Стійке виявлення фішингових URL-адрес за умов обфускаційних атак

Robust Detection of Phishing URLs Under Obfuscation Attacks

Ростислав Фещенко

Corresponding author: курсант, e-mail feshenkorostislav@gmail.com, ORCID ID: https://orcid.org/0009-0006-4025-455X

Катерина Вітвіцька

курсантка, e-mail: vitvitskakateryna320@gmail.com, ORCID ID: https://orcid.org/0009-0005-9932-1289

Василь Лучик

доктор економічних наук, професор, професор кафедри, e-mail: feshenkorostislav@gmail.com, ORCID ID: https://orcid.org/0000-0001-6007-910X

Rostyslav Feshchenko

Corresponding author: Cadet, e-mail feshenkorostislav@gmail.com, ORCID ID: https://orcid.org/0009-0006-4025-455X

Kateryna Vitvitska

Cadet, e-mail: vitvitskakateryna320@gmail.com, ORCID ID: https://orcid.org/0009-0005-9932-1289

Vasyl Luchyk

Dr of Economic Sciences, Professor, Professor of the Department, e-mail: feshenkorostislav@gmail.com, ORCID ID: https://orcid.org/0000-0001-6007-910X

Харківський національний університет внутрішніх справ, м. Кам'янець-Подільський, Україна

Kharkiv National University of Internal Affairs, Kamianets-Podilskyi, Ukraine

Received: August 08, 2026 | Revised: August 26, 2026 | Accepted: August 31, 2026

УДК 004.056.53:004.85

DOI: https://doi.org/10.33445/sds.2026.16.4.10

Мета роботи. На основі підходу з використанням двох представлень розроблено та експериментально перевірено протокол лексичного виявлення фішингових вебадрес, стійкий до синтаксичних перетворень, що зберігають цільовий вузол у межах прийнятої моделі загроз.

Методи дослідження. Дослідження має емпіричний характер. Використано відкритий набір фішингових і легітимних веб-адрес Prasad і Chandra. Після очищення сформовано вибірку з 235 366 унікальних записів. Навчальні, валідаційні та тестові вибірки розділено за зареєстрованими доменами. Проведено порівняння символьної базової моделі, моделі з двома представленнями та запропонованого варіанта з навчальною аугментацією. Модель загроз охоплювала зміну регістру вузла, додавання явного типового порту, відсоткове кодування, додавання фрагмента, кінцевої крапки домену та префікса з інформацією про користувача, вилучуваного сегмента шляху, а також композиції зазначених перетворень. Якість моделей оцінено за точністю, повнотою, F1-мірою, площею під ROC-кривою та часткою хибнопозитивних спрацювань.

Результати дослідження. На неперетинній за доменами тестовій вибірці без перетворень базова та аугментована моделі досягли значень F1-міри 0,9962 і 0,9960 відповідно; у сценарії transfer-тестування, адаптивного до базової моделі, — 0,9943 і 0,9961 відповідно.

Теоретична цінність дослідження. Результати дослідження розширюють методичні підходи до оцінювання лексичних детекторів фішингу та обґрунтовують доцільність спільного використання необроблених і канонізованих представлень веб-адрес, а також інваріантного навчання в межах формалізованої моделі обфускаційного порушника.

Практична цінність дослідження. Запропонований протокол може застосовуватися для локального попереднього фільтрування веб-адрес у шлюзах електронної пошти, розширеннях браузерів і системах моніторингу без виконання мережевих запитів, а також для підтримки управлінських рішень щодо політик безпекового фільтрування веб-адрес.

Оригінальність. Встановлено, що тестування лише на незмінених веб-адресах є недостатнім для висновку про стійкість моделі в межах прийнятої моделі загроз. Показано, що навчальна аугментація підвищує інваріантність до досліджуваного класу семантично еквівалентних обфускацій без практично значущого погіршення якості на неперетворених даних.

Обмеження дослідження. Перед упровадженням запропонованого підходу в реальних системах необхідне додаткове оцінювання часової стійкості, міжнабірної узагальнюваності та залежності результатів від особливостей браузерної обробки веб-адрес.

Тип статті. Емпірична.

Purpose. Based on a dual-view approach, a lexical detection protocol for phishing web addresses was developed and experimentally validated under syntactic transformations that preserve the destination host within the adopted threat model.

Method. The study is empirical. An open dataset of phishing and legitimate web addresses compiled by Prasad and Chandra was used. After data cleaning, 235,366 unique records remained. The training, validation, and test subsets were partitioned by registrable domain. A character-level baseline model, a dual-view model, and the proposed model trained with data augmentation were compared. The threat model included host case variation, explicit default ports, percent encoding, fragments, trailing dots, user-info prefixes, removable path segments, and combinations of these transformations. Model performance was assessed using precision, recall, F1-score, area under the receiver operating characteristic curve (ROC-AUC), and false-positive rate.

Findings. On the clean, domain-disjoint test set, the baseline and augmentation-trained models achieved F1-scores of 0.9962 and 0.9960, respectively; under the baseline-adaptive transfer-testing scenario, the corresponding values were 0.9943 and 0.9961.

Theoretical implications. The study extends methodological approaches to evaluating lexical phishing detectors by substantiating the joint use of raw and canonical web address representations and invariance-oriented training within a formalized obfuscation-adversary model.

Practical implications. The proposed protocol is suitable for local pre-filtering of web addresses in email gateways, browser extensions, and monitoring systems without requiring network queries. It can also support security-policy decisions concerning web-address filtering.

Originality/value. The study demonstrates that performance on clean data alone is insufficient to substantiate claims of robustness within the adopted threat model. It further shows that training-time augmentation improves invariance to the studied class of semantically equivalent obfuscations without causing a practically significant degradation in performance on clean data.

Limitations/future research. Further temporal, cross-dataset, and browser-dependent evaluations are required before deployment in operational systems.

Paper type. Empirical.

Ключові слова аргументація даних; канонізація; машинне навчання; обфускація; фішинг; веб-адреса.

Key words. Data Augmentation; Canonicalization; Machine Learning; Obfuscation; Phishing; Web Address.

Вступ

Фішинг залишається одним із найпоширеніших методів отримання несанкціонованого доступу до облікових записів та інформаційних систем. За даними Anti-Phishing Working Group, у 2025 році було зафіксовано майже 3,8 млн фішингових атак, із них 853 244 – у четвертому кварталі (Anti-Phishing Working Group, 2026). Згідно з доповіддю Європейського агентства з кібербезпеки, фішинг, малспам, вішинг і малвертайзинг становлять близько 60 % початкових векторів вторгнення (Європейське агентство з кібербезпеки, 2025). Це свідчить не лише про значне поширення та стійкість цієї загрози, а й про дедалі вищий рівень індустріалізації та автоматизації фішингових кампаній. Готові комплекти фішингових сторінок, автоматизована реєстрація доменів і моделі “фішинг як послуга” скорочують час підготовки атаки.

Для органів публічної безпеки, підрозділів кіберзахисту правоохоронних органів та операторів критичної інфраструктури це зумовлює потребу в локальних, відтворюваних і керованих відповідно до політик безпеки механізмах попереднього фільтрування вебадрес ще до завантаження вебсторінки. URL-адреса є індикатором, який засіб безпеки може оцінити до звернення до відповідного ресурсу. Її аналіз не потребує виконання скриптів, рендерингу вебсторінки, здійснення WHOIS-запитів або передавання адреси зовнішнім сервісам. Це робить лексичні детектори вебадрес корисними для поштових шлюзів, розширень браузерів, систем DNS-фільтрації та моніторингу.

