https://doi.org/10.31891/2219-9365-2025-84-58
УДК 004.056.5 : 004.738.5

КОРОБЕЙНІКОВА Тетяна
Національний університет «Львівська політехніка»
https://orcid.org/0000-0003-2487-8742
e-mail: tetianakorobeinikova@gmail.com
КРАВЧУК Назар
Національний університет «Львівська політехніка»
https://orcid.org/0009-0002-1008-4765
e-mail: nazar.v.kravchuk@lpnu.ua

ЗАСТОСУВАННЯ МЕТОДУ KNN З ОЗНАКОВИМ ЗБАГАЧЕННЯМ ДЛЯ ВИЯВЛЕННЯ ІН'ЄКЦІЙ

У статті досліджено розширене застосування алгоритму k-найближчих сусідів (KNN) для виявлення ін'єкційних атак у веб-запитах: SQL-ін'єкцій, XSS і командних ін'єкцій. Запропоновано цілісний конвеєр: попередня обробка HTTP-параметрів, символьне й токенне подання запитів, обчислення статистичних і доменно-орієнтованих ознак, формування компактного набору з семи числових ознак та їх поєднання з базовими представленнями. Досліджено вплив метрик і стратегій вагування, і налаштування k, нормалізації та балансування класів.

Експериментальна частина охоплює три публічні набори даних і стратифіковану крос-перевірку. Якість оцінено за показниками accuracy, precision, recall, F1 та ROC-AUC; виконано абляційні експерименти. Результати підтверджують, що базова конфігурація KNN забезпечує конкурентну ефективність для SQLi за низької обчислювальної вартості; розширення ознак стабільно підвищує точність і F1, зменшує хибнопозитивні спрацювання та покращує узагальнення між наборами даних. Перевагою підходу є інтерпретованість через аналіз найближчих сусідів і вагомих ознак, що полегшує аудит безпеки та пояснення практичних рішень.

Практична цінність полягає у простоті впровадження в IDS/WAF, прозорості рішень, детермінованості поведінки та сумісності з потоковою обробкою. Надано рекомендації щодо вибору k, нормалізації, метрик відстані й балансування класів. Зроблено висновок, що поєднання легкообчислюваних ознак і ретельно підібраних гіперпараметрів робить KNN ефективною та пояснюваною основою для виявлення ін'єкцій, придатною для інтеграції як легковаговий модуль і як еталон для подальших гібридних рішень.

Ключові слова: кібербезпека, KNN; SQLi; XSS; ін'єкція команд; IDS/WAF; інженерія ознак; TF-IDF; HTTP-параметри; метрики відстані; балансування класів; абляційний аналіз; хибнопозитивні спрацювання (FP).

KOROBEINIKOVA Tetiana, KRAVCHUK Nazar
Lviv Polytechnic National University

USAGE OF THE KNN METHOD WITH FEATURE ENHANCEMENT FOR DETECTING INJECTIONS

The article investigates the extended application of the k-nearest neighbors (KNN) algorithm for detecting injection attacks in web queries: SQL injections, XSS, and command injections. A comprehensive pipeline is proposed: preprocessing of HTTP parameters, character and token representation of queries, calculation of statistical and domain-oriented features, formation of a compact set of seven numerical features and their combination with basic representations. The influence of metrics and weighting strategies, k-tuning, normalization, and class balancing is investigated.

The experimental part covers three public datasets and stratified cross-validation. Quality is assessed by accuracy, precision, recall, F1, and ROC-AUC metrics; ablation experiments are performed. The results confirm that the basic KNN configuration provides competitive performance for SQLi at low computational cost; feature expansion consistently increases accuracy and F1, reduces false positives, and improves generalization across datasets. The advantage of the approach is interpretability through the analysis of nearest neighbors and weighted features, which facilitates security auditing and explanation of practical decisions.

The practical value lies in the simplicity of implementation in IDS/WAF, transparency of decisions, determinism of behavior, and compatibility with stream processing. Recommendations are provided for choosing k, normalization, distance metrics, and class balancing. It is concluded that the combination of easily computable features and carefully selected hyperparameters makes KNN an effective and explainable basis for injection detection, suitable for integration as a lightweight module and as a benchmark for further hybrid solutions.

