УДК 004 DOI https://doi.org/10.35546/kntu2078-4481.2025.2.2.30

А. О. НІКІТЕНКО
аспірант кафедри прикладної математики та інформатики
Донецький національний технічний університет
ORCID: 0009-0006-1363-2324

ПРОЦЕС СТВОРЕННЯ АВТОРСЬКОГО ДАТАСЕТУ ДЛЯ ДОНАВЧАННЯ МОДЕЛЕЙ СИСТЕМ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ

У статті висвітлено процес створення авторського набору реальних мережевих потокових даних, призначеного для донавчання моделей систем виявлення мережевих вторгнень (NIDS), побудованих на методах глибокого навчання. Зважаючи на обмежену доступність сучасних, повноцінних і різноманітних датасетів, автори розробили комплексну методологію формування такого набору в контрольованому середовищі, яке імітує реалістичні умови функціонування корпоративної мережі. Для генерації трафіку використовувались як легітимні дії звичайних користувачів (перегляд вебсторінок, обмін файлами, електронна пошта тощо), так і різноманітні кіберзагрози, зокрема DDoS, DoS, розвідка (Reconnaissance), атаки на вебзастосунки, Brute-force та інші. Інструменти типу metasploit, hping3, slowloris, Hydra та інші були задіяні для симуляції атак, а легітимний трафік створювався шляхом автоматизованої активності з кількох вузлів, що імітували поведінку звичайних користувачів. Особливу увагу приділено адаптації процесу збору даних до обмежених ресурсів, без втрати якості чи репрезентативності зібраного трафіку. Запропоновано підхід до балансування обсягів легітимного та аномального трафіку, що дозволяє ефективно використовувати датасет для задач навчання, донавчання, валідації й тестування моделей NIDS. Крім того, акцент зроблено на структурованості, мітках і метаінформації в зібраних даних, що спрощує подальшу обробку та аналіз. Розроблений датасет може стати цінним ресурсом для дослідницької спільноти в галузі кібербезпеки, особливо в контексті розробки адаптивних та інтелектуальних систем виявлення вторгнень, здатних ефективно розпізнавати як відомі, так і нові типи атак. Отримані результати підтверджують надійність запропонованої методології, високу якість зібраного трафіку та його відповідність сучасним викликам у сфері мережевої безпеки.

Ключові слова: *мережева безпека, NIDS, глибоке навчання, набір даних, мережевий трафік, кіберзагрози, симуляція атак.*

A. O. NIKITENKO
Postgraduate Student at the Department of Applied Mathematics and Informatics
Donetsk National Technical University
ORCID: 0009-0006-1363-2324

THE PROCESS OF CREATING AN AUTHOR'S DATASET FOR RETRAINING MODELS OF NETWORK INTRUSION DETECTION SYSTEMS BASED ON DEEP LEARNING

The article describes the process of creating an author's set of real network streaming data intended for training models of network intrusion detection systems (NIDS) based on deep learning methods. Given the limited availability of modern, complete, and diverse datasets, the authors have developed a comprehensive methodology for generating such a set in a controlled environment that simulates realistic corporate network conditions. To generate traffic, they used both legitimate actions of ordinary users (web browsing, file sharing, e-mail, etc.) and various cyber threats, including DDoS, DoS, reconnaissance, attacks on web applications, Brute-force, and others. Tools such as Metasploit, Hping3, Slowloris, Hydra, and others were used to simulate attacks, and legitimate traffic was created by automated activity from multiple nodes that mimicked the behavior of ordinary users. Particular attention is paid to adapting the data collection process to limited resources, without losing the quality or representativeness of the collected traffic. An approach to balancing the amount of legitimate and abnormal traffic is proposed, which allows the dataset to be effectively used for training, retraining, validation, and testing of NIDS models. In addition, emphasis is placed on the structuredness, labels, and meta-information in the collected data, which simplifies further processing and analysis. The developed dataset can become a valuable resource for the cybersecurity research community, especially in the context of developing adaptive and intelligent intrusion detection systems that can effectively recognize both known and new types of attacks. The obtained results confirm the reliability of the proposed methodology, the high quality of the collected traffic, and its relevance to modern network security challenges.

Key words: *network security, NIDS, deep learning, dataset, network traffic, cyber threats, attack simulation.*

Постановка проблеми

У сучасних умовах стрімкої цифровізації обсяги мережевого трафіку невпинно зростають, що супроводжується ескалацією кіберзагроз. Системи виявлення мережевих вторгнень (Network Intrusion Detection Systems – NIDS) відіграють ключову роль у захисті інформаційних ресурсів, здійснюючи аналіз мережевого трафіку в режимі реального часу з метою виявлення аномальної або шкідливої активності. Ефективність таких систем значною мірою залежить від якості, різноманітності та репрезентативності наборів даних (датасетів), що використовуються для їх навчання та тестування. Через складність отримання реальних мережевих трас із надійним маркуванням, більшість публічних датасетів для NIDS створюються в контрольованих умовах. Такий підхід дає змогу точно моделювати атаки та керувати параметрами середовища, проте не завжди забезпечує достатню різноманітність і автентичність трафіку, що може обмежувати здатність моделей узагальнюватися на реальні сценарії. Це породжує потребу у вдосконаленні методів генерації навчальних даних для підвищення якості моделей виявлення вторгнень. Попри певні переваги використання публічних датасетів, подібні дані не завжди достовірно відображають реалістичну мережеву поведінку та загрози [1]. З метою подолання цих обмежень у дослідженні представлено новий авторський датасет мережевого трафіку, який поєднує реалістичну активність користувачів із широким спектром сучасних атак. Для його створення розгорнута спеціалізована експериментальна інфраструктура, що включає мережевий пристрій MikroTik, системи атакувальників і жертв на базі операційної системи Ubuntu, а також засоби збору та обробки трафіку – зокрема *tshark* [2] і *CICFlowMeter* [3]. Симуляція атак здійснювалася з використанням провідних інструментів, серед яких *sqlmap* [4], *nmap* [5], *hydra* [6], *hping3* [7], Python-скрипти, *slowhttptest* [8], *fping* [9], *beef-xss* [10] і *msfvenom* [11].

Аналіз останніх досліджень і публікацій

У рамках дослідження оглянуто низку наукових публікацій, що висвітлюють сучасні підходи, проблеми та тенденції у створенні мережевих датасетів для NIDS. Представлений аналіз дозволяє оцінити переваги й недоліки існуючих рішень, а також визначити вимоги до побудови власного набору даних.

