МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЗАЄЦЬ Олександр Юрійович

Програмна система аналізу мережевих логів з використанням NLP-підходу для виявлення аномалій / Software System for Network Log Analysis Using an NLP Approach for Anomaly Detection

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
О. Ю. Заєць

_____

Науковий керівник:
д.т.н., професор, М. П. Комар

_____

Кваліфікаційну роботу допущено
до захисту
«___»_____ 2026 р.

Завідувач кафедри
_____ Н.В. Дзюбановська

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
ЗАЙЦЮ Олександру Юрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Програмна система аналізу мережевих логів з використанням NLP-підходу для виявлення аномалій / Software System for Network Log Analysis Using an NLP Approach for Anomaly Detection
керівник роботи д.т.н., професор М. П. Комар
затверджені наказом по університету від 01 грудня 2025 року № 894.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити
- проаналізувати предметну область виявлення аномалій у мережевих логах;
- дослідити особливості мережевих логів як джерела даних для інтелектуального аналізу;
- виконати аналіз існуючих підходів і методів виявлення аномалій, зокрема на основі машинного навчання, глибокого навчання та NLP;
- сформувати вимоги до програмної системи аналізу мережевих логів;
- розробити архітектуру програмної системи та структуру її основних модулів;
- розробити алгоритм попереднього оброблення логів і перетворення їх у текстове подання, придатне для NLP-аналізу;
- реалізувати модуль виявлення аномалій на основі вибраного NLP-підходу;
- провести експериментальне оцінювання якості роботи системи;
- сформувати висновки щодо ефективності запропонованого рішення.
5. Перелік графічного матеріалу в роботі
- архітектура програмної системи аналізу мережевих логів;
- структура модуля попереднього оброблення та текстового подання логів;

– структура модуля NLP-аналізу та виявлення аномалій;
– схема алгоритму завантаження та попереднього оброблення мережевих логів;
– схема алгоритму формування текстового подання мережевого логу;
– схема алгоритму NLP-класифікації мережевого логу.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2026 р.	

Студент_____ О. Ю. Заєць
підпис

Керівник роботи_____ д.т.н., професор М. П. Комар
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмна система аналізу мережевих логів з використанням NLP-підходу для виявлення аномалій» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 68 сторінок і містить 15 ілюстрацій, 8 таблиць, 2 додатки та 38 використаних джерел.

Метою роботи є розроблення програмної системи аналізу мережевих логів з використанням NLP-підходу для виявлення аномальних подій у мережевому трафіку та оцінювання ефективності її функціонування.

Методи дослідження включають методи системного аналізу, попереднього оброблення даних, нормалізації мережевих логів, формування текстового подання записів, токенізації, векторизації, машинного навчання, NLP-класифікації, а також методи моделювання та експериментального оцінювання якості програмної системи.

Розроблено програмну систему аналізу мережевих логів, яка забезпечує завантаження вхідних логових записів, перевірку їх структури, очищення та нормалізацію даних, відбір інформативних атрибутів, формування стандартизованого текстового подання мережевої події, токенізацію текстових записів, класифікацію подій за допомогою NLP-моделі, а також фіксацію результатів у вигляді мітки класу «норма» або «аномалія» та показників якості класифікації.

Результати дослідження можуть бути використані для створення та вдосконалення програмних засобів моніторингу мережевої інфраструктури, автоматизації аналізу мережевих логів, своєчасного виявлення підозрілих подій, зменшення навантаження на фахівців з кібербезпеки та підвищення ефективності реагування на потенційні інциденти.

Ключові слова: МЕРЕЖЕВІ ЛОГИ, NLP-ПІДХІД, АНОМАЛІЇ, КІБЕРБЕЗПЕКА, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ, ПОПЕРЕДНЄ ОБРОБЛЕННЯ ДАНИХ, ВИЯВЛЕННЯ ВТОРГНЕНЬ.

ANNOTATION

Qualification work on the topic «Software System for Network Log Analysis Using an NLP Approach for Anomaly Detection» for Bachelor's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 68 pages and it contains 15 figures, 8 tables, 2 annexes and 38 sources.

The purpose of the work is to develop a software system for network log analysis using an NLP-based approach to detect anomalous events in network traffic and evaluate the effectiveness of its operation.

The research methods include methods of system analysis, data preprocessing, network log normalization, formation of textual representations of records, tokenization, vectorization, machine learning, NLP-based classification, as well as methods of modeling and experimental evaluation of the software system quality.

A software system for network log analysis has been developed. It provides loading of input log records, verification of their structure, data cleaning and normalization, selection of informative attributes, formation of a standardized textual representation of a network event, tokenization of text records, classification of events using an NLP model, and recording of results in the form of a class label "normal" or "anomaly" and classification quality indicators.

The research results can be used to create and improve software tools for monitoring network infrastructure, automating network log analysis, timely detection of suspicious events, reducing the workload of cybersecurity specialists, and increasing the efficiency of response to potential incidents.

Keywords: NETWORK LOGS, NLP-BASED APPROACH, ANOMALIES, CYBERSECURITY, MACHINE LEARNING, CLASSIFICATION, DATA PREPROCESSING, INTRUSION DETECTION.

ЗМІСТ

ВСТУП

Стрімке зростання обсягів мережевого трафіку, ускладнення архітектури інформаційно-комунікаційних систем і постійна еволюція кіберзагроз зумовили потребу в розробленні ефективних засобів автоматизованого аналізу мережевих подій. У сучасних умовах мережеві логи є одним із головних джерел інформації про стан інфраструктури, характер взаємодії між вузлами мережі, спроби несанкціонованого доступу, відхилення від типової поведінки та ознаки потенційних атак. Саме тому своєчасне виявлення аномалій у логах набуло важливого значення для забезпечення кібербезпеки, підтримання працездатності сервісів і зниження ризику порушення цілісності, конфіденційності та доступності даних [9, 18].

Традиційні системи виявлення вторгнень значною мірою спираються на сигнатурні, статистичні або класичні алгоритмічні підходи. Такі методи демонструють належну ефективність у випадках, коли атака вже відома або має стабільні характерні ознаки. Водночас їх застосування часто супроводжується низкою обмежень. По-перше, сигнатурні механізми погано адаптуються до нових або модифікованих атак. По-друге, значна кількість правил і параметрів ускладнює масштабування системи. По-третє, висока частота хибних спрацьовувань знижує практичну цінність результатів моніторингу та перевантажує фахівців з безпеки [1, 9, 20]. У наукових джерелах також відзначено, що мінімізація хибнопозитивних випадків залишається однією з ключових проблем розвитку сучасних IDS [9].

У зв’язку з цим значного поширення набули методи машинного навчання та глибокого навчання, які дають змогу виявляти приховані закономірності в мережевих даних і формувати моделі розпізнавання складних аномальних сценаріїв. Для задач виявлення вторгнень застосовуються дерева рішень, Random Forest, XGBoost, методи градієнтного бустингу, згорткові та рекурентні нейронні мережі, а також гібридні архітектури [4, 7, 16, 24, 26]. Такі підходи забезпечують вищу гнучкість порівняно з класичними сигнатурними системами, однак також мають певні недоліки. Насамперед ідеться про залежність від якості

підготовки ознак, чутливість до дисбалансу класів, складність інтерпретації результатів і необхідність адаптації до нових джерел логових даних [12, 20].

Останніми роками особливий інтерес викликає використання методів оброблення природної мови в задачах кібербезпеки. Поява великих мовних моделей суттєво посилила інтерес до цього напряму, оскільки вони продемонстрували здатність працювати зі складними контекстними залежностями, текстовими шаблонами та семантичними зв'язками [2, 3, 8, 17, 23].

У сучасних дослідженнях простежується тенденція до інтеграції генеративних моделей і мовних підходів у процес аналізу кіберінцидентів, підтримки конфігурації, пояснення рішень і виявлення аномалій за логами [3, 6, 8, 17, 22, 23]. Разом із тим у більшості практичних реалізацій зберігаються проблеми вартості обчислень, складності розгортання, неоднорідності логових форматів і необхідності попереднього перетворення даних у придатне для текстового аналізу подання [2, 8].

Для мережевих логів ця проблема є особливо актуальною. На відміну від звичайного тексту, логові записи мають напівструктуровану природу, містять IP-адреси, порти, часові мітки, коди подій, службові атрибути, фрагменти системних повідомлень та інші елементи, що потребують спеціального оброблення перед поданням у модель. Отже, ефективність такого підходу визначається не лише вибором алгоритму класифікації, а й якістю попереднього очищення, нормалізації, токенізації, перетворення ознак у текстове представлення та формування кінцевого набору даних для аналізу. Саме цей етап часто визначає практичну придатність усієї системи.