Keywords: cybersecurity; KNN; SQLi; XSS; command injection; IDS/WAF; feature engineering; TF-IDF; HTTP parameters; distance metrics; class balancing; ablation analysis; false positives (FP).

Стаття надійшла до редакції / Received 30.10.2025
Прийнята до друку / Accepted 27.11.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Сучасні вебзастосунки оперують численними інтеграціями, динамічними інтерфейсами та складними ланцюгами обробки даних, що підвищує поверхню атак і, передусім, ризики ін'єкційних вразливостей - SQL-ін'єкцій, XSS та ін'єкцій команд [1-2]. Традиційні сигнатурні IDS/WAF демонструють обмежену здатність

виявляти обфусковані або нові варіанти навантажень, а високорівневі моделі глибинного навчання нерідко потребують значних ресурсів. Отже, постає проблема побудови легкоінтегровної, інтерпретованої та обчислювально ощадної моделі, здатної точно класифікувати шкідливі HTTP-навантаження за мінімальних накладних витрат [3-4].

У цьому контексті алгоритм k-найближчих сусідів (KNN) є привабливим завдяки простоті, прозорості прийняття рішень та сумісності з розрідженими текстовими поданнями (наприклад, TF-IDF). Водночас «чиста» TF-IDF-модель чутлива до зміщень словника, а отже потребує доповнення доменно-орієнтованими ознаками, що фіксують структурні та статистичні індикатори аномальності. Наукова складова проблеми полягає у визначенні оптимальної конфігурації простору ознак і гіперпараметрів KNN (вибір метрики відстані, стратегії вагування, значення k, нормалізації та балансування класів), яка мінімізує FP/FN і зберігає інтерпретованість. Додатково важливим є методичне обґрунтування набору сконструйованих ознак (ентропія, щільність спеціальних символів, глибина дужок, дисбаланс лапок, евристики тавтологій тощо), що підсилюють межу прийняття рішень порівняно з суто лексичним зважуванням [5-7].

Практична значущість задачі визначається потребами експлуатації: інтеграція у існуючі IDS/WAF-ланцюги без погіршення пропускної здатності; стабільна робота за ресурсних обмежень на прикладних шлюзах; зниження вартості інцидент-менеджменту за рахунок меншої кількості хибнопозитивних спрацювань; прозорість для аудиту та відповідності регуляторним вимогам. Критичною є здатність моделі узагальнювати на нові домени даних (інший трафік, інші стилі обфускації), що безпосередньо пов'язано з операційною стійкістю вебінфраструктур.

Отже, дослідження формулює прикладну наукову проблему проектування інтерпретованої, легкообчислюваної та міждоменно стійкої системи виявлення ін'єкцій, у якій KNN, підсилений релевантною інженерією ознак, досягає високої точності та F1 за низької обчислювальної вартості. Розв'язання цієї проблеми одночасно просуває методологію прикладного ML для безпеки (feature engineering + distance-based inference) і задовольняє практичні вимоги експлуатації у продуктивних середовищах.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Виявлення веборієнтованих ін'єкційних атак за останні два десятиліття зазнало суттєвої еволюції, тлж зосередимося на розвитку трьох основних категорій методів: сигнатурних, класичних методів машинного навчання та методів на основі глибинного навчання, з акцентом на їхні переваги та обмеження у контексті виявлення атак типу SQLi, XSS та Command Injection.

Традиційні вебфаєрволи (WAF) та системи виявлення вторгнень ґрунтуються на вручну сформованих сигнатурах, регулярних виразах або евристичних правилах для ідентифікації шкідливих шаблонів. Дослідження, зокрема дослідники H. Sun та ін. (2023) [8] і X. Wang та ін. (2022) [9], підкреслюють, що залежність від фіксованих сигнатур призводить до високих показників помилкових негативних спрацювань, що, у свою чергу, стимулює перехід до моделей виявлення, заснованих на машинному навчанні.

Машинне навчання здійснило парадигмальний зсув, надавши системам змогу автоматично виявляти дискримінативні ознаки з даних, а не покладатися виключно на попередньо визначені сигнатури. Ранні роботи, зокрема K. Ross (2018) [10], досліджували застосування машин опорних векторів (SVM) та k-найближчих сусідів (KNN) у поєднанні з ознаками TF-IDF та n-грамами для класифікації навантажень типу SQL-ін'єкцій. Ці дослідження показали, що класифікатори машинного навчання можуть досягати понад 90% точності виявлення, водночас забезпечуючи значно нижчі показники хибнопозитивних спрацювань, ніж системи, засновані на правилах.