У роботі [12] запропоновано фреймворк для генерації мережевих наборів даних з реальних трас трафіку, що можуть бути використані у дослідженнях систем виявлення мережевих вторгнень (NIDS). Запропонований підхід дозволяє формувати реалістичні сценарії, які раніше були недоступні. Основною перевагою є можливість поєднання фонового мережевого трафіку, зібраного в реальному середовищі, з трафіком, що містить ознаки атак. Оскільки ці дані не зазнали анонімізації, вони зберігають повну інформацію, що підвищує їхню достовірність. Додатково, створені інструменти сприяють ефективному витягу характеристик мережевих сеансів. До недоліків віднесено залежність від доступності великого обсягу захоплених мережевих трас.

У дослідженні [13] представлено модель CNN-LSTM на основі SFL, що застосовує методи глибокого навчання до аналізу корисного навантаження HTTP-трафіку в умовах високопродуктивного обчислювального середовища. Розроблена система AI-IDS виявляє аномальний трафік, зокрема нові патерни, які недоступні для традиційних NIDS, що базуються на сигнатурах. Це дозволяє автоматично формалізувати нові атаки і вдосконалювати правила сигнатур. Проте система має недолік у вигляді високої кількості хибнопозитивних спрацьовувань, що потребує повторної верифікації подій, тому наразі рекомендовано використовувати її як допоміжну.

У роботі [14] здійснено поглиблений аналіз семи найбільш цитованих еталонних наборів даних для NIDS, у результаті чого виявлено шість характерних «запахів дизайну даних» – повторюваних патернів, які можуть свідчити про недоліки в конструюванні наборів даних. Автори зазначають, що неочевидні дизайнерські рішення можуть істотно вплинути на експериментальні результати. Запропоновано шість евристик для вимірювання цих недоліків. Встановлено, що наявність «запахів» корелює з недостатньою різноманітністю даних, нечітким маркуванням і слабкими критеріями узагальнення. У роботі надано рекомендації для підвищення якості створення і використання наборів даних, що має на меті стимулювати критичний перегляд наявних ресурсів у спільноті дослідників.

Метою роботи [15] є створення нового великого набору даних з атаками на IoT-пристрої. У межах експериментальної інфраструктури з 105 пристроїв було реалізовано 33 атаки, які класифіковано у сім категорій: DDoS, DoS, розвідка, веб-атаки, атаки грубою силою, спуфінг та Mirai. Набір даних надається у відкритому доступі на платформі CIC Dataset. Його особливістю є фокус на атаки між IoT-пристроями, що підвищує актуальність у контексті сучасних загроз.

У роботі [16] проведено огляд наявних публічних наборів даних та визначено проблематику відсутності єдиного підходу до валідації міток, що ускладнює порівнянність результатів і знижує довіру до моделей. Автори закликають до створення динамічних наборів даних з прозорими процедурами маркування та перевірки, що дозволить об’єктивніше оцінювати ефективність підходів до виявлення атак. Основним недоліком є відсутність реалізації запропонованих підходів у вигляді прототипу чи проєкту.

Дослідження [17] висуває нові вимоги до створення IDS-наборів даних: наявність інформації про корисне навантаження, істинні мітки (ground truth), шифрування, анонімізацію, а також забезпечення практичної реалізації. Автори створили новий датасет HIKARI-2021, який містить зашифрований трафік, базується на ознаках

з CICIDS-2017 і доповнений додатковими характеристиками, такими як IP-адреси та порти. Датасет орієнтований на реалістичні сценарії атаки, що дозволяє розширити межі аналізу.

У роботі [18] представлено датасет LITNET-2020, зібраний у реальній академічній мережі. Він містить приклади нормального та зловмисного трафіку, 85 ознак мережевого потоку та 12 типів атак. Було проведено кластерний і статистичний аналіз для оцінки ефективності ознак. Датасет є відкритим для досліджень, проте його недоліком є обмежене охоплення сценаріїв поза академічним середовищем.

У роботі [19] розглянуто можливість використання повністю синтетичних даних, згенерованих за допомогою генеративних змагальних мереж (GAN), для тренування моделей машинного навчання в NIDS. Було використано дані з трьох наборів: UNSW-NB15, NSL-KDD та BoT-IoT. Експерименти показали, що моделі, натреновані на синтетичних даних, досягають точності, порівнянної або вищої за ті, що тренувались на реальному трафіку (до 100 % на BoT-IoT). Це свідчить про потенціал GAN як засобу подолання проблеми обмеженої доступності якісних реальних даних. Автори планують подальше дослідження адаптивних GAN, здатних еволюціонувати разом із новими загрозами.

Аналіз сучасних підходів до створення датасетів для NIDS засвідчує еволюцію від синтетичних, штучно створених наборів до більш реалістичних, динамічних і технічно повних рішень. Класичні датасети (наприклад, KDD’99) часто зазнають критики через застарілість, низьку репрезентативність і наявність артефактів, що призводить до перенавчання моделей і їх неспроможності працювати у реальному середовищі [16]. У відповідь на ці виклики в новітніх дослідженнях акцент зміщується в бік використання або симуляції реального трафіку. Зокрема, у CICIoT2023 [15] та LITNET-2020 [18] трафік записується в умовах, наближених до продуктивних мереж, із широким охопленням атак і збереженням реальних мережевих характеристик.

Типовий процес побудови NIDS-датасету включає кілька ключових етапів: проєктування сценаріїв легітимної активності та атак, налаштування контрольованого середовища (емулятора або фізичної лабораторії), збір трафіку (у форматі PCAP або NetFlow), екстракція ознак (features), маркування з’єднань (наприклад, як «атака» чи «нормальна активність»), а також валідація та попередній аналіз.

Щодо джерел трафіку, умовно їх поділяють на три категорії: реальні, синтетичні та гібридні. Перші базуються на даних із продуктивних або напівреальних середовищ (як-от у LITNET-2020), другі – моделюються у віртуалізованих лабораторіях або генеруються за допомогою моделей штучного інтелекту, зокрема GAN (наприклад, SYN-GAN [19]), тоді як гібридні поєднують справжній фоновий трафік із навмисно створеними атаками [17]. Ефективність моделей глибокого навчання значною мірою залежить від якості та повноти ознак: сучасні датасети містять не лише статистичні характеристики з’єднань (IP, порти, тривалість, кількість пакетів), але й контекстні ознаки, флаги протоколів, інтервали між пакетами тощо.

