УДК 504.75

ОГЛЯД МЕТОДІВ КЛАСИФІКАЦІЇ ЕЛЕКТРОННИХ ЛИСТІВ

А. В. Гаврилова[1], Г. О. Яйлимова[1]

[1] Навчально-науковий Фізико-технічний інститут

Анотація

У роботі розглянуто методи виявлення фішингових електронних листів на основі сучасних технологій глибокого навчання. Особливу увагу приділено архітектурам нейронних мереж, зокрема CNN, LSTM і трансформерам (BERT), що демонструють високу точність при класифікації фішингових, спамових і безпечних листів. Також досліджено ефективність традиційних методів, таких як сигнатурний аналіз, чорні списки, правила фільтрації, а також алгоритми машинного навчання. Розкрито залежність точності класифікації від якості датасетів і важливість постійного оновлення моделей.

Ключові слова: фішинг, електронна пошта, глибоке навчання, нейронні мережі, сигнатурний аналіз, класифікація, BERT

Вступ

Фішинг є одним із найпоширеніших методів соціальної інженерії, який широко використовується зловмисниками для викрадення персональних або корпоративних даних [1]. Зазвичай атаки реалізуються через електронну пошту, оскільки це масовий і відносно незахищений канал комунікації. У роботі розглядаються підходи до виявлення таких листів, зокрема за допомогою класичних методів (правила, списки) [2], сигнатурного аналізу [3] та новітніх технологій машинного та глибокого навчання [4, 5].

1. Методи виявлення фішингових листів

1.1. Сигнатурні методи

Сигнатурні методи є одними з найдавніших підходів до виявлення фішингових атак і базуються на пошуку заздалегідь визначених шаблонів або ознак, характерних для шкідливих повідомлень [3]. Ці ознаки – своєрідні «відбитки» фішингових листів – включають типові фрази, доменні шаблони та елементи HTML-структури. Наприклад, системи можуть реагувати на наявність таких фраз у тексті повідомлення, як «verify your account», «update payment details» або «click here to unlock», що часто використовуються зловмисниками для спонукання користувача до дії.

Ознакою фішингу можуть бути підозрілі доменні імена, що імітують легітимні адреси, але ведуть на шахрайські сайти (наприклад, security-bank.com.verify-login.au). Також варто звертати увагу на HTML-структуру листа – приховані редиректи чи шкідливий JavaScript можуть виконувати несанкціоновані дії.

Сигнатурний підхід вирізняється простотою, швидкодією та низькими вимогами до ресурсів, що дозволяє ефективно фільтрувати велику кількість пошти [2]. Водночас він не здатен виявляти нові або модифіковані атаки, що знижує його ефективність в умовах постійної еволюції фішингових методів [5].

Для покращення цих підходів сучасні системи часто комбінують сигнатури з регулярними виразами, евристичними правилами та fuzzy matching [3, 6]. Попри це, самі по собі сигнатурні методи не здатні забезпечити належний рівень захисту, тому на практиці використовуються переважно як складова багаторівневих систем фільтрації.

1.2. Класичні евристичні методи

Класичні евристичні методи виявлення фішингових повідомлень базуються на заздалегідь визначених правилах, логіці поведінки системи й емпіричних спостереженнях за шкідливою активністю. Одним із найбільш уживаних механізмів є застосування чорних списків (blacklists), до яких включаються IP-адреси, доменні імена або ключові слова, що були помічені у фішинговій активності [1]. Якщо новий електронний лист містить хоча б один елемент із такого списку, система може автоматично заблокувати його або помістити в карантин. Подібні списки формуються на основі попередніх інцидентів і можуть бути як локальними (в межах організації), так і глобальними (формуються спільнотами або спеціалізованими сервісами).

