Застосування нейронних мереж для вибору інструментів для тестування на проникнення

Application of Neural Networks for Selecting Tools for Penetration Testing

Андріан Піскозуб [A]
Corresponding author: к. тех. Н., доцент кафедри захисту інформації, e-mail: azpiskozub@gmail.com, ORCID: 0000-0002-3582-2835

Анастасія Журавчак [A]
аспірантка, асистентка кафедри захисту інформації, e-mail: anastasiia.y.tolkachova@lpnu.ua, ORCID: 0000-0002-8196-7963

Даниїл Журавчак [A]
доктор філософії, асистент кафедри захисту інформації, e-mail: danyil.y.zhuravchak@lpnu.ua, ORCID: 0000-0003-4989-0203

Юрій Журавчак [A]
магістр, кафедра інформаційних систем та технологій, e-mail: yurii.zhuravchak.mitis.2023@lpnu.ua, ORCID: 0009-0003-2378-5365

Ігор Беляєв [B]
аспірант, асистент кафедри кібербезпеки, e-mail: igor@beliaiev.com, ORCID: 0009-0005-8130-7972

Andriian Piskozub [A]
Corresponding author: PhD. in Engineering, Associate Professor, Department of Information Security, e-mail: azpiskozub@gmail.com, ORCID: 0000-0002-3582-2835

Anastasiia Zhuravchak [A]
PhD student, Associate Assistant, Department of Information Security, e-mail: anastasiia.y.tolkachova@lpnu.ua, ORCID: 0000-0002-8196-7963

Danyil Zhuravchak [A]
PhD, Associate Professor, Department of Information Security, e-mail: danyil.y.zhuravchak@lpnu.ua, ORCID: 0000-0003-4989-0203

Yurii Zhuravchak [A]
Master's Degree, Institute of Computer Science and Information Technology, e-mail: yurii.zhuravchak.mitis.2023@lpnu.ua, ORCID: 0009-0003-2378-5365

Igor Beliaiev [B]
PhD student, Associate Assistant, Department of Cybersecurity, e-mail: igor@beliaiev.com, ORCID: 0009-0005-8130-7972

[A] Національний університет "Львівська політехніка", м. Львів, Україна
[B] Львівський Національний університет імені Івана Франка, м. Львів, Україна

[A] Lviv Polytechnic National University, Lviv, Ukraine
[B] Ivan Franko National University of Lviv, Lviv, Ukraine

Received: June 13, 2025 | Revised: August 23, 2025 | Accepted: August 31, 2025

DOI: https://doi.org/10.33445/sds.2025.15.4.12

Мета роботи. Розробити метод автоматизованого вибору інструментів для тестування на проникнення веб-додатків за допомогою нейронних мереж.

Метод дослідження. Побудова нейронної мережі прямого поширення з навчанням за методом зворотного поширення помилки на основі даних експертів та користувачів, поданих у вигляді матриці характеристик інструментів. Реалізація моделі через веб-сервіс з використанням стеку LAMP і бібліотеки FANN.

Результати дослідження. Створено веб-застосунок, що дозволяє користувачам задавати критерії до інструментів тестування, а система надає відповідні рекомендації. Навчена нейронна мережа демонструє ефективність у виборі утиліт згідно з вхідними параметрами, що підтверджується експериментами з утилітами Acunetix, Nessus та Nexpose. Система враховує як експертні дані, так і користувацький зворотний зв'язок, що забезпечує її динамічну адаптацію.

Теоретична цінність дослідження. Обґрунтовано ефективність застосування нейромереж для автоматизованого підбору інструментів у сфері кібербезпеки, що відкриває нові підходи до інтеграції машинного навчання у процеси тестування на проникнення.

Практична цінність дослідження. Розроблений веб-сервіс може використовуватись як допоміжний інструмент тестувальниками безпеки, зокрема початківцями, для швидкого та обґрунтованого вибору засобів тестування.

Цінність дослідження. Дослідження демонструє, що застосування нейромереж дозволяє підвищити ефективність вибору інструментів та спростити процес прийняття рішень при тестуванні веб-додатків.

