

УДК 621.39

DOI <https://doi.org/10.32782/2663-5941/2025.6.1/14>**Рибак А.В.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**Карпенко А.О.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ В ЕМУЛЯЦІЙНІ СЕРЕДОВИЩА GNS3 ДЛЯ АВТОМАТИЗОВАНОЇ ДІАГНОСТИКИ МЕРЕЖ

Розвиток інформаційних технологій супроводжується зростанням складності комп'ютерних мереж, що забезпечують функціонування сучасної цифрової інфраструктури. Питання ефективного моделювання, тестування та діагностики таких мереж набуває особливої актуальності в умовах упровадження технологій IoT, хмарних обчислень і штучного інтелекту. У статті здійснено аналіз сучасних симуляційних та емуляційних платформ, зокрема Cisco Packet Tracer, GNS3, NS-3, OMNeT++, QualNet, Cooja, Mininet та інших, визначено їхні переваги та обмеження у контексті освітнього й практичного використання. Показано, що, незважаючи на широкі функціональні можливості, більшість із них не мають механізмів автоматичного виявлення конфігураційних помилок та інтелектуального аналізу логів. Відсутність таких інструментів ускладнює процес виявлення аномалій, оптимізації топологій і тестування складних мережевих сценаріїв.

У роботі обґрунтовано доцільність інтеграції технологій штучного інтелекту у середовища мережевого моделювання. Запропоновано концепцію веб-платформи "tolka.lab", створеної на базі Flask, MySQL, GNS3 та GPT-4, яка реалізує інтелектуальний підхід до аналізу логів і автоматичного пояснення результатів роботи мережі. Проведене експериментальне тестування довело, що використання системи tolka.lab дозволяє скоротити час виявлення помилок, підвищити якість кінцевих конфігурацій, знизити кількість повторних помилок та мінімізувати потребу у зовнішніх джерелах інформації.

Отримані результати підтверджують, що впровадження штучного інтелекту у засоби емуляції підвищує ефективність діагностики, навчання й практичного адміністрування мереж. Визначено, що подальший розвиток цього напрямку пов'язаний зі створенням спеціалізованих алгоритмів, здатних враховувати особливості протоколів, динаміку мережевих топологій і специфіку поведінки пристроїв у реальному часі. Розроблена концепція tolka.lab відкриває нові можливості для вдосконалення систем навчання, дослідження та проектування сучасних телекомунікаційних мереж, забезпечуючи їхню надійність, масштабованість і стійкість до збоїв.

Ключові слова: емуляція мереж, симуляція, штучний інтелект, GNS3, Cisco Packet Tracer, мережеве моделювання, tolka.lab, Flask, GPT-4.

Постановка проблеми. Сучасні комп'ютерні мережі стають дедалі складнішими, що потребує ефективних засобів їхнього тестування та діагностики без використання фізичного обладнання. Існуючі симулятори та емулятори, попри широкі функціональні можливості, не забезпечують автоматизованого виявлення помилок і глибокого аналізу конфігурацій. Відсутність інтелектуальних систем у таких середовищах знижує ефективність проектування, навчання та практичного адміністрування мереж. Тому актуальною є розробка інструментів, які поєднують можливості мережевого моделювання та штучного інтелекту для підвищення точності діагностики і якості навчального процесу.

Аналіз останніх досліджень і публікацій. Проблематика мережевого моделювання висвітлюється у працях, присвячених використанню симуляторів і емуляторів, таких як Cisco Packet Tracer, NS-3, OMNeT++, QualNet, Mininet, GNS3 тощо [1]. Відомі дослідження зосереджуються на порівнянні функціональності, продуктивності та точності цих інструментів, однак питання автоматизації діагностики залишаються

© Рибак А.В., Карпенко А.О., 2025

Стаття поширюється на умовах ліцензії CC BY 4.0

недостатньо розробленими [1, 2]. Сучасні підходи до використання штучного інтелекту у мережевій сфері здебільшого орієнтовані на аналіз трафіку, виявлення атак або прогнозування навантажень, але майже не охоплюють сферу інтеграції AI у симуляційні та емуляційні середовища. Відсутність універсальних алгоритмів, здатних адаптуватися до особливостей протоколів і динамічної поведінки мережесистем, створює наукову прогалину, яку покликана заповнити дана робота.