Актуальність теми роботи зумовлена потребою в розробленні програмних засобів, здатних виконувати інтелектуальний аналіз мережевих логів із використанням сучасних NLP-підходів для виявлення аномалій. Така система має поєднувати переваги традиційного підходу до моніторингу мережевих подій із можливостями сучасних мовних і класифікаційних моделей. Крім того, важливим є врахування напрацювань у сфері виявлення кібератак, представлених у роботах українських і зарубіжних дослідників, зокрема

досліджень, присвячених нейромережевим системам кіберзахисту, зменшенню розмірності даних, адаптивним системам детекції та ієрархічній класифікації атак [5, 10–15, 28–33].

Метою роботи є розроблення програмної системи аналізу мережевих логів з використанням NLP-підходу для виявлення аномальних подій у мережевому трафіку та оцінювання ефективності її функціонування.

Для досягнення поставленої мети в роботі сформовано такі завдання дослідження:

– проаналізувати предметну область виявлення аномалій у мережевих логах;

– дослідити особливості мережевих логів як джерела даних для інтелектуального аналізу;

– виконати аналіз існуючих підходів і методів виявлення аномалій, зокрема на основі машинного навчання, глибокого навчання та NLP;

– сформувати вимоги до програмної системи аналізу мережевих логів;

– розробити архітектуру програмної системи та структуру її основних модулів;

– розробити алгоритм попереднього оброблення логів і перетворення їх у текстове подання, придатне для NLP-аналізу;

– реалізувати модуль виявлення аномалій на основі вибраного NLP-підходу;

– провести експериментальне оцінювання якості роботи системи;

– сформувати висновки щодо ефективності запропонованого рішення.

Об’єктом дослідження є процес аналізу мережевих логів у задачах виявлення аномалій у комп’ютерних мережах.

Предметом дослідження є методи, моделі та програмні засоби оброблення мережевих логів на основі NLP-підходу для автоматизованого виявлення аномальних записів.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах IV Всеукраїнської науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі» (ІКСМ), м.

Тернопіль, ЗУНУ, 20 травня 2026 р. (додаток Б).

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Характеристика мережевих логів як джерела даних для виявлення аномалій

Предметна область дослідження охоплює процеси збору, зберігання, оброблення та аналізу мережевих логів з метою виявлення аномальних подій, що можуть свідчити про збої, несанкціоновані дії, вторгнення або інші небезпечні стани мережевої інфраструктури. У сучасних інформаційно-комунікаційних системах мережеві логи є одним із основних джерел даних для контролю стану мережі, аналізу поведінки вузлів, фіксації службових подій і виявлення потенційних кіберзагроз [9, 18].

Мережева інфраструктура постійно генерує значні обсяги журналів подій. Такі журнали формуються маршрутизаторами, комутаторами, міжмережевими екранами, серверами, системами виявлення вторгнень, проксі-серверами, засобами автентифікації та прикладним програмним забезпеченням. У результаті утворюється великий потік записів, які містять відомості про джерело й призначення з’єднання, часову мітку, протокол, номер порту, тип події, статус запиту, службові параметри сеансу та інші характеристики. Саме ця інформація є основою для подальшого аналізу мережевої активності [20, 21].

На відміну від багатьох інших типів даних, мережеві логи мають напівструктурований характер. Частина полів у них є чітко формалізованою, наприклад IP-адреса, порт, тип протоколу або код відповіді. Інша частина подається у вигляді текстових повідомлень, службових описів або комбінованих рядків. Така особливість ускладнює безпосереднє застосування стандартних алгоритмів аналізу, оскільки потребує попереднього очищення, нормалізації та узгодження формату представлення даних. Крім того, в логах можуть бути присутні дублікати, пропущені значення, шум, нетипові службові фрагменти та різнорідні шаблони запису. Це суттєво впливає на якість подальшого виявлення аномалій [20, 29].

У загальному вигляді мережевий лог можна розглядати як послідовність

записів про події, що виникають у процесі функціонування мережі. Кожен такий запис описує окрему дію або стан: встановлення з'єднання, передавання пакета, спробу доступу до ресурсу, помилку маршрутизації, зміну конфігурації, відмову автентифікації або іншу подію, яка має значення для моніторингу. Сукупність цих записів дає змогу сформувати уявлення про типову поведінку системи. Відхилення від цієї поведінки розглядаються як потенційні аномалії.

Аномалія в контексті мережевого аналізу – це подія, запис або послідовність записів, що істотно відрізняються від нормального профілю функціонування системи. До аномалій можуть належати різкі зміни інтенсивності трафіку, нехарактерні поєднання адрес і портів, повторювані невдалі спроби входу, нетипові часові шаблони активності, а також послідовності подій, пов'язані з відомими або новими атаками. Особливість таких відхилень полягає в тому, що частина з них має явні ознаки, тоді як інші можуть бути прихованими й виявлятися лише через складні залежності між параметрами [1, 9, 18].

У системах кіберзахисту мережеві аномалії зазвичай пов'язуються з кількома групами подій. До першої групи належать атаки, що мають ознаки сканування, перебору або розвідки мережі. До другої – події, пов'язані з несанкціонованим доступом, ескалацією привілеїв або використанням вразливостей. До третьої – аномалії, викликані шкідливим програмним забезпеченням, автоматизованими ботами, перевантаженням сервісів чи маніпуляціями з трафіком. Окремо доцільно враховувати технічні аномалії, зумовлені не атакою, а збоями обладнання, помилками конфігурації, деградацією сервісу або некоректною роботою прикладних компонентів. Таким чином, задача аналізу логів не зводиться лише до фіксації атак, а охоплює ширший клас відхилень у поведінці мережевого середовища [12, 20].

У практичному аспекті аналіз мережевих логів виконує кілька важливих функцій. Насамперед він забезпечує моніторинг стану мережі в режимі, близькому до реального часу. Крім того, він дає змогу виявляти ознаки атак на ранніх стадіях, фіксувати підозрілі сценарії поведінки, накопичувати статистику про події безпеки та формувати основу для подальшого реагування. У разі

правильної організації такого процесу логовий аналіз перетворюється на один із центральних елементів системи підтримки кіберзахисту [11, 13, 14].

Разом із цим предметна область має низку характерних проблем. По-перше, обсяги мережевих логів є дуже великими, тому ручний аналіз швидко стає неефективним. По-друге, дані надходять із різнорідних джерел і мають відмінний формат. По-третє, типові шаблони поведінки мережі можуть істотно змінюватися залежно від часу доби, навантаження, ролі користувачів і режиму роботи сервісів. По-четверте, реальні атаки нерідко маскуються під нормальну активність, через що традиційні правила або жорсткі порогові механізми не завжди дають надійний результат. Усе це зумовило активний розвиток інтелектуальних підходів до оброблення логів [1, 12, 18, 20].

Класичні системи виявлення вторгнень переважно спиралися на сигнатурний або статистичний аналіз. Сигнатурний підхід є ефективним у разі наявності заздалегідь відомих ознак атаки, однак погано працює для нових або модифікованих сценаріїв. Статистичний підхід орієнтований на виявлення відхилень від середніх характеристик мережі, однак чутливий до вибору параметрів і часто генерує хибні спрацьовування [1, 9]. У зв’язку з цим дедалі ширше застосовуються методи машинного навчання, які дають змогу формувати моделі нормальної та аномальної поведінки на основі даних.

У задачах виявлення аномалій у мережевих логах використовуються дерева рішень, Random Forest, XGBoost, ансамблеві моделі, глибокі нейронні мережі, а також гібридні архітектури, що поєднують переваги кількох підходів [4, 7, 16, 24, 26]. Такі методи демонструють вищу адаптивність до складних закономірностей у даних, проте їх ефективність значною мірою залежить від якості сформованих ознак. Саме тому питання представлення мережевого логу стає центральним у всій предметній області.

Останній етап розвитку цієї галузі пов’язаний із залученням NLP-підходів. Якщо традиційно лог розглядався як набір полів для числового або категоріального аналізу, то в межах NLP-підходу логовий запис трактується як спеціалізований текстовий об’єкт. Такий підхід дає змогу враховувати не лише окремі значення полів, а й контекст, послідовність, зв’язки між подіями та

текстові патерни, притаманні журналам системних і мережевих повідомлень [2, 3, 23].