Ще одним поширеним засобом є фільтрація на основі правил (rule-based filters), які задаються вручну адміністраторами або системними аналітиками. Наприклад, може бути встановлено правило, що всі листи, які надходять із зовнішніх доменів і містять виконувані файли (вкладення з розширенням .exe), автоматично переміщуються до карантину або по-

значаються як потенційно небезпечні. Такі правила дозволяють швидко реагувати на відомі загрози й адаптувати поведінку системи до специфіки організації [4].

Крім того, аналізується структура листа: система може перевіряти відповідність між заголовками листа, виявляти нетипову або підозрілу структуру HTML, наявність прихованих елементів або навіть повну відсутність видимого тексту в тілі повідомлення. Наприклад, деякі фішингові листи можуть мати справжній заголовок, але містити лише одне посилання в тілі без пояснень – така аномалія може бути ознакою загрози.

Незважаючи на їхню ефективність у певних випадках, класичні евристичні методи мають суттєві обмеження. Основною проблемою є необхідність постійного оновлення правил і списків – зловмисники швидко змінюють свої підходи, обходячи існуючі фільтри. Також ці методи часто не можуть впоратися зі складними або новими сценаріями атак, які не підпадають під жодне з наявних правил. У результаті система або пропускає загрозу, або помилково класифікує легітимний лист як небезпечний, що може знижувати загальну ефективність захисту та викликати недовіру з боку користувачів.

1.3. Алгоритми машинного навчання

Застосування алгоритмів машинного навчання дозволяє моделювати статистичні закономірності у фішингових і спамових повідомленнях, базуючись на аналізі вмісту листів і супутніх метаданих. Серед класичних методів одним із найпоширеніших є наївний байєсівський класифікатор (Naive Bayes), який використовує ймовірнісну модель для визначення належності листа до певного класу [1]. Алгоритм оцінює, наскільки ймовірно поява конкретних слів у тексті в межах кожної з категорій (наприклад, «фішинг», «спам», «безпечний»), і обирає найімовірнішу. Цей підхід є дуже швидким і добре працює для текстових задач, проте має лінійну природу й не враховує складні залежності між словами.

Іншим популярним методом є алгоритм опорних векторів (Support Vector Machines, SVM), який шукає гіперплощину в багатовимірному просторі ознак, що максимально відділяє приклади різних класів. Ефективність SVM залежить від якості обраних ознак – чим краще вони відображають відмінності між класами, тим точнішим буде розділення [4]. У задачах класифікації тексту SVM часто демонструє високу точність, особливо при використанні векторного представлення на основі TF-IDF.

Ще один відомий підхід – дерева рішень (Decision Trees), які моделюють процес класифікації як послідовність логічних перевірок [4]. Наприклад, дерево може спочатку перевірити, чи містить лист вкладення, далі – чи є в тексті слово «urgent», і на основі відповідей обрати клас повідомлення. Цей метод добре інтерпретується, оскільки кожен крок логічно пояснюваний, однак він схильний до перенавчання, особливо на малих вибірках або за відсутності регуляризації.

Класичні алгоритми машинного навчання прості й зрозумілі, але потребують ручного формування ознак, що обмежує їх здатність узагальнювати та ефективно виявляти нові види фішингових атак.

1.4. Глибоке навчання (Deep Learning)

Глибоке навчання відкриває нові можливості для автоматичної класифікації електронних листів завдяки здатності самостійно виявляти складні шаблони та залежності у вхідних даних без необхідності ручного виділення ознак [5]. Однією з базових архітектур у цьому напрямі є багатошаровий перцептрон (MLP), що працює з векторизованими текстовими представленнями, наприклад, на основі one-hot- або TF-IDF-кодування. Хоча MLP демонструє непогані результати, він не враховує порядок слів, що може бути критичним у задачах розпізнавання фішингу.

Значно більших результатів досягають згорткові нейронні мережі (Convolutional Neural Networks, CNN), які добре виявляють локальні текстові шаблони, типові для фішингових листів, наприклад фрази на кшталт «Click here to...», нетипові структури доменних імен чи HTML-теги [6]. CNN-мережі ефективно працюють із символьними та словесними представленнями тексту й здатні виявляти навіть неявні ознаки загроз.