Постановка завдання. Метою статті є розробка та дослідження концепції веб-платформи “tolka.lab”, що інтегрує технології штучного інтелекту з емуляційним середовищем GNS3 для автоматизованого аналізу логів і діагностики мережесистем конфігурацій. Для досягнення мети поставлено такі завдання:

- провести аналіз сучасних симуляторів та емуляторів комп’ютерних мереж і визначити їхні обмеження;
- обґрунтувати необхідність інтеграції штучного інтелекту у процес діагностики мережесистем;
- розробити архітектуру веб-платформи “tolka.lab” на базі Flask, MySQL, GNS3 та GPT-4;
- здійснити тестування ефективності роботи платформи порівняно з класичним середовищем GNS3;
- оцінити перспективи подальшого розвитку інтелектуальних систем для мережевого моделювання.

Виклад основного матеріалу. Розвиток інформаційних технологій супроводжується ускладненням комп’ютерних мереж, що забезпечують функціонування цифрової інфраструктури, передачу даних та впровадження інноваційних технологій – IoT, хмарних обчислень і штучного інтелекту. Це зумовлює актуальність планування та тестування мереж для гарантування їхньої надійності та продуктивності. Для аналізу роботи мереж використовуються симулятори та емулятори, що дають змогу моделювати топології, тестувати конфігурації й знаходити помилки без фізичного обладнання. Проте сучасні інструменти обмежені в автоматизованій діагностиці та інтеграції інтелектуальних систем, що вимагає нових підходів до моделювання на основі сучасних технологій.

Інструменти моделювання займають ключове місце у тестуванні мереж. Симулятори працюють із математичними моделями та дискретними подіями, що забезпечує можливість відтворення, але знижує реалістичність [3]. Структуровану класифікацію симуляторів наведено на рисунку 1. Емулятори ж працюють із реальним кодом протоколів, використовуючи ресурси операційної сис-

теми для створення мереж, наближених до реальних, хоча їхні можливості залежать від доступних апаратних ресурсів. Структуровану класифікацію емуляторів наведено на рисунку 2.

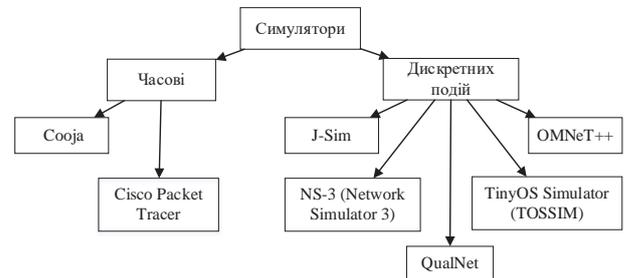


Рис. 1. Класифікація симуляторів мереж

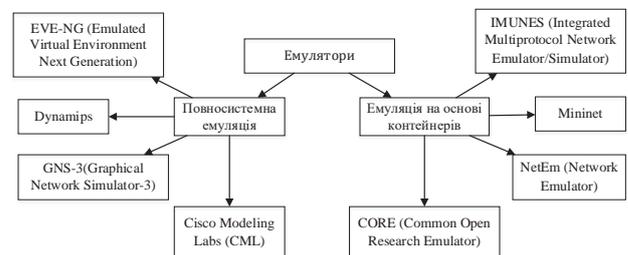


Рис. 2. Класифікація емуляторів мереж

Cisco Packet Tracer (CPT) – середовище для моделювання мереж, IoT і кібербезпеки без фізичного обладнання; використовується у навчанні та підготовці до CCNA/CCNP [4]; NS-3 – симулятор дискретних подій на C++/Python для дослідження мобільних, бездротових, IoT-, 5G- і SDN-мереж та аналізу протоколів [5]; OMNeT++ – об’єктно-орієнтоване середовище моделювання протоколів і розподілених систем, застосовується для складних моделей бездротових, сенсорних, IoT- і 5G-мереж; QualNet – комерційний симулятор для проектування, тестування й оптимізації телекомунікаційної інфраструктури [6]; Cooja – симулятор ОС Contiki для бездротових сенсорних мереж з візуалізацією роботи вузлів; J-Sim – Java-симулятор для дослідження мобільних, сенсорних, супутникових і Mesh-мереж; TOSSIM – симулятор TinyOS, що моделює роботу на низькому рівні для аналізу енергоспоживання й протоколів IoT; GNS3 – безкоштовний емулятор, що працює з образами Cisco IOS/JunOS для побудови віртуальних топологій і тестування конфігурацій [7]; EVE-NG – універсальна платформа для емуляції мереж різних виробників і створення складних гібридних топологій [8]; IMUNES – емулятор на FreeBSD для високоточної симуляції великих мереж і тестування протоколів; CORE – емулятор реального часу для побудови гібридних топологій з віртуальних і фізичних вуз-

лів; Cisco Modeling Labs (CML) – платформа Cisco для моделювання складних мережевих сценаріїв і навчання кібербезпеці [3]; NetEm – Linux-емулятор трафіку для імітації затримок, втрат і інших параметрів мережі; Dynatips – симулятор маршрутизаторів Cisco з реальними IOS-образами; Mininet – легкий емулятор для створення масштабованих мереж і дослідження SDN та нових протоколів.