Для прикладу, запис логу з полями типу source IP, destination IP, port, protocol, status і timestamp може бути перетворений у текстову конструкцію, яка описує сутність події у більш цілісному вигляді. У такому разі модель аналізує не лише окремі атрибути, а й цілісне представлення події як послідовності токенів. Саме ця ідея лежить в основі сучасних підходів до логового аналізу з використанням великих мовних моделей та спеціалізованих NLP-моделей [3, 23, 27].

На рисунку 1.1 показано місце NLP-підходу в загальній схемі аналізу мережевих логів.

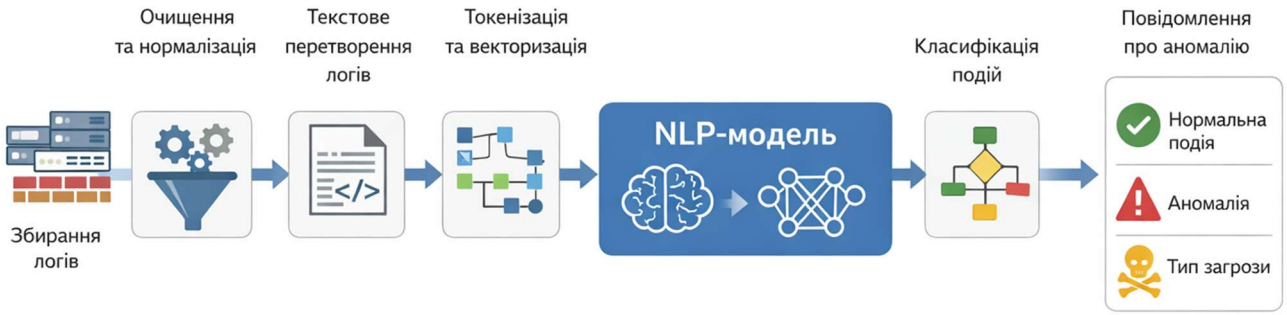

Рисунок 1.1 – Місце NLP-підходу в процесі аналізу мережевих логів

Водночас використання NLP-підходу в цій предметній області не усуває всіх проблем автоматично. Ефективність такого рішення залежить від якості попереднього оброблення, способу формування текстового представлення, збалансованості набору даних, вибору моделі та способу інтерпретації результатів. Крім того, для реальних умов функціонування важливими залишаються питання швидкодії, масштабованості та стійкості до шуму в логах [2, 8, 23].

Отже, предметна область аналізу мережевих логів є складною, багатокомпонентною та такою, що потребує поєднання методів кібербезпеки, аналізу даних і оброблення природної мови.

1.2 Аналіз сучасних підходів до інтелектуального аналізу мережевих логів

Існуючі підходи до виявлення аномалій у мережевих логах доцільно поділити на шість груп: сигнатурні, статистичні, методи класичного машинного навчання, методи глибокого навчання, гібридні моделі та NLP-орієнтовані підходи. Вони відрізняються способом подання даних, механізмом прийняття рішення, рівнем адаптивності до нових атак і вимогами до попереднього оброблення [1, 9, 20, 21].

Сигнатурні методи є найдавнішими та найбільш формалізованими. Їх робота ґрунтується на порівнянні поточних подій із заздалегідь визначеними шаблонами атак. Перевагою такого підходу є висока точність для відомих загроз і відносна простота інтерпретації результату. Основний недолік полягає в слабкій здатності виявляти нові, модифіковані або комбіновані атаки. Крім того, зі збільшенням кількості правил система ускладнюється в супроводі, а проблема хибних спрацьовувань залишається актуальною [9, 18].

Окремим напрямом розвитку сигнатурних IDS стали нечіткі сигнатурні моделі. У них шаблони задаються не жорстко, а з урахуванням нечітких меж ознак, що дає змогу краще працювати з частково зміненими сценаріями атак. Такий підхід підвищує гнучкість сигнатурного аналізу, але не усуває його основного обмеження – залежності від наперед відомих шаблонів [18]. Проблему зменшення кількості хибних спрацьовувань у сигнатурних системах детально розглянуто в [9].

Статистичні методи базуються на формуванні профілю нормальної поведінки мережі та виявленні відхилень від нього. Їх перевага полягає у здатності фіксувати нетипову активність без жорстко заданих сигнатур. Водночас такі методи чутливі до способу побудови базового профілю, залежать від стабільності мережевого середовища та можуть помилково трактувати легітимні зміни навантаження як аномалію. Через це статистичні підходи рідко використовуються ізольовано та здебільшого входять до складу складніших моделей [1, 20].

Значно ширше в сучасних роботах представлені методи машинного

навчання. Їх основна ідея полягає у побудові класифікатора, який розділяє нормальні та аномальні події на основі навчальних даних. До найуживаніших алгоритмів належать дерева рішень, Random Forest, XGBoost, Extreme Learning Machine та інші ансамблеві або швидкі класифікаційні моделі [1, 4, 7, 24]. Вони є придатними для задач IDS завдяки добрій швидкодії, можливості працювати з числовими й категоріальними ознаками та відносно простому впровадженню.

Дерева рішень характеризуються простою логікою побудови та високою інтерпретованістю, однак схильні до перенавчання та залежать від структури вибірки [24]. Random Forest зменшує цей недолік завдяки ансамблю дерев і забезпечує стабільніші результати для задач класифікації мережевих подій [7]. XGBoost дає змогу підвищити точність за рахунок бустингу слабких моделей, проте потребує точнішого налаштування параметрів і якісного формування ознак [4]. Подібні властивості мають і сучасні бустингові ансамблі, зокрема Dual-IDS, які орієнтовані на покращення точності виявлення мережевих вторгнень [16]. Extreme Learning Machine розглядається як швидкий варіант нейромережевої класифікації, придатний для інтелектуальних IDS, хоча його ефективність також визначається якістю підготовлених даних [1].

Обмеження класичного машинного навчання пов’язані насамперед із необхідністю ручного або напівавтоматичного формування ознак. Якщо представлення мережевого логу є слабким, навіть ефективний класифікатор не забезпечує надійного результату. Саме тому розвиток цієї галузі поступово змістився в бік глибокого навчання, яке здатне самостійно виявляти складні нелінійні залежності в даних [20].

Методи глибокого навчання застосовуються тоді, коли мережеві події мають складну внутрішню структуру або часову залежність. Для задач виявлення аномалій у мережевому трафіку використовуються глибокі нейронні мережі, згорткові моделі, рекурентні мережі та комбіновані архітектури. Зокрема, CNN-BiLSTM поєднує переваги локального виділення патернів і моделювання послідовностей, що є корисним для аналізу логів і трафіку як часових послідовностей [26]. У роботах українських дослідників також представлено підходи на основі глибоких нейронних мереж, багатоканальних

детекторів, зменшення розмірності даних та адаптивних систем кіберзахисту [5, 10–15, 28–31].

Перевагою глибокого навчання є здатність працювати зі складними багатовимірними залежностями та виявляти приховані шаблони атак. Недоліками залишаються потреба в значних обсягах даних, вища обчислювальна вартість і нижча пояснюваність результату. Саме проблема інтерпретації рішень стала основою окремого напряму досліджень – explainable intrusion detection systems. У таких роботах увагу зосереджено не лише на точності виявлення, а й на можливості пояснити, чому конкретний запис або сесія були віднесені до аномальних [20]. Для практичного використання це має принципове значення, оскільки система без зрозумілих пояснень ускладнює подальше реагування на інцидент.

Перспективним напрямом є гібридні моделі, у яких поєднуються кілька механізмів: сигнатурний аналіз, статистичне оцінювання, класифікація на основі машинного навчання, пояснюваність та адаптація до контексту. Саме такі підходи дають змогу частково компенсувати недоліки окремих методів. У наукових працях представлено рішення, що поєднують адаптивні системи кіберзахисту, штучні нейронні мережі, імунні механізми, багаторівневі детектори та ієрархічну класифікацію атак [11–15, 31–33]. Для практичних систем такий підхід є виправданим, оскільки реальне мережеве середовище рідко може бути надійно описане одним методом.

Останній етап розвитку цієї галузі пов’язаний із застосуванням NLP-підходів і великих мовних моделей. На відміну від класичних методів, де мережевий лог подається як набір числових або категоріальних полів, NLP-підхід розглядає логовий запис як спеціалізований текст. Це дає змогу враховувати контекст, послідовність подій, характер текстових повідомлень і взаємозв’язки між елементами запису [2, 3, 17, 23].