Попри прогрес, проблемними залишаються питання узгодженості форматів, етичності збору, репрезентативності трафіку та актуальності типів атак. Як показано в роботах [13] та [16], навіть сучасні набори часто не охоплюють Zero-Day-атаки або нові вектори загроз. Це створює потребу у відкритих, добре документованих, оновлюваних та достовірних датасетах, які дозволять формувати стійкі до змін моделі нового покоління NIDS. Таким чином, створення авторських датасетів, що враховують недоліки існуючих рішень, є актуальним завданням дослідників у сфері мережевої безпеки.

Формулювання мети дослідження

Метою дослідження є пошук можливості створення якісного, достовірного та репрезентативного набору мережевих даних, а також визначення універсальної методології, яка дозволяє генерувати сучасні датасети за умов обмежених обчислювальних ресурсів. Отриманий в результаті проведених експериментів датасет покликан усунути недоліки наявних рішень та сприяти розвитку інноваційних підходів до побудови NIDS, заснованих на методах глибокого навчання.

Викладення основного матеріалу дослідження

Розробка якісного та репрезентативного набору даних для навчання NIDS вимагає ретельного планування як інфраструктури, так і процесів збору та обробки інформації. Далі описано запропоновану методологію побудови датасету, яка базується на моделюванні реалістичного мережевого середовища, симуляції різних типів атак і генерації фонової активності звичайних користувачів.

Процес формування набору охоплює кілька ключових етапів: проєктування та налаштування мережевої топології, вибір інструментів для імітації атак і нормального трафіку, організацію збору мережевих потоків за допомогою *tshark*, а також обробку зібраних даних для подальшого виділення ознак потоків. Особливу увагу приділено оптимізації використання обчислювальних ресурсів без шкоди для якості даних, що дає змогу відтворювати сучасні загрози в контрольованому середовищі навіть за обмежених інфраструктурних можливостей.Запропонована методологія спрямована на забезпечення високої точності, актуальності та універсальності даних, необхідних для досліджень у сфері кібербезпеки. Загальна структура експериментального середовища схематично подана на рис. 1.

Короткий огляд кібератак

У межах експериментального дослідження було згенеровано як нормальний (benign), так і шкідливий мережевий трафік. Нормальний трафік не містить ознак атак чи аномалій і використовується як еталон для навчання

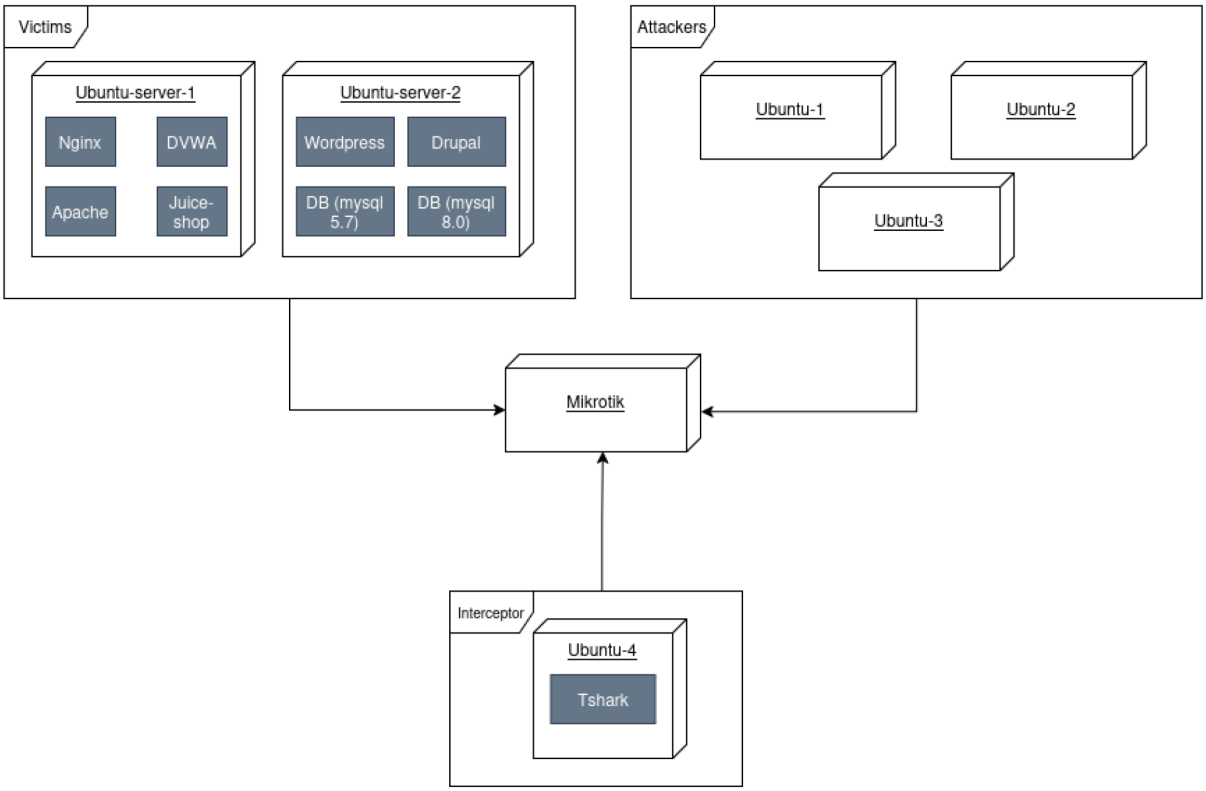

Рис. 1. Загальна інфраструктура експериментального середовища

та тестування NIDS. Шкідливий трафік охоплює кілька категорій атак, зокрема DoS/DDoS, розвідувальні (reconnaissance) та web-атаки.

До DoS-атак належать такі типи перевантаження систем: HTTP Flood, UDP Flood і UDP Fragmentation – атаки, спрямовані на виснаження ресурсів сервера шляхом надсилання великої кількості, іноді фрагментованих, пакетів. Атаки типу SYN Flood, TCP Flood, ACK Fragmentation і ICMP Flood порушують функціонування TCP/IP-стека, ускладнюючи встановлення з'єднань і обробку трафіку. Атаки Slowloris та Slow POST реалізують повільну передачу HTTP-запитів, блокуючи можливість обробки нових з'єднань.