Симуляційні та емуляційні платформи забезпечують можливість тестування роботи протоколів, пристроїв і сервісів без залучення фізичного обладнання, що особливо важливо як у навчальних цілях, так і для професійної сертифікації та тестування складних мережевих архітектур.

Вибір інструментів симуляції та емуляції є важливим етапом у проектуванні й тестуванні мереж. Порівняння Cisco Packet Tracer і GNS3 є актуальним, оскільки це найбільш поширені інструменти у сфері мережевих технологій. Аналіз проводиться за критеріями: функціональність, точність моделювання, зручність роботи, підтримка протоколів, продуктивність і вимоги до обладнання.

Packet Tracer має навчальну спрямованість: він дозволяє створювати базові топології та налаштовувати основні мережеві сервіси (VLAN, DHCP, DNS, NAT, VPN, IoT), проте відтворює лише логічну модель роботи пристроїв без реальних операційних систем [3]. На відміну від нього, GNS3 використовує повноцінні образи IOS та інших ОС, що забезпечує високий рівень достовірності емуляції маршрутизаторів, комутаторів, серверів і засобів безпеки, а також дає змогу моделювати поведінку мереж під навантаженням і перевіряти сумісність різних платформ. Інтерфейс Packet Tracer характеризується простотою та орієнтований на студентів і початківців, тоді як GNS3 вимагає глибших знань у сфері віртуалізації, але підтримує побудову складних гібридних топологій із застосуванням VMware, Docker чи фізичного обладнання. Packet Tracer реалізує базовий набір протоколів (RIP, OSPF, EIGRP, DHCP, VLAN), тоді як GNS3 забезпечує повноцінну підтримку мережевих стеків Cisco, Juniper, Fortinet, Palo Alto, VyOS і Linux, включно з BGP, MPLS, IPSec VPN та іншими корпоративними технологіями [5]. У продуктивності Packet Tracer демонструє стабільну роботу навіть на малопотужних ПК, тоді як GNS3 значно більш ресурсомісткий і часто потребує серверного чи хмарного середовища. Таким чином, Packet Tracer вирізняється доступністю та простотою, але має суттєві обмеження у точності моделювання та підтримці протоколів. Натомість GNS3 забезпечує професійний рівень емуляції та гнучкість конфігурацій, що робить його придатним

для дослідження складних мережевих інфраструктур, попри більшу складність освоєння та підвищені апаратні вимоги.

Попри стрімкий розвиток засобів моделювання комп'ютерних мереж, проблема ефективної діагностики помилок у симуляційних та емуляційних середовищах залишається актуальною. Навіть сучасні платформи, такі як Cisco Packet Tracer та GNS3, попри широкі можливості, мають обмеження у сфері автоматизованого виявлення та аналізу помилок під час налаштувань або моделювання аварійних сценаріїв [12].

Головним недоліком Cisco Packet Tracer є спрощена симуляція, що ґрунтується на програмному відтворенні логіки пристроїв. Через це частина конфігураційних помилок не виявляється, оскільки не всі процеси мережевого обладнання моделюються повністю. CLI містить індикатор синтаксичних помилок, який підкреслює некоректні команди (рис. 3), однак інструменти візуалізації трафіку та система перевірки завдань спрямовані переважно на базові команди й не забезпечують глибокий аналіз налаштувань.

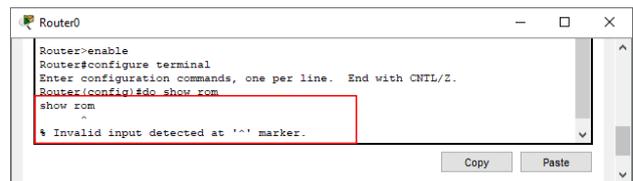


Рис. 3. Індикатор помилок в Command Line Interface

У GNS3, попри емуляцію реальних операційних систем і повну підтримку протоколів, процес діагностики залишається складним. Система не має механізмів автоматичного виявлення конфігураційних помилок, тому користувач самостійно перевіряє сервіси, аналізує журнали та застосовує стандартні команди (show, debug, ping, traceroute). У складних топологіях це вимагає високої кваліфікації та значних часових витрат. Приклад такого аналізу наведено на рис. 4, де GNS3 використовує реальний образ Cisco IOS.



