РОЗРОБКА ТА ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ML-КЕРОВАНОГО ВЕБ-СКАНЕРА ДЛЯ DAST

Долгова Наталя
к.т.н, доцент
Єфімов Михайло
здобувач вищої освіти магістерського рівня
Кафедра кібербезпеки та інформаційних технологій
Харківський національний економічний університет імені Семена Кузнеця

Анотація. У статті розглядається програмна реалізація та результати випробувань веб-сканера безпеки, у якому машинне навчання виконує керівну роль у виявленні та пріоритизації потенційних вразливостей. Запропоновано трирівневу організацію системи (джерела даних, ядро обробки з ML-аналізом, прикладна інфраструктура) і чотириетапний робочий конвеєр, що поєднує пасивне ранжування, ML-керований фаззинг і доказову активну валідацію. Експериментальні прогони на навчальному стенді демонструють істотне скорочення часу перевірок без втрати повноти виявлення критичних класів атак.

Ключові слова: веб-безпека, динамічний аналіз (DAST), машинне навчання, ансамблеві моделі, фаззинг, активна валідація, SQL-ін’єкція, XSS.

Вступ. Динаміка сучасних веб-систем (мікросервіси, SPA, багатоканальні API) підвищує різноманіття параметризованих ендпоінтів і ускладнює контексти взаємодії. У цих умовах традиційні сканери часто накопичують хибнопозитивні спрацьовування та вимагають значного часу на прогін. Запропонований підхід інтегрує машинне навчання як керувальний механізм: ML визначає пріоритети перевірок, скорочує обсяг активних запитів і підвищує доказовість висновків, спираючись на класичні моделі scikit-learn [1] та послідовнісні моделі у Keras [2].

Додатковий чинник актуальності — стійкість «класичних» загроз на кшталт SQL-ін’єкцій і XSS [3]. Для подолання зазначених обмежень застосовано підхід, у якому машинне навчання не відокремлений детектор, а керувальний механізм: воно визначає пріоритети перевірок, скорочує обсяг активних запитів і підвищує доказовість висновків.

Цільовий дизайн розглядає систему як послідовний потік від «сирих» подій до звітів. На вхідному рівні акумулюються результати краулінгу, журнали запитів/відповідей, корпуси пейлоадів і довідкова інформація з відкритих реєстрів. Центральне ядро виконує ETL-перетворення та ML-аналіз, а прикладний рівень забезпечує дашборд, базу знахідок і формування звітів для інтеграції з процесами розробки. Модуль машинного навчання організовано як ансамбль, що поєднує «класичні» моделі з послідовнісною, а метапередбачувач із м’яким голосуванням стабілізує вердикт на різнорідних даних. Робочий конвеєр складається з чотирьох етапів: краулінг, пасивне ML-ранжування URL/параметрів, ML-керований фаззинг пейлоадів і багатопотокова активна валідація з коректним оновленням CSRF-токенів та диференційним порівнянням сторінок.

Мета та задачі дослідження. Розробити та експериментально оцінити ML-керований веб-сканер для DAST, який скорочує час повного прогона та підвищує доказовість результатів. Для досягнення мети сформульовано такі задачі: спроєктувати послідовний конвеєр перевірок із чотирма етапами — краулінг, пасивне ML-ранжування, ML-керований фаззинг і активна валідація; реалізувати модуль машинного навчання з двома гілками підготовки даних (класичною та послідовнісною) та ансамблевим об'єднанням оцінок; забезпечити відтворюваність і інтеграцію результатів через версіонування артефактів, журналювання та прикладну інфраструктуру звітів; провести експериментальні випробування на навчальному стенді, порівнявши режим з ML-фільтрацією пейлоадів і «агресивний» режим без фільтрації за часом прогона та повнотою виявлення.