Водночас зловмисник має значний контроль над більшістю компонентів вхідного рядка вебадреси, зокрема над субдоменом, шляхом, параметрами, фрагментом, регістром окремих компонентів і способом кодування символів. Тому висока якість класифікації на статичній тестовій вибірці сама по собі ще не доводить стійкості моделі до навмисних синтаксичних перетворень у межах прийнятої моделі загроз.

Набір даних *PhiUSIIL* містить 235 795 записів із численними лексичними, мережевими та контентними характеристиками, а також необробленими URL-адресами. Його автори продемонстрували високу якість інкрементного виявлення фішингу (Prasad & Chandra, 2024), а версію 2 набору даних було опубліковано в Mendeley Data (Prasad & Chandra, 2023). Проте використання готових характеристик і випадковий розподіл даних між вибірками можуть приховувати два методологічні ризики.

Перший ризик полягає в тому, що записи з близькими або ідентичними доменами можуть одночасно потрапити до навчальної та тестової вибірок. За таких умов модель може не засвоювати загальні закономірності, а лише запам’ятовувати специфічні шаблони доменів. Другий ризик полягає в тому, що характеристики, обчислені для початкового рядка вебадреси, не дають змоги оцінити, як зміниться рішення класифікатора після застосування семантично еквівалентної обфускації. У цьому дослідженні використовували лише URL-адреси та відповідні мітки класів, а навчальну, валідаційну й тестову вибірки формували за зареєстрованими доменами.

Під семантично еквівалентною обфускацією в цьому дослідженні розуміють синтаксичну модифікацію URI, яка в межах прийнятої моделі загроз не змінює цільового вузла або ресурсу. До таких модифікацій належать зміна регістру DNS-імені, зазначення порту за замовчуванням, відсоткове кодування незарезервованого символу, додавання фрагмента, кінцевої крапки доменного імені, інформації про користувача перед незмінним вузлом і сегмента шляху, який може бути вилучений, а також поєднання зазначених перетворень. Таке визначення є вужчим за підходи, що передбачають заміну символів або реєстрацію подібного домену, оскільки в останніх випадках створюється новий мережевий об’єкт, що потребує інших припущень у межах моделі

загроз. Таке обмеження дає змогу оцінити інваріантність класифікатора без підміни предмета дослідження питаннями доступності або реєстрації домену.

Мета статті – підвищити стійкість виявлення фішингових URL-адрес до досліджуваного класу семантично еквівалентних обфускацій шляхом застосування запропонованого протоколу з подвійним представленням, що поєднує два символьні представлення – необроблене та канонізоване – з аугментацією фішингових адрес у процесі навчання в межах формалізованої моделі загроз.

Для досягнення поставленої мети визначено такі **завдання дослідження**: очистити набір даних *PhiUSIIL* і сформувати неперетинні за зареєстрованими доменами навчальну, валідаційну й тестову вибірки; реалізувати консервативну канонізацію відповідно до синтаксису URI; побудувати базову модель, модель із подвійним представленням та аугментовану модель із застосуванням однакового алгоритму навчання; визначити сім одиничних обфускаційних атак, складену атаку та *baseline-adaptive transfer*-атаку; порівняти якість моделей на неперетворених даних, величину зниження F1-міри, довірчі інтервали та парні помилки класифікації.

Наукова новизна полягає в розробленні та експериментальній перевірці цілісного протоколу лексичного виявлення фішингових вебадрес, який комплексно поєднує формування неперетинних за зареєстрованими доменами вибірок, паралельне використання необробленого (*raw*) і канонізованого (*canonical*) представлень та навчання з аугментацією (*augmentation-based training*) у межах формалізованої моделі обфускаційного порушника, а також передбачає єдиний *baseline-adaptive transfer*-тест для оцінювання стійкості моделей до досліджуваного класу семантично еквівалентних обфускацій.

Огляд літератури

Огляд методів машинного виявлення шкідливих URL-адрес, проведений D. Sahoo, C. Liu та S. C. H. Hoi, охоплює лексичні ознаки, ознаки вузлів і мережеві ознаки, пакетні та потокові алгоритми, а також проблему виявлення нових адрес, відсутніх у чорних списках (Sahoo et al., 2017). У пізнішому систематичному огляді A. Safi та S. Singh показано, що машинне навчання є найпоширенішою групою методів, однак порівняння результатів різних досліджень ускладнюється через відмінності в джерелах даних, правилах їх очищення та способах формування вибірок (Safi & Singh, 2023). Отже, забезпечення відтворюваності потребує не лише оприлюднення програмного коду моделі, а й точного опису процедури формування незалежної тестової вибірки.

Традиційні підходи ґрунтуються на явно сконструйованих характеристиках рядка URL. O. K. Sahingoz та співавтори порівняли сім алгоритмів, що використовують ознаки URL, і продемонстрували конкурентоспроможність алгоритму випадкового лісу (Sahingoz et al., 2019). B. B. Gupta та співавтори запропонували компактний набір із дев’яти лексичних ознак, придатний для роботи в режимі реального часу (Gupta et al., 2021). Перевагами таких систем є пояснюваність і незначна затримка оброблення, проте заздалегідь сформований вручну перелік ознак може не охоплювати нових способів маскування. Крім того, окремі ознаки можуть бути суттєво пов’язані з конкретним джерелом легітимних адрес.

Моделі символьного рівня зменшують залежність від ручного конструювання ознак. A. Aljofey та співавтори використали символьну згорткову нейронну мережу для автоматичного виявлення локальних шаблонів (Aljofey et al., 2020). L. Tang і Q. H. Mahmoud поєднали чорні та білі списки з рекурентною моделлю у браузерній системі (Tang & Mahmoud, 2022). P. Maneriker та співавтори в моделі URLTran застосували трансформери й окремо дослідили роботу системи за дуже низьких рівнів хибнопозитивних спрацювань, що важливо для практичного застосування (Maneriker et al., 2021). Такі архітектури підвищують репрезентативну здатність моделей, однак не усувають їхньої залежності від способу токенізації та розподілу навчальних даних.

У GramBeddings символьні *n*-грами поєднано зі згортковими, рекурентними компонентами та компонентами механізму уваги; автори також оприлюднили великий набір URL-адрес (Bozkir et al., 2023). H. Ghalechyan та співавтори провели емпіричне дослідження нейромережевих моделей і наголосили на важливості канонізації, а також контролю довжини та глибини адрес (Ghalechyan et al., 2024). M. A. Tamal та співавтори опублікували окремий набір із 247 950 записів, доповнений внутрішніми ознаками URL (Tamal et al., 2024). Розширення та урізноманітнення наборів даних є важливими для розвитку відповідних методів, однак значний обсяг вибірки сам по собі не гарантує незалежності її елементів: тисячі адрес одного вузла можуть створювати лише видимість різноманіття на рівні окремих рядків.