Рис. 4. Індикатор помилок в GNS3 з підтримкою образу Cisco IOS

Додатковим ускладненням є відсутність комплексної моделі моделювання низькорівневих мере-

жевих проблем, таких як затримки, втрати пакетів, перевантаження каналів, вплив QoS-політик або деградація продуктивності при аваріях. Через це інженери не завжди можуть перевірити поведінку мережі в екстремальних або нестандартних умовах до її впровадження. Окремо варто наголосити на відсутності інтеграції інтелектуальних систем підтримки прийняття рішень. Більшість емуляторів не використовують алгоритми штучного інтелекту, які могли б аналізувати топології, виявляти типові помилки, пропонувати оптимальні налаштування чи попереджати про ризики. Наявність таких інструментів значно полегшила б роботу мережеских інженерів.

Застосування ШІ могло б забезпечити автоматичну перевірку логіки налаштувань протоколів, розпізнавання конфігураційних помилок, оптимізацію топологій за критеріями продуктивності й безпеки, моделювання роботи мережі в аварійних сценаріях і раннє виявлення вразливостей під час зростання навантаження. Втім, інтеграція ШІ у мережеві емулятори потребує спеціалізованих алгоритмів, адаптованих до специфіки протоколів та динаміки мережевої взаємодії. Нині відсутні готові модулі, які можна безпосередньо підключити до GNS3 чи Cisco Packet Tracer. Більшість open-source рішень орієнтовані на обробку тексту, зображень або даних, але не на мережеві сценарії. Тому створення інтелектуальної діагностичної системи потребує власних моделей ШІ, здатних аналізувати мережеву поведінку, виявляти аномалії та оптимізувати конфігурації. Для реалізації такого підходу була розроблена веб-платформа “tolka.lab”, яка побудована на основі мікрофреймворка для вебдодатків – Flask, який створений на основі мови програмування Python [9]. Завдяки даному фреймворку, ми змогли побудувати гнучку та просту веб-платформу з можливістю масштабування та підключення зовнішніх сервісів.

Основною задачею платформи є забезпечення користувачів зручним інтерфейсом для реєстрації та входу в систему, управління персональними профілями, публікацією постів на тематику мереж, а також для інтеграції з системою віртуалізації мережеских топологій GNS3 та автоматичного аналізу логів за допомогою інтегрованої моделі штучного інтелекту.

Архітектура системи відповідає класичній моделі «клієнт–сервер–база даних» із використанням зовнішніх API-сервісів (рис. 5). Клієнтська частина реалізована у вигляді веб-інтерфейсу, що взаємодіє із сервером через HTTPS із використанням SSL-сертифікатів. Для її побудови засто-

совано HTML5, CSS3 та JavaScript [10]. Серверна частина представлена застосунком на Flask, який обробляє HTTP-запити, формує веб-сторінки та забезпечує доступ до бази даних. У серверній логіці використано Python 3 та відповідні бібліотеки: flask (маршрутизація, сесії, шаблони), flask_mysql (інтеграція з MySQL), g4f.client (взаємодія з моделями GPT-4), hashlib (хешування), datetime, os, werkzeug.utils (безпечна обробка файлів), MySQLdb.cursors (словникові результати SQL-запитів), base64 (кодування), requests (зовнішні API-запити), gc (очищення логів після AI-обробки). База даних – реляційна система управління базами даних MySQL, у якій зберігаються усі ключові дані для входу користувачів в систему, відображення постів та проєктів. Розроблена на основі phpmyadmin (рис. 6). Файлова система використовується для зберігання завантажених зображень. Інтегровані зовнішні сервіси: GNS3 Server API – для роботи з мережевими топологіями, та GPT-4 (через g4f.client) – для автоматизованого генерування пояснень логів [11].