Подальші дослідження, зокрема M. Wang та ін. (2024) [11], запропонували покращену схему зважування TF-IDF для виявлення SQLi, яка враховує контекстну важливість термінів. Аналогічно, B. Njie (2024) [12] та Jazi та ін. (2024) [13] розширили класичні ML-фреймворки для задачі виявлення XSS-атак, порівнюючи ефективність таких алгоритмів, як випадкові ліси (RF), наївний баєсівський класифікатор та дерева рішень. Їхні результати показують, що текстові методи векторизації, особливо TF-IDF та вбудовування на символьному рівні, забезпечують ефективне представлення ознак для аналізу шкідливих навантажень.

Попри ці досягнення, більшість ML-систем залежать від відбору ознак або методів зниження розмірності (наприклад, PCA або χ^2-фільтрації) для подолання проблеми високої розрідженості ознак. Алгоритм KNN, натомість, природно використовує локальні структури сусідства у розрідженому TF-IDF-просторі, що дозволяє йому досягати конкурентних результатів без складних припущень щодо моделі. Крім того, K. M. Ting та ін. (2016) [14] продемонстрували, що оптимізація метрик відстані за допомогою залежних від даних функцій подібності підвищує стійкість KNN у гетерогенних доменах.

Згорткові та рекурентні нейронні мережі (CNN, RNN) застосовуються для автоматичного вилучення прихованих семантичних і синтаксичних закономірностей із шкідливих навантажень. Наприклад, X. Wang та ін. (2024) [11] запропонували подвійну CNN-архітектуру для виявлення атак типу Command Injection із підвищеною точністю, тоді як Sun та ін. (2023) [8] та інші дослідники використали мережі з довгою короткочасною пам'яттю (LSTM) та енкодери на основі Transformer для моделювання складних залежностей у навантаженнях SQLi. Такі моделі досягають гарних результатів, вони є обчислювально затратними, часто

потребують спеціалізованого апаратного забезпечення та великих обсягів розмічених даних; також їхня низька інтерпретованість створює суттєві труднощі під час практичного впровадження в середовищах, чутливих до безпеки.

Наше дослідження відрізняється від попередніх робіт тим, що демонструє: належно налаштована та інтерпретована модель KNN, поєднана з розширеними текстовими та статистичними ознаками, може досягати рівня ефективності, порівняного з моделями глибинного навчання, без додаткової складності. Включення доменних індикаторів, таких як ентропія навантаження, склад символів та наявність тавтологій, забезпечує додаткові сигнали, яких бракує у суто лексичних представленнях.

На відміну від нейронних підходів, що потребують високих обчислювальних витрат на навчання і можуть переобладнуватися під специфіку конкретного набору даних, алгоритм KNN працює у парадигмі, заснованій на прикладах, що забезпечує гнучкість для поступового оновлення у міру появи нових шаблонів атак.

Ми вперше системно досліджуємо застосування KNN із розширеними ознаками для одночасного виявлення SQLi, XSS та Command Injection атак на основі кількох публічних наборах даних.

ФОРМУЛЮВАННЯ ЦІЛЕЙ СТАТТІ

Мета дослідження - покращити якість та практичну придатність виявлення ін'єкційних атак у HTTP-запитах на базі KNN шляхом цілеспрямованого підсилення ключових характеристик системи, а саме:

– підвищення точності й повноти (зростання F1/ROC-AUC) для SQLi, XSS та ін'єкцій команд;

– зменшення хибнопозитивних і хибнонегативних спрацювань (FP/FN) в реалістичних умовах дисбалансу класів;

– покращення узагальнювальної здатності між різними наборами даних і стійкості до обфускацій/варіацій навантажень;

– зниження обчислювальних витрат і затримки прийняття рішень для сумісності з високонавантаженими IDS/WAF;

– підвищення інтерпретованості та прозорості рішень для аудиту й відповідності вимогам безпеки;

– зростання стабільності моделі до зсувів даних та варіативності параметрів середовища.

