих технологій для інтеграції з новими ШІ-рішеннями, оцінку рівня захищеності мережі, ідентифікацію потенційних "вузьких місць", таких як недостатня обчислювальна потужність або низький рівень автоматизації. Важливо врахувати, що не всі компанії мають внутрішню готовність до інтеграції технологій ШІ. У таких випадках слід залучати зовнішніх консультантів.

Для оцінки готовності часто використовуються спеціалізовані фреймворки, як-от AI Maturity Model, які допомагають структурувати підхід до підготовки організації, розбиваючи впровадження на етапи.

Розробка плану впровадження

Впровадження ШІ потребує детального планування. Цей план має включати:

- Цілі проєкту: визначення конкретних завдань, які повинен вирішувати ШІ, наприклад, скорочення часу виявлення фішингових атак або підвищення точності класифікації.
- Технічні етапи: розбиття процесу на підзадачі, такі як інтеграція з існуючими платформами захисту, тестування моделей та оптимізація алгоритмів.
- Тимчасові рамки: встановлення чітких термінів виконання кожного етапу, що дозволяє відстежувати прогрес.

Це також включає визначення потенційних ризиків впровадження, таких як затримки у розробці, проблеми з якістю даних або можливі технічні збої, і розробку відповідних стратегій їх мінімізації.

Тестування в пілотному режимі

Впровадження ШІ слід починати з пілотного проєкту, у рамках якого технології будуть протестовані на невеликій частині даних або в окремому середовищі. Це дозволяє оцінити ефективність ідентифікації фішингових загроз, а також оптимізувати модель до її впровадження у повному масштабі. Пілотні проєкти мінімізують ризик помилок, що можуть виникнути внаслідок неврахованих факторів.

Для цього створюється тестове середовище, яке імітує реальні умови організації. Дані, використані під час тестування, мають бути якомога ближчими до бойових умов, оскільки це гарантує релевантність результатів.

Моніторинг та адаптація

Після запуску системи в роботу необхідно організувати безперервний моніторинг її продуктивності. Це включає аналіз метрик, таких як точність виявлення фішингових атак, частота помилкових спрацьовувань, а також відгуки користувачів. Отримані дані використовуються для внесення змін у модель, наприклад, для переобучення або оптимізації алгоритмів.

Крім того, система має бути гнучкою, щоб адаптуватися до нових типів загроз. Це може вимагати впровадження оновлень, інтеграції нових джерел даних або змін у конфігурації інструментів моніторингу.

3.2.2. Стратегії інтеграції з існуючими системами безпеки

Для інтеграції методів виявлення фішингових атак на базі ШІ з існуючими системами безпеки важливо враховувати архітектурну сумісність, наявність функціональних перетинів, а також можливості подальшої автоматизації. Поєднання цих технологій має ґрунтуватися на багаторівневій структурі інтеграції, яка охоплює технічні аспекти, процеси обміну даними та управління кіберризиками.

Основним етапом інтеграції є створення спільного каналу для передачі та обробки даних. Наприклад, дані, отримані від ШІ-моделей, можуть бути інтегровані до системи SIEM, яка надає аналітику подій у режимі реального часу. Це дає змогу відразу виявляти аномалії в мережевій активності, пов'язані з фішинговими атаками. До цього слід додати використання форматів обміну даними, таких як JSON чи XML, які забезпечують стандартизовану передачу інформації.

Наступний рівень - це об'єднання з існуючими політиками багаторівневого захисту, включно з мережевими міжмережевими екранами, антивірусними програмами та хмарними сервісами для захисту електронної пошти. Це включає налаштування автоматичних тригерів для відповідних реакцій, таких як блокування підозрілих URL-адрес, видалення небезпечних листів чи інформування співробітників про потенційні загрози. Додатково, API таких систем, як Office 365 Advanced Threat Protection, дозволяють впроваджувати аналітику в існуючі платформи без потреби створення нових інфраструктурних рішень.

Ключовим аспектом успішної інтеграції є аналіз обмежень наявної інфраструктури. Наприклад, при використанні систем, які вже мають затримки у виконанні, додаткове навантаження від ШІ-рішень може призвести до їхньої нестабільної роботи. Тому варто оцінити обчислювальні потужності серверів, швидкість мережевих каналів і наявність резервів для масштабування. У випадках перевантаження існуючих систем рекомендується використовувати хмарні рішення для зберігання та обробки великих обсягів даних, що зменшує тиск на локальну інфраструктуру.