Поширення ChatGPT та споріднених моделей спричинило появу нових робіт, присвячених використанню LLM у задачах кібербезпеки. Частина таких досліджень розглядає генеративні моделі як інструмент пояснення, підтримки конфігурації, аналізу загроз і допомоги в DevSecOps-процесах [2, 8, 17, 19, 22].

Інша частина безпосередньо орієнтована на виявлення аномалій за логами або подіями. Зокрема, у роботі LogGPT запропоновано підхід до логового аналізу з використанням ChatGPT-подібної моделі, а в HuntGPT поєднано аномалійне виявлення, пояснюваність та LLM-компонент [3, 23]. У суміжних дослідженнях також розглянуто можливість використання механізмів уваги, характерних для сучасних мовних моделей, у задачах виявлення аномальних подій [27].

Перевага NLP-підходу полягає в тому, що він краще працює з напівструктурованими логами, де важливими є не лише значення окремих полів, а і спосіб їх поєднання в одному записі. Крім того, такий підхід дає можливість перейти від ручного формування ознак до текстового подання подій, яке є ближчим до реальної структури логів. Водночас повноцінне використання LLM у практичній IDS пов’язане з обмеженнями швидкодії, вартості, складності інтеграції та нестабільності результату в разі неякісного попереднього оброблення [2, 3, 8, 23]. Отже, доцільнішим для кваліфікаційної роботи є не пряме використання великої генеративної моделі як універсального засобу, а побудова програмної системи, у якій мережеві логи спочатку перетворюються в текстове подання, після чого аналізуються за допомогою придатної NLP-моделі класифікації.

Для узагальнення основних груп методів сформовано таблицю 1.1.

Таблиця 1.1 – Порівняльна характеристика підходів до виявлення аномалій у мережевих логах

Група підходів	Основна перевага	Основний недолік	Доцільність використання
Сигнатурні	Висока точність для відомих атак	Не виявляють нові атаки	Фіксація типових загроз
Статистичні	Виявляють відхилення від норми	Чутливі до зміни профілю мережі	Базовий контроль аномалій
Машинне навчання	Добра точність і швидкодія	Залежність від ознак	Класифікація типових подій

Продовження таблиці 1.1

Глибоке навчання	Виявлення складних залежностей	Висока обчислювальна вартість	Аналіз складних шаблонів
Гібридні	Поєднання сильних сторін різних методів	Складність реалізації	Практичні IDS-системи
NLP-підходи	Робота з текстовими та напівструктурованими логами	Вимогливість до підготовки даних	Аналіз логів як текстових подій

Як видно з таблиці 1.1, жоден із розглянутих підходів не є універсальним. Сигнатурні методи залишаються корисними для відомих сценаріїв атак, статистичні – для базового контролю відхилень, моделі машинного навчання – для ефективної класифікації, глибоке навчання – для складних залежностей, а гібридні системи – для практичного поєднання цих можливостей. Проте саме NLP-підхід є найбільш перспективним для задачі, у якій основним джерелом даних виступають мережеві логи як напівструктуровані текстові записи.

Отже, аналіз наукових джерел показав, що сучасний розвиток систем виявлення аномалій у мережевих логах відбувається в напрямі переходу від жорстких правил і вручну сформованих ознак до інтелектуальних моделей, здатних враховувати контекст подій і працювати з текстовим представленням логів.

1.3 Постановка задачі дослідження

Проведений аналіз предметної області та існуючих підходів показав, що мережеві логи є цінним, але складним джерелом даних для виявлення аномалій. Основна проблема полягає в тому, що логові записи надходять із різних джерел, мають напівструктурований формат, містять службові поля, текстові фрагменти, числові параметри та часові мітки. За таких умов традиційні сигнатурні та

статистичні методи не завжди забезпечують належну гнучкість, а класичні ML-підходи потребують якісного формування ознак. Це зумовлює доцільність побудови програмної системи, у якій мережеві логи розглядаються як спеціалізовані текстові об’єкти, придатні для подальшого NLP-аналізу.

У межах кваліфікаційної роботи досліджується задача автоматизованого виявлення аномалій у мережевих логах на основі їх попереднього оброблення, текстового представлення та подальшої класифікації за допомогою NLP-підходу. Практичний зміст цієї задачі полягає у створенні програмної системи, здатної приймати набір логових записів, перетворювати їх у стандартизоване подання, аналізувати ознаки подій і формувати результат у вигляді класифікації запису як нормального або аномального.

Таким чином, основну задачу дослідження сформовано так: розробити програмну систему аналізу мережевих логів, яка забезпечує автоматизоване виявлення аномальних подій на основі NLP-підходу та дає змогу оцінювати якість прийнятих рішень за стандартними метриками класифікації.

Для розв’язання поставленої задачі необхідно визначити вхідні дані, вихідні результати, функціональні вимоги до системи, загальну логіку її роботи та основні обмеження.

Вхідними даними системи є мережеві логи, подані у вигляді набору записів про мережеві події. Такі записи можуть містити часову мітку, IP-адресу джерела, IP-адресу призначення, номер порту, протокол, тип події, статус відповіді, тривалість сесії, обсяг переданих даних, службове повідомлення та інші атрибути. Частина полів може бути відсутньою, надлишковою або поданою в неоднорідному форматі. Саме тому система повинна підтримувати етап попереднього очищення та узгодження структури логів.

Вихідним результатом системи є рішення щодо кожного аналізованого запису або групи записів. У найпростішому випадку результатом є віднесення події до одного з двох класів: нормальна подія або аномалія. У розширеному варіанті система може додатково формувати мітку типу загрози, рівень ризику або коротке текстове пояснення результату. Для кваліфікаційної роботи достатнім є базовий сценарій бінарної класифікації з можливістю подальшого

розширення.

Для формалізації задачі доцільно подати її у вигляді послідовності функціональних етапів. На першому етапі виконується завантаження логових записів із джерела даних. На другому – очищення, нормалізація та приведення полів до єдиного формату. На третьому – формування текстового представлення кожного запису. На четвертому – перетворення тексту у форму, придатну для оброблення NLP-моделлю. На п’ятому – класифікація події. На шостому – формування результату та оцінювання якості роботи системи.

З огляду на сформульовану задачу, програмна система повинна відповідати низці функціональних вимог. По-перше, має бути забезпечено завантаження та оброблення логів у табличному або текстовому форматі. По-друге, система повинна виконувати попередню обробку даних без ручного втручання для базового сценарію роботи. По-третє, має бути реалізовано механізм перетворення логового запису в текстове представлення. По-четверте, система повинна містити модуль класифікації, здатний визначати належність запису до нормального або аномального класу. По-п’яте, має бути забезпечено формування результатів у зручному для аналізу вигляді.

Крім функціональних вимог, доцільно врахувати і нефункціональні вимоги. До них належать зрозуміла структура системи, модульність реалізації, можливість роботи з типовими наборами логів, прийнятна швидкодія для експериментального аналізу та можливість масштабування окремих компонентів у подальшому. Для бакалаврської роботи не ставиться задача промислового розгортання, тому акцент робиться не на високонавантаженій експлуатації, а на коректності архітектурного рішення, логіці оброблення даних і якості класифікації.

Задача дослідження також передбачає обґрунтування вибору саме NLP-підходу. Такий вибір пояснюється тим, що мережеві логи не є повністю структурованими числовими даними. Вони часто містять текстові фрагменти, шаблони повідомлень, змішані формати записів і контекстні зв’язки між атрибутами. Через це представлення логів як текстових послідовностей є обґрунтованим і дає змогу застосувати методи токенізації, векторизації та

класифікації, характерні для задач оброблення природної мови [2, 3, 23]. У цьому випадку логовий запис аналізується не лише як набір окремих полів, а як цілісний опис події.

У межах роботи доцільно орієнтуватися на таку концепцію системи:

– джерело логів формує вхідний набір записів;

– модуль попереднього оброблення очищає та нормалізує дані;

– модуль текстового подання перетворює кожен запис у стандартизований текстовий шаблон;

– NLP-модуль виконує класифікацію;

– модуль оцінювання формує кількісні показники якості.