Розвідувальні атаки, такі як: Ping Sweep, OS Scan, Port Scan, Vulnerability Scan, Host Discovery спрямовані на збір інформації про мережеву інфраструктуру шляхом активного сканування IP-адрес, портів і операційних систем. Такі дії зазвичай передують складнішим етапам вторгнення.

Web-атаки включають SQL Injection і Command Injection – методи ін'єкції шкідливих команд у запити до веб-додатків з метою компрометації баз даних або серверів. Також реалізовано атаки через File Upload, Reverse Shell (backdoor malware) та Dictionary Brute Force – зокрема, спроби підбору облікових даних або встановлення прихованих каналів керування сервером. Наведені сценарії дають змогу всебічно оцінити здатність NIDS виявляти широкий спектр сучасних загроз.

Для подальшого дослідження було відібрано атаки на основі чотирьох ключових критеріїв: по-перше, забезпечено повне охоплення основних тактик із матриці MITRE ATT&CK, зокрема Denial of Service (T1498), Reconnaissance (TA0043), Execution через Command and Scripting Interpreter (T1059), а також Credential Access (Brute Force); по-друге, враховано наявність відповідних типів атак у популярних загальнодоступних NIDS-датасетах, таких як CIC-IDS2017 та UNSW-NB15; по-третє, обрані атаки характеризуються високою відтворюваністю в локальному середовищі, що підтверджено попередніми дослідженнями з відтворюваності кіберекспериментів; по-четверте, охоплено ключові класи загроз – Reconnaissance (Ping Sweep, Port Scan, OS Scan), Web-атаки (SQL Injection, XSS, CSRF, Command Injection) згідно з класифікацією OWASP Top 10, а також Dictionary Brute Force. Відбір атак базувався на їхній наявності у щорічних звітах CERT-IST, CERT-EU Threat Landscape Report та ANSSI Cyber Threat Overview, що підтверджує їхню практичну значущість і актуальність для оцінювання ефективності NIDS.

Опис розробленої інфраструктури

Розгорнута інфраструктура (рис. 1) складається з трьох основних компонентів: машини-атакуючі (**Attackers**), машин-цілей (**Victims**) та машини-перехоплювача (**Interceptor**). Кожен з елементів відіграє специфічну роль у генерації, прийомі та зборі мережевого трафіку, необхідного для формування якісного набору даних.

Машини-атакуючі (Attackers)

Основне призначення цих машин – генерація синтетичного атакувального трафіку. Було розгорнуто три машини з іменами ubuntu-1, ubuntu-2 та ubuntu-3. На кожній із них встановлено типовий набір інструментів для реалізації атак, зокрема: *iperf3*, *sqlmap*, *nmap*, *hydra*, *hping3*, *slowhttptest*, *fping*, *beef-xss*, *msfvenom*, а також Python-інтерпретатор із підтримкою виконання авторських скриптів.

Крім того, на кожній машині розміщено спеціалізовані Bash та Python скрипти, створені для автоматизованого проведення атак та генерації трафіку. Зокрема, використані такі скрипти:

- Benign.sh – генерація нормального фоновго трафіку,
- Bruteforce.sh – атаки типу Brute Force,
- DDoS.sh / DoS.sh – атаки на відмову в обслуговуванні,
- Slowloris.py – повільні HTTP-атаки,
- Spoofing.sh – атаки з підробкою IP-адрес,
- Web.sh – атаки на веб-додатки.

Кожен скрипт містить попереднє налаштування середовища, перевірку наявності залежностей та запуск відповідної атаки.

Машини-цілі (Victims)

Машини цього класу призначені для розгортання вразливих веб-додатків та прийому трафіку від машин-жертв. Було використано дві машини: ubuntu-server-1 та ubuntu-server-2. На обох серверах додатки розгорнуто із використанням контейнеризації через технологію Docker.

На ubuntu-server-1 запущено:

- веб-сервери nginx та apache,
- навчальні уразливі веб-додатки DVWA (Damn Vulnerable Web Application) та OWASP Juice Shop.

На ubuntu-server-2 розгорнуто:

- CMS-додатки WordPress та Drupal,
- дві версії СУБД MySQL: 5.7 та 8.0.

Процес розгортання автоматизовано за допомогою Bash-скриптів setup_server_1.sh та setup_server_2.sh, які виконують:

1. Встановлення Docker (якщо він відсутній),
2. Створення конфігураційного файлу docker-compose.yml,
3. Ініціалізацію контейнерів за допомогою docker-compose up.

Машина-перехоплювач (Interceptor)

Ця машина виконує функцію моніторингу та перехоплення мережевого трафіку, що передається між машинами-жертвами та цілями. На машині встановлено лише інструмент **tshark**, за допомогою якого весь трафік записується у форматі *.pcap*. Для кожного експериментального запуску атаки генерується окремий файл, що забезпечує точну відповідність між типом трафіку та збереженим мережевим знімком.

Таблиця 1

Конфігурації обчислювальних машин, задіяних в експериментальній інфраструктурі

Назва машини	Характеристики
Attackers	
ubuntu-1	Операційна система: Ubuntu 24.04, intel core i3-9100F 3.60GHz, ОЗУ: 16 ГБ
ubuntu-2	Операційна система: Ubuntu 24.04, intel core i3-6100, 3.70GHz, ОЗУ: 8 ГБ
ubuntu-3	Операційна система: Ubuntu 24.04, intel core i3-7100, 3.90GHz, ОЗУ: 8 ГБ
Victims	
ubuntu-server-1	Операційна система: Ubuntu-server 24.04, intel core i3-9100F 3.60GHz, ОЗУ: 16 ГБ
ubuntu-server-2	Операційна система: Ubuntu-server 24.04, intel core i3-9100F 3.60GHz, ОЗУ: 16 ГБ
Interceptor	
ubuntu-4	Операційна система: Ubuntu 24.04, intel core i3-4150, 3.50GHz, ОЗУ: 12 ГБ

Також не менш важливою частиною експериментального середовища є використання мережевого обладнання, що забезпечує фізичне та логічне з'єднання між усіма компонентами інфраструктури. Експеримент проводився у локальній мережі організації, тому особливу увагу приділено питанню ізоляції середовища для запобігання поширенню атак за межі тестового стенду.