Особливу увагу слід приділити інтеграції через хмарні сервіси, які дозволяють легко масштабувати потужності підвищеної обробки даних. Наприклад, сервіси Amazon Web Services (AWS) чи Google Cloud можуть забезпечувати обробку потоків фішингових даних та їх перевірку за допомогою попередньо навчених моделей. Цей підхід ефективний у великих організаціях з

великою кількістю користувачів і розподіленими офісами, де важливо забезпечити захист у різних регіонах.

Не менш важливим є налаштування моніторингу та керування системою інтеграції. Сюди входять регулярні перевірки точності роботи моделі та аналіз фідбеку від співробітників і системи. Використання дашбордів та звітності дозволяє оперативно ідентифікувати проблеми, оцінювати ефективність інтеграції та адаптувати модель до змінних загроз.

Інтеграція також потребує розробки політик безпеки, які враховують специфіку використання ШІ. Наприклад, у правила роботи з електронною поштою можна включити процес обов'язкової перевірки всіх підозрілих повідомлень через ШІ-систему перед тим, як вони досягнуть кінцевого користувача. Додатково, співробітники мають бути проінформовані про зміни в системах безпеки та нові протоколи.

Для забезпечення успішної інтеграції системи виявлення фішингових атак з іншими рішеннями безпеки важливо врахувати питання відповідності програмного забезпечення, апаратної інфраструктури та політик доступу. Наприклад, системи антивірусного захисту або шлюзи безпеки для веб-запитів можуть бути ефективними партнерами для роботи ШІ-алгоритмів. Важливо забезпечити, щоб API, якими користуються системи, були взаємосумісними і дозволяли обмін даними без втрат продуктивності.

Крім того, розробка універсального протоколу інтеграції є ключовим завданням. Багато існуючих рішень використовують різні формати збереження даних, такі як лог-файли, бази даних або потокові дані з серверів. Для поєднання всіх цих джерел необхідно створити проміжний шар обробки даних, який трансформуватиме інформацію в уніфікований формат. Наприклад, системи типу Splunk чи ELK Stack можуть виконувати функцію цього прошарку, забезпечуючи агрегацію даних перед їхньою передачею в систему виявлення загроз.

Одним із ключових аспектів інтеграції є налаштування функцій автоматизації дій у відповідь на виявлені загрози. Наприклад, якщо модель ідентифікує потенційно фішингове посилання, система може автоматично блокувати його на рівні шлюзу або надсилати сповіщення користувачу. Це дозволяє значно скоротити час реакції на загрози, підвищуючи рівень захищеності мережі.

Для організацій, які використовують хмарні сервіси, інтеграція з системами безпеки хмари, такими як Microsoft Defender для Office 365 чи Google Workspace Security, забезпечує додатковий рівень захисту. Наприклад, ці платформи можуть використовувати функції ШІ для аналізу метаданих листів або контенту, а потім передавати дані для поглибленої аналітики в інтегровану систему.

Оптимізація архітектури інтеграції також включає розподіл ресурсів між компонентами системи. Наприклад, дані про підозрілу активність можуть бути розділені між різними рівнями аналізу: поверхневим для швидкого виявлення фішингу та глибоким - для підтвердження загроз. Це дозволяє зменшити навантаження на основні системи безпеки, зберігаючи їхню продуктивність на високому рівні.

3.2.3. Навчання персоналу для роботи з ШІ-методами

Навчання персоналу для роботи з методами штучного інтелекту (ШІ) вимагає системного підходу, який охоплює технічні аспекти, питання безпеки, а також практичну підготовку до взаємодії із системами ШІ. Ефективна програма навчання повинна враховувати рівень підготовки співробітників, специфіку їхніх завдань і роль у процесі забезпечення безпеки організації.

Основним завданням є навчання технічного персоналу основам функціонування моделей ШІ, їхньому налаштуванню та моніторингу. Для цього варто організувати спеціалізовані тренінги, на яких фахівці зможуть вивчити принципи роботи нейронних мереж, методи машинного навчання та способи

інтеграції ШІ в інфраструктуру організації. Окрім теоретичної частини, програма навчання повинна включати практичні модулі, які дозволять співробітникам працювати з реальними даними та сценаріями кіберзагроз.