Окремо необхідно визначити межі дослідження. У цій роботі не розглядається побудова повноцінної промислової IDS із потоковим аналізом у реальному часі, розподіленим розгортанням або повним покриттям усіх типів мережевих атак. Також не ставиться задача універсального аналізу будь-яких журналів подій без попередньої адаптації структури. Дослідження обмежується розробленням прототипу програмної системи для аналізу мережевих логів із використанням NLP-підходу в межах експериментального набору даних. Таке обмеження є виправданим, оскільки воно дає змогу зосередитися на ключовому науково-практичному аспекті – перетворенні логів у текстове подання та автоматизованому виявленні аномалій.

2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ NLP-ПІДХОДУ

2.1 Розроблення архітектури програмної системи

Архітектуру програмної системи сформовано з урахуванням послідовності оброблення мережевих логів: від надходження вхідних записів до отримання результату класифікації. Основна вимога до такої архітектури полягає у відокремленні етапів підготовки даних, текстового подання, NLP-аналізу та формування результатів. Це спрощує реалізацію, тестування та подальше вдосконалення окремих модулів системи [34].

У межах роботи програмну систему доцільно побудувати як модульну структуру, що складається з п’яти основних компонентів [34]:

– модуля завантаження логів;
– модуля попереднього оброблення;
– модуля текстового подання логів;
– модуля NLP-класифікації;
– модуля формування та відображення результатів.

Модуль завантаження логів забезпечує приймання вхідних даних із файлу або підготовленого набору записів. На цьому етапі логові дані зчитуються, перетворюються у внутрішню структуру зберігання та передаються до наступного модуля. Основне завдання цього компонента полягає в тому, щоб уніфікувати початковий формат записів і підготувати їх до подальшого оброблення.

Модуль попереднього оброблення виконує очищення та нормалізацію даних. На цьому етапі усуваються порожні або некоректні значення, узгоджуються назви полів, видаляються дублікати, за потреби виконуються перетворення часових міток, адрес, портів та інших атрибутів. Саме цей модуль формує впорядкований набір записів, придатний для подальшого аналізу.

Модуль текстового подання є ключовим елементом архітектури. Його призначення полягає у перетворенні структурованого або напівструктурованого логового запису в текстовий шаблон. У результаті кожна мережева подія

подається як цілісний текстовий опис, що містить основні атрибути запису в уніфікованому форматі. Такий підхід дає змогу застосувати методи NLP-аналізу до логів як до спеціалізованих текстових об’єктів.

Модуль NLP-класифікації виконує основне аналітичне завдання системи. На його вхід надходять текстові представлення логів, які після токенізації або векторизації подаються в модель класифікації. Результатом роботи модуля є визначення класу події: нормальна або аномальна. У межах архітектури цей компонент є центральним, оскільки саме він реалізує інтелектуальне виявлення відхилень у логових даних.

Модуль формування та відображення результатів забезпечує подання отриманих висновків у зручному вигляді. У ньому зберігаються результати класифікації, обчислюються підсумкові показники якості та формуються повідомлення для користувача. Для експериментального сценарію достатньо табличного відображення записів, передбаченого класу та узагальнених метрик.

Загальну архітектуру системи представлено на рисунку 2.1.

Рисунок 2.1 – Архітектура програмної системи аналізу мережевих логів [34]

Взаємодія модулів є послідовною. Кожен наступний компонент отримує результат роботи попереднього. Така схема є простою, зрозумілою та придатною для прототипної реалізації. Крім того, вона дає змогу окремо вдосконалювати кожен модуль без зміни всієї системи. Наприклад, можна змінити спосіб текстового представлення логів або модель класифікації без перебудови інших частин архітектури.

Отже, архітектуру програмної системи сформовано як модульну послідовну структуру, орієнтовану на оброблення мережевих логів, їх перетворення у текстове подання та подальшу NLP-класифікацію. Така побудова

відповідає поставленій задачі дослідження та створює основу для детальнішого проєктування окремих модулів системи.

2.2 Проєктування підсистеми попереднього оброблення мережевих логів

Модуль попереднього оброблення та текстового подання логів спроєктовано для перетворення сирих мережевих записів у уніфіковану форму, придатну для подальшого NLP-аналізу. Його роль є ключовою, оскільки якість класифікації значною мірою залежить не лише від моделі, а й від того, наскільки коректно підготовлено вхідні дані.

На вхід модуля надходять мережеві логи у вигляді таблиці або набору рядків із полями події. Такі записи можуть містити часову мітку, адресу джерела, адресу призначення, порт, протокол, тривалість сесії, обсяг переданих даних, тип події та інші атрибути. У реальних наборах даних частина полів є пропущеною, надлишковою або поданою в неоднорідному форматі. Саме тому першим етапом роботи модуля визначено очищення даних.

На етапі очищення передбачено видалення дублікатів, оброблення пропущених значень, усунення службових або порожніх записів і приведення назв полів до єдиного формату. Якщо окремий атрибут не має аналітичної цінності або містить лише технічний шум, він не передається далі. Такий підхід дає змогу зменшити надлишковість даних і підвищити стійкість системи до випадкових помилок у логах.

Після очищення виконується нормалізація полів. На цьому етапі часові мітки переводяться до єдиного представлення, текстові значення очищаються від зайвих символів, назви протоколів та статусів уніфікуються, а числові поля перетворюються у стандартизовану форму. Для IP-адрес, портів і службових кодів не виконується складна семантична інтерпретація, однак забезпечується їх коректне збереження в структурі запису. У результаті формується узгоджений набір логів, де однакові типи подій мають однаковий формат подання.

Після цього виконується відбір атрибутів для подальшого аналізу. До структури текстового подання включаються лише ті поля, які реально впливають

на опис мережевої події. Для задачі виявлення аномалій доцільно використовувати часову мітку, IP-адресу джерела, IP-адресу призначення, порт, протокол, статус події, тривалість і обсяг переданих даних. Інші атрибути можуть додаватися лише за наявності практичної потреби. Такий підхід зменшує розмір вхідного тексту і водночас зберігає його інформативність.

Основною особливістю модуля є перетворення логового запису у текстове подання. Для цього кожна подія подається не як набір розрізнених полів, а як короткий стандартизований текстовий шаблон. Наприклад, запис із джерелом, призначенням, протоколом і портом може бути поданий як опис мережевої взаємодії між вузлами з конкретними параметрами. Такий формат є зручним для токенізації та подальшої класифікації NLP-моделлю.

Під час проєктування модуля важливо було забезпечити однакову структуру всіх текстових записів. Це потрібно для того, щоб модель працювала не з випадково сформованими рядками, а з послідовностями єдиного типу. Тому текстове представлення формується за шаблоном, у якому порядок основних атрибутів є фіксованим. Завдяки цьому модель отримує стабільний вхід, а варіативність зосереджується саме в значеннях полів, а не в способі побудови тексту.

Загальну логіку роботи модуля подано на рисунку 2.2.

Рисунок 2.2 – Структура модуля попереднього оброблення та текстового подання логів

Результатом роботи модуля є набір текстових описів мережевих подій, підготовлених для токенізації та подальшої класифікації. Саме цей результат передається до наступного модуля системи, який виконує NLP-аналіз і визначає належність події до нормального або аномального класу.

Отже, модуль попереднього оброблення та текстового подання логів спроєктовано як послідовність операцій очищення, нормалізації, відбору атрибутів і формування уніфікованого текстового шаблону. Така структура дає змогу підготувати якісний вхід для моделі класифікації та є необхідною умовою ефективного виявлення аномалій у мережевих логах.

2.3 Проєктування підсистеми NLP-аналізу та класифікації аномалій

Модуль NLP-аналізу та виявлення аномалій спроєктовано як центральний компонент програмної системи, що виконує класифікацію підготовлених логових записів. Його призначення полягає у прийманні текстового подання мережевих подій, перетворенні цього подання у форму, придатну для оброблення моделлю, та визначенні належності запису до нормального або аномального класу.

На вхід модуля надходять текстові описи логів, сформовані на попередньому етапі. Такі описи мають єдину структуру, тому модуль працює не з довільним текстом, а зі стандартизованими послідовностями, що відображають основні характеристики мережевої події. Це спрощує подальшу токенізацію та зменшує вплив випадкових відмінностей у форматі запису.

Першим етапом роботи модуля є перетворення тексту у числове подання. Для цього використовується токенізація, у межах якої текстовий запис розбивається на окремі елементи, після чого формується послідовність індексів або векторних ознак. Саме цей етап забезпечує перехід від текстового опису до машинно-оброблюваного формату. Якість такого перетворення безпосередньо впливає на точність подальшої класифікації.