A. Hannousse та S. Yahiouche сформулювали рекомендації щодо формування еталонних наборів даних і показали, що походження даних, їх збалансованість та відтворюваність процедури підготовки впливають на результати дослідження не менше, ніж вибір класифікатора (Hannousse & Yahiouche, 2021). Їхні результати підтверджують необхідність оприлюднення правил очищення даних і недопущення змішування спостережень, які не могли б бути незалежними в умовах реальної експлуатації. Для URL-адрес природною одиницею групування є зареєстрований домен, хоча навіть такий спосіб поділу не усуває часової залежності або можливості використання спільного набору фішингових сторінок.

Особливого значення набуває моделювання дій активного порушника. B. Sabir та співавтори створили URLBUG і перевірили 50 детекторів: медіана коефіцієнта кореляції Метьюза знизилася з 0,92 до 0,02 на згенерованих адресах, призначених для обходу детекторів (Sabir et al., 2022). M. J. Pillai та співавтори дослідили атаки ухилення та відповідні механізми захисту класифікаторів вебфішингу (Pillai et al., 2024). H. Kwon і J. Lee у 2026 році продемонстрували, що навіть до п’яти візуально подібних замін можуть бути використані для експлуатації слабких місць *subword*-токенізації трансформерів (Kwon & Lee, 2026). Ці результати свідчать про те, що оцінювання моделей виключно на неперетворених даних може систематично завищувати їхню практичну стійкість.

Загальна теорія *adversarial machine learning* передбачає чітке визначення знань, можливостей і цілей порушника. B. Biggio і F. Roli наголошують, що механізм захисту не можна оцінювати виключно за допомогою тієї атаки, на якій його було навчено (Biggio & Roli, 2018). D. Arp та співавтори виокремили типові помилки застосування машинного навчання у сфері комп’ютерної безпеки: невідповідність моделі загроз, залежність між навчальною і тестовою вибірками, некоректне визначення базових моделей та формулювання висновків на основі однієї метрики (Arp et al., 2022). Тому в цьому дослідженні для всіх моделей використано однаковий атакований тестовий набір, а обфускацію визначено відносно зафіксованої базової моделі (*baseline-adaptive transfer*), а не окремо для кожної захищеної моделі.

Витік даних є давно відомою причиною отримання надмірно оптимістичних оцінок якості моделей. S. Kaufman та співавтори визначили його як використання під час навчання інформації, яка була б недоступною на момент реального прогнозування (Kaufman et al., 2012). S. Kapoor і A. Narayanan показали зв’язок між витоком даних і кризою відтворюваності у прикладному машинному навчанні (Kapoor & Narayanan, 2023). У сфері кібербезпеки додатковим чинником є часовий вимір: TESSERACT демонструє, що просторове та часове змішування даних може суттєво викривляти оцінки якості класифікації шкідливого програмного забезпечення (Pendlebury et al., 2019). Набір *PhiUSIIL* не містить придатної часової мітки для кожної URL-адреси, тому поділ за доменами є необхідним, але недостатнім наближенням до перспективного тестування.

Аналіз наведених досліджень дає змогу визначити практичну потребу в компактному протоколі, який одночасно: працює лише з рядковим представленням URL; не потребує попередньо сформованого словника; контролює витік даних між доменами; зберігає необроблену інформацію; формує інваріантне представлення; передбачає навчання на допустимих обфускаціях; оцінюється на однакових неперетворених та атакованих тестових

вибірках. Окремі складові такого підходу — канонізація, символьні n-грами, аугментація, *adversarial*-оцінювання та груповий поділ (*group-based splitting*) — уже представлені в науковій літературі. Водночас їх комплексне поєднання з формалізованою моделлю обфускаційного порушника та єдиним *baseline-adaptive transfer*-тестом залишається недостатньо опрацьованим. Саме таку комбінацію реалізовано в подальшому дослідженні, а порівняльний синтез ключових методологічних характеристик наведено в табл. 1а.

Таблиця 1а: Порівняння досліджуваних підходів за ключовими методологічними вимірами

Підхід / робота	Domain-disjoint	Канонізація	Raw + canonical	Adversarial augmentation	Adaptive evaluation
Sahingoz et al., 2019	–	–	–	–	–
Maneriker et al., 2021	частково	–	–	–	–
Sabir et al., 2022	залежить від детектора	–	–	+	+
Prasad & Chandra, 2024	не акцентовано	+	–	–	–
Запропонований протокол	+	+	+	+	+ (baseline-adaptive transfer)

Джерело: *розроблено авторами за результатами огляду літератури.*

Методологія дослідження

Набір даних і запобігання витоку

Для аналізу використано набір даних *PhiUSIIL Phishing URL Dataset* (ідентифікатор UCI – 967; версія набору даних: DOI 10.17632/shwpxscxy2.2) (Prasad & Chandra, 2023, 2024). Початковий набір містив 235 795 записів. Для дослідження використано лише поле URL та мітку класу; 54 наявні ознаки не використовувалися і повторно не обчислювалися.

У вихідному наборі даних значення 1 позначає легітимний вебсайт, а значення 0 – фішинговий. У програмній реалізації мітки було інвертовано, тому позитивному класу відповідає фішинговий вебсайт.

Перед поділом набору даних було видалено порожні й аномально довгі рядки, точні дублікати, а також URL-адреси з конфліктними мітками, тобто випадки, коли однакова адреса належала до різних класів. Після очищення залишилося 235 366 унікальних URL-адрес, серед яких 100 516 були фішинговими, а 134 850 – легітимними. Кількість видалених точних дублікатів становила 429; URL-адрес із конфліктними мітками не виявлено.

Для кожної URL-адреси локально, без виконання DNS-запитів, було визначено ім’я хоста та зареєстрований домен. Для цього використано вбудований знімок *Public Suffix List*. Записи, для яких не вдалося коректно визначити ім’я хоста, було віднесено до окремої хешованої групи. Кожен зареєстрований домен розподілено до однієї з частин вибірки за допомогою 64-бітної хеш-функції BLAKE2b із фіксованою сіллю. Фактичний розподіл даних наведено в табл. 1б.

Таблиця 1б: Склад доменно-неперетинних частин PhiUSIIL

Частина	URL	Фішингові	Легітимні	Домени
Навчальна	158 786	64 174	94 612	122 907
Валідаційна	43 787	23 534	20 253	26 574
Тестова	32 793	12 808	19 985	26 028

Джерело: *розроблено авторами за даними (Prasad & Chandra, 2023, 2024).*

Валідаційна частина використовувалася тільки для вибору порога, що максимізує F1-міру серед значень від 0,05 до 0,95 із кроком 0,005. Після цього поріг фіксували. Тест не застосовувався ні для налаштування класифікатора, ні для вибору атаки під час навчання. Детерміноване гешування не забезпечує точно однакової частки класів, але усуває ручний добір сприятливого розбиття.

Канонізація URL-адрес

Синтаксичною основою є RFC 3986, який визначає компоненти URI, зарезервовані й незарезервовані символи та правила вилучення *dot*-сегментів (Berners-Lee et al., 2005). Семантика HTTP, зокрема типові порти й роль фрагмента, узгоджується з RFC 9110 (Fielding et al., 2022). Реалізована канонізація не робить мережевих запитів і не розкриває перенаправлення, тому є придатною для швидкого та приватного попереднього аналізу.