Рекомендації з навчання повинні також охоплювати аспекти використання інструментів автоматизації, таких як спеціалізовані платформи для виявлення фішингових атак. Наприклад, персонал має бути ознайомлений з роботою таких систем, як IBM Watson, Google AI чи Azure Machine Learning. Практична підготовка може включати налаштування цих інструментів, їх тестування та аналіз результатів роботи.

Для працівників, які не є технічними спеціалістами, але взаємодіють із системами ШІ, необхідно розробити програму навчання, орієнтовану на розуміння процесів виявлення загроз і мінімізацію людського фактору. Наприклад, для співробітників відділів підтримки чи HR важливо навчитися розпізнавати фішингові атаки, використовуючи інструменти ШІ, а також реагувати на інциденти відповідно до протоколів.

Особливу увагу варто приділити формуванню навичок роботи з великими обсягами даних, які аналізуються моделями ШІ. Співробітники повинні бути обізнані з інструментами візуалізації даних, такими як Tableau чи Power BI, які дозволяють оцінити ефективність роботи моделі. Також доцільно навчати персонал основам роботи з мовами програмування, зокрема Python, що забезпечить глибше розуміння принципів роботи ШІ-моделей.

Однією з ключових рекомендацій є регулярне оновлення знань співробітників. Технології ШІ постійно розвиваються, і для ефективної роботи необхідно забезпечити доступ до актуальних курсів, семінарів і тренінгів. Це можуть бути як внутрішні програми навчання, організовані в компанії, так і зовнішні курси від провідних постачальників, таких як Coursera, Udemy чи Microsoft Learn.

Варто врахувати, що навчання повинно бути інтерактивним і залучати співробітників до активного обговорення. Для цього можна використовувати ігрові сценарії, моделювання ситуацій, пов'язаних з реальними кіберзагрозами, а також конкурси серед працівників. Це не тільки підвищить ефективність навчання, але й мотивуватиме співробітників до вдосконалення своїх навичок.

Інтеграція методів ШІ в роботу організації також вимагає адаптації внутрішніх політик і процедур. У рамках навчання співробітники повинні ознайомитися з новими стандартами роботи, наприклад, правилами конфіденційності даних, принципами реагування на інциденти та етичними аспектами використання ШІ. Це допоможе уникнути конфліктів і забезпечити прозорість процесів.

Для успішної реалізації програм навчання важливо також визначити ключових співробітників, які стануть амбасадорами впровадження ШІ. Вони повинні глибше розуміти принципи функціонування моделей і допомагати колегам у вирішенні проблем. Такі спеціалісти можуть бути залучені до розробки внутрішніх інструкцій, проведення тренінгів і моніторингу ефективності навчання.

Навчання персоналу для роботи з ШІ-рішеннями повинно враховувати кілька ключових аспектів, включаючи технічну, організаційну, і етичну підготовку. Нижче наведено основні рекомендації, які допоможуть забезпечити ефективну інтеграцію цих технологій у роботу організації:

1. Ознайомлення з основами ШІ та машинного навчання. Перший етап навчання включає введення в базові концепції роботи ШІ, такі як типи алгоритмів, принципи навчання моделей, і методи обробки даних. Особливу увагу слід приділити інструментам виявлення фішингових атак, таким як використання NLP для аналізу текстів або CNN для обробки графічних матеріалів.
2. Підготовка до роботи з платформами для навчання та аналізу моделей. Співробітникам слід навчитися працювати з платформами, такими як TensorFlow, Scikit-learn, або PyTorch. Це дозволить їм налаштовувати та

перевіряти моделі, адаптувати їх до специфічних потреб організації та інтегрувати з існуючими системами.

3. Симуляція реальних сценаріїв роботи. У навчанні важливо використовувати симуляції можливих сценаріїв атак і роботи моделей ШІ в реальних умовах. Це допомагає персоналу навчитися швидко реагувати на інциденти та адаптувати свої дії до умов роботи.
4. Постійне оновлення знань. З огляду на стрімкий розвиток технологій, організація має забезпечити співробітникам доступ до найсучаснішої інформації про загрози та методи їх виявлення. Це може включати участь у конференціях, вебінарах і курсах підвищення кваліфікації.
5. Етичний аспект роботи з ШІ. Важливо навчати персонал дотримання етичних принципів роботи з ШІ, включаючи захист конфіденційних даних, уникнення дискримінації в алгоритмах і прозорість дій системи. Це забезпечить довіру до технологій і їх використання.
6. Оцінка результатів навчання. Організація повинна розробити механізми оцінки ефективності навчання співробітників, наприклад, через тести, практичні кейси або інтерактивні симуляції. Це дозволить коригувати програми навчання, забезпечуючи їх відповідність поточним завданням.