Результати дослідження і їх обговорення. Програмне ядро побудовано як набір узгоджених компонентів із чіткими контрактами. Краулер формує карту поверхні атаки з урахуванням автентифікації та динамічних параметрів і передає артефакти до сховища. Пасивний модуль оцінює ризик без ін'єкцій у пакетному режимі; вихідні бали використовуються як фільтр для наступних етапів. Менеджер фаззингу працює з великими пулами пейлоадів і перед активною перевіркою виконує пакетну інференцію ансамблем, аби залишити лише верхній зріз найризиковіших комбінацій. Модуль активної валідації багатопотоково виконує перевірки, автоматично оновлює CSRF-токени та застосовує диференційне порівняння сторінок, що дозволяє підтверджувати «сліпі» ін'єкції без залежності від специфічних повідомлень про помилки.

Ансамблеве ML-ядро включає п'ять базових моделей (Random Forest, Multinomial Naive Bayes, Linear SVM, XGBoost, LSTM). «Класична» гілка працює з TF-IDF і інженерією ознак (довжина параметра, частка спеціальних символів, маркери SQL/XSS, ентропія); «нейромережева» — із символьними послідовностями фіксованої довжини для LSTM із вбудовуваннями та регуляризацією. Метапередбачувач усереднює апостеріорні ймовірності, зменшуючи чутливість до представлення даних і коливань окремих моделей. Усі артефакти (моделі, токенізатори, конфігурації) версіонуються для відтворюваності й регресійного аналізу (рис.1).