Ще однією потужною архітектурою є довгі короткочасні пам'яті (Long Short-Term Memory, LSTM) – тип рекурентних нейронних мереж, що дозволяє враховувати контекст не лише на рівні поточних слів, а й у довгих послідовностях. Це дає змогу врахувати семантичні зв'язки між різними частинами повідомлення, навіть якщо вони розділені десятками слів або речень [5].

Нарешті, сучасні трансформерні моделі, зокрема BERT (Bidirectional Encoder Representations from Transformers), забезпечують найвищу точність серед доступних архітектур [5]. Завдяки механізму самоуваги (self-attention), BERT здатна враховувати як локальний, так і глобальний контекст у тексті, аналізуючи зв'язки між усіма словами одночасно. Це дозволяє виявляти фішингові інтенції навіть у граматично правильних і стилістично витончених листах, що не мають очевидних «тригерних» фраз.

За наявності великих і якісних навчальних датасетів глибокі моделі демонструють точність класифікації, що може сягати 98–99% [4, 5]. Саме завдяки цим показникам глибоке навчання все активніше використовується в сучасних системах фільтрації електронної пошти, витісняючи класичні алгоритми.

1.5. Оцінка моделей і практичні аспекти

Оцінювання ефективності моделей класифікації електронних листів є ключовим етапом у процесі їхньої розробки та впровадження. Щоб визначити, яка модель найкраще справляється з виявленням фішингових повідомлень, застосовуються кілька загальновживаних метрик [1, 4]. Однією з основних

є загальна точність (accuracy), що відображає частку правильно класифікованих прикладів серед усіх. Проте ця метрика може бути недостатньо інформативною в умовах незбалансованих датасетів, де, наприклад, більшість листів – безпечні.

У зв'язку з цим особливу увагу приділяють precision – точності класифікації позитивного класу (тобто фішингових або спамових листів), яка показує, яку частку з усіх виявлених як «загроза» повідомлень справді становлять шкідливі. Інша важлива метрика – recall (чутливість або повнота), що характеризує здатність моделі виявляти всі наявні загрози у вибірці. Для збалансованої оцінки між цими двома метриками використовують F1-міру (F1-score), яка є гармонічним середнім між precision і recall і особливо корисна в задачах із нерівномірним розподілом класів [4].

Крім того, часто аналізуються показники помилок першого та другого роду – false positive rate (FPR) і false negative rate (FNR) відповідно. Високий показник FPR свідчить про те, що модель надмірно агресивна й класифікує занадто багато легітимних листів як фішингові, що може викликати обурення користувачів. Натомість високий FNR вказує на те, що система пропускає справжні загрози, що є критично небезпечним у корпоративному середовищі.

Окрім формальних метрик, у реальних умовах важливими є також практичні аспекти: швидкодія моделі (час класифікації одного листа), здатність до адаптації до нових типів загроз (наприклад, через донавчання), а також ресурсоємність – тобто обсяг оперативної пам'яті та обчислювальної потужності, необхідних для роботи системи [3]. Саме поєднання високих оціночних показників із задовільними експлуатаційними характеристиками визначає придатність моделі до реального впровадження.

1.6. Комплексні системи

Розробляються гібридні моделі (CNN+LSTM, BERT+MLP), що поєднують переваги кількох підходів [5, 6]. Також застосовуються багатомодульні фреймворки з аналізом тексту, URL-адрес, вкладень і заголовків листів [2].

2. Огляд існуючих продуктів

На ринку представлено низку рішень для автоматичного виявлення фішингових листів. Зокрема, продукти Microsoft Defender for Office 365 [7] та Google Workspace Security [8] використовують перевірку заголовків, доменів (SPF, DKIM, DMARC), сигнатурне сканування та алгоритми машинного навчання.