Для побудови мережі використано маршрутизатор MikroTik Cloud Router Switch CRS326-24G-2S+RM, налаштований у режимі маршрутизатора з підтримкою VLAN. Створено три логічно ізольовані підмережі:

- **VLAN10** – для машин-атакуючих (192.168.10.0/24)
- **VLAN20** – для машин-жертв (192.168.20.0/24)

- **VLAN30** – для сервера перехоплення трафіку (192.168.30.0/24)

Порти маршрутизатора налаштовано з відповідними PVID для автоматичного призначення VLAN-тегів вхідному трафіку. Для кожної VLAN було піднято DHCP-сервер, який автоматично призначає IP-адреси відповідним пристроям. Додатково на маршрутизаторі активовано packet sniffing – передачу копій мережевого трафіку на віддалений сервер через протокол стрімінгу для його подальшого аналізу.

Підключення до віртуального стенду здійснювалося окремим ноутбуком через SSH, що дозволяло керувати експериментом без загрози проникнення атак до основної мережі університету. Ізоляція досягалася як фізичним від'єднанням, так і логічною сегментацією через VLAN. Побудовану інфраструктуру наведено на рис. 2.

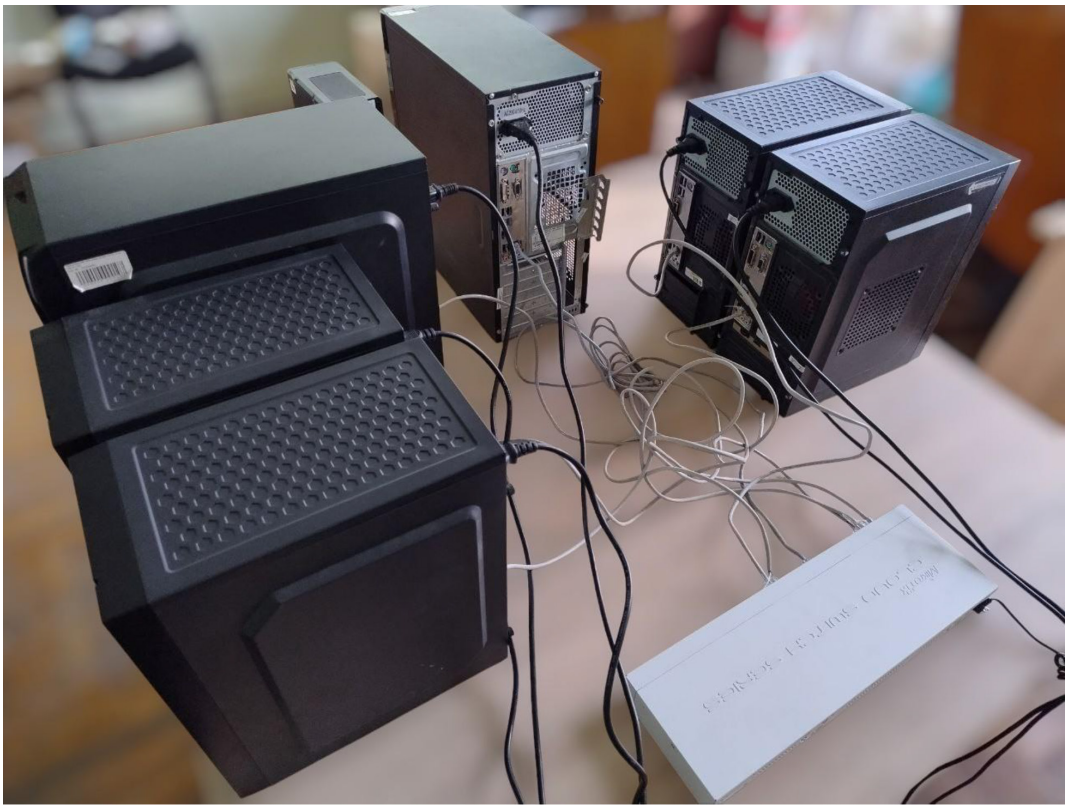

Рис. 2. Фото інфраструктури експериментального середовища

Проведення експериментів

Після налаштування інфраструктури та встановлення всіх необхідних залежностей здійснено серію експериментів з генерації атакуючого та легітимного мережевого трафіку. На кожній з атакуючих машин (ubuntu-1, ubuntu-2, ubuntu-3) були заздалегідь підготовлені bash-скрипти, що автоматизували запуск конкретних атак, включно з попереднім встановленням необхідних бібліотек та запуском відповідних інструментів.

Керування запуском сценаріїв здійснювалося централізовано – з окремої машини-оператора через SSH-з'єднання. Це дозволяло віддалено ініціювати атаки в контрольованій послідовності та синхронізовано керувати експериментальним процесом.

Тривалість кожного експерименту становила орієнтовно 30–60 хвилин, що забезпечувало достатній обсяг даних для статистичного аналізу та подальшої генерації ознак (features) на основі мережевих потоків. Перелік реалізованих атак разом із відповідними обсягами згенерованого мережевого трафіку, що були використані для формування дослідного набору даних, представлено в таблиці 2.

Порядок проведення експериментальних досліджень структуровано відповідно до категорій атак і наведено в таблиці 3.

Під час проведення кожного експерименту мережевий трафік записувався у форматі.pcap за допомогою інструмента **tshark**, що дозволило надалі здійснити глибокий аналіз потоків і виконати екстракцію ключових ознак для побудови моделей NIDS. Таким чином сформовано.pcap-файли, які чітко відповідали кожному типу атаки.