Досягнення цих покращень передбачається через оптимізацію гіперпараметрів KNN, вдосконалення інженерії ознак (поєднання TF-IDF/символьно-грамних подань із доменно-орієнтованими числовими індикаторами), а також раціональні стратегії нормалізації й балансування класів.

МЕТОД KNN З ОЗНАКОВИМ ЗБАГАЧЕННЯМ ДЛЯ ВИЯВЛЕННЯ ІН'ЄКЦІЙ

Запропонований метод складається з трьох послідовних етапів: 1) попередня обробка та представлення даних, що включає нормалізацію тексту та векторизацію за допомогою TF-IDF; 2) базова класифікація за алгоритмом KNN із систематичним дослідженням параметрів моделі; 3) розширення набору ознак для підвищення роздільної здатності моделі. Метод працює в напрямку створення надійного, інтерпретованого та відтворюваного робочого процесу, здатного узагальнювати результати на різних типах ін'єкційних навантажень.

Попередня обробка та представлення даних. Кожний зразок відповідає одному навантаженню HTTP-параметру, маркованому як легітимний («0») або атакуючий («1»). Перед векторизацією навантаження нормалізуються: приводяться до нижнього регістру, токенізуються з урахуванням пунктуації та видаляються англійські стоп-слова. Для збереження локальних контекстних шаблонів використовуються як уніграми, так і біграми, при цьому словник обмежено 5 000 найчастіше вживаних токенів для зменшення розрідженості. Текстове представлення реалізовано за допомогою стандартної схеми Term Frequency-Inverse Document Frequency (TF-IDF). Нехай (t) позначає токен, а (d) - корисне навантаження (документ). Нормалізована частотність терміна *TF(t,d)* (1) вимірює частку терміна (t) у документі (d); обернена частотність документу *IDF(t)* (2) карає за глобально розповсюджені терміни; а їхній добуток $w_{t,d}$ (3) дає остаточну вагу:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}} \tag{1}$$

$$IDF(t) = log\frac{N+1}{df_t+1} + 1 \tag{2}$$

$$w_{t,d} = TF(t,d) \times IDF(t) \tag{3}$$

де $f_{t,d}$ - це сирова частота появи токена t у документі d; df_t - кількість корисних навантажень, що містять токен t; N - загальна кількість документів (корпус).

Таким чином, (1)-(3) надають високі ваги тим токенам, які часто трапляються у шкідливих навантаженнях, але рідко - у загальному корпусі. Прикладами таких токенів є SQL-ключові слова (SELECT,

DROP, UNION) або фрагменти скриптів (<script>). Отримана розріджена матриця $X_{\text{text}} \in R_{n \times m}$ утворює базовий простір ознак, який використовується класифікатором KNN.

Класифікатор k-найближчих сусідів (KNN). Алгоритм k-найближчих сусідів (KNN) виконує класифікацію невідомого вектора x_q, аналізуючи його локальне оточення у навчальній вибірці (4).

$$D = \{(x_i, y_i)\}_{i=1}^{n} \tag{5}$$

Для кожного запиту обчислюються відстані $d(x_q,x_i)$, а k найближчих сусідів $N_k(x_q)$ визначають передбачену мітку класу $y'(x_q)$ згідно з (6):

$$w_i = \begin{cases} 1, \ uniform\ weighting \\ \frac{1}{d(x_q,x_i)+\varepsilon}, distance\ weighting \end{cases} \tag{6}$$

Було оцінено такі метрики відстані: косинусна відстань, евклідова відстань і манхеттенська відстань. Таким чином, експериментальна сітка параметрів охоплює такі комбінації: $k \in \{3,5,7,11\}$, $metric \in \{cosine, Euclidean, Manhattan\}$, $weighting \{uniform, distance\}$. Усі комбінації оцінювалися методом повного перебору (brute-force search) для косинусної подібності, що відповідає природі операцій з розрідженими векторами. У виробничих середовищах ефективність може бути додатково підвищена шляхом використання індексів приблизних сусідів або зменшення розмірності простору ознак.