Майбутні дослідження. Удосконалення архітектури моделі, пояснюваність рішень нейромережі, масштабування системи на більші обсяги даних та розширення спектра інструментів.

Тип статті. Прикладне дослідження.

Purpose. To develop a method for the automated selection of penetration testing tools for web applications using neural networks.

Method. Construction of a feedforward neural network trained with the backpropagation algorithm using expert and user data represented as a matrix of tool characteristics. Implementation of the model through a web service using the LAMP stack and FANN library.

Findings. A web application was developed that allows users to specify criteria for testing tools, and the system provides appropriate recommendations. The trained neural network demonstrates effectiveness in selecting utilities based on input vectors, confirmed by experiments with Acunetix, Nessus, and Nexpose. The system incorporates both expert data and user feedback, ensuring its dynamic adaptation.

Theoretical implications. The study substantiates the effectiveness of neural networks for the automated selection of tools in cybersecurity, paving the way for new approaches to integrating machine learning into penetration testing processes.

Practical implications. The developed web service can be used as an auxiliary tool by security testers, especially beginners, for fast and justified selection of testing tools.

Value. The study shows that the application of neural networks increases the efficiency of tool selection and simplifies decision-making during web application testing.

Future research. Improving the model architecture, explainability of neural network decisions, scaling the system to larger datasets, and expanding the toolset.

Paper type. Applied study.

Ключові слова: штучний інтелект, тестування на проникнення, інструменти безпеки, нейронні мережі, автоматизація, машинне навчання.

Key words: artificial intelligence, penetration testing, security tools, neural networks, automation, machine learning.

Вступ

У даній науковій статті розглядається актуальна проблема вибору інструментальних засобів для проведення тестування на проникнення веб-додатків. Тестування на проникнення спрямоване на виявлення та усунення вразливостей, а використання інструментальних засобів дозволяє автоматизувати рутинні операції. Однак, тестувальники, особливо початківці, стикаються зі складнощами у виборі найбільш підходящих утиліт для конкретних завдань безпеки, оскільки існує багато схожих інструментів. Стаття починається з аналізу проблематики вибору утиліт для тестування безпеки, зокрема процесів формування власних наборів інструментів досвідченими тестувальниками. Автори пропонують метод вирішення цієї проблеми за допомогою створення веб-сервісу, що використовує нейромережу для аналізу критеріїв вибору та рекомендацій інструментів. Нейромережу навчають на основі даних, зібраних експертами з тестування безпеки, у вигляді матриці утиліт та їхніх характеристик. Користувач сервісу може зазначити критерії для пошуку інструментів, а система надасть кілька найвідповідніших варіантів. Також враховується можливість коригування вибору користувачем, що дозволяє розширювати базу навчальних даних. Запропонований підхід включає створення двох моделей нейромереж: одна навчається виключно на даних експертів, а інша враховує також дані користувачів. Це забезпечує динамічну адаптацію системи до нових вимог і зворотного зв’язку. Результати дослідження демонструють, що використання нейромереж для вибору інструментів дозволяє підвищити ефективність тестування безпеки. Запропонований метод може бути застосований не лише для вибору засобів тестування веб-додатків, а й для інших сфер програмного забезпечення.

Теоретичні основи дослідження

Сьогодні широкого використання набули інструменти для автоматизації тестування на проникнення. Вони дозволяють зменшити обсяг рутинних завдань і підвищити ефективність роботи. Водночас різноманітність таких інструментів і їхніх функціональних можливостей створює проблему вибору оптимального рішення для конкретних завдань. Наприклад, одні сканери чудово знаходять вразливості, пов’язані з конфігурацією серверів, але пропускають помилки логіки застосунку. Інші можуть мати високий відсоток хибнопозитивних спрацювань [1]. Таким чином, вибір інструментів залишається критичною точкою в проведенні тестувань, який потребує значного часу на аналіз доступних рішень та їхнє порівняння. Використання нейромереж може стати вирішенням цієї проблеми. Вони здатні обробляти великі обсяги даних та підбирати найкращий інструмент для тестування. Нейромережі також можуть допомагати в класифікації вразливостей, прогнозуванні можливих атак і покращенні ефективності використання існуючих інструментів [2]. У сучасних умовах проблема вибору ефективних інструментів для тестування набуває дедалі більшої актуальності. Існуючі рішення часто мають обмежену сферу застосування, а їхній вибір потребує значних витрат часу та експертних знань. Водночас автоматизація цього процесу із застосуванням методів нейромереж залишається недостатньо дослідженою, що стримує її впровадження у практику.