3.3. Оцінка ефективності та постійне удосконалення методики

Оцінка ефективності впровадження ШІ-рішень для виявлення фішингу вимагає системного підходу, що враховує не лише технологічні аспекти, але й організаційні. Основними етапами цього процесу є аналіз результатів роботи системи, порівняння їх з початковими цілями, а також оптимізація та адаптація інтегрованих рішень для досягнення більшої результативності. Нижче наведено методи оцінки ефективності та шляхи вдосконалення впроваджених рішень:

1. Кількісні метрики результативності. Одним із найпоширеніших методів є оцінка точності, повноти та специфічності моделі. Для цього аналізують такі показники, як рівень правильних виявлень (True Positive Rate), рівень помилкових спрацьовувань (False Positive Rate) та середній час виявлення загроз. Такі метрики дозволяють об'єктивно оцінити продуктивність системи та її вплив на захист організації.
2. Якісний аналіз помилок. Після впровадження важливо провести детальний аналіз помилок класифікації, які допускає модель. Наприклад, можна аналізувати кейси, де фішингові атаки залишилися невиявленими (False Negatives), або випадки хибного спрацювання на безпечні елементи (False Positives). Такий аналіз дозволяє зрозуміти причини помилок, які можуть бути пов’язані як з недостатньою якістю навчальних даних, так і з недоліками самої моделі.
3. Моніторинг у реальному часі. Для оцінки реальної ефективності важливо інтегрувати інструменти моніторингу продуктивності у виробниче середовище. Це може включати аналіз логів, відстеження поведінки користувачів і реакцію системи на нові загрози. Моніторинг допомагає виявити, наскільки система адаптивна до змін у середовищі атак.
4. Порівняння з альтернативними рішеннями. Щоб оцінити вартість впровадженого ШІ-рішення, корисно провести A/B тестування з іншими методами або технологіями, які використовуються в організації. Це дозволяє визначити, чи забезпечує інтегроване рішення кращий рівень захисту, ніж традиційні підходи або альтернативні системи.
5. Вплив на загальну безпеку організації. Ефективність ШІ-рішення повинна оцінюватися не лише за технічними параметрами, але й за тим, як воно сприяє загальній стратегії кібербезпеки організації. Наприклад, слід аналізувати, наскільки впровадження ШІ допомогло знизити кількість інцидентів безпеки,

підвищити рівень довіри користувачів до системи чи оптимізувати витрати на інші засоби захисту.

6. Зворотний зв'язок від користувачів і персоналу. Інтеграція нових рішень часто викликає зміни у робочих процесах. Тому важливо отримувати регулярний зворотний зв'язок від співробітників, які використовують систему, і враховувати їхні зауваження для оптимізації.
7. Автоматизація процесу оцінки. Впровадження автоматизованих інструментів для аналізу ефективності може значно спростити цей процес. Наприклад, системи аналітики, які інтегруються з ШІ-моделями, дозволяють автоматично збирати статистику, генерувати звіти та надавати рекомендації для вдосконалення.

Ключовим аспектом вдосконалення інтегрованих рішень є адаптивність до змінюваних умов. Це досягається шляхом регулярного оновлення моделей, використання більш сучасних алгоритмів і розширення обсягів навчальних даних. Постійний аналіз результатів роботи і вдосконалення методів дозволяє не тільки підвищити ефективність, але й забезпечити відповідність системи новим викликам у сфері кібербезпеки.

Щоб забезпечити максимальну ефективність оцінки та вдосконалення інтегрованих рішень на основі отриманих даних, важливо впроваджувати комплексний підхід до збору й аналізу інформації. Використання великих масивів історичних і реальних даних дозволяє розробляти не лише точніші моделі, а й створювати детальніший контекст для оцінювання роботи системи. Організації можуть інтегрувати платформи управління даними, які автоматично збирають метрики ефективності та забезпечують їхню кореляцію з реальними інцидентами, такими як успішні або невиявлені фішингові атаки. Це створює базу для ітеративного навчання моделей і впровадження змін на основі поточного середовища кіберзагроз.