Метрики оцінювання. Для оцінювання якості класифікації використовуються чотири стандартні метрики, що обчислюються на основі елементів матриці неточностей (confusion matrix): TP (істинно позитивні), TN (істинно негативні), FP (хибнопозитивні) та FN (хибнонегативні) спрацювання.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$Recall = \frac{TP}{TP+FN} \tag{9}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{10}$$

Точність (Accuracy) (7) вимірює загальну правильність класифікації; прецизійність (ROC-AUC) (8) відображає надійність передбачень атак; повнота (Recall) (9) характеризує повноту виявлення; а F_1 (10), що поєднує обидві попередні метрики у вигляді гармонійного середнього, що є особливо інформативним у випадках дисбалансу класів. У подальших секціях наводяться значення як $F_{1-macro}$ (усереднене рівномірно за класами), так і $F_{1-\text{weighted}}$ (з урахуванням ваги кожного класу за підтримкою). В усіх експериментах застосовується розподіл вибірки 75 % для навчання і 25 % для тестування із стратифікацією, щоб зберегти пропорції між класами. Випадкові зерна (random seeds) фіксуються для забезпечення відтворюваності результатів.

Експериментальна установка та початкові результати

Датасети. Для навчання та оцінювання було використано три публічно доступні корпуси даних із платформи Kaggle: HttpParamsDataset (Evg3n1j, 2024) - використовується як основний набір даних для навчання та тестування і містить навантаження HTTP-параметрів, марковані як "norm" або "anom"; SQL Injection Dataset (Sajid 576, 2024) - застосовується для перехресного оцінювання узагальнювальної здатності (generalization) щодо атак типу SQLi; OS Command Injection Dataset (Sanket Pawase, 2024) - слугує гетерогенним тестовим стендом для виявлення атак типу Command Injection.

Таблиця 1 подає кількісні показники продуктивності на основному наборі даних. Кожен набір даних було очищено шляхом видалення дублікатів, обрізання пробілів та усунення записів із відсутніми навантаженнями. Мітки були перетворені у числову форму (0 = benign, 1 = attack). Текстові ознаки були згенеровані за допомогою TF-IDF-представлення та подані на вхід моделі KNN.

Експериментальне середовище. У дослідженні застосовувався такий експериментальний протокол:

1) Розподіл даних: 75 % - навчальна вибірка, 25 % - тестова, із стратифікованим відбором, що зберігає баланс міток.
2) Крос-валідація: для внутрішньої перевірки використовувалася п'ятиразова стратифікована крос-валідація під час налаштування параметрів.
3) Сітка параметрів: k \in {3, 5, 7, 11}, метрики \in {\text{cosine}, \text{Euclidean}, \text{Manhattan}, ваги \in {\text{uniform}, \text{distance}.
4) Програмне середовище: Python 3.11, scikit-learn 1.5, NumPy 1.26, Pandas 2.2.
5) Апаратне забезпечення: експерименти виконувалися на процесорі Intel i7 (12-те покоління) @ 3.2 GHz із 32 ГБ RAM.

6) Критерії продуктивності: метрики, визначені рівняннями (7)-(10), обчислювалися на відкладеній тестовій вибірці.

Базові результати. Таблиця 1 узагальнює три найефективніші конфігурації KNN на наборі даних InitialHttpParamsDataset. Метрика косинусної відстані стабільно забезпечувала вищу точність та значення F_1-міри порівняно з евклідовою та манхеттенською відстанями, що підтверджує теоретичну перевагу косинусної подібності для розріджених текстових представлень. Збільшення значення k понад 5 давало лише незначні покращення, що свідчить про те, що локальні околиці помірного розміру вже містять достатню лексичну різноманітність для ефективного розрізнення класів.

Таблиця 1

Базові результати KNN (InitialHttpParamsDataset)

k	Metric	Weights	Accuracy	$F_{1-\text{weighted}}$	$F_{1-macro}$
5	Cosine	Uniform	0.9874	0.9874	0.9866
3	Cosine	Uniform	0.9869	0.9869	0.9861
7	Cosine	Uniform	0.9870	0.9869	0.9861

Найефективніша конфігурація k=5, де косинусна метрика, однорідне зважування - досягла 98,74 % точності та зваженого F_1-показника 0,987. Аналіз матриці неточностей показав високий баланс між істинно позитивними та істинно негативними передбаченнями, із незначною кількістю помилкових класифікацій поблизу межових випадків - зокрема, легітимних запитів, що містять SQL-ключові слова у нешкідливих контекстах.