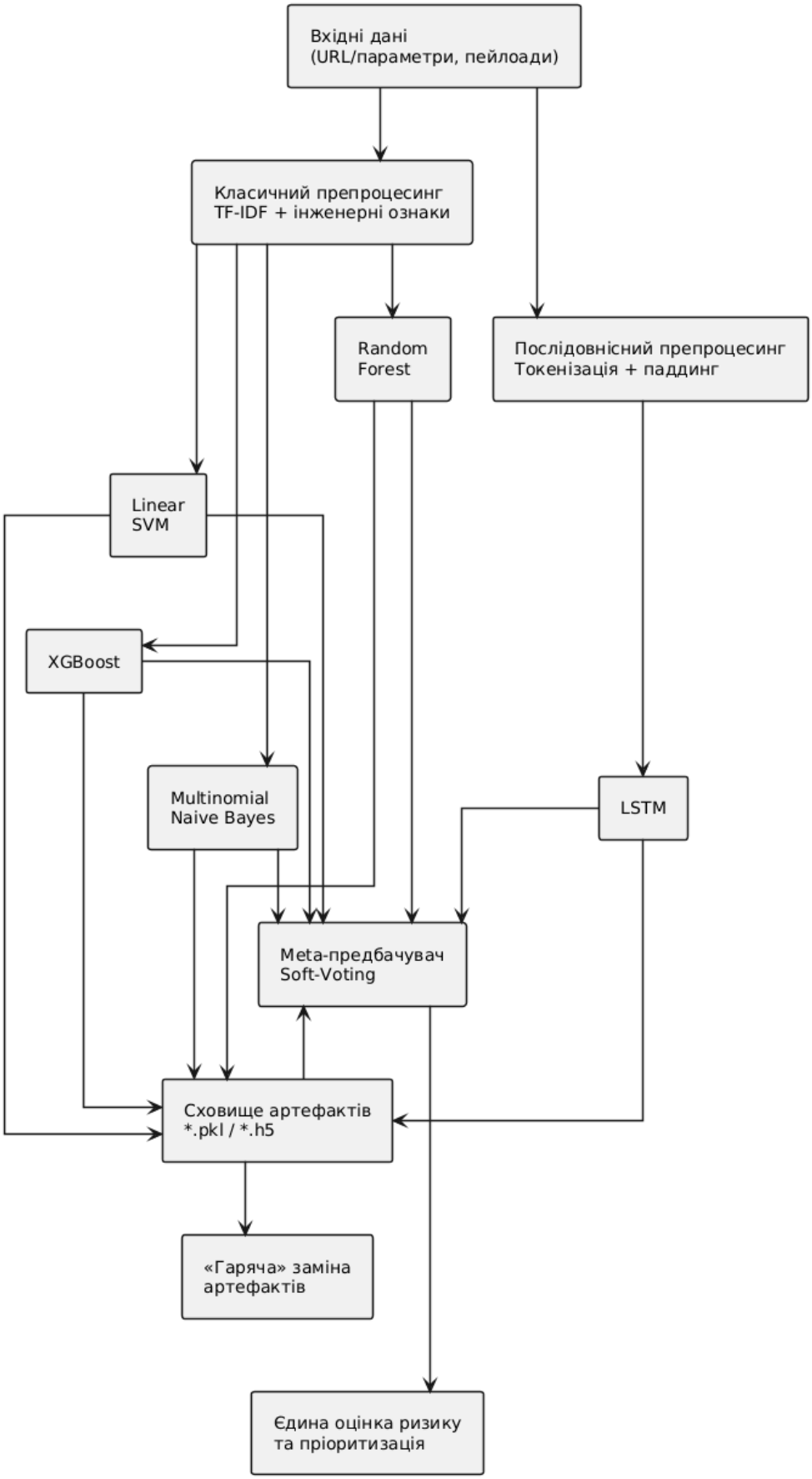

Рисунок 1. Ансамблеве ML-ядро та дві гілки препроцесингу

Для оцінювання підходу випробування проведено на навчальному стенді з автентифікацією та варійованими рівнями складності. Порівнювалися два режими: «агресивний» перебір без попередньої фільтрації та ML-керований конвеєр. У другому випадку великий пул пейлоадів (понад 30 тис.) спершу проходив ML-відсів, унаслідок чого до активної валідації потрапляла компактна підмножина; це скорочувало загальний час повного прогона приблизно у п’ятнадцять разів без втрати повноти виявлення для SQL-ін’єкцій і XSS. Додатково пакетна інференція зменшувала власні витрати ML-модуля: обчислення виконувалися порціями зі спільними буферами та попередньо побудованими матрицями ознак, що мінімізувало I/O-накладні витрати. Логіку впливу кожного компоненту і сумарний ефект показано на рис. 2: тривалість повного прогона знижується з ≈90 хв до ≈14 хв 25 с, а кількість пейлоадів для активної перевірки — з 31 067 до ≈22 700. Окрема позначка «Пакетна інференція» відображає внесок саме ML-модуля: час його обробки всього пулу скорочується з ~40 хв до <2 хв. Таким чином, поєднання пасивного ранжування та пакетної інференції формує керований конвеєр, що зменшує «шум», тиск на ціль і час перевірок, забезпечуючи при цьому повноту виявлення.

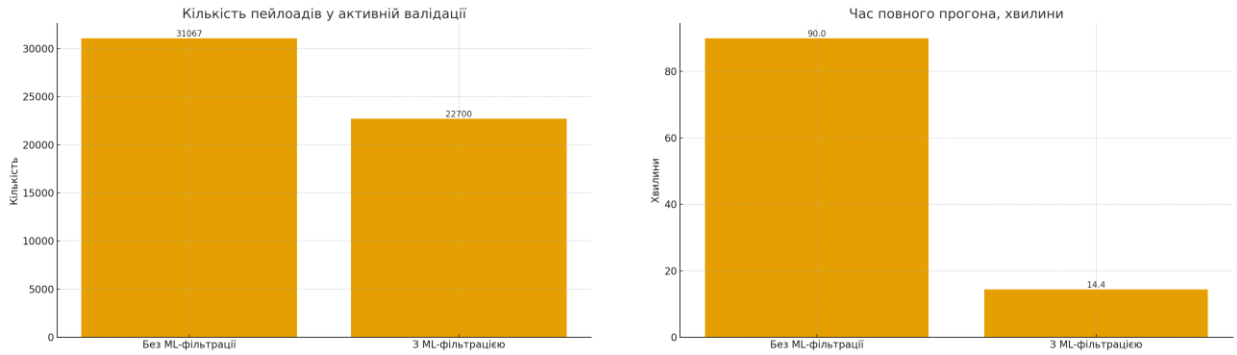

Рисунок 2. Порівняння продуктивності сканування

Практична придатність системи підтверджується коректною роботою в автентифікованих сесіях, автоматичним оновленням захисних токенів та веденням журналів доказів. Усі проміжні артефакти — скорингові таблиці, підмножини пейлоадів, результати диференційного аналізу — зберігаються в сховищі й прив’язуються до конкретних конфігурацій запуску. Це спрощує аудит, порівняння прогонів і поетапне удосконалення моделей. Інтеграція з довідковими записами (наприклад, CVE) дозволяє додавати контекст до знайдених проблем і формувати пріоритети усунення, що важливо для впровадження в інженерні процеси організації.