Оптимізація інтегрованих рішень вимагає дотримання циклу безперервного вдосконалення, що включає повторювані етапи тестування, аналізу та налаштування. Одним із важливих напрямків є ідентифікація системних проблем, які можуть виникати через неповноту або надлишкову залежність моделі від певних ознак. Наприклад, якщо система виявляє тенденцію надто часто класифікувати конкретні типи електронних листів як фішингові через спільні ознаки, це може свідчити про необхідність оптимізації алгоритму або додаткової аугментації даних. Також значну роль відіграє масштабування моделі, що включає її налаштування для роботи з більшими обсягами даних чи більш складними сценаріями атак, які раніше не були представлені в навчальних вибірках.

Додатковий фактор, що впливає на оцінку результативності, — це інтеграція з іншими системами безпеки. Це дозволяє автоматизувати реагування на інциденти, підвищуючи ефективність виявлення загроз і скорочуючи час на реагування. Водночас важливо уникати дублювання функціоналу або конфліктів між інструментами, які можуть призвести до хибної ідентифікації атак або затримки в їх обробці. Використання централізованої аналітики, що базується на потоках даних із кількох джерел, дає можливість порівнювати продуктивність різних систем і виявляти зони, що потребують доопрацювання.

У третьому розділі сформовано методичні рекомендації для впровадження системи виявлення фішингових атак у корпоративне середовище. Описано процес оцінки ризиків, підготовки інфраструктури та інтеграції інструментів ШІ в існуючі системи безпеки. Запропоновано схеми навчання персоналу, механізми моніторингу та постійного вдосконалення системи. Особливу увагу приділено аналізу ефективності впровадження, а також розробці стратегій реагування на нові типи атак. Ці рекомендації спрямовані на підвищення стійкості організацій до кіберзагроз і забезпечення ефективної роботи системи в довгостроковій перспективі.

ВИСНОВКИ

Розроблена методика для виявлення фішингових атак з використанням штучного інтелекту показала значний потенціал для інтеграції в сучасні системи кіберзахисту. Особливістю цього підходу є врахування специфіки обробки даних у реальному часі, адаптивності до нових загроз і можливості масштабування відповідно до потреб організації. Використання ШІ дозволяє значно підвищити ефективність виявлення фішингових атак за рахунок автоматизації аналізу даних, що надходять, і визначення аномалій на основі виявлених шаблонів. Наприклад, впровадження систем на базі глибокого навчання, таких як нейронні мережі або алгоритми класифікації, сприяє ідентифікації складних і прихованих загроз, які традиційні методи не здатні розпізнати.

У рамках запропонованої методики було визначено ключові кроки для ефективного впровадження, включаючи попередню підготовку інфраструктури організації. Це охоплює аналіз поточних ресурсів, визначення технічних вимог і створення умов для безперебійної роботи нових рішень. Наприклад, організації необхідно забезпечити достатню обчислювальну потужність для роботи алгоритмів, інтегрувати їх у наявну архітектуру безпеки й налаштувати моніторинг для відстеження результатів роботи моделей у реальному часі. Особлива увага приділялася забезпеченню відповідності вимогам конфіденційності та захисту даних, щоб впровадження не суперечило існуючим стандартам і регуляторним нормам.

Ключовим аспектом методики є можливість безшовної інтеграції з іншими системами безпеки, такими як антивірусні програми чи системи виявлення вторгнень. Це дозволяє побудувати єдину екосистему захисту, де кожен компонент підсилює інші, створюючи багаторівневий бар'єр для кіберзагроз. Використання API для обміну даними між системами та автоматизація реагування на інциденти

дозволяють значно скоротити час між виявленням фішингової атаки та її нейтралізацією, підвищуючи загальну стійкість організації до атак.

У роботі досліджено методи машинного навчання, які використовуються для аналізу фішингових сайтів, листів, та інших видів шкідливого контенту. Було систематизовано джерела даних для навчання моделей, включаючи відкриті бази даних, API та спеціалізовані сервіси, такі як PhishTank, OpenPhish, VirusTotal та інші. Увага приділялася передобробці даних, яка є ключовим етапом для забезпечення точності та стабільності роботи моделей. Зокрема, детально розглянуто процес очищення, нормалізації та трансформації даних для побудови відповідних форматів для навчання.