Головною проблемою, що стоїть перед дослідниками, є недостатня адаптованість нейромереж до змінюваних умов і завдань пентесту. Також залишається відкритим питання інтеграції таких рішень у процес пентесту без втрати ефективності та точності. Ці виклики потребують комплексного аналізу та пошуку інноваційних рішень, що дозволять значно покращити якість тестування на проникнення.

Постановка проблеми

У сучасних умовах кіберзагроз тестування на проникнення (penetration testing) стало ключовим елементом забезпечення безпеки веб-додатків. Одним із критичних аспектів цього

процесу є правильний вибір інструментів для виявлення вразливостей, адже від нього залежить як повнота, так і точність аналізу. Ринок пропонує велику кількість утиліт, що мають схожі функціональні можливості, однак різну ефективність у залежності від типу атак, що ускладнює вибір, особливо для початківців.

У роботі [2] P. Gallus та ін. досліджується застосування генеративних нейронних мереж, таких як ChatGPT, для автоматизації процесів тестування на проникнення веб-додатків. Основну увагу приділено використанню цих моделей для генерації сценаріїв атак і автоматичного виявлення вразливостей у популярних платформах, зокрема WordPress. Авторами підкреслено ефективність таких підходів у підвищенні швидкості аналізу безпеки та зниженні вимог до експертних знань тестувальників. Однак, робота обмежується розглядом лише генеративних моделей і не охоплює їх інтеграцію з іншими методами тестування на проникнення, такими як аналіз поведінкових моделей або розширення до багатокомпонентних архітектур.

У дослідженні [3] K. Pozdniakov та ін. запропоновано підхід до автоматизації аудиту безпеки з використанням методів підкріплюючого навчання у поєднанні з глибокими нейронними мережами. Основна ідея роботи полягає у застосуванні алгоритму Q-навчання для визначення оптимальних стратегій тестування, що дозволяє автоматично адаптуватися до різних систем і середовищ. Авторами відзначено значний потенціал запропонованого підходу в підвищенні ефективності тестування на проникнення, зокрема у складних і динамічних середовищах. Однак, у статті відсутній глибокий аналіз обмежень, пов’язаних із продуктивністю моделі, коли вона застосовується до великомасштабних корпоративних систем або середовищ із високим рівнем варіативності вразливостей.

У роботі [4] M. Pawlicki та ін. представлено детальний огляд викликів та можливостей використання штучних нейронних мереж у сфері кібербезпеки, зокрема для систем виявлення вторгнень (IDS). У статті розглянуто проблеми налаштування гіперпараметрів, забезпечення збалансованості датасетів і захисту моделей від атак. Додатково автори акцентують увагу на нетехнічних аспектах, таких як етичні, правові та суспільні виклики, що виникають при впровадженні цих технологій. У роботі також наведено приклади реальних впроваджень, які підтверджують ефективність нейронних мереж у виявленні аномалій і загроз. Однак, ключовою проблемою, відзначеною авторами, є обмежена інтерпретованість моделей, що може ускладнювати їх прийняття у практичних сценаріях.

Підсумовуючи ці дослідження, можна зазначити, що сучасні підходи до застосування нейронних мереж у тестуванні на проникнення демонструють значний потенціал у підвищенні ефективності та автоматизації процесів кібербезпеки. Генеративні моделі, такі як ChatGPT [5], дозволяють створювати сценарії атак, а методи підкріплюючого навчання сприяють адаптації систем до динамічних середовищ і складних системних архітектур. Водночас систематичні огляди підкреслюють необхідність вирішення таких викликів, як налаштування гіперпараметрів, збалансування даних та забезпечення захисту самих моделей від атак.