Для забезпечення відповідності структури трафіку характеристикам набору **CIC-IDS-2018** було використано інструмент **CICFlowMeter**, який дозволяє формувати *traffic flow* кожної сесії. Далі, за допомогою мови програмування **Python**, було здійснено попередню обробку, маркування даних відповідно до типів атак, а також об'єднання

Таблиця 2

Перелік атак та обсяг згенерованого мережевого трафіку

Type	Attack	Rows
Benign	Normal traffic	1008224
DDoS	HTTP Flood	90583
	UDP Flood	37632
	UDP Fragmentation	40142
	SYN Flood	38487
	ICMP Flood	35932
	ACK Fragmentation	39698
	TCP Flood	40165
	Slowloris	8725
DoS	UDP Flood	9878
	HTTP Flood	220233
	SYN Flood	122936
	Slow POST	141910
Reconnaissance	Ping Sweep	6894
	OS Scan	
	Vulnerability Scan	
	Port Scan	
	Host Discovery	
Web	Sql injection	47335
	Command injection	7658
	Backdoor malware (Reverse Shell)	14986
	File upload	14365
	XSS	15237
	Browser hijacking	14834
	CSRF	12637
Brute-force	Dictionary Brute force (Web)	3620

Таблиця 3

Параметри експериментальних атак

№	Тип атаки	Підтип атаки	Машини-атакуючі	Цільові IP та порти	Інструменти
1	Benign	–	ubuntu-1,2,3	192.168.30.251, 192.168.30.253,	Locust, selenium, iperf3
2	DoS	UDP Flood	ubuntu-1,2,3	192.168.30.251:8081, 8082; 192.168.30.253:8081	hping3
3	DoS	SYN Flood	ubuntu-1,2,3	192.168.30.251:8081, 8082; 192.168.30.253:8085	hping3
4	DoS	ICMP Flood	ubuntu-1,2,3	192.168.30.251:8081, 8082; 192.168.30.253:8082	hping3
5	DoS	HTTP DoS	ubuntu-1,2,3	192.168.30.251:8081, 8082; 192.168.30.253:8083	slowhttptest
5	DoS	SlowHTTPTest	ubuntu-1,2,3	192.168.30.251:8081, 8082; 192.168.30.253:8085	slowhttptest
6	DDoS	UDP Flood	ubuntu-1,2,3	192.168.30.251:8081	hping3
7	DDoS	SYN Flood	ubuntu-1,2,3	192.168.30.251:8082	hping3
8	DDoS	ICMP Flood	ubuntu-1,2,3	192.168.30.253:8085	hping3
9	DDoS	Slowloris	ubuntu-1,2,3	192.168.30.253:8081	python
10	DDoS	HTTP Flood	ubuntu-1,2,3	192.168.30.253:8082	slowhttptest
11	DDoS	ACK Fragmentation	ubuntu-1,2,3	192.168.30.253:8082	hping3
12	DDoS	UDP Fragmentation	ubuntu-1,2,3	192.168.30.253:8083	hping3
13	Reconnaissance	Ping Sweep	ubuntu-1,2,3	192.168.30.251:8081	fping
14	Reconnaissance	OS Scan	ubuntu-1,2,3	192.168.30.251:8081	nmap
15	Reconnaissance	Vulnerability Scan	ubuntu-3	192.168.30.253:8085	nmap
16	Reconnaissance	Port Scan	ubuntu-3	192.168.30.251:8082	nmap
17	Reconnaissance	Host Discovery	ubuntu-3	192.168.30.253:8082	nmap
18	Web Attack	SQL Injection	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	sqlmap
19	Web Attack	Command Injection	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	bash
20	Web Attack	Backdoor Malware	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	msfvenom
21	Web Attack	File Upload Attack	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	bash
22	Web Attack	XSS	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	bash
23	Web Attack	Browser Hijacking	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	beef-xss
24	Web Attack	CSRF	ubuntu-1,2,3	192.168.30.251:8081 192.168.30.253:8082	python
25	Brute Force	Bruteforce (Web)	ubuntu-1	192.168.10.254	hydra

результатів у два окремі файли: attack.csv та benign.csv. Файл attack.csv містить 21 клас, оскільки види атак типу «Reconnaissance» об'єднані в один клас через обмежену кількість згенерованого трафіку.

Ці набори даних використовувались для донавчання авторської моделі [10]. Для оцінювання ефективності моделі застосовано класичні метрики якості класифікації: **accuracy**, **precision**, **recall**, **F1-score** та **FPR** (*False Positive Rate*).

Точне налаштування (fine tuning) моделі

Авторську модель [20] попередньо навчено на відібраних ознаках з набору даних CIC-IDS-2018, після чого збережено у форматі.keras. Наступним етапом стала підготовка власного набору даних для донавчання цієї моделі з метою адаптації до нових умов та розширення простору ознак.

На етапі попередньої обробки даних проводиться очищення пропущених та некоректних значень. Нескінченні значення (inf, -inf) замінюються на NaN, після чого відсутні дані або заповнюються середніми значеннями по відповідних стовпцях, або видаляються – залежно від контексту. Строкові стовпці (за винятком цільової змінної Label) перетворюються на числові представлення, при цьому помилкові рядки також трансформуються у NaN з подальшим очищенням.

Далі застосовується нормалізація числових ознак методом Min-Max масштабування, що дозволяє привести значення до єдиного діапазону. Класові мітки у стовпці Label кодуються у числову форму за допомогою LabelEncoder. Після завершення етапу попередньої обробки дані поділяються на вхідні ознаки (features) та цільову змінну (target).

Подальше точне налаштування (fine-tuning), або ж трансферне навчання (transfer learning), базується на використанні попередньо натренованої моделі як основи для розв'язання нової, близької задачі [21]. Це дозволяє скористатися вже вивченими абстракціями, мінімізуючи обчислювальні витрати на повторне навчання з нуля.

Для реалізації цього підходу завантажено збережену модель, у якій всі шари, за винятком двох останніх (Dropout та Dense), заморожено тобто їх ваги не змінюються в процесі донавчання. На основі виходу передостаннього шару сформованоновий вихідний шар типу Dense, кількість нейронів у якому відповідає кількості класів нового набору (21 клас).

Нову модель побудовано з тими самими вхідними параметрами, але з оновленою структурою виходу. Для навчання модель компілюється з оптимізатором Adam, функцією втрат categorical_crossentropy та метриками оцінювання accuracy і AUC. Це дозволило ефективно провести донавчання моделі на новому датасеті, адаптувавши її до особливостей нових атак і специфіки трафіку, зібраного в межах запропонованої інфраструктури. Отримані показники наведено у таблиці 4.