Алгоритм додає схему http, якщо вона відсутня; переводить схему й вузол у нижній регістр; прибирає кінцеву крапку DNS-імені та префікс www; перетворює міжнародне доменне ім'я у форму IDNA; вилучає порт 80 для http і 443 для https; декодує відсотково закодовані незарезервовані символи; стискає повторні похилі риски; прибирає прості сегменти "."; сортує параметри запиту; повністю вилучає фрагмент. Нетиповий порт, зарезервовані коди й значущі частини шляху зберігаються. Усі операції обмежено локальним синтаксичним рівнем, щоб не виконувати потенційно небезпечний ресурс.

Канонічний рядок не замінює початковий. Деякі відмінності, які нормалізація усуває, можуть самі бути сигналом обману, наприклад надлишкове кодування або незвичний порт. Тому запропоновано двопредставну схему: одна половина простору ознак обчислюється для raw URL, друга -для canonical URL. Класифікатор одночасно бачить ознаки підозрілої форми й ознаки стабілізованої сутності. Це відрізняє метод від попередньої нормалізації, після якої початковий рядок безповоротно втрачається.

Моделі та навчання

У кожному представленні URL перетворюється на символьні n-грами довжини від 3 до 5. Перед гешуванням векторизатор переводить символи в нижній регістр; тому позначення raw означає відсутність URL-канонізації, а не побайтове збереження регістру початкового рядка. Застосовано feature hashing у 131 072 вимірів на представлення без словника й з невід'ємними значеннями. Гешування ознак дає фіксовану пам'ять і дає змогу обробляти великий словник *n*-грам (Weinberger et al., 2009). Вектори нормалізуються за евклідовою нормою. Для raw-моделі розмір становить 131 072, для двопредставних -262 144.

Класифікатором є лінійна логістична модель, навчена стохастичним градієнтним спуском із L2-регуляризацією, усередненням ваг, *class_weight=balanced* та *seed* = 20260724. Максимальна кількість проходів -35, критерій зупинки -10^{-4}. Такий вибір навмисно відокремлює внесок представлення й аугментації від складності нейромережі. Крім того, лінійна модель відтворюється на центральному процесорі та придатна для систем з обмеженою затримкою.

Порівнювали три варіанти. Raw baseline навчався лише на початкових рядках і одному гешованому просторі. Dual view навчався на початкових прикладах, але використовував конкатенацію *raw* і *canonical*. *Robust augmented* мав ту саму двопредставну архітектуру, що й *dual view*, а до навчальної частини для кожної фішингової адреси додавався один детерміновано обраний обфускований варіант. Легітимні адреси не дублювалися, тому аугментація цілеспрямовано моделювала дії порушника. Фактична навчальна множина *robust*-моделі містила 222 960 рядків проти 158 786 у двох інших.

Модель порушника та обфускації

Порушник має чорний доступ до оцінки базового класифікатора, контролює текст фішингової URL-адреси, але не може змінити мітку або змусити дослідницький код звернутися до мережі. Його мета – мінімізувати оцінку фішингового класу за допомогою обмеженого

набору перетворень. Порушник не змінює легітимні адреси, тому хибні спрацювання в атакованому тесті вимірюються на тих самих негативних прикладах.

Визначено такі сімейства. Host case чергує регістр символів DNS-вузла. *Default* port явно додає :80 або :443 відповідно до схеми. Percent path кодує один незарезервований символ шляху. Fragment додає текст після “#”, який не передається серверу в HTTP-запиті. *Trailing dot* додає кореневу крапку до повного DNS-імені. *User-info* вставляє оманливий префікс виду “secure.trusted.example@” перед незмінним вузлом призначення. Dot segment додає вилучуваний сегмент “/./” у шлях. *Composite* послідовно поєднує зміну регістру, типовий порт, кінцеву крапку, *percent-encoding*, *dot*-сегмент і фрагмент.

Для сценарію адаптивного до базової моделі transfer-тестування для кожного позитивного тестового URL генерували всі вісім фіксованих варіантів. *Raw baseline* оцінював їх пакетно, після чого до спільного атакованого тесту включався варіант із найменшою ймовірністю фішингу. Важливо, що цей самий набір без повторного добору подавався dual-view і robust-моделям. Отже, атака є адаптивною відносно *baseline* і *transfer*-атакою відносно *Dual/Robust*: порівняння не надає захисту слабшої атаки та відображає перенесення оптимізованих проти baseline перетворень. Розподіл фактично обраних варіантів наведено в табл. 2.

Таблиця 2: Варіанти, обрані в baseline-adaptive transfer-сценарії

Перетворення	Кількість	Частка, %
Типовий порт	5 004	39,07
Складена	3 745	29,24
Кінцева крапка	1 544	12,05
Фрагмент	1 216	9,49
Percent-encoding	866	6,76
Регістр вузла	232	1,81
User-info	146	1,14
Dot-сегмент	55	0,43

Джерело: *розроблено авторами.*

Деякі перетворення мають протокольні застереження. Інформація користувача може вплинути на заголовок Authorization у специфічному клієнті; нормалізація dot-сегментів залежить від реалізації проміжного сервера; сервер може по-різному трактувати повторне кодування. Тому експеримент перевіряє рядкову інваріантність за прийнятою моделлю, а не гарантує байт-у-байт однакову відповідь довільного вебстека. Це обмеження враховано при інтерпретації.

Метрики й статистична перевірка

Позитивним класом є фішинг. *Precision* показує частку справжніх фішингових адрес серед попереджень; *recall* – частку виявлених фішингових адрес; F1 – їх гармонійний баланс. Частку хибних спрацювань обчислювали як відношення легітимних адрес, позначених фішингом, до всіх легітимних адрес. ROC-AUC характеризує ранжування незалежно від порога (Fawcett, 2006), але за дисбалансу або дуже малих робочих FPR її слід читати разом із *precision*–*recall* показниками (Saito & Rehmsmeier, 2015). У цій редакції як основні метрики наведено Precision, Recall, F1, ROC-AUC і FPR; додавання PR-AUC / *Average Precision* планується в подальших експериментах, зокрема для калібрування за реалістичною поширеністю фішингу.

Для кожної пари “модель — сценарій” побудовано 95-відсотковий percentile bootstrap інтервал F1 за 500 повторних вибірок тесту з поверненням (Efron & Tibshirani, 1993). Однаковий

seed забезпечив відтворюваність. Для парного порівняння *raw baseline* і *robust augmented* на *baseline-adaptive transfer*-тесті застосовано точний двобічний критерій Мак-Немара за кількістю незгодних передбачень (McNemar, 1947). Нульова гіпотеза полягає в однаковій ймовірності унікальної правильної відповіді кожної моделі. Bootstrap на рівні рядків є допустимим, але не ідеальним через можливу залежність URL одного домену; у подальшій роботі доцільно збільшити кількість повторів до 1000–5000 і застосувати *group bootstrap* за реєстрованим доменом.

Результати

Підготовку даних, навчання та повне оцінювання виконано в Python 3.14.6 із використанням NumPy 2.5.1, pandas 3.0.5, SciPy 1.18.0 і scikit-learn 1.9.0. Загальний час виконання – від завантаження даних до збереження результатів – становив 467,0 с на локальній системі. Вихідні метрики записано у форматі CSV, а параметри та версії програмного забезпечення – у форматі JSON. Це дає змогу без ручного переписування звірити кожне наведене у статті числове значення з результатом машинного обчислення.