Таким чином, можна помітити існуючу необхідність у подальших дослідженнях для забезпечення інтеграції цих методів у реальних корпоративних середовищах. Зокрема, потребує розвитку питання пояснюваності моделей, розширення наборів даних для різноманітних сценаріїв атак, а також створення комплексних рішень, що об’єднують різні підходи, такі як генеративні нейромережі, підкріплююче навчання та традиційні методи кібербезпеки.

Методологія

Навчання нейромережі відбувається на основі даних, отриманих від експертів у сфері тестування на проникнення. У загальному вигляді дані можуть бути описані таким чином.

$$M = \begin{cases} m_{11,} m_{12,} \dots, m_{1n,} \\ m_{21,} m_{22,} \dots, m_{2n,} \\ m_{k1,} m_{k2,} \dots, m_{kn,} \end{cases} \quad (1)$$

де M – матриця значень критеріїв інструментальних засобів;
m – значення критерію засобу;
k – розмір множини критеріїв;
n – розмір множини утиліт.

Фрагмент даних, які можуть бути використані для навчання наведено в таблиці 1. Ці дані було отримано в результаті перевірки кількох сканерів на тестовому майданчику. За допомогою сканерів перевіряли характерні для Web-додатків вектори атак.

Таблиця 1 – Дані для перевірки сканерів у тестовому середовищі

Критерії	Acunetix	Nexpose	Nessus
Server-Side Template Injection	1	1	0
Stored Cross Site Scripting	1	1	1
Reflect Cross Site Scripting	1	0	0
Cross Site Request Forgery	1	0	0
Weak CORS	1	1	1
HTML injection	0	1	0
CSS Injection	0	1	0
Host Header Injection	1	1	1
Server-Side Request Forgery	1	0	0
XML Injection	1	1	1
HTTP Splitting Smuggling	1	0	0
HTTP Incoming Requests	1	0	1
XPath Injection	0	0	1
SQL Injection	1	1	0
SSI Injection	1	0	0

Як зазначалося раніше, множину навчальних прикладів можна розширити, враховуючи думки користувачів щодо відповідності вектора вимог та утиліт, які були запропоновані як найбільш відповідні за заданим вектором. Такі дані не можна використовувати без попередньої перевірки на наявність суперечностей. Під суперечностями розуміється відповідність однакових наборів вхідних даних різним наборам вихідних даних. Під час навчання на суперечливих прикладах результат роботи нейромережі буде непередбачуваним. Для вирішення поставленого завдання використовується нейромережа прямого поширення, а для навчання з учителем застосовується метод зворотного поширення помилки [6].

Архітектура веб-сервісу. Для реалізації завдання вибору інструментів для тестування на проникнення пропонується створити веб-сервіс, основною частиною логіки якого на серверній стороні буде нейронна мережа. Розроблений сайт допомагатиме початківцям у тестуванні на проникнення обирати інструменти. Для реалізації нейронної мережі пропонується використовувати бібліотеку FANN (Fast Artificial Neural Network) [7] у поєднанні з традиційним програмним стеком LAMP для веб-застосунків (Linux + Apache + MySQL + PHP). Усі використані технології є безкоштовними. Загальний вигляд архітектури веб-застосунку наведено на рисунку 1.

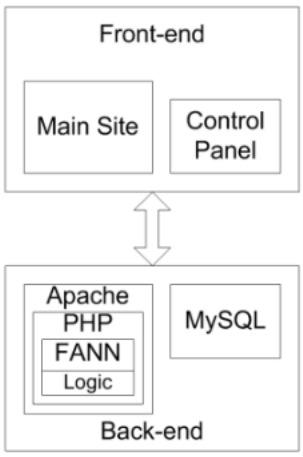

Рисунок 1 – Короткий опис архітектури

Основна частина сайту доступна всім користувачам, і за її допомогою користувач відповідатиме на запитання, формуючи таким чином вектор вимог для пошуку інструментів. Панель управління є закритою та доступна лише адміністратору сайту. За допомогою панелі управління адміністратор виконує такі дії:

1. Додавання нових інструментів у систему.

Технології розвиваються, з'являються нові уразливості та нові напрями атак. Інструменти для виявлення цих уразливостей також повинні оновлюватися, щоб ідентифікувати актуальні проблеми безпеки веб-застосунків. Якщо випускається новий інструмент, якого ще немає в системі, адміністратор може додати його до списку інструментів.