Виявлено, що інтеграція кількох методів — аналізу URL, семантичного аналізу тексту та перевірки метаданих — дозволяє досягти кращих результатів у порівнянні з використанням окремих методів. Це підтверджує важливість комплексного підходу до виявлення фішингу.

Також, важливим стало розроблення рекомендацій для інтеграції системи на основі штучного інтелекту в існуючі інфраструктури організацій. Було визначено стратегії поєднання нових методів із традиційними засобами кібербезпеки, такими як системи захисту електронної пошти, міжмережеві екрани та SIEM-системи. Окрема увага приділялася навчанню персоналу організацій для забезпечення їхньої здатності взаємодіяти з новими технологіями. Це включало рекомендації щодо проведення тренінгів, створення документації та побудови сценаріїв реагування на інциденти.

Під час аналізу слабких місць у сучасних стандартах кібербезпеки, таких як ISO/IEC, NIST та рекомендацій ENISA, виявлено низку обмежень. Традиційні підходи часто фокусуються на статичних методах і регламентованих процедурах, що не дозволяють оперативно реагувати на нові види атак. У цьому контексті запропонована модель забезпечує більшу гнучкість та адаптивність, що робить її придатною для використання в реальних умовах.

Одним із ключових результатів роботи стала розробка методичних рекомендацій щодо впровадження та оцінки ефективності систем виявлення фішингу. Було запропоновано підхід до оптимізації моделей на основі аналізу результатів тестування. Зокрема, виявлення та аналіз помилок класифікації допомогли вдосконалити алгоритми для зменшення кількості хибно-позитивних і хибно-негативних результатів. На основі цих вдосконалень було розроблено універсальні принципи для підвищення ефективності моделей.

Отримані результати мають як наукове, так і практичне значення. У науковому плані запропонований метод розширює існуючі знання про застосування штучного інтелекту в кібербезпеці та забезпечує основу для подальших досліджень. Практичне значення полягає в можливості реального впровадження розробленої системи для виявлення фішингових атак у компаніях, що прагнуть мінімізувати ризики, пов'язані з кіберзагрозами.

Рекомендації щодо подальшого розвитку включають вдосконалення методів обробки великих даних, розширення джерел навчальних даних та створення платформ для постійного оновлення моделей. Це дозволить підвищити ефективність системи в умовах динамічних загроз.

1. У роботі виконано аналітичний огляд методів виявлення фішингових атак. Досліджено методи, які використовуються для аналізу фішингових сайтів, листів, та інших видів шкідливого контенту з використанням штучного інтелекту.

2. Розроблена модель виявлення фішингових атак з використанням штучного інтелекту на підприємствах. Що дозволяє підвищити ефективність системи кібербезпеки в умовах динамічних загроз

3. Розроблені рекомендації щодо впровадження системи виявлення фішингу з використанням штучного інтелекту в існуючі системи безпеки організації.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Comparative Analysis of Blacklist-Based and AI-Based Detection Methods // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
2. A Systematic Review on Deep-Learning-Based Phishing Email Detection // MDPI. – 2023. – URL: https://www.mdpi.com (дата звернення: 11.09.2024).
3. Random Forest and Ensemble Methods for Phishing URL Detection // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
4. Phishing Email Detection Based on Natural Language Processing Techniques // MDPI. – URL: https://www.mdpi.com (дата звернення: 15.11.2024).
5. Голубєва Л. А. Інформаційна безпека у кіберпросторі: сучасні виклики та загрози // Наукові записки НаУКМА. – URL: https://ekmair.ukma.edu.ua (дата звернення: 13.10.2024).
6. Phishing Email Detection Using Machine Learning: A Critical Review // IEEE Xplore. – URL: https://ieeexplore.ieee.org (дата звернення: 15.11.2024).
7. Phishing Email Detection Using Machine Learning: A Critical Review // IEEE Xplore. – 2024. – URL: https://ieeexplore.ieee.org (дата звернення: 05.09.2024).
8. Моделі виявлення фішингових атак: аналітичний огляд // Вісник Харківського національного університету радіоелектроніки. – URL: https://journals.hnure.edu.ua (дата звернення: 06.10.2024).
9. Phishing Email Detection Based on Natural Language Processing Techniques // MDPI. – URL: https://www.mdpi.com (дата звернення: 15.11.2024).
10. Штучний інтелект у кібербезпеці: сучасний стан і перспективи розвитку // Журнал "Кібернетика і системний аналіз". – URL: https://cyberleninka.ru (дата звернення: 22.10.2024).