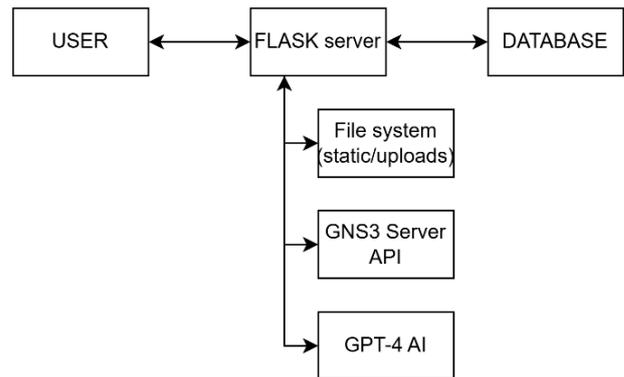


Рис. 5. Архітектура системи

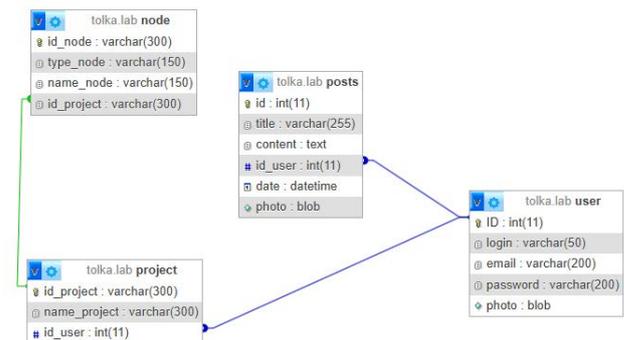


Рис. 6. Схема даних БД “tolka.lab”

У системі реалізовано комплекс механізмів безпеки. Паролі зберігаються у вигляді SHA-256 хешів, що унеможливує їх відновлення у разі компрометації. Цілісність і конфіденційність сесій забезпечує секретний ключ Flask. Під час завантаження файлів перевіряється їх тип, що запобігає

розміщенню шкідливого контенту. Передавання даних між клієнтом і сервером відбувається через захищене SSL-з'єднання.

Веб-платформа підтримує реєстрацію та автентифікацію користувачів: під час створення облікового запису пароль хешується, а під час входу відбувається перевірка логіна та хешу. Користувач може переглядати й оновлювати дані профілю, змінювати аватар (що зберігається у форматі Base64) та оновлювати пароль. Платформа дозволяє створювати та переглядати публікації, що містять заголовок, текст і додаткові медіа. Усі матеріали зберігаються в базі даних із зазначенням автора. Окремий модуль забезпечує роботу з мережевими проєктами через інтеграцію з GNS3: система отримує перелік проєктів, завантажує дані про вузли та передає логи моделі GPT-4, яка формує автоматизовані пояснення щодо стану мережевої інфраструктури.

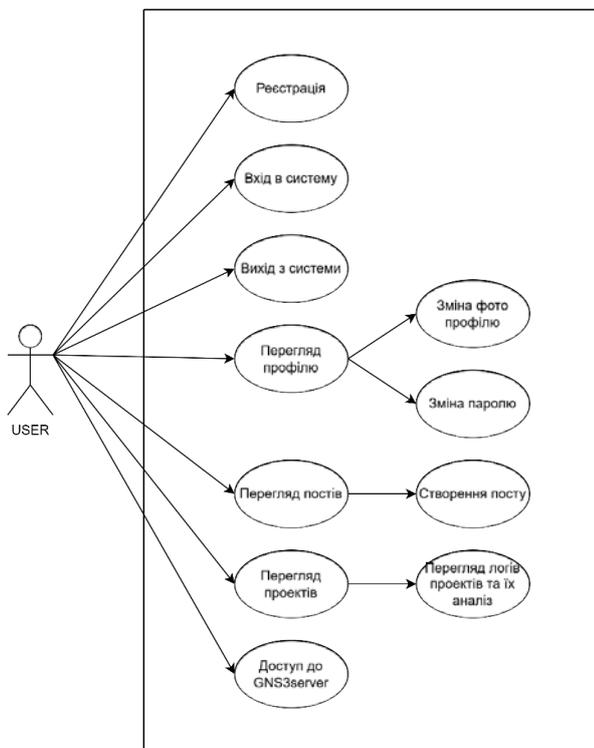


Рис. 7. Use-case діаграма користувача веб-платформи

Для оцінювання функціональності та працездатності розробленої веб-платформи її необхідно розгорнути на Ubuntu 24.04 Desktop (або Ubuntu Server; Desktop обрано для зручності роботи з інструментами розробки). Попередньо створюється віртуальне середовище у Oracle VirtualBox з використанням ISO-образу ОС. Використання віртуалізації забезпечує ізоляцію середовища, контрольовані умови тестування та зменшення впливу зовнішніх чинників.