2. Додавання нових характеристик інструментів.

Кожен інструмент визначається певним набором характеристик. Якщо новий інструмент має унікальні характеристики, адміністратор може додати їх до загального списку характеристик.

3. Визначення значень характеристик інструментів.

На основі таблиці вхідних даних адміністратор вводить відповідні значення для кожної характеристики кожного інструмента.

Інформація про інструменти та їх характеристики зберігається у базі даних. Ці дані використовуються для навчання нейронної мережі. Крім того, результати вибору інструментів користувачами також зберігаються у базі даних. Це дозволяє користувачам зберігати результати свого вибору, якщо їхній вибір відрізняється від запропонованого.

Приклад реалізації нейронної мережі

Розглянемо приклад створення нейронної мережі та її навчання на даних, наведених у таблиці 1. Параметри нейронної мережі:

кількість вхідних нейронів – 15 (відповідає кількості критеріїв утиліт);

кількість вихідних нейронів – 3 (відповідає кількості утиліт);

кількість прихованих шарів – 2;

кількість нейронів у прихованих шарах – 9 і 6 відповідно.

Немає суворих правил для вибору кількості нейронів у прихованих шарах, зазначені вище значення отримані за допомогою правила геометричної піраміди.

Результати

У таблиці 2 наведено декілька прикладів вхідних даних та результат роботи нейронної мережі. Результат складається з трьох чисел, що відповідають трьом утилітам. Чим ближче значення

до одиниці, тим більша впевненість нейронної мережі в тому, що утиліта відповідає заданому вектору вимог.

Таблиця 2 – Результати роботи нейромережі

#	Вхідні дані	Результат
1	1; 1; 1; 1; 1; 0; 0; 1; 1; 1; 1; 1; 0; 1; 1	0.978; 0.234; 0.002
2	1; 1; 0; 0; 1; 1; 1; 1; 0; 1; 0; 0; 0; 1; 0	0.923; 0.124; -0.078
3	0; 1; 0; 0; 1; 0; 0; 1; 0; 1; 0; 1; 1; 0; 0	0.854; 0.000; 0.612

У першому прикладі нейронна мережа обрала лише першу (0.978) і другу (0.234) утиліти, оскільки їхні результати є додатними. Третя утиліта (0.002) не була обрана, оскільки її значення є нижчим за поріг прийняття рішення. Вхідний вектор: (1; 1; 1; 1; 0; 0; 1; 1; 1; 1; 1; 0; 1; 1).

У другому прикладі нейронна мережа обрала лише першу утиліту (0.923), оскільки результати для другої (-0.124) і третьої (-0.078) утиліт були негативними. Це вказує на те, що заданий вектор (1; 1; 0; 0; 1; 1; 1; 1; 0; 1; 0; 0; 1; 0) відповідав лише першій утиліті.

У третьому прикладі нейронна мережа обрала першу (0.854) і третю (0.612) утиліти. Друга утиліта (0.000) не була обрана через відсутність позитивного значення. Вхідний вектор: (0; 1; 0; 0; 1; 0; 1; 0; 1; 1; 0; 1; 0; 0). Для практичного використання нейронна мережа повинна бути навчена на даних про 50-100 утиліт. Наведений вище приклад є лише демонстрацією її можливостей.

Обговорення

У ході дослідження авторами запропоновано метод вирішення завдання вибору інструментальних засобів для тестування веб-застосунків на проникнення. Описано вхідні дані для створення веб-сервісу, що реалізує запропонований метод.

Перевага використання нейронних мереж полягає в простоті реалізації порівняно з детермінованими алгоритмами; як метрика використовується кількість рядків коду. До недоліків можна віднести необхідність експериментального підбору параметрів нейронної мережі. Додатковою складністю є пошук даних для навчання через високі вимоги до експертів, які надають навчальні дані для нейронної мережі.

Напрямок подальших досліджень полягає у вивченні впливу параметрів нейронної мережі на результати її роботи.