11. The Role of Neural Networks in Cyber Threat Mitigation // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 15.11.2024).
12. Challenges in Training AI Models for Cybersecurity Applications // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 15.11.2024).
13. Comparative Analysis of Blacklist-Based and AI-Based Detection Methods // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
14. Deep Learning Techniques for Cybersecurity: Applications and Challenges // Elsevier. – URL: https://www.elsevier.com (дата звернення: 05.11.2024).
15. Real-Time Phishing Detection Using Hybrid Models // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
16. Фішинг як актуальна загроза інформаційній безпеці // Науково-технічний журнал "Інформаційна безпека". – URL: https://www.isjournal.org.ua (дата звернення: 09.09.2024).
17. Random Forest and Ensemble Methods for Phishing URL Detection // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
18. Advanced URL-Based Phishing Detection Using CNNs // Elsevier. – URL: https://www.elsevier.com (дата звернення: 15.11.2024).
19. Random Forest and Ensemble Methods for Phishing URL Detection // Springer. – URL: https://link.springer.com (дата звернення: 25.09.2024).
20. AI-Driven Solutions for Cyber Threat Intelligence // ACM Digital Library. – URL: https://dl.acm.org (дата звернення: 14.10.2024).
21. Applications of LSTM Networks in Real-Time Phishing Detection // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 15.11.2024).
22. Feature Engineering for Phishing Website Detection // Springer. – URL: https://link.springer.com (дата звернення: 14.10.2024).
23. The Role of Neural Networks in Cyber Threat Mitigation // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 13.10.2024).

24. Кіберзахист: роль машинного навчання у попередженні атак // Вісник Одеського національного університету. – URL: https://visnyk.onu.edu.ua (дата звернення: 22.09.2024).
25. Технології виявлення загроз на основі поведінкового аналізу // Вісник Київського політехнічного інституту. – URL: https://journal.kpi.ua (дата звернення: 07.09.2024).
26. Scalable Machine Learning Models for Phishing Detection // Elsevier. – URL: https://www.elsevier.com (дата звернення: 30.10.2024).
27. Машинне навчання та нейронні мережі у виявленні фішингових атак // Журнал "Сучасні інформаційні системи". – URL: https://ais.khpi.edu.ua (дата звернення: 02.11.2024).
28. Applications of LSTM Networks in Real-Time Phishing Detection // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 10.10.2024).
29. Evaluating the Effectiveness of SVM Models in Phishing Detection // ACM Digital Library. – URL: https://dl.acm.org (дата звернення: 17.10.2024).
30. Інструменти аналізу шкідливих URL-адрес: огляд та оцінка // Журнал "Технології програмування". – URL: https://programmingtech.ua (дата звернення: 21.09.2024).
31. Hybrid Approaches to Cybersecurity with AI and Human Expertise // Taylor & Francis. – URL: https://www.tandfonline.com (дата звернення: 18.10.2024).
32. Benchmarking Phishing Detection Algorithms // Springer. – URL: https://link.springer.com (дата звернення: 13.10.2024).
33. Context-Aware Machine Learning Models for Phishing // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 15.11.2024).
34. Експертна система для аналізу кіберзагроз // Науковий журнал "Інформаційні технології і комп’ютерна інженерія". – URL: https://journals.nure.ua (дата звернення: 27.10.2024).