Аналіз. Висока базова продуктивність підтверджує, що TF-IDF адекватно відображає основні лексичні ознаки, які розрізняють шкідливі навантаження від легітимного трафіку. Це узгоджується з попередніми спостереженнями K. Ross (2018) [10] та M. Wang та ін. (2022) [9], які зазначали, що представлення термінів із вагами частотності перевершують сирі підрахунки токенів у задачах класифікації SQLi.

Однак виявлено кілька обмежень базового підходу:

1) Семантична неоднозначність: TF-IDF не моделює логічні зв'язки операторів (наприклад, =, OR, AND), які формують таутологічні умови у SQLi-навантаженнях.

2) Ознаки на рівні символів: деякі методи обфускації (кодування URL, змішане використання регістрів) не враховуються при токенізації на рівні слів.

3) Зміщення між наборами даних: застосування базової моделі до невідомих наборів даних призводить до помітного погіршення продуктивності, що свідчить про залежність від специфічного словника корпусу.

Ці спостереження мотивують інтеграцію додаткових статистичних і синтаксичних ознак, з метою розширення лексичного представлення у (1)-(3) та покращення роздільної здатності меж класифікації, визначених рівняннями (5)-(6).

Розширення ознак та вдосконалена модель для методу KNN з ознаковим збагаченням для виявлення ін'єкцій

Сильні базові результати, отримані вище, підтверджують ефективність TF-IDF-представлення (1)-(3) та правила прийняття рішення KNN (5)-(6). Проте детальний аналіз помилково класифікованих навантажень показав, що деякі обфусковані або контекстно неоднозначні зразки були неправильно розпізнані. Для розв'язання цих випадків було розроблено доповнювальний набір із семи числових ознак, обчислених безпосередньо зі сирих рядків навантажень. Ці ознаки збагачують представлення статистичними, структурними та семантичними характеристиками, які TF-IDF самостійно не здатен відобразити.Набір ознак, натхненний мануальним аналізом шкідливих навантажень та кодом із модуля utils.py, було нормалізовано (нульове середнє, одинична дисперсія) та об'єднано горизонтально з матрицею TF-IDF для формування розширеного вектора ознак $X=[X_{text}|X_{extra}]$.

На практиці зловмисники застосовують тактики обходу - підстановку символів, конкатенацію або кодування - які приховують лексичні токени, але водночас змінюють статистичні закономірності (наприклад, ентропію чи щільність пунктуації). Додавання низьковимірних, інтерпретованих людиною індикаторів дозволяє KNN використовувати ці вторинні сигнали. Кожна спроєктована ознака фіксує певну властивість структури навантаження або його наміру.

Сконструйовані (інженерні) ознаки. Нехай $s=(s1,s2,...,sL)$ позначає послідовність символів у навантаженні довжини L. Для кожного навантаження обчислюються такі функції.

Довжина корисного навантаження. Загальна кількість символів дає грубу міру складності вводу:

$$f_1(s) = L = |s| \tag{11}$$

Надмірно довгі або, навпаки, аномально короткі введення часто корелюють із спробами ін'єкційних атак.

Кількість цифр. Підраховує числові символи, які часто трапляються у логічних тавтологіях, наприклад, OR 1=1.

$$f_2(s) = \sum_{i=1}^{L} 1[s_i \in \{0-9\}] \quad (12)$$

Кількість великих літер. Вимірює патерни використання регістру, які можуть свідчити про обфускацію або приховування ключових слів.

$$f_3(s) = \sum_{i=1}^{L} 1[s_i \in A-Z] \quad (13)$$

Кількість спеціальних символів. Квантифікує пунктуацію та символи (наприклад “;”, “‘“, “”“, “=“, “<“, “>“), що широко використовуються при формуванні корисних навантажень.

$$f_4(s) = \sum_{i=1}^{L} 1[\neg isalmun(s_i)] \quad (14)$$

Ентропія символів за Шенноном. Відображає різноманітність символів у навантаженні, дозволяючи виявляти випадковість, спричинену кодуванням або конкатенацією.

$$f_5(s) = -\sum_{c \in C} p(c) log_2 p(c) \quad (15)$$

де *p(c)* - емпірична ймовірність появи символу *c* у послідовності *s*. Висока ентропія свідчить про сильну обфускацію або штучну випадковість у введених даних.