Після токенізації дані подаються до моделі NLP-класифікації. Для кваліфікаційної роботи доцільно використовувати компактну модель текстової класифікації, орієнтовану на бінарний результат: нормальна подія або аномалія. Такий підхід є достатнім для демонстрації працездатності системи та експериментального оцінювання якості. У межах проєктування модуль не прив’язується жорстко до однієї конкретної архітектури, однак передбачає

використання моделі, здатної працювати з короткими спеціалізованими текстами.

Логіка роботи модуля базується на тому, що текстове представлення мережевого логу містить ключові ознаки події у послідовному вигляді. Модель виявляє характерні комбінації токенів, зв’язки між атрибутами та шаблони, властиві нормальній або аномальній поведінці. На відміну від класичних підходів, де поля часто аналізуються ізольовано, тут запис оцінюється як цілісна текстова конструкція.

Результатом роботи модуля є прогноз класу події. Додатково може формуватися числова оцінка впевненості моделі. Якщо значення перевищує заданий поріг, подія відноситься до аномальних. Якщо ні – до нормальних. Така схема є простою в реалізації й водночас придатною для подальшого розширення, наприклад для багатокласової класифікації типів загроз.

Структуру модуля доцільно подано на рисунку 2.3.

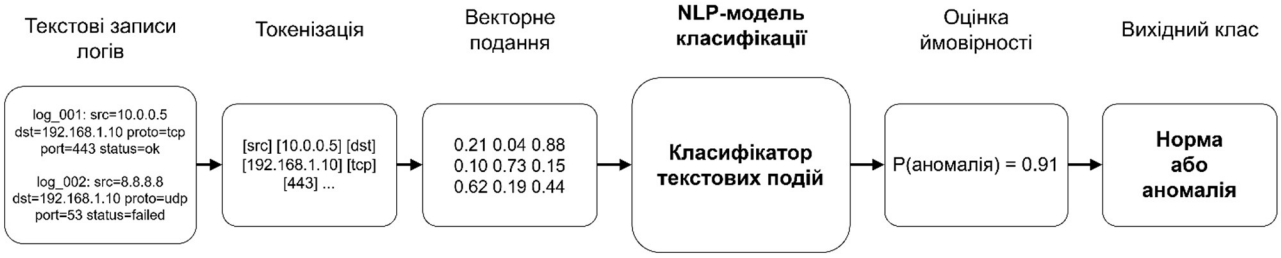

Рисунок 2.3 – Структура модуля NLP-аналізу та виявлення аномалій

У модулі доцільно передбачити два режими роботи. Перший режим – навчання моделі на розмічених даних. На цьому етапі система отримує текстові записи та правильні мітки класів, після чого формує параметри моделі. Другий режим – виявлення аномалій, тобто класифікація нових логів без зміни параметрів моделі. Таке розділення є стандартним і дає змогу чітко відокремити етап побудови моделі від етапу практичного використання.

Окрему увагу під час проєктування приділено вибору вихідного формату. Для експериментальної системи доцільно зберігати не лише передбачений клас, а й правильну мітку, якщо вона відома, а також значення прогнозу моделі. Це необхідно для подальшого обчислення метрик accuracy, precision, recall і F1-

score.

Перевагою спроєктованого модуля є його незалежність від конкретного джерела логів. Якщо попередній модуль забезпечує уніфіковане текстове подання, NLP-компонент може працювати з різними типами мережевих записів без зміни загальної логіки. Це підвищує гнучкість архітектури та спрощує заміну моделі в майбутньому.

Отже, модуль NLP-аналізу та виявлення аномалій спроєктовано як послідовність операцій токенізації, векторизації, класифікації та формування результату. Його основна функція полягає у визначенні належності текстового представлення мережевого логу до нормального або аномального класу. Така структура відповідає меті роботи та створює основу для подальшого опису алгоритмів функціонування системи.

2.4 Алгоритмічне забезпечення функціонування системи

Функціонування програмної системи реалізовано як послідовність взаємопов’язаних алгоритмів, що охоплюють завантаження логів, їх підготовку, текстове подання, NLP-класифікацію та формування результату. Такий підхід дає змогу чітко розмежувати етапи оброблення даних і забезпечити узгоджену роботу всіх модулів системи.

У межах проєктування доцільно виділити три основні алгоритми:

– алгоритм завантаження та попереднього оброблення логів;
– алгоритм формування текстового подання;
– алгоритм NLP-класифікації та формування результату.

Алгоритм завантаження та попереднього оброблення логів забезпечує підготовку вхідних даних до подальшого аналізу. Його призначення полягає у зчитуванні мережевих логів, перевірці їх структури, очищенні та нормалізації основних полів.

Послідовність кроків алгоритму:

1) завантажити набір мережевих логів із джерела даних;
2) перевірити наявність обов’язкових полів у записах;

3) видалити дублікати та порожні записи;
4) обробити пропущені значення;
5) нормалізувати назви полів і формат атрибутів;
6) сформувати очищений набір логів;
7) передати підготовлені записи до модуля текстового подання.

На рисунку 2.4 подано схему цього алгоритму.

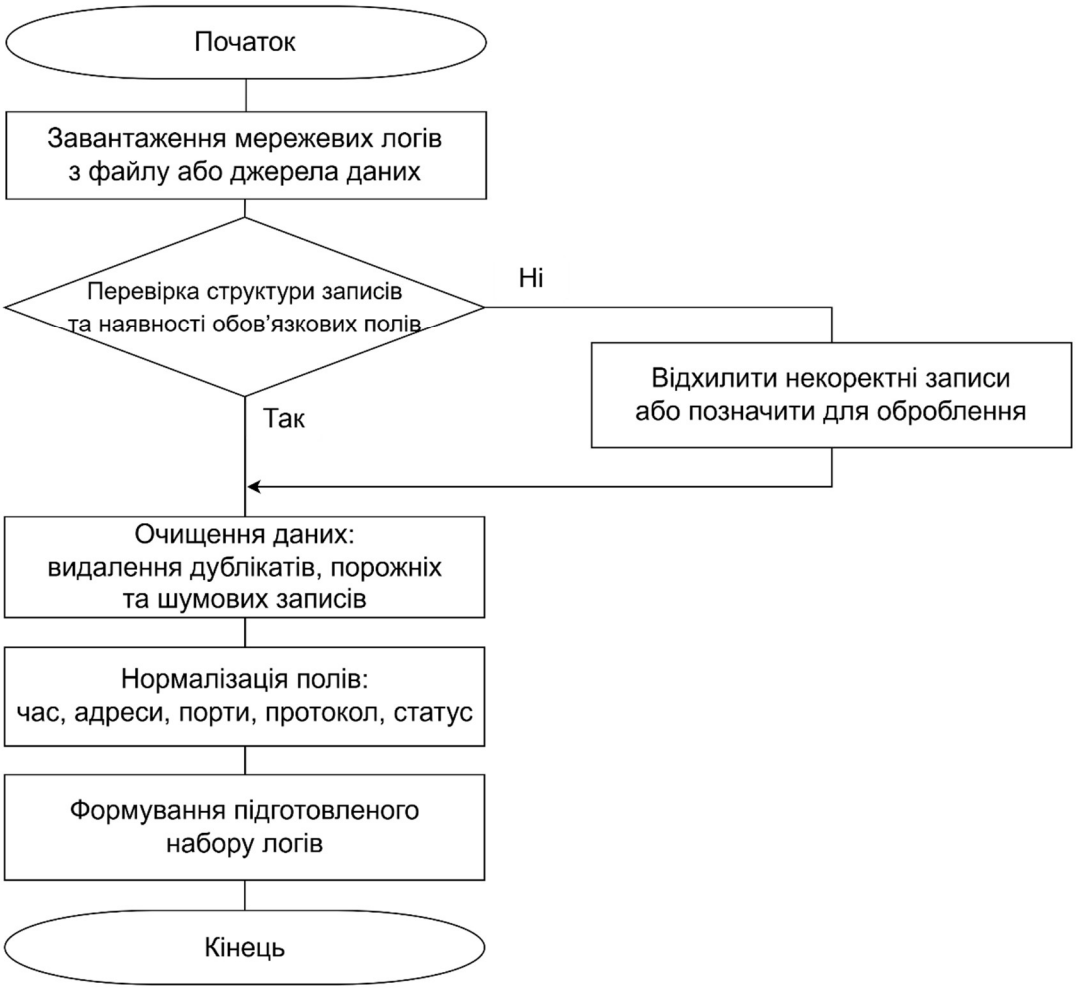

Рисунок 2.4 – Схема алгоритму завантаження та попереднього оброблення мережевих логів

Алгоритм формування текстового подання логів виконує перетворення очищених логових записів у стандартизований текстовий формат. Це необхідно для подальшої токенізації та класифікації засобами NLP.