Таблиця 4

Порівняння показників моделі до та після донавчання

Метрика	Optimized CNN-BiGRU-Attention [20]	Дотренована CNN-BiGRU-Attention
Accuracy (%)	99.41	99.44
Precision (%)	99.45	99.47
Recall (%)	99.41	99.44
F1-score (%)	99.42	99.45
FPR	0,0035	0,0033

Для порівняння використано тренувальні підмножини даних. Як видно з таблиці 4, донавчання моделі на запропонованому датасеті забезпечило певне покращення показників її ефективності. Це свідчить про наявність у сформованому наборі даних, попри його обмежений обсяг, релевантної інформації, здатної адаптувати модель до нових ознак і умов функціонування. Водночас помірний характер покращень вказує на те, що зібраного трафіку недостатньо для суттєвого підвищення якості класифікації. Проте навіть такі результати підтверджують перспективність обраного підходу: за умов обмежених обчислювальних ресурсів можливо сформувати повноцінний датасет, придатний для донавчання моделей NIDS. Основною метою цього дослідження було формування методології, що забезпечує ефективне генерування навчальних даних для систем виявлення мережевих вторгнень у локальному середовищі. Попри те, що створений набір даних наразі поступається за масштабом і різноманіттям широко визнаним публічним датасетам, таким як CIC-IDS2017 та UNSW-NB15, він має потенціал до подальшого розвитку. Зокрема, майбутнє масштабування експериментів із розширенням кількості вузлів, типів атак і сценаріїв функціонування мережі може сприяти створенню повноцінного сучасного датасету, релевантного до умов реального середовища інформаційної безпеки.

Висновки

Досліджено широкий спектр сучасних типів кібератак, розроблено компактну експериментальну інфраструктуру для їх реалізації в ізольованому контрольованому середовищі, а також здійснено збір мережевого трафіку для створення власного набору даних. Основною метою дослідження було формування методології, яка дозволяє з використанням обмежених обчислювальних ресурсів генерувати навчальні дані для тренування моделей глибокого навчання, що застосовуються в NIDS.

Серед ключових досягнень даного дослідження доцільно виокремити розгортання модульної експериментальної інфраструктури, що забезпечує централізоване управління процесами ініціювання атак і збору трафіку; генерацію релевантного мережевого трафіку із застосуванням широкого спектру інструментів, зокрема *hping3*, *nmap*, *sqlmap*, *slowhttptest* та інших; екстракцію ознак трафіку за допомогою *CICFlowMeter* з подальшим формуванням структурованого датасету, придатного для використання у задачах машинного навчання; а також успішне донавчання попередньо натренованої моделі на зібраних даних, що забезпечило позитивну, хоча й помірну динаміку основних метрик якості класифікації.

Попри позитивні результати, отримані в ході експериментів, існують певні обмеження. Зокрема, обмежена кількість нових класів атак, обсяг зібраного трафіку, а також невисока варіативність мережевого середовища можуть стримувати потенціал моделі до генералізації.

У подальших дослідженнях планується масштабувати експериментальне середовище до рівня розподіленої мережевої інфраструктури, зокрема з використанням декількох логічно об'єднаних підмереж (наприклад, різні VLAN-и чи сегменти університетської мережі), збільшити кількість цільових вузлів і атакуючих машин, що дозволить згенерувати більший обсяг трафіку з більшою різноманітністю, інтегрувати нові типи атак, включаючи сучасні приклади з області AI-driven атак, атак на IoT-пристрої, fileless-malware, багатовекторні DDoS-атаки, автоматизувати процес генерації, збереження та розмітки трафіку з метою пришвидшення збору та обробки нових наборів даних.

Розроблена методологія може стати основою для побудови повноцінної платформи тестування та навчання моделей виявлення вторгнень у реалістичних умовах, що відповідають сучасним вітчизняним викликам у сфері кібербезпеки.

Список використаної літератури

1. Molina-Coronado B., Rosero-Montalvo P. D., Calafate C. T. Survey of Network Intrusion Detection Methods From the Perspective of the Knowledge Discovery in Databases Process // *IEEE Transactions on Network and Service Management*. 2020. Vol. 17, no. 4. P. 2451–2479. DOI: https://doi.org/10.1109/tnsm.2020.3016246 (дата звернення: 08.06.2025).

2. Jo M., Jeong H., Song B., Jo H. Encrypted Traffic Decryption Tools: Comparative Performance Analysis and Improvement Guidelines // Electronics. 2024. Vol. 13, no. 14. P. 2876. URL: https://doi.org/10.3390/electronics13142876 (дата звернення: 08.06.2025).

3. CICFlowMeter (formerly ISCXFlowMeter) [Електронний ресурс]. URL: https://www.unb.ca/cic/research/applications.html (дата звернення: 08.06.2025).

4. *SQLMap* [Електронний ресурс]. URL: https://sqlmap.org/ (дата звернення: 08.06.2025).

5. Lyon G. *Nmap Security Scanner* [Електронний ресурс]. URL: http://nmap.org/ (дата звернення: 08.06.2025).

6. Maciejak D. *Hydra* [Електронний ресурс]. URL: https://github.com/vanhauser-thc/thc-hydra (дата звернення: 08.06.2025).

7. Kali Tools. *hping3 Package Description* [Електронний ресурс]. URL: https://www.kali.org/tools/hping3 (дата звернення: 08.06.2025).

8. Kali Tools. *Slowhttptest Usage Example* [Електронний ресурс]. URL: https://www.kali.org/tools/slowhttptest/ (дата звернення: 08.06.2025).

9. *Fping* [Електронний ресурс]. URL: https://fping.org/ (дата звернення: 08.06.2025).

10. *The Browser Exploitation Framework (BeEF)* [Електронний ресурс]. URL: https://beefproject.com (дата звернення: 08.06.2025).

11. Rapid7. *Introducing msfvenom* [Електронний ресурс]. URL: https://www.rapid7.com/blog/post/2011/05/24/introducing-msfvenom/ (дата звернення: 08.06.2025).

12. Velarde-Alvarado P., Coronado E. V., Ponce H. et al. A Novel Framework for Generating Personalized Network Datasets for NIDS Based on Traffic Aggregation // *Sensors*. 2022. Vol. 22, no. 5. P. 1847. DOI: https://doi.org/10.3390/s22051847 (дата звернення: 08.06.2025).

13. Kim A., Park M., Lee D. H. AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection // *IEEE Access*. 2020. Vol. 8. P. 70245–70261. DOI: https://doi.org/10.1109/access.2020.2986882 (дата звернення: 08.06.2025).

14. Flood R., Shiaeles S., Kolokotronis N. et al. Bad Design Smells in Benchmark NIDS Datasets // *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, Vienna, Austria, 8–12 July 2024. P. 658–675. DOI: https://doi.org/10.1109/eurosp60621.2024.00042 (дата звернення: 08.06.2025).