Після налаштування віртуальної машини, розпочинаємо встановлювати програмні пакети, але перед цим обов'язково оновити існуючі системні пакети для забезпечення актуальності програмного забезпечення та усунення потенційних вразливостей, використовуючи наступні команди:

```
apt update && sudo apt upgrade -y
```

```
apt install mc nano apache2 nginx mysql-server python3 python3-pip python3-venv git php php-cli php-json php-gd php-ldap php-odbc php-pear php-xml php-xmlrpc php-mbstring php-snmp php-soap php-zip php-imap php php-fpm libapache2-mod-php php-mysql mariadb-server phpmyadmin -y
```

З метою забезпечення шифрованого з'єднання HTTPS налаштовується Apache на порту 8080. Для цього редагуємо файл `/etc/apache2/ports.conf`. Змінюємо `Listen 80` на `Listen 8080`. Також у файлі `/etc/apache2/sites-available/000-default.conf` змінюємо `<VirtualHost *:80>`. Встановлюється Stunnel (`apt install stunnel -y`), який працює як SSL-проксі, забезпечуючи шифрування трафіку. SSL-сертифікати створюються командою `openssl req -newkey rsa:8192 -x509 -sha256 -days 365 -nodes -out cert.pem -keyout cert.pem`.

Після запуску служби баз даних `systemctl start mariadb` проводиться її базова безпечна конфігурація (`mysql_secure_installation`) – задається пароль адміністратора, видаляються анонімні користувачі, забороняється віддалений root-доступ тощо. Перезапускається веб-сервер `systemctl restart nginx`, після чого phpMyAdmin доступний за HTTPS-адресою.

Після встановлення необхідних компонентів створюється структура проєкту. Веб-платформа розташовується в системному каталозі `/var/www/tolka`, який прийнято використовувати для зберігання веб-застосунків у більшості Linux-систем.

Для ізоляції залежностей проєкту та зручності розробки створюється віртуальне середовище за допомогою команди `python3 -m venv venv`. Команда створює окремий інтерпретатор Python з незалежними бібліотеками, що дозволяє уникнути конфліктів версій пакетів у глобальній системі. Активація віртуального середовища здійснюється командою `source venv/bin/activate`. Після активації всі встановлення бібліотек через `pip` будуть виконуватись у межах цього віртуального середовища.

Створюється файл `app.py`, в якому буде прописано backend веб-платформи.

Також створюються SSL-сертифікати для Flask, які підключаються безпосередньо в самому коді файлу `app.py`.

```
mkdir ~/ssl
cd ~/ssl
```

```
openssl genpkey -algorithm RSA -out server.key
openssl req -new -key server.key -out server.csr
openssl x509 -req -in server.csr -signkey server.
key -out server.crt
```

Для безпосереднього написання та супроводу програмного коду використовується середовище розробки Visual Studio Code (VS Code). Це сучасний багатофункціональний редактор, що підтримує численні розширення, інтеграцію з системами контролю версій, дебагінг, автодоповнення коду та багато інших функцій, що підвищують продуктивність розробника.

Для моделювання комп'ютерних мереж та отримання логів у межах практичного дослідження був використаний open source емулятор мережевого програмного забезпечення GNS3. Для його встановлення в систему було використано наступні команди:

```
apt update && sudo apt upgrade -y
add-apt-repository ppa:gns3/ppa
apt update
apt install gns3-server gns3-gui -y
```

У процесі інсталяції система може вимагати підтвердження встановлення залежностей – на всі підтвердження дається відповідь YES. Для перевірки успішної інсталяції виконується команда перегляду версій серверної та клієнтської частин:

```
gns3server --version
gns3 --version
```

Після встановлення створюється конфігураційна директорія для серверної частини GNS3:

```
mkdir -p ~/.config/GNS3
```

У даному каталозі створюється файл конфігурації сервера `gns3_server.conf`:

```
nano ~/.config/GNS3/gns3_server.conf
```

У файл конфігурації вносяться наступні параметри:

```
[Server]
host = 0.0.0.0
port = 3080
local = false
auth = false
console_start_port_range = 5000
console_end_port_range = 10000
```

Після цього запускається сервер GNS3 з вказаною конфігурацією:

```
gns3server -config ~/.config/GNS3/gns3_server.conf
```

Таким чином, було успішно розгорнуто та налаштовано локальне серверне оточення для емуляції та моделювання мережевих сценаріїв у середовищі GNS3.