На чистій тестовій вибірці без перетину доменів модель *raw baseline* продемонструвала F1 = 0,9962, recall = 0,9923, ROC-AUC = 0,9987 і FPR = 0,00000. Модель *dual view* мала F1 = 0,9962, а *robust augmented* — F1 = 0,9960. Відмінності на чистих даних є незначними: запропонована аугментація не спричинила практично значущого погіршення результатів. Водночас значення, близькі до граничних, не слід тлумачити як свідчення розв'язання проблеми фішингу загалом: набір даних має специфічний спосіб формування, а поділ за доменами не відтворює майбутніх фішингових кампаній.

Сценарій адаптивного до базової моделі *transfer*-тестування зменшив F1 базової моделі до 0,9943, тобто абсолютна зміна відносно чистого тесту становила 0,0018. Для *dual view* отримано F1=0,9942, для *robust augmented* — F1=0,9961. 95-відсотковий інтервал robust-моделі дорівнював [0,9953; 0,9969]. Повні значення основних сценаріїв наведено в табл. 3.

Таблиця 3: Якість моделей у вибраних сценаріях

Модель	Сценарій	F1	95 % CI	AUC	FPR
Raw	Чистий	0,9962	[0,9954; 0,9970]	0,9987	0,00000
Raw	Типовий порт	0,9947	[0,9939; 0,9956]	0,9992	0,00000
Raw	Кінцева крапка	0,9947	[0,9939; 0,9957]	0,9990	0,00000
Raw	Baseline-adaptive	0,9943	[0,9934; 0,9954]	0,9984	0,00000
Dual	Чистий	0,9962	[0,9954; 0,9969]	0,9988	0,00005
Dual	Типовий порт	0,9947	[0,9938; 0,9956]	0,9992	0,00005
Dual	Кінцева крапка	0,9941	[0,9932; 0,9951]	0,9986	0,00005
Dual	Baseline-adaptive	0,9942	[0,9933; 0,9951]	0,9985	0,00005
Robust	Чистий	0,9960	[0,9953; 0,9968]	0,9988	0,00010
Robust	Типовий порт	0,9989	[0,9985; 0,9993]	1,0000	0,00010
Robust	Кінцева крапка	0,9960	[0,9952; 0,9968]	0,9996	0,00010
Robust	Baseline-adaptive	0,9961	[0,9953; 0,9969]	0,9988	0,00010

Джерело: *розроблено авторами.*

На *baseline-adaptive transfer*-тесті лише базова модель дала правильний результат у 6 випадках, а лише *robust*-модель — у 50 випадках. Загалом виявлено 56 незгодних пар. Точне двобічне p-значення становило $1{,}018 \times 10^{-9}$. За рівня значущості 0,05 нульову гіпотезу відхилено.

Виявлена статистично значуща різниця доповнює, але не замінює оцінювання величини ефекту. Абсолютний приріст F1 становить приблизно 0,18 відсоткового пункту (0,9961 − 0,9943). Тому для ухвалення практичного рішення важливо також ураховувати абсолютну кількість пропущених атак і хибних попереджень.

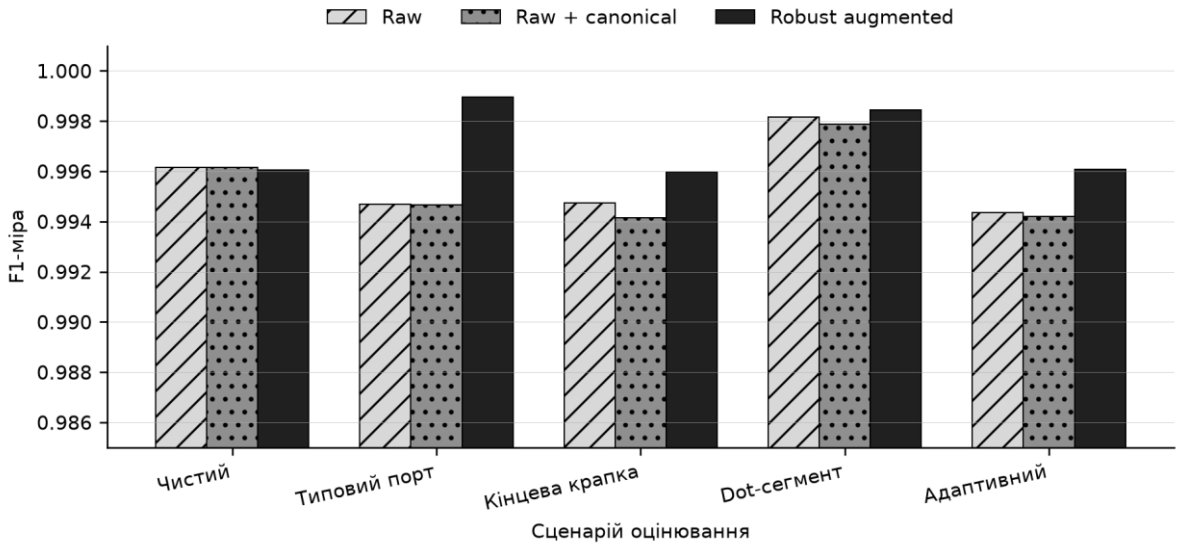

Рисунок 1: F1-міра трьох моделей у чистому й обфускаційних сценаріях
Джерело: розроблено авторами.

Вхідними даними для рисунка є значення F1-міри моделей *Raw*, *Dual* і *Robust*, обчислені на тому самому тестовому наборі, що й показники, наведені в табл. 3, для п'яти сценаріїв оцінювання: "Чистий", "Типовий порт", "Кінцева крапка", "Dot-сегмент" і "Baseline-adaptive". Сценарій "Dot-сегмент" до табл. 3 не увійшов.

На рис. 1 показано, чи є ефект системним для кількох перетворень, чи він зосереджений у конкретному варіанті. Окрема модифікація, яка випадково додає «підозрілу» *n*-граму, може навіть підвищити оцінку ймовірності фішингу. Саме тому в сценарії *baseline-adaptive* вибирається мінімальна оцінка відносно *baseline*, а не будь-яка зміна заздалегідь визначається як така, що забезпечує обхід моделі. Такий підхід запобігає хибному висновку про вразливість моделі, зробленому лише на підставі синтаксичної відмінності.