Поєднання пасивного ML-ранжування та цільового фаззингу переводить акцент з «грубої сили» на кероване дослідження поверхні атаки. Ансамбль зменшує ризик деградації якості на конкретних підмножинах даних, а пакетна обробка мінімізує часові витрати. Відтворюваність забезпечується версіонуванням моделей і даних, чіткими контрактами між етапами та повною трасованістю рішень. Разом це створює підґрунтя для масштабування на інші

класи вразливостей (поза SQL/XSS), а також для інтеграції в конвеєри CI/CD із контрольованою «агресивністю» перевірок і формалізованими стоп-умовами.

Висновки

Представлено й випробувано ML-керований веб-сканер, у якому машинне навчання виконує функції пріоритизації та управління навантаженням у чотириетапному конвеєрі. Експерименти засвідчили істотне скорочення часу повного прогона (приблизно у 15 разів) без втрати повноти виявлення SQL-ін'єкцій і XSS, підвищення пропускної здатності за рахунок пакетної інференції та зменшення «шуму» завдяки пасивному ранжуванню. Ансамблеве ядро стабілізує якість на різнорідних даних, а інфраструктурні рішення (автентифікація, CSRF-оновлення, диференційна валідація, версіонування артефактів) забезпечують відтворюваність і практичну придатність.

Подальший розвиток бачиться в розширенні класів уразливостей і сценаріїв тестування (з урахуванням сучасних SPA/API), у стандартизації бенчмаркінгу з порівняннями «лоб-у-лоб» із промисловими засобами, а також у формалізації політик «безпечного тестування» для продуктивних середовищ. Отримані результати підтверджують доцільність застосування машинного навчання як керувального елемента в DAST-системах і окреслюють шлях до промислової інтеграції з контрольованими витратами часу та ресурсів.

Список використаних джерел

1. Scikit-learn. (n.d.). *scikit-learn: Machine Learning in Python*. Retrieved November 1, 2025, from https://scikit-learn.org/
2. Keras. (n.d.). *Keras: The Python Deep Learning API*. Retrieved November 1, 2025, from https://keras.io/
3. MITRE. (n.d.). *Common Vulnerabilities and Exposures (CVE)*. Retrieved November 1, 2025, from https://cve.mitre.org/

Відомості про авторів, які взяли участь у 4-ій Міжнародній науково-практичній конференції «Achievements of Science and Applied Research» (10–12 листопада 2025 | Дублін, Ірландія)

Прізвище ім’я по-батькові	Автор 1	Долгова Наталя Геннадіївна
	Автор 2	Єфімов Михайло Юрійович
Для сертифікату: Ім’я та Прізвище латинськими літерами	Автор 1	Natalya Dolgova
	Автор 2	Mykhailo Yefimov
Заклад вищої освіти або організація	Автор 1	Харківський національний економічний університет імені Семена Кузнеця
	Автор 2	Харківський національний економічний університет імені Семена Кузнеця
Науковий ступінь та вчене звання (Для здобувачів вищої освіти рівень: бакалавр/магістр)	Автор 1	кандидат технічних наук
	Автор 2	магістр
Назва кафедри	Автор 1	Кафедра кібербезпеки та інформаційних технологій
	Автор 2	Кафедра кібербезпеки та інформаційних технологій
E-mail	Автор 1	natalya.dolgova@hneu.net
	Автор 2	mishaefimov2002@gmail.com
Назва секції (обов’язково)	Інформаційні технології, кібербезпека та комп’ютерна інженерія	
Паперова версія сертифікату з мокрою печаткою та підписом (оплачується окремо)	Автор 1	ні
	Автор 2	ні