Після написання backend та frontend та підключення всіх зовнішніх залежностей, ми можемо

протестувати веб-платформу.

Під час тестування участь взяли 10 респондентів, умовно поділених на дві рівні групи по 5 осіб. Кожен учасник мав базові навички роботи з GNS3 та однаковий рівень підготовки. Перша група працювала з GNS3, виконуючи конфігурування мережевої топології, аналіз логів вручну та усунення типових помилок. Друга група виконувала аналогічні завдання, але в середовищі `tolka.lab`, яка надавала пояснення щодо поточного стану мережі та вказувала на можливі джерела проблем. Результати оцінки за основними параметрами представлені в табл. 1.

Таблиця 1

Результати тестування

Параметр оцінки	GNS3	tolka.lab
Середній час виявлення помилок (хв)	11,2	4,6
Кількість допущених повторних помилок	7	2
Якість фінальної конфігурації (макс. 10 балів)	6,4	9,1
Суб'єктивна оцінка зручності (від 1 до 5)	3,0	4,8
Потреба в додатковому пошуку в інтернеті	Висока (усі учасники)	Мінімальна (1 учасник)

Отримані результати підтверджують підвищення ефективності роботи в середовищі `tolka.lab`. Автоматизований аналіз логів зменшив час діагностики, підвищив якість конфігурацій та зручність роботи завдяки інтеграції з GNS3 і використанню моделей GPT. Тестування показало, що застосування III в емуляційних платформах підвищує продуктивність, зменшує кількість критичних помилок, покращує практичні навички та знижує залежність від зовнішніх джерел, що забезпечує `tolka.lab` перевагу над традиційним GNS3 у навчальних і прикладних завданнях.

Розроблена платформа, що поєднує Flask, GNS3 та GPT-4, забезпечує інтелектуальний аналіз логів і формування текстових пояснень стану мережі, частково усуваючи недоліки класичних емуляторів. За результатами тестування користувачі `tolka.lab` продемонстрували швидше виявлення помилок, меншу кількість повторних неточностей, кращу якість фінальних конфігурацій та вищу суб'єктивну зручність роботи [13].

Водночас встановлено, що подальша інтеграція III в мережеве моделювання потребує розробки алгоритмів, здатних враховувати специфіку протоколів і динамічні зміни топологій. Це визначає перспективний напрям розвитку інструментів

моделювання, навчання й тестування.

Висновки. У результаті проведеного дослідження проаналізовано сучасні симуляційні та емуляційні інструменти моделювання комп'ютерних мереж, зокрема Cisco Packet Tracer, GNS3, NS-3, OMNeT++, QualNet, Cooja, Mininet та інші. Встановлено, що, попри високий рівень функціональності, жоден із розглянутих інструментів не забезпечує повноцінної автоматизованої діагностики конфігураційних помилок і не інтегрує інтелектуальні алгоритми для аналізу мережевих процесів. Це обмежує їх застосування у складних сценаріях тестування та навчальних середовищах.

Запропоновано новий підхід до підвищення ефективності моделювання мереж шляхом інтеграції технологій штучного інтелекту в емуляційні платформи. Реалізована веб-платформа "tolka.lab", створена на базі Flask, MySQL, GNS3 та GPT-4, продемонструвала практичну ефективність у задачах автоматичного аналізу логів та пояснення станів мережевих топологій.

Результати експериментального тестування засвідчили, що використання "tolka.lab" дозволяє скоротити середній час виявлення помилок більш ніж удвічі, підвищити якість фінальних конфігурацій, зменшити кількість повторних помилок і мінімізувати залежність користувачів від зовнішніх джерел інформації.

Отримані результати підтверджують перспективність впровадження штучного інтелекту у процес моделювання та навчання мережевих технологій. Подальші дослідження мають бути спрямовані на розробку спеціалізованих алгоритмів ШІ, здатних аналізувати поведінку складних топологій у реальному часі, прогнозувати відмови та автоматично оптимізувати конфігурації. Реалізація таких рішень сприятиме створенню нового покоління інтелектуальних систем мережевого моделювання, що поєднуюватимуть точність емуляції, гнучкість налаштувань і високий рівень автоматизації.