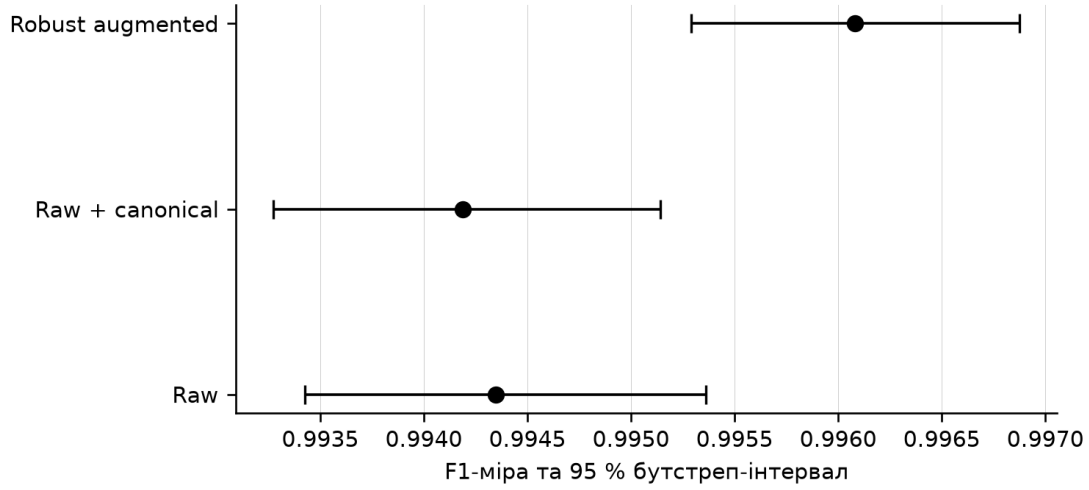

Рисунок 2: F1-міра на baseline-adaptive transfer-тесті з 95-відсотковими бутстреп-інтервалами
Джерело: *розроблено авторами.*

Вхідними даними для рисунка є значення F1-міри та межі 95-відсоткового бутстреп-інтервалу для моделей *Raw*, *Dual* і *Robust* у сценарії "Baseline-adaptive", наведені у стовпцях "F1" і "95 % CI" табл. 3.

Обговорення

Перше спостереження стосується експериментального протоколу. Навіть після вилучення точних дублікатів випадковий поділ рядків залишав би близькі адреси одного домену в різних частинах. Доменне групування є суворішим, бо змушує модель переносити лексичні закономірності на нові реєстровані домени. При цьому результати лишилися високими, що свідчить про сильний сигнал у PhiUSIIL, але водночас вимагає обережності: фішингові й легітимні URL могли надходити з джерел із різною структурою. Майже граничні значення F1 можуть відображати не лише високу дискримінативну здатність запропонованих представлень, а й *dataset-specific signal*, пов'язаний зі способом формування позитивного та негативного класів PhiUSIIL.

Друге спостереження -канонізацію доцільно використовувати як додаткове, а не єдине представлення. Якщо залишити лише canonical URL, детектор втратить сам факт надлишкового кодування, фрагмента або *user-info*, який часто є корисною ознакою. Якщо залишити лише raw URL, семантично тотожні рядки можуть опинитися далеко у просторі *n*-грам. Конкатенація зберігає обидва типи інформації. Однак dual view без аугментації не обов'язково набуває інваріантності: оптимізатор може й надалі покладатися переважно на raw-координати.

Третє спостереження — аугментація задає моделі бажану поведінку. Кожний фішинговий навчальний URL представлено також одним перетвореним варіантом, тоді як легітимний клас не дублюється. Завдяки цьому одночасно змінюються invariance training, розмір навчальної множини, баланс класів і частота певних *n*-грам. Частина ефекту потенційно може бути наслідком зміни *distribution/class exposure*, а не лише обфускаційної інваріантності. Ідеальним контрольним експериментом було б додати size-matched модель (Dual + дублювання *phishing* без обфускації); у цій редакції такий контроль не виконано і зафіксовано як обмеження. Зафіксований чистий поріг запобігає додатковому налаштуванню під атаку.

Четверте спостереження стосується FPR. У поштовому або браузерному продукті навіть частка у кілька десятих відсотка може створювати тисячі зайвих попереджень. Тому поріг, що максимізує F1 на відносно збалансованій вибірці, не є автоматично оптимальним для експлуатації. Перед розгортанням його слід калібрувати за реальною базовою часткою фішингу, допустимим навантаженням на аналітика та вартістю пропуску. URLTran також підкреслює потребу оцінювати дуже низькі FPR (Maneriker et al., 2021).

П'яте спостереження — стійкість має бути відносною до загрозової моделі. Отриманий результат слід інтерпретувати як стійкість до досліджуваного класу семантично еквівалентних обфускацій, а не як загальну adversarial robustness. Наші перетворення не охоплюють реєстрацію *typo*- або *homograph*-домену, редиректи, скорочувачі, QR-посилання, IP-літерали, *Unicode confusables* і спільно керовані набори доменів. Атаки URLBUG (Sabir et al., 2022) та HG-BS (Kwon & Lee, 2026) ширші й можуть вимагати запитів до реєстратора, DNS або моделі-трансформера. Тому отриманий ефект не можна переносити на всі можливі маніпуляції.

Шосте спостереження — адаптивність у роботі обмежена фіксованим набором кандидатів і доступом до ймовірності baseline. Тому сценарій коректніше називати *baseline-adaptive transfer evaluation*: для *robust*-моделі атака вже не є повністю адаптивною. Порушник із довільною кількістю запитів може комбінувати символи, піддомени й параметри та знайти сильніший варіант. З іншого боку, однаковий *transfer*-тест робить порівняння трьох моделей чесним: *robust*-модель не отримує спеціально слабші приклади. Майбутня робота має додати *query-budget* і незалежний генератор, який не використовувався в аугментації.

Практична перевага запропонованого протоколу — відсутність мережевих залежностей. *Feature hashing* не зберігає словника, канонізація має лінійну складність від довжини рядка, а лінійний класифікатор виконує розріджений скалярний добуток. Систему можна розмістити перед дорожчим модулем, який аналізує DOM, сертифікат, скриншот або

граф інфраструктури. Таке каскадування дозволяє спрямувати сумнівні адреси до глибшого аналізу й не розкривати всі URL зовнішньому сервісу. З погляду управління безпекою це підтримує багаторівневу політику фільтрації: локальний лексичний екран зменшує навантаження на аналітиків і знижує залежність від хмарних репутаційних сервісів.

Відтворюваність забезпечено завдяки фіксованому значенню *seed*, детермінованому гешуванню доменів, відсутності онлайн-ознак і збереженню інформації про версії бібліотек. Водночас сервіс UCI може змінити спосіб надання файлу, а вбудований список *Public Suffix List* – оновитися в новій версії *tldextract*. Для точного відтворення результатів необхідно архівувати отриманий файл, його криптографічну контрольну суму та конфігурацію середовища пакетів.

У поточному комплекті збережено програмний код, вихідні метрики та файл JSON з інформацією про версії. Визначення контрольної суми вихідного архіву потребує окремого завантаження файлу, оскільки *ucimlrepo* повертає декодовану таблицю.

Обмеження валідності

Внутрішня валідність обмежена тим, що аугментація збільшує тільки позитивний клас і може частково навчати модель артефактам функцій перетворення; відсутність *size-matched* контролю не дає повністю відокремити ефект обфускації від ефекту збільшення кількості позитивних прикладів. Ми зменшили цей ризик детермінованим вибором одного з кількох сімейств, однак незалежна атака й контрольна модель потрібні. Крім того, HashingVectorizer допускає колізії, хоча великий простір робить їх вплив обмеженим. *Bootstrap* виконано на рівні рядків; через спільні кампанії залишкова залежність може звузити інтервали.

Зовнішня валідність обмежена одним набором і відсутністю часової мітки. Доменне неперетинання не гарантує незалежності організації, шаблону сторінки або джерела збору. Реальна частка фішингу значно нижча, ніж у тесті, а легітимні адреси різноманітніші. Потрібна перевірка на потоках *PhishTank/OpenPhish* і незалежному списку популярних або реально відвіданих адрес, сформованому після навчального періоду.