15. Neto E. C. P., Silva L. C., Rocha D. M. et al. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment // *Sensors*. 2023. Vol. 23, no. 13. P. 5941. DOI: https://doi.org/10.3390/s23135941 (дата звернення: 08.06.2025).

16. Guerra J. L., Catania C., Veas E. Datasets are not Enough: Challenges in Labeling Network Traffic // *Computers & Security*. 2022. P. 102810. DOI: https://doi.org/10.1016/j.cose.2022.102810 (дата звернення: 08.06.2025).

17. Ferriyan A., Hermawan D., Suhartono D. Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic // *Applied Sciences*. 2021. Vol. 11, no. 17. P. 7868. DOI: https://doi.org/10.3390/app11177868 (дата звернення: 08.06.2025).

18. Damasevicius R., Krilavicius T., Maskeliunas R. et al. LITNET-2020: An Annotated Real-World Network Flow Dataset for Network Intrusion Detection // *Electronics*. 2020. Vol. 9, no. 5. P. 800. DOI: https://doi.org/10.3390/electronics9050800 (дата звернення: 08.06.2025).

19. Rahman S., Chowdhury M. J. M., Jahan I. et al. SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security // *Internet of Things*. 2024. Vol. 26. P. 101212. DOI: https://doi.org/10.1016/j.iot.2024.101212 (дата звернення: 08.06.2025).

20. А. О. Нікітенко і Є. О. Башков. Оптимізація системи виявлення мережевих вторгнень на основі глибокого навчання із використанням методу зменшення розмірності та метаевристичних алгоритмів // *Наукові праці Вінницького національного технічного університету*. 2025. № 1. DOI: https://doi.org/10.31649/2307-5376-2025-1-86-98 (дата звернення: 08.06.2025).

21. Sanmorino A., Puspasari D. R., Wibowo A. Feature Extraction vs Fine-tuning for Cyber Intrusion Detection Model // *Jurnal INFOTEL*. 2024. Vol. 16, no. 2. P. 302–315. DOI: https://doi.org/10.20895/infotel.v16i2.996 (дата звернення: 08.06.2025).

References

1. Molina-Coronado, B., Mori, U., Mendiburu, A., & Miguel-Alonso, J. (2020). Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Transactions on Network and Service Management, 17*(4), 2451–2479. https://doi.org/10.1109/TNSM.2020.301624

2. Jo, M., Jeong, H., Song, B., & Jo, H. (2024). Encrypted traffic decryption tools: Comparative performance analysis and improvement guidelines. *Electronics, 13*(14), 2876. https://doi.org/10.3390/electronics13142876

3. Canadian Institute for Cybersecurity. (n.d.). *CICFlowMeter (formerly ISCXFlowMeter)*. University of New Brunswick. https://www.unb.ca/cic/research/applications.htm

4. SQLMap. (n.d.). *sqlmap.org*. https://sqlmap.org/

5. Lyon, G. F. (n.d.). *Nmap security scanner*. https://nmap.org/

6. Maciejak, D. (n.d.). *Hydra*. https://github.com/vanhauser-thc/thc-hydra

7. Kali Linux Tools. (n.d.). *hping3 package description*. https://www.kali.org/tools/hping3/

8. Kali Linux Tools. (n.d.). *Slowhttptest usage example*. https://www.kali.org/tools/slowhttptest/

9. fping.org. (n.d.). *fping*. https://fping.org/

10. BEEF Project. (n.d.). *The Browser Exploitation Framework*. https://beefproject.com/

11. Rapid7. (2011, May 24). *Introducing msfvenom*. https://www.rapid7.com/blog/post/2011/05/24/introducing-msfvenom/

12. Velarde-Alvarado, P., Gonzalez, H., Martínez-Peláez, R., Mena, L. J., Ochoa-Brust, A., Moreno-García, E., Félix, V. G., & Ostos, R. (2022). A novel framework for generating personalized network datasets for NIDS based on traffic aggregation. *Sensors, 22*(5), 1847. https://doi.org/10.3390/s22051847

13. Kim, A., Park, M., & Lee, D. H. (2020). AI-IDS: Application of deep learning to real-time web intrusion detection. *IEEE Access, 8*, 70245–70261. https://doi.org/10.1109/ACCESS.2020.2986882

14. Flood, R., Engelen, G., Aspinall, D., & Desmet, L. (2024). Bad design smells in benchmark NIDS datasets. In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)* (pp. 658–675). IEEE. https://doi.org/10.1109/EuroSP60621.2024.00042

15. Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors, 23*(13), 5941. https://doi.org/10.3390/s23135941

16. Guerra, J. L., Catania, C., & Veas, E. (2022). Datasets are not enough: Challenges in labeling network traffic. *Computers & Security, 115*, 102810. https://doi.org/10.1016/j.cose.2022.102810

17. Ferriyan, A., Thamrin, A. H., Takeda, K., & Murai, J. (2021). Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. *Applied Sciences, 11*(17), 7868. https://doi.org/10.3390/app11177868

18. Damasevicius, R., Venckauskas, A., Grigaliunas, S., Toldinas, J., Morkevicius, N., Aleliunas, T., & Smuikys, P. (2020). LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. *Electronics, 9*(5), 800. https://doi.org/10.3390/electronics9050800

19. Rahman, S., Pal, S., Mittal, S., Chawla, T., & Karmakar, C. (2024). SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security. *Internet of Things, 26*, 101212. https://doi.org/10.1016/j.iot.2024.101212

20. Nikitenko, A., & Bashkov, Y. (2025). Optymizatsiia systemy vyiavlennia merezhevykh vtorhnen na osnovi hlybokoho navchannia iz vykorystanniam metodu zmenshennia rozmirnosti ta metaevrystychnoho alhorytmiv [In Ukrainian]. *Scientific Works of Vinnytsia National Technical University, 1*, 86–98. https://doi.org/10.31649/2307-5376-2025-1-86-98

21. Sanmorino, A., Suryati, S., Gustriansyah, R., Puspasari, S., & Ariati, N. (2024). Feature extraction vs fine-tuning for cyber intrusion detection model. *JURNAL INFOTEL, 16*(2), 302–315. https://doi.org/10.20895/infotel.v16i2.996