Висновки

У результаті дослідження запропоновано метод автоматизованого вибору інструментів для тестування на проникнення веб-додатків, що базується на використанні нейронних мереж. Метод дозволяє ефективно аналізувати критерії вибору інструментів та надавати рекомендації, зважаючи на дані, отримані від експертів та користувачів. Основними перевагами підходу є простота реалізації та можливість динамічної адаптації до змін у вимогах тестування безпеки. Запропонований метод забезпечує підвищення ефективності процесу вибору інструментів, зменшуючи час і складність завдання для користувачів. Проте дослідження виявило ряд викликів, зокрема:

необхідність експериментального налаштування параметрів нейронної мережі;

складність у зборі якісних навчальних даних через високі вимоги до експертних знань.

Фінансування

Це дослідження не отримало конкретної фінансової підтримки.

Конкуруючі інтереси

Автори заявляють, що у них немає конкуруючих інтересів.

Список використаних джерел

1. Tolkachova, A., & Piskozub, A. (2024). Methods for testing the security of web applications. *Cybersecurity: Education, Science, Technique, 2*(26), 115–122. https://doi.org/10.28925/2663-4023.2024.26.668
2. Chowdhary, A., Jha, K., & Zhao, M. (2023). Generative adversarial network (GAN)-based autonomous penetration testing for web applications. *Sensors, 23*(18), 8014. https://doi.org/10.3390/s23188014
3. Pozdniakov, K., Alonso, E., Stankovic, V., Tam, K., & Jones, K. (2020, June 15–19). Smart security audit: Reinforcement learning with a deep neural network approximator. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)* (pp. 1–8). IEEE. https://doi.org/10.1109/CyberSA49311.2020.9139683
4. Pawlicki, M., Kozik, R., & Choraś, M. (2022, June). A survey on neural networks for (cyber-)security and (cyber-) security of neural networks. *Neurocomputing.* https://doi.org/10.1016/j.neucom.2022.06.002
5. Aljanabi, M., & ChatGPT. (2023, January). ChatGPT: Future directions and open possibilities. *Mesopotamian Journal of Cyber Security*, 16–17. https://doi.org/10.58496/mjcs/2023/003
6. Yam, Y. F., & Chow, T. W. S. (1993). Extended backpropagation algorithm. *Electronics Letters, 29*(19), 1701–1702. https://doi.org/10.1049/el:19931131
7. Wicht, B., Fischer, A., & Hennebert, J. (2018). DLL: A fast deep neural network library. In *Artificial Neural Networks in Pattern Recognition* (pp. 54–65). Springer International Publishing. https://doi.org/10.1007/978-3-319-99978-4_4

References

1. Tolkachova, A., & Piskozub, A. (2024). Methods for testing the security of web applications. *Cybersecurity: Education, Science, Technique, 2*(26), 115–122. https://doi.org/10.28925/2663-4023.2024.26.668
2. Chowdhary, A., Jha, K., & Zhao, M. (2023). Generative adversarial network (GAN)-based autonomous penetration testing for web applications. *Sensors, 23*(18), 8014. https://doi.org/10.3390/s23188014
3. Pozdniakov, K., Alonso, E., Stankovic, V., Tam, K., & Jones, K. (2020, June 15–19). Smart security audit: Reinforcement learning with a deep neural network approximator. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)* (pp. 1–8). IEEE. https://doi.org/10.1109/CyberSA49311.2020.9139683
4. Pawlicki, M., Kozik, R., & Choraś, M. (2022, June). A survey on neural networks for (cyber-)security and (cyber-) security of neural networks. *Neurocomputing.* https://doi.org/10.1016/j.neucom.2022.06.002
5. Aljanabi, M., & ChatGPT. (2023, January). ChatGPT: Future directions and open possibilities. *Mesopotamian Journal of Cyber Security*, 16–17. https://doi.org/10.58496/mjcs/2023/003
6. Yam, Y. F., & Chow, T. W. S. (1993). Extended backpropagation algorithm. *Electronics Letters, 29*(19), 1701–1702. https://doi.org/10.1049/el:19931131
7. Wicht, B., Fischer, A., & Hennebert, J. (2018). DLL: A fast deep neural network library. In *Artificial Neural Networks in Pattern Recognition* (pp. 54–65). Springer International Publishing. https://doi.org/10.1007/978-3-319-99978-4_4