Послідовність кроків алгоритму:

1) отримати очищений логовий запис;
2) вибрати інформативні атрибути події;

3) розмістити атрибути у фіксованому порядку;
4) сформувати текстовий шаблон запису;
5) перевірити коректність сформованого тексту;
6) зберегти текстове представлення;
7) передати текстовий запис до модуля класифікації.

На рисунку 2.5 подано схему цього алгоритму.

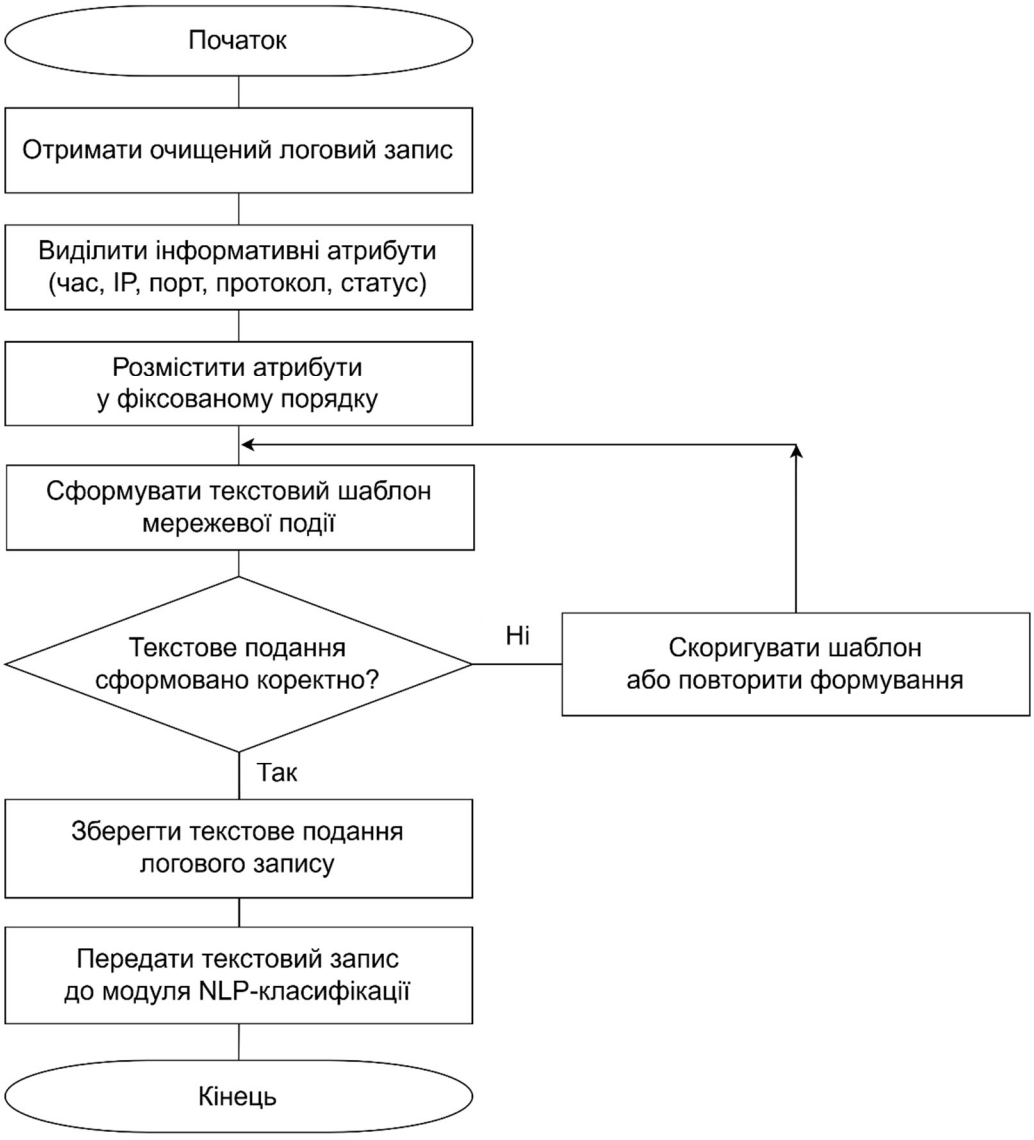

Рисунок 2.5 – Схема алгоритму формування текстового подання мережевого логу

Алгоритм NLP-класифікації та формування результату виконує основне аналітичне завдання системи – віднесення події до нормального або аномального класу. Саме на цьому етапі реалізується використання NLP-підходу для виявлення аномалій.

Послідовність кроків алгоритму:

1) отримати текстове представлення логового запису;
2) виконати токенізацію тексту;
3) сформувати числове подання вхідної послідовності;
4) подати дані на вхід NLP-моделі;
5) отримати прогноз класу;
6) визначити належність події до класу «норма» або «аномалія»;
7) зберегти результат класифікації;
8) за наявності еталонних міток обчислити метрики якості.

На рисунку 2.6 подано схему цього алгоритму.

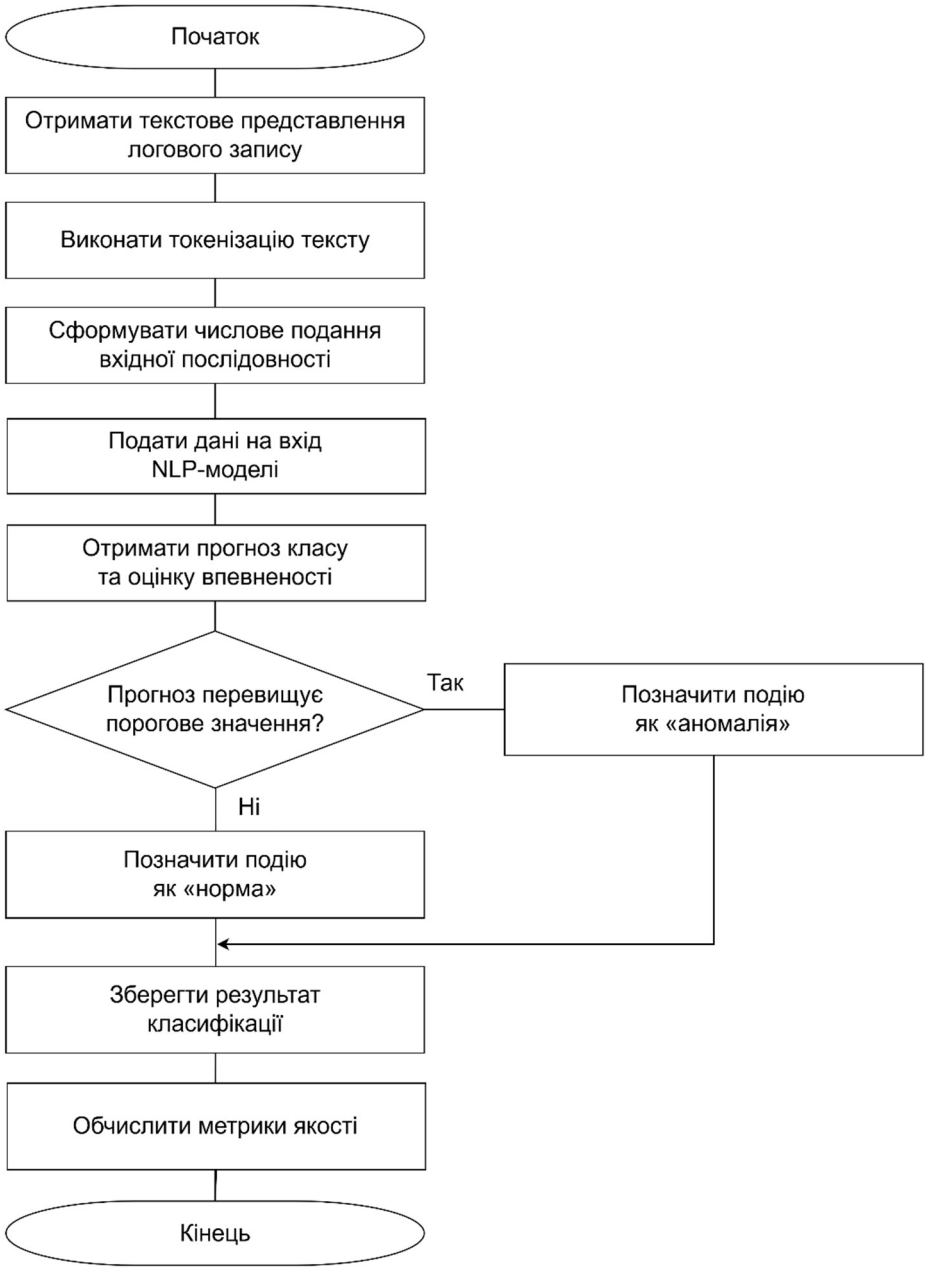

Рисунок 2.6 – Схема алгоритму NLP-класифікації мережевого логу

У межах системи ці алгоритми працюють послідовно. Результат кожного попереднього алгоритму є вхідними даними для наступного. Така логіка забезпечує цілісність оброблення та узгодженість між модулями. Водночас кожен алгоритм може бути вдосконалений окремо без порушення загальної структури системи.