Список літератури:

1. Gomez J., Kfoury E. F., Crichigno J., Srivastava G. A Survey on Network Simulators, Emulators, and Testbeds used for Research and Education: *Computer Networks*. 2023. Vol. 237. Art. 110054. DOI: 10.1016/j.comnet.2023.110054.
2. Ergün S. A survey on how network simulators serve reinforcement learning: *Computer Networks*. 2023. Vol. 234. Art. 109934. DOI: 10.1016/j.comnet.2023.109934.
3. Tufail M., Javaid N., Khan Z. A. Teaching Computer Networks using Cisco Packet Tracer: *International Journal of Modern Education and Computer Science*. 2020. Vol. 12, No 5. P. 1–10.
4. Cisco Networking Academy. Introduction to Networks v7.0. Cisco Press, 2021. P. 420.
5. ns-3: A Discrete-Event Network Simulator for Internet Systems. URL: <https://www.nsnam.org>.
6. Qualnet. Une entreprise canadienne qui se consacre à l'installation et à l'entretien d'infrastructures de réseaux sans fil, ainsi qu'à la construction des structures associées. URL: <https://qualnet.ca/en/>.
7. GNS3 Documentation. GNS3 official user guide. URL: <https://docs.gns3.com>.
8. EVE-NG. Official website of the network emulation tool. URL: <https://www.eve-ng.net>.
9. Flask. Encyclopedic Dictionary of Archaeology. Cham, 2021. P. 484. URL: https://doi.org/10.1007/978-3-030-58292-0_60201
10. Oleshchenko L. M. Fundamentals of Web Programming. Kyiv : Igor Sikorsky Kyiv Polytechnic Institute, 2021. P. 138.
11. Gaithersburg. National Institute of Standards and Technology. 2023. P. 96. URL: <https://nvlpubs.nist.gov/nistpubs/AI/NIST.AI.100-1.pdf>.
12. Карпенко А., Толстова А. Сучасний стан та перспективи розвитку емуляторів мереж. Матеріали III Міжнародної науково-практичної конференції "Кібербезпека державних інституцій та подолання кризових станів" (Том 1), м. Київ, 14 листопада 2024 р. Київ: ІСЗЗІ КПІ ім. Ігоря Сікорського, 2024. С. 337–338.
13. Карпенко А., Толстова А. Особливості застосування засобів моделювання мереж. Матеріали VIII науково-практичної конференції "Актуальні питання застосування спеціальних інформаційно-комунікаційних систем", м. Київ, 18 червня 2025 р. Київ : ІСЗЗІ КПІ ім. Ігоря Сікорського, 2025. С. 348–349.

Rybak A.V., Karpenko A.O. INTEGRATION OF ARTIFICIAL INTELLIGENCE INTO GNS3 EMULATION ENVIRONMENTS FOR AUTOMATED NETWORK DIAGNOSTICS

The development of information technologies is accompanied by an increase in the complexity of computer networks that support the functioning of modern digital infrastructure. The issue of effective modelling, testing, and diagnostics of such networks is becoming particularly relevant in the context of the introduction of IoT technologies, cloud computing, and artificial intelligence. The article analyses modern simulation and emulation platforms, including Cisco Packet Tracer, GNS3, NS-3, OMNeT++, QualNet, Cooja, Mininet and

others, and identifies their advantages and limitations in the context of educational and practical use. It is shown that, despite their wide range of functionalities, most of them do not have mechanisms for automatic detection of configuration errors and intelligent analysis of logs. The lack of such tools complicates the process of detecting anomalies, optimising topologies and testing complex network scenarios.

The paper justifies the feasibility of integrating artificial intelligence technologies into network modelling environments. It proposes the concept of the *tolka.lab* web platform, created on the basis of Flask, MySQL, GNS3 and GPT-4, which implements an intelligent approach to log analysis and automatic explanation of network performance results. Experimental testing has proven that using the *tolka.lab* system reduces error detection time, improves the quality of final configurations, reduces the number of repeat errors, and minimises the need for external sources of information.

The results confirm that the introduction of artificial intelligence into emulation tools increases the efficiency of diagnostics, training, and practical network administration. It has been determined that further development in this area is linked to the creation of specialised algorithms capable of taking into account the characteristics of protocols, the dynamics of network topologies and the specific behaviour of devices in real time. The developed concept of *tolka.lab* opens up new opportunities for improving the systems of training, research, and design of modern telecommunications networks, ensuring their reliability, scalability, and resistance to failures.

Key words: network emulation, simulation, artificial intelligence, GNS3, Cisco Packet Tracer, network modelling, *tolka.lab*, Flask, GPT-4.

Дата надходження статті: 29.11.2025

Дата прийняття статті: 18.12.2025

Опубліковано: 30.12.2025