35. Automated Feature Extraction for Cybersecurity Applications // MDPI. – URL: https://www.mdpi.com (дата звернення: 16.09.2024).
36. Перспективи використання штучного інтелекту в боротьбі з фішингом // Журнал "Кібернетика України". – URL: https://cybernetics.org.ua (дата звернення: 15.11.2024).
37. Comparative Analysis of Blacklist-Based and AI-Based Detection Methods // Springer. – URL: https://link.springer.com (дата звернення: 13.11.2024).
38. Real-Time Phishing URL Detection Using Ensemble Methods // ACM Digital Library. – URL: https://dl.acm.org (дата звернення: 13.11.2024).
39. The Evolution of Phishing Detection Techniques with AI // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 13.11.2024).
40. Challenges in Training AI Models for Cybersecurity Applications // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 15.11.2024).
41. Adaptive Learning in Phishing Detection Systems // Elsevier. – URL: https://www.elsevier.com (дата звернення: 11.11.2024).
42. Metrics for Evaluating AI-Based Security Systems // Springer. – URL: https://link.springer.com (дата звернення: 11.11.2024).
43. Detection of Phishing Websites Using Machine Learning Techniques // Hindawi. – URL: https://www.hindawi.com (дата звернення: 10.11.2024).
44. The Future of AI in Cybersecurity // Harvard Business Review. – URL: https://hbr.org (дата звернення: 10.11.2024).
45. Advanced Neural Network Models for Phishing Detection // IEEE Transactions on Cybernetics. – URL: https://ieeexplore.ieee.org (дата звернення: 12.11.2024).
46. Comprehensive Review on Cybersecurity Applications of AI // Springer. – URL: https://link.springer.com (дата звернення: 14.11.2024).
47. Challenges in Scaling AI Models for Security Applications // Elsevier. – URL: https://www.elsevier.com (дата звернення: 10.11.2024).

48. Phishing Email Detection Based on Natural Language Processing Techniques // MDPI. – URL: https://www.mdpi.com (дата звернення: 11.11.2024).
49. AI-Powered Phishing Mitigation Techniques // Taylor & Francis. – URL: https://www.tandfonline.com (дата звернення: 12.11.2024).
50. Deep Reinforcement Learning for Cyber Threat Prevention // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 14.11.2024).
51. Integrating AI with Human Expertise for Phishing Detection // Elsevier. – URL: https://www.elsevier.com (дата звернення: 14.11.2024).
52. Real-Time Phishing Detection Using Hybrid Models // Springer. – URL: https://link.springer.com (дата звернення: 14.11.2024).
53. Efficient Feature Selection for Machine Learning-Based Phishing Detection // Wiley Online Library. – URL: https://onlinelibrary.wiley.com (дата звернення: 14.11.2024).
54. Phishing URL Detection via Transfer Learning // ACM Digital Library. – URL: https://dl.acm.org (дата звернення: 15.11.2024).
55. The Role of Ensemble Learning in Phishing Detection // Hindawi. – URL: https://www.hindawi.com (дата звернення: 15.11.2024).
56. Comparative Study on AI Models for Cyber Threat Analysis // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 15.11.2024).
57. Evaluating Cybersecurity Metrics for AI Systems // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
58. Behavioral Analysis in AI-Based Phishing Detection Systems // Taylor & Francis. – URL: https://www.tandfonline.com (дата звернення: 15.11.2024).
59. Low-Cost Solutions for Phishing Detection Using AI // MDPI. – URL: https://www.mdpi.com (дата звернення: 15.11.2024).
60. Optimization Algorithms for AI in Cybersecurity // Elsevier. – URL: https://www.elsevier.com (дата звернення: 15.11.2024).

61. AI Techniques for Securing Digital Transactions // IEEE. – URL: https://ieeexplore.ieee.org (дата звернення: 15.11.2024).
62. Comprehensive Metrics for Evaluating Phishing Detection Tools // Springer. – URL: https://link.springer.com (дата звернення: 15.11.2024).
63. Detecting Phishing Websites Using Machine Learning Techniques // PLOS ONE. – URL: https://journals.plos.org (дата звернення: 15.11.2024).
64. AI Models in the Detection of Phishing Links // Hindawi. – URL: https://www.hindawi.com (дата звернення: 15.11.2024).
65. Reinforcement Learning Approaches for Phishing Mitigation // IEEE Access. – URL: https://ieeexplore.ieee.org (дата звернення: 15.11.2024).
66. ISO/IEC 27001:2013. Information technology – Security techniques – Information security management systems – Requirements // ISO. – URL: https://www.iso.org/standard/54534.html (дата звернення: 15.11.2024).
67. ISO/IEC 27002:2022. Information security, cybersecurity and privacy protection – Information security controls // ISO. – URL: https://www.iso.org/standard/74978.html (дата звернення: 15.11.2024).
68. ISO/IEC 27005:2018. Information technology – Security techniques – Information security risk management // ISO. – URL: https://www.iso.org/standard/75281.html (дата звернення: 15.11.2024).
69. NIST Cybersecurity Framework (CSF) 1.1. A Framework for Improving Critical Infrastructure Cybersecurity // NIST. – URL: https://www.nist.gov/cyberframework (дата звернення: 15.11.2024).