Отже, алгоритмічне забезпечення програмної системи сформовано як послідовність трьох основних процесів: підготовки логів, формування текстового подання та NLP-класифікації. Це створює завершену логіку функціонування системи й забезпечує основу для подальшої програмної реалізації та експериментального дослідження.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ СИСТЕМИ

3.1 Реалізація основних модулів програмної системи

Програмну систему аналізу мережевих логів реалізовано як модульний програмний прототип, призначений для послідовного оброблення мережевих записів, їх перетворення у текстове подання та подальшої класифікації засобами NLP. Такий підхід дав змогу відокремити етапи роботи з даними, спростити тестування окремих компонентів і забезпечити зрозумілу логіку функціонування системи.

У межах реалізації система працює з мережевими логами, поданими у табличному форматі. Для експериментального дослідження використано CSV-файли набору CICIDS2018 [35], які містять опис мережевих потоків і мітки класів. Кожен запис розглядається як окрема мережева подія, що надалі проходить послідовні етапи підготовки та аналізу.

Загальна схема реалізації системи включає такі етапи:

– завантаження вхідних логів;
– очищення та нормалізація даних;
– формування текстового подання записів;
– токенізація і перетворення тексту у числовий формат;
– класифікація подій;
– збереження результатів та обчислення метрик.

Перший компонент системи реалізує завантаження даних. Його функція полягає у зчитуванні CSV-файлу, перевірці структури таблиці та формуванні внутрішнього подання набору логів. На цьому етапі виконується контроль наявності ключових полів, зокрема часової мітки, IP-адрес, портів, протоколу, числових характеристик потоку та мітки класу. Якщо деякі обов’язкові поля відсутні, запис або файл не допускається до подальшого аналізу.

На рисунку 3.1 наведено приклад завантаження вхідних мережевих логів у програмній системі. Скріншот демонструє кількість завантажених записів, кількість атрибутів, формат файлу та фрагмент таблиці з основними полями

мережевих подій.

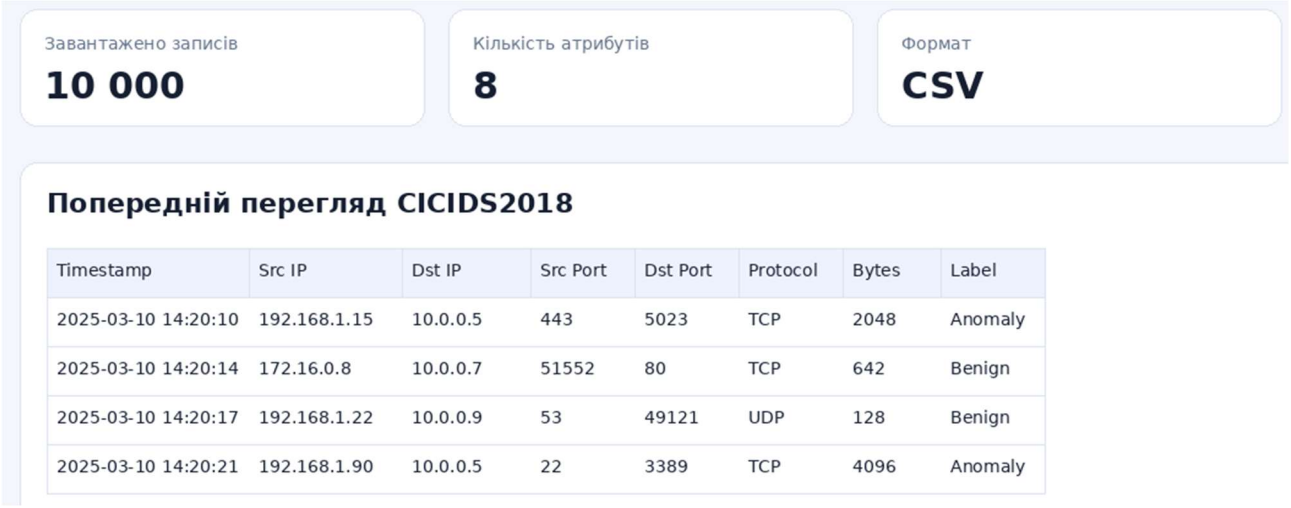

Рисунок 3.1 – Завантаження вхідних мережевих логів

Другий компонент системи реалізує попереднє оброблення мережевих логів. На цьому етапі усуваються дублікати, видаляються порожні або некоректні записи, виконується оброблення пропущених значень і нормалізація типів даних. Числові поля приводяться до числового формату, текстові значення очищаються від зайвих символів, а назви атрибутів уніфікуються. У результаті формується підготовлений набір даних, у якому всі записи мають узгоджену структуру.

Наступний етап пов’язаний із формуванням текстового подання логів. Саме він є ключовим для всієї роботи, оскільки забезпечує перехід від структурованих мережевих даних до NLP-орієнтованого формату. Для цього реалізовано шаблонне формування тексту: із кожного запису вибираються основні атрибути події, після чого вони підставляються у фіксовану текстову конструкцію. Такий підхід забезпечує однаковий порядок подання ознак, що є важливим для подальшої токенізації та класифікації.

Після формування текстових описів реалізується підготовка даних до NLP-моделі. На цьому етапі тексти перетворюються у послідовності токенів, а потім – у числові вектори або індексні послідовності. Для всіх записів задається однакова довжина вхідної послідовності. Коротші тексти доповнюються службовими значеннями, а довші – обрізаються до заданого розміру. Це дає змогу уніфікувати вхід моделі та забезпечити стабільність її роботи.

Модуль класифікації реалізовано окремо від підсистеми підготовки даних. У межах прототипу він працює у двох режимах: навчання і прогнозування. У режимі навчання система отримує текстові записи разом із правильними мітками класів і формує модель бінарної класифікації. У режимі прогнозування використовується вже навчена модель, яка визначає належність нового запису до класу «норма» або «аномалія». Така схема дала змогу відокремити побудову моделі від її практичного використання.

Після завершення класифікації система формує результати аналізу. Для кожного запису зберігається передбачений клас, а для тестової вибірки додатково обчислюються основні метрики якості: accuracy, precision, recall і F1-score. На основі цих даних надалі будуються матриця помилок і порівняльні графіки, подані у підрозділах 3.3–3.4.

Для реалізації системи доцільно використовувати мову програмування Python, оскільки вона має зручний інструментарій для роботи з табличними даними, текстами і моделями машинного навчання. У межах прототипу функціонально використовуються:

– pandas для завантаження й оброблення таблиць;
– numpy для числових операцій;
– scikit-learn для поділу вибірок і розрахунку метрик;
– tensorflow або keras для побудови моделі текстової класифікації;
– matplotlib для формування графіків результатів.

Для узгодженої організації програмної частини сформовано узагальнену структуру файлів проєкту (таблиця 3.1, рисунок 3.2). Вона відображає логіку реалізації системи та відокремлення програмних компонентів, наборів даних, моделі та результатів.

Подана структура є зручною для експериментальної реалізації, оскільки дає змогу чітко відокремити вхідні дані, проміжні результати, програмні модулі, навчену модель і підсумкові матеріали дослідження.

Для узагальнення складу реалізації сформовано таблицю 3.2, у якій зафіксовано основні програмні компоненти системи.

Таблиця 3.1 – Узагальнена структура файлів програмної системи

Елемент структури	Призначення
data/raw/	зберігання вхідних мережевих логів
data/processed/	зберігання очищених і текстово перетворених даних
src/load_data.py	завантаження логів
src/preprocess.py	попереднє оброблення та нормалізація