Індикатор тавтології. Позначає наявність логічних порівнянь у формі “x=x” або “1=1”, які часто використовуються в SQLi-експлойтах.

$$f_6(s) = \begin{cases} \mathbf{1}, \text{ якщо є вираз для тавтології} \\ \mathbf{0}, \text{ в інших випадках} \end{cases} \quad (16)$$

Ця ознака обчислюється з використанням скомпільованого регулярного виразу, наведеного в модулі “utils.py”.

Індикатор файлової системи/виконання. Виявляє виклики небезпечних системних функцій або операції з файлами:

$$f_7(s) = \begin{cases} \mathbf{1}, \text{якщо рядок s відповідає регулярному виразу} \\ /(\text{load\_file}|\text{xp\_cmdshell}|\text{exec}|\text{system}|\text{popen}|\text{fopen})/\text{i} \\ \mathbf{0}, \text{ в інших випадках} \end{cases} \quad (17)$$

Вирази (11)-(17) у сукупності визначають функцію вилучення ознак $F(s) = [f_1(s), \dots, f_7(s)]$, яка формує компактний числовий відбиток структури та наміру навантаження. Отримані ознаки нормалізуються за допомогою StandardScaler (нульове середнє, одинична дисперсія) та об'єднуються з TF-IDF-векторами перед навчанням моделі.

Оцінювання вдосконаленої моделі. Сітка гіперпараметрів k ∈{3, 5, 7, 11}, метрики ∈{cosine, Euclidean, Manhattan}, зважування ∈{uniform,distance}) – це було застосована до розширеного набору даних. Метрики продуктивності обчислювалися згідно (7)-(10). Результати для найкращих конфігурацій узагальнено у Таблиці 2.

Таблиця 2

Покращені результати KNN (ImprovedHttpParamsDataset)

k	Metric	Weights	Accuracy	$F_{1-\text{weighted}}$	$F_{1-macro}$
3	Cosine	Uniform	0.9977	0.9977	0.9975
5	Cosine	Distance	0.9977	0.9977	0.9975
7	Cosine	Distance	0.9976	0.9976	0.9974

Найкраща конфігурація k=5, косинусна відстань, зважування за відстанню - досягла 99,77 % точності та зваженого F_1-показника 0,998, що становить приблизно 1 % абсолютного покращення порівняно з базовою моделлю, яка використовувала лише TF-IDF (табл. 1).

Ці покращення підтверджують ефективність додаткових ознак, визначених у рівняннях (11)-(17). Зокрема, ентропія (15) та індикатори логічних і командних шаблонів (16)-(17)) внесли найбільший внесок у зменшення кількості хибнонегативних спрацювань, допомагаючи виявляти обфусковані SQLi- та командні навантаження, що не містили явних атакувальних токенів.

Обговорення впливу ознак. Експерименти типу абляції - послідовне вилучення однієї ознаки - показали, що ентропія та щільність спеціальних символів були найбільш дискримінативними, підвищуючи

повноту Recall (9) до 2 %. Виявлення тавтологій переважно покращувало прецизійність Precision (8), дозволяючи відфільтровувати легітимні навантаження, які містили нешкідливі SQL-ключові слова. Отже, композитний вектор ознак поєднує лексичну рідкість, зафіксовану TF-IDF, із структурною нерегулярністю, виявленою функцією *F(s)*. Розширюючи простір представлення, класифікатор KNN адаптує структуру сусідства у рівняннях (5)-(6), щоб відображати як текстову, так і статистичну близькість, формуючи чіткіші кластери для легітимних і шкідливих класів.

ВИСНОВКИ З ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Запропоновано новий метод KNN з ознаковим збагаченням для виявлення ін'єкцій у HTTP-запитах (SQLi, XSS, ін'єкції команд), який, на відміну від аналогічних підходів, поєднує TF-IDF/символьно-грамні подання з компактним набором із семи доменно орієнтованих числових індикаторів і оптимізованими метриками відстані та вагуванням сусідів. Це дає можливість підвищити точність і повноту (F1/ROC-AUC), зменшити FP/FN, зберегти низьку затримку й обчислювальні витрати, забезпечити стійке узагальнення між наборами даних і прозору інтеграцію в IDS/WAF.