Конструктивна валідність обмежена лексичною постановкою. Мітка “фішинг” описує намір сторінки, але URL не завжди містить достатній сигнал. Компрометований легітимний домен може мати звичайний шлях, а довгий корпоративний URL-виглядати підозріло. Тому модель не є доказом шкідливості та не повинна одноосібно блокувати критичні ресурси. Вона генерує ризикову оцінку для подальшої політики.

Статистична валідність підтримана парним тестом і бутстрепом, але велика тестова множина робить навіть малу різницю статистично значущою. У висновках пріоритет надано абсолютній F1-зміні, кількості помилок і FPR. Ми не виконували багаторазового пошуку гіперпараметрів, тому не коригували p-значення за множинні порівняння; критерій Мак-Немара застосовано лише до наперед визначеної основної пари на *baseline-adaptive transfer*-тесті.

Висновки

Відповідно до сформульованої мети статті розроблено й експериментально перевірено двопредставний протокол лексичного виявлення фішингових URL, що поєднує необроблене та канонізоване символьні представлення з аугментацією семантично еквівалентними обфускаціями. На відміну від оцінки за випадковим поділом, усі частини сформовано за неперетинними реєстрованими доменами; тому модель не бачила під час навчання доменів тесту. Після очищення використано 235 366 адрес, а тест містив 32 793 записів. Твердження про відтворюваність стосується локально відтворюваного експериментального комплекту; публічний сталий архів коду й результатів має супроводжувати фінальну публікацію.

На чистому тесті robust augmented досяг F1=0,9960 за FPR=0,00010, не продемонструвавши практично значущого погіршення порівняно з *raw baseline*, для якого F1=0,9962 (формально чистий F1 зменшився на 0,0002, а FPR зріс з 0 до 0,00010). На спільному

baseline-adaptive transfer-тесті F1 базової моделі становила 0,9943, двопредставної без аугментації — 0,9942, запропонованої — 0,9961. Точний критерій Мак-Немара дав $p=1{,}018\times10^{-9}$; рішення щодо рівності парних помилок сформульовано вище. Отже, основний прикладний результат полягає у підвищенні інваріантності до визначеного класу семантично еквівалентних обфускацій без практично значущого погіршення чистої якості.

Наукова новизна роботи полягає в розробленні та експериментальній перевірці протоколу стійкого лексичного виявлення фішингових URL, що комплексно поєднує доменно-неперетинне формування вибірок, паралельне *raw/canonical* представлення URL та *augmentation-based* навчання в межах формалізованої моделі обфускаційного порушника, а також у застосуванні однакового *baseline-adaptive transfer*-тесту. Практичне значення полягає в можливості реалізувати легкий локальний фільтр без DNS, WHOIS або завантаження сторінки та використати його як елемент багаторівневої політики кібербезпеки.

Подальші дослідження мають використати часовий зовнішній тест, group bootstrap за доменами (1000–5000 повторів), *size-matched* контроль аугментації, метрику PR-AUC, незалежні *homograph* і *redirect*-атаки, аналіз стабільності за різних *Public Suffix List* та калібрування порога за реальною поширеністю фішингу. Для експлуатації доцільно поєднати лексичну оцінку з репутацією домену, TLS, вмістом і поясненням причини попередження.

Внесок авторів

Фещенко Р.: Conceptualization, Methodology, Software, Investigation, Formal Analysis, Writing – Original Draft. Вітвіцька К.: Validation, Literature Review, Data Curation, Writing – Review & Editing. Лучик В.: Supervision, Conceptualization, Writing – Review & Editing; остаточне схвалення рукопису перед поданням.

Подяки

Немає.

Фінансування

Дослідження не отримувало зовнішнього фінансування.

Конфлікт інтересів

Автори заявляють про відсутність конфлікту інтересів.

Доступність даних і коду

PhiUSIIL доступний у UCI Machine Learning Repository, identifier 967, та Mendeley Data, DOI 10.17632/shwpxscxy2.2. Код експерименту, таблиця метрик, JSON-параметри, seed, версія Public Suffix List / середовища та контрольні суми опубліковано у сталому архіві Zenodo: https://doi.org/10.5281/zenodo.21864061 (DOI 10.5281/zenodo.21864061).

Декларація про використання штучного інтелекту

Під час підготовки робочої версії рукопису використано Cursor із моделлю GPT-5.6 Sol для пошуку потенційних джерел та мовного редагування. Бібліографічні метадані перевірено через Crossref, DataCite, сторінки видавців і офіційні документи; усі числові результати отримано локальним виконанням відкритого коду, а не згенеровано моделлю. Інструмент ШІ не є автором і не несе відповідальності. Відповідно до політики видання автори зобов’язані перевірити та усвідомлено відредагувати кожне твердження, повторно виконати або верифікувати аналіз і взяти повну відповідальність за остаточний текст.

Список використаних джерел

[1] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J.-P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. Electronics, 9(9), 1514. https://doi.org/10.3390/electronics9091514.

[2] Anti-Phishing Working Group. (2026). Phishing activity trends report: 4th quarter 2025. URL : https://docs.apwg.org/reports/apwg_trends_report_q4_2025.pdf.

[3] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In 31st USENIX Security Symposium (pp. 3971–3988). URL : https://www.usenix.org/conference/usenixsecurity22/presentation/arp.

[4] Berners-Lee, T., Fielding, R., & Masinter, L. (2005). Uniform resource identifier (URI): Generic syntax (RFC 3986). RFC Editor. https://doi.org/10.17487/RFC3986.

[5] Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023.

[6] Bozkir, A. S., Dalgic, F. C., & Aydos, M. (2023). GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings. Computers & Security, 124, 102964. https://doi.org/10.1016/j.cose.2022.102964.

[7] Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. Chapman & Hall. https://doi.org/10.1201/9780429246593.

[8] European Union Agency for Cybersecurity. (2025). ENISA threat landscape 2025. Publications Office of the European Union. https://doi.org/10.2824/1946374.

[9] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.

[10] Fielding, R., Nottingham, M., & Reschke, J. (2022). HTTP semantics (RFC 9110). RFC Editor. https://doi.org/10.17487/RFC9110.

[11] Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. Scientific Reports, 14, 25134. https://doi.org/10.1038/s41598-024-74725-6.

[12] Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Computer Communications, 175, 47–57. https://doi.org/10.1016/j.comcom.2021.04.023.

[13] Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. Engineering Applications of Artificial Intelligence, 104, 104347. https://doi.org/10.1016/j.engappai.2021.104347.

[14] Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), 100804. https://doi.org/10.1016/j.patter.2023.100804.

[15] Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining. ACM Transactions on Knowledge Discovery from Data, 6(4), 1–21. https://doi.org/10.1145/2382577.2382579.

[16] Kwon, H., & Lee, J. (2026). Crafting evasive phishing URLs: Exploiting tokenizer vulnerabilities in transformer-based detection systems. International Journal of Machine Learning and Cybernetics, 17(8), 390. https://doi.org/10.1007/s13042-026-03239-6.

[17] Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021). URLTran: Improving phishing URL detection using transformers. In MILCOM 2021—IEEE Military Communications Conference (pp. 197–204). https://doi.org/10.1109/MILCOM52596.2021.9653028.

[18] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2), 153–157. https://doi.org/10.1007/BF02295996.