Системи на кшталт Proofpoint [9] і Mimecast [10] орієнтовані на виявлення цільових атак (spear phishing, BEC) і враховують контекст спілкування та поведінкові шаблони.

Серед рішень із відкритим кодом вирізняються SpamAssassin [11] і MailScanner [12], які поєднують евристичні правила, сигнатури та антивірусну перевірку. База фішингових доменів PhishTank [13] використовується як допоміжне джерело даних.

Окрему категорію становлять інтелектуальні системи, як-от IRONSCALES [14] чи Cofense [15], які застосовують глибоке навчання, колективну інтелектуальну обробку загроз і адаптивну фільтрацію.

Усі ці продукти поєднують кілька підходів, проте залишаються залежними від оновлень даних і налаштування під конкретне середовище. Це підтверджує актуальність розробки локальних моделей, адаптованих до специфіки організації.

3. Експериментальна частина

Для оцінки ефективності методів виявлення фішингових листів була реалізована система класифікації електронних повідомлень з використанням глибокої нейронної мережі. Основною метою експерименту було дослідити можливість автоматичного розпізнавання фішингових листів серед реальних поштових даних.

Було зібрано датасет електронних листів у форматі .eml, який містив три класи: фішинг, спам і безпечні (легітимні) листи. Кількість даних була обмеженою, однак структура була збалансована: близько 50% листів – спам, 35.6% – фішинг, 14.4% – безпечні. Листи мали різне походження: як реальні корпоративні приклади, так і шаблонні повідомлення з відкритих ресурсів.

Перед навчанням було здійснено парсинг файлів: виділено ключові заголовки (From, To, Subject, Reply-To, Authentication-Results), а також тіло листа. Усі текстові дані були об'єднані в єдиний рядок для подальшої векторизації. Для обробки використовувались стандартні інструменти Python – зокрема, модулі `email`, `os`, `sklearn`, `tensorflow`.

3.1. Технічна реалізація моделі класифікації

Система реалізована в середовищі Python із використанням бібліотеки TensorFlow. Основні етапи виглядали так:

- **Векторизація тексту:** застосовано шар `TextVectorization`, що трансформує текстові повідомлення в послідовності чисел із обмеженням на кількість токенів (до 20 000) та довжину (до 1 000 символів).
- **Побудова моделі:** створено модель нейронної мережі типу Embedding + GlobalAveragePooling + Dense:
  - Embedding(20000, 128) – представлення слів у 128-вимірному просторі,
  - GlobalAveragePooling1D – агрегація векторів слів у фіксовану ознаку,
  - Dense(64, relu) – прихований шар,
  - Dense(3, softmax) – класифікація на три класи.
- **Процес навчання:** модель навчалась протягом 100 епох, при цьому застосовувались функція втрат `sparse_categorical_crossentropy` та оптимізатор `Adam`. Навчальна та тестова вибірки були розділені в пропорції 80/20.
- **Оцінка:** точність перевірялась на тестових даних, а також за метриками precision, recall,

F1-score. Було сформовано звіт із використанням `classification_report()` із бібліотеки `sklearn.metrics`.

3.2. Результати експерименту

Загальна точність класифікації становила близько 71%. Найкращі результати досягнуті для класу спам – recall понад 90% та F1-score 0.75. Для фішингових листів recall склав 56%, precision – 81%, що свідчить про схильність моделі до консервативної класифікації (більше false negatives). Безпечні листи мали високу precision (1.00), але низький recall (50%), тобто модель рідко помилялася, коли класифікувала лист як безпечний, але часто не змогла розпізнати шкідливий.

Причини помірного рівня якості класифікації, досягнутого в експерименті, пояснюються кількома важливими факторами. По-перше, модель навчалась на обмеженому обсязі прикладів фішингових листів. Нестача даних цього класу негативно впливає на здатність моделі узагальнювати та розпізнавати нові або нетипові атаки, оскільки вона не бачила достатньої кількості варіацій таких повідомлень у тренувальній вибірці.