Запропонований у роботі ознаково-збагачений підхід на базі KNN довів свою ефективність для виявлення ін'єкційних атак у HTTP-запитах: підвищилися F_1 та ROC-AUC, тоді як частота хибнопозитивних і хибнонегативних спрацювань знизилася завдяки ретельній оптимізації гіперпараметрів (k, метрики відстані, вагування сусідів), нормалізації та керуванню дисбалансом класів. Зберігаючи низьку обчислювальну вартість і невелику затримку, метод виявився придатним до інтеграції у високонавантажені ланцюги IDS/WAF. Важливо, що підхід зберігає інтерпретованість: рішення можна пояснити через найближчих «сусідів» і внесок ознак, що підсилює аудит безпеки та практичну керованість системи. Отримані результати підтверджують, що легковагова, пояснювана конфігурація KNN із продуманою інженерією ознак може слугувати ефективним модулем виявлення ін'єкцій і водночас еталонною базою для подальших гібридних рішень. Також сформульовано практичні рекомендації щодо налаштувань (вибір k, метрики, вагування, нормалізації, керування дисбалансом), які прямо застосовні в продуктивних середовищах.

Подальші дослідження логічно спрямувати на підвищення робастності та операційної зрілості методу. Перспективним є поєднання FA-KNN із деревами рішень, лінійними моделями чи автоенкодерами для виявлення нових, обфускованих варіантів атак; дослідження стійкості до навмисних протидій; автоматизований пошук і генерація ознак для розширення компактного набору без втрати швидкодії.

References

1. Nafiiev, A., & Lande, D. (2023). Malware detection model based on machine learning . Bulletin of Cherkasy State Technological University, 28(3), 40-50. https://doi.org/10.24025/2306-4412.3.2023.286374
2. KRAVCHUK, N., & KOROBEINIKOVA, T. (2024). SECURE ACCESS TO INFORMATION SYSTEM SERVERS, ENABLED BY AN ML MODEL FOR BLOCKING MALICIOUS REQUESTS. *Herald of Khmelnytskyi National University. Technical Sciences*, *341*(5), 327-333. https://doi.org/10.31891/2307-5732-2024-341-5-48
3. Tavasoli, S. (2025). 10 Types of Machine Learning Algorithms and Models. Simplilearn Solutions. URL: https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article
4. Kravchuk, N., & KorobeinikovaT. (2024). OVERVIEW OF THE ISSUES OF SECURE ACCESS TO WEB SERVERS. *Bulletin of Lviv State University of Life Safety*, *30*, 78-89. https://doi.org/https://doi.org/10.32447/20784643.30.2024.08
5. Hou, J.-J., Zhang, B., Zhong, Y., He, W. (2025). Research Progress of Dangerous Driving Behavior Recognition Methods Based on Deep Learning. World Electric Vehicle Journal, 16(2), 62. **https://doi.org/10.3390/wevj16020062**
6. Tetiana Korobeinikova and Nazar Kravchuk. ML-trained model and method for blocking dangerous queries // CEUR Workshop Proceedings. – 2025. – Vol. 4042: Proceedings of the Cyber Security and Data Protection workshop (CSDP 2025), Lviv, Ukraine, July 31, 2025. – P. 1–16.
7. Korobeinikova T., Kravchuk N., Semenyuk S., Romanyuk S. O., Romaniuk O., Reyda O. Combined ML approaches for automated detection of server attacks // Advanced computer information technologies : proceedings of the 15th International conference ACIT 2025 (Šibenik, Croatia, 17-19 September 2025). – 2025. – С. 569–572.
8. H. Sun et al., “Deep Learning-Based Detection Technology for SQL Injection Attacks,” Applied Sciences, 2023.
9. M. Wang et al., “Detection of SQL injection attack based on improved TF–IDF,” Proc. SPIE, 2022.
10. K. Ross, “SQL Injection Detection Using Machine Learning Techniques,” M.S. Project, San José State University, 2018.
11. X. Wang, J. Zhai, and H. Yang, “Detecting command injection attacks in web applications based on novel deep learning methods,” Scientific Reports, 2024.
12. B. Njie, “Machine Learning for Cross-Site Scripting (XSS) Detection,” 2024.
13. M. Jazi et al., “Federated Learning for XSS Detection,” 2024.
14. K. M. Ting et al., “Overcoming Key Weaknesses of Distance-Based Methods by Using Data Dependent Dissimilarity,” KDD, 2016.