[19] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2019). TESSERACT: Eliminating experimental bias in malware classification across space and time. In 28th USENIX Security Symposium (pp. 729–746). URL : https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury

[20] Pillai, M. J., Remya, S., Devika, V., Ramasubbareddy, S., & Cho, Y. (2024). Evasion attacks and defense mechanisms for machine learning-based web phishing classifiers. IEEE Access, 12, 19375–19387. https://doi.org/10.1109/ACCESS.2023.3342840.

[21] Prasad, A., & Chandra, S. (2023). PhiUSIIL phishing URL dataset (Version 2) [Data set]. Mendeley Data. https://doi.org/10.17632/shwpxscxy2.2.

[22] Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 136, 103545. https://doi.org/10.1016/j.cose.2023.103545.

[23] Sabir, B., Babar, M. A., Gaire, R., & Abuadbba, A. (2022). Reliability and robustness analysis of machine learning based phishing URL detectors. IEEE Transactions on Dependable and Secure Computing. https://doi.org/10.1109/TDSC.2022.3218043.

[24] Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. Journal of King Saud University—Computer and Information Sciences, 35(2), 590–611. https://doi.org/10.1016/j.jksuci.2023.01.004.

[25] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029.

[26] Sahoo, D., Liu, C., & Hoi, S. C. H. (2017). Malicious URL detection using machine learning: A survey. arXiv. https://doi.org/10.48550/arXiv.1701.07179.

[27] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432.

[28] Tamal, M. A., Islam, M. K., Bhuiyan, T., & Sattar, A. (2024). Dataset of suspicious phishing URL detection. Frontiers in Computer Science, 6, 1308634. https://doi.org/10.3389/fcomp.2024.1308634.

[29] Tang, L., & Mahmoud, Q. H. (2022). A deep learning-based framework for phishing website detection. IEEE Access, 10, 1509–1521. https://doi.org/10.1109/ACCESS.2021.3137636.

[30] Weinberger, K., Dasgupta, A., Langford, J., Smola, A., & Attenberg, J. (2009). Feature hashing for large scale multitask learning. In Proceedings of the 26th International Conference on Machine Learning (pp. 1113–1120). https://doi.org/10.1145/1553374.1553516.

References

[1] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J.-P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. Electronics, 9(9), 1514. https://doi.org/10.3390/electronics9091514.

[2] Anti-Phishing Working Group. (2026). Phishing activity trends report: 4th quarter 2025. https://docs.apwg.org/reports/apwg_trends_report_q4_2025.pdf.

[3] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In 31st USENIX Security Symposium (pp. 3971–3988). https://www.usenix.org/conference/usenixsecurity22/presentation/arp.

[4] Berners-Lee, T., Fielding, R., & Masinter, L. (2005). Uniform resource identifier (URI): Generic syntax (RFC 3986). RFC Editor. https://doi.org/10.17487/RFC3986.

[5] Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 84, 317–331. https://doi.org/10.1016/j.patcog.2018.07.023.

[6] Bozkir, A. S., Dalgic, F. C., & Aydos, M. (2023). GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings. Computers & Security, 124, 102964. https://doi.org/10.1016/j.cose.2022.102964.

[7] Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. Chapman & Hall. https://doi.org/10.1201/9780429246593.

[8] European Union Agency for Cybersecurity. (2025). ENISA threat landscape 2025. Publications Office of the European Union. https://doi.org/10.2824/1946374.

[9] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010.

[10] Fielding, R., Nottingham, M., & Reschke, J. (2022). HTTP semantics (RFC 9110). RFC Editor. https://doi.org/10.17487/RFC9110.

[11] Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. Scientific Reports, 14, 25134. https://doi.org/10.1038/s41598-024-74725-6.

[12] Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Computer Communications, 175, 47–57. https://doi.org/10.1016/j.comcom.2021.04.023.

[13] Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. Engineering Applications of Artificial Intelligence, 104, 104347. https://doi.org/10.1016/j.engappai.2021.104347.

[14] Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. Patterns, 4(9), 100804. https://doi.org/10.1016/j.patter.2023.100804.

[15] Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining. ACM Transactions on Knowledge Discovery from Data, 6(4), 1–21. https://doi.org/10.1145/2382577.2382579.

[16] Kwon, H., & Lee, J. (2026). Crafting evasive phishing URLs: Exploiting tokenizer vulnerabilities in transformer-based detection systems. International Journal of Machine Learning and Cybernetics, 17(8), 390. https://doi.org/10.1007/s13042-026-03239-6.

[17] Maneriker, P., Stokes, J. W., Lazo, E. G., Carutasu, D., Tajaddodianfar, F., & Gururajan, A. (2021). URLTran: Improving phishing URL detection using transformers. In MILCOM 2021—IEEE Military Communications Conference (pp. 197–204). https://doi.org/10.1109/MILCOM52596.2021.9653028.

[18] McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika, 12(2), 153–157. https://doi.org/10.1007/BF02295996.

[19] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., & Cavallaro, L. (2019). TESSERACT: Eliminating experimental bias in malware classification across space and time. In 28th USENIX Security Symposium (pp. 729–746). https://www.usenix.org/conference/usenixsecurity19/presentation/pendlebury

[20] Pillai, M. J., Remya, S., Devika, V., Ramasubbareddy, S., & Cho, Y. (2024). Evasion attacks and defense mechanisms for machine learning-based web phishing classifiers. IEEE Access, 12, 19375–19387. https://doi.org/10.1109/ACCESS.2023.3342840.

[21] Prasad, A., & Chandra, S. (2023). PhiUSIIL phishing URL dataset (Version 2) [Data set]. Mendeley Data. https://doi.org/10.17632/shwpxscxy2.2.

[22] Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers & Security, 136, 103545. https://doi.org/10.1016/j.cose.2023.103545.

[23] Sabir, B., Babar, M. A., Gaire, R., & Abuadbba, A. (2022). Reliability and robustness analysis of machine learning based phishing URL detectors. IEEE Transactions on Dependable and Secure Computing. https://doi.org/10.1109/TDSC.2022.3218043.

[24] Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. Journal of King Saud University—Computer and Information Sciences, 35(2), 590–611. https://doi.org/10.1016/j.jksuci.2023.01.004.

[25] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029.

[26] Sahoo, D., Liu, C., & Hoi, S. C. H. (2017). Malicious URL detection using machine learning: A survey. arXiv. https://doi.org/10.48550/arXiv.1701.07179.

[27] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3), e0118432. https://doi.org/10.1371/journal.pone.0118432.

[28] Tamal, M. A., Islam, M. K., Bhuiyan, T., & Sattar, A. (2024). Dataset of suspicious phishing URL detection. Frontiers in Computer Science, 6, 1308634. https://doi.org/10.3389/fcomp.2024.1308634.

[29] Tang, L., & Mahmoud, Q. H. (2022). A deep learning-based framework for phishing website detection. IEEE Access, 10, 1509–1521. https://doi.org/10.1109/ACCESS.2021.3137636.

[30] Weinberger, K., Dasgupta, A., Langford, J., Smola, A., & Attenberg, J. (2009). Feature hashing for large scale multitask learning. In Proceedings of the 26th International Conference on Machine Learning (pp. 1113–1120). https://doi.org/10.1145/1553374.1553516.