По-друге, складні фішингові повідомлення часто стилістично та структурно дуже подібні до легітимних листів. Вони можуть бути граматично правильними, мати корпоративний стиль оформлення та містити реалістичні заклики до дії. Така текстова схожість значно ускладнює завдання класифікації, оскільки навіть людина іноді не одразу здатна відрізнити фішинг від звичайної службової кореспонденції.

По-третє, модель використовувала лише текстову складову листа без урахування додаткових ознак, таких як аналіз вкладень, URL-посилань, структури HTML або вбудованих зображень. Ці елементи часто містять критичну інформацію для виявлення фішингової активності, тому їхня відсутність значно обмежує глибину аналізу. Врахування таких додаткових параметрів у майбутніх реалізаціях могло б істотно підвищити точність класифікації.

Подальші плани

Планується:

- розширити датасет, зокрема за рахунок реальних листів користувачів і синтетичних зразків;
- протестувати інші архітектури (наприклад, LSTM, BiLSTM, BERT);
- інтегрувати структурні та поведінкові ознаки (довжина листа, частка посилань, аналіз вкладень);
- реалізувати attention-механізми для глибшого аналізу контексту листа.

Висновки

Глибоке навчання забезпечує високу точність виявлення фішингових листів. Найкращі результати демонструють гібридні моделі (CNN+LSTM+attention, BERT). Важливими залишаються якість і актуальність датасетів, баланс класів, швидкодія моделі. У майбутньому – розширення даних, застосування attention-механізмів, візуальний аналіз вкладень.

Перелік використаних джерел

1. Email Spam: A Comprehensive Review of Optimize Detection Methods / Z. Alshingiti [та ін.] // IEEE Access. — 2024.
2. *Smadi S., Aslam N.* Detection of Phishing Emails using Neural Network Trained with RL // Proceedings of the SAI Intelligent Systems Conference. — 2018.
3. D-Fence: A Comprehensive Phishing Email Detection Framework / J. Lee [та ін.] // Security and Communication Networks (SCN). — 2020.
4. Email Spam Classification Based on Deep Learning Methods: A Review / E. Tusher [та ін.] // International Journal of Computer Science and Management (IJCSM). — 2025.
5. A Systematic Review on Deep-Learning-Based Phishing Email Detection / K. Thakur [та ін.] // Electronics. — 2023.
6. *Manaswini P.* Themis: Multi-level Email Phishing Detection with Attention // IEEE Access. — 2019.
7. *Microsoft.* Microsoft Defender for Office 365. — 2024. — Accessed: 2025-05-11. https://learn.microsoft.com/en-us/microsoft-365/security/office-365-security/defender-for-office-365?view=o365-worldwide.
8. *Google.* Google Workspace Security. — 2024. — Accessed: 2025-05-11. https://workspace.google.com/security/.
9. *Proofpoint.* Proofpoint Email Protection. — 2024. — Accessed: 2025-05-11. https://www.proofpoint.com/us/products/email-protection.
10. *Mimecast.* Mimecast Email Security. — 2024. — Accessed: 2025-05-11. https://www.mimecast.com/products/email-security-with-targeted-threat-protection/.
11. *Apache Software Foundation.* Apache SpamAssassin Project. — 2024. — Accessed: 2025-05-11. https://spamassassin.apache.org/.
12. *MailScanner Team.* MailScanner Email Security System. — 2024. — Accessed: 2025-05-11. https://mailscanner.info/.
13. *PhishTank.* PhishTank Database. — 2024. — Accessed: 2025-05-11. https://phishtank.org/.
14. *IRONSCALES.* IRONSCALES Email Security. — 2024. — Accessed: 2025-05-11. https://ironscales.com/.
15. *Cofense.* Cofense Phishing Defense. — 2024. — Accessed: 2025-05-11. https://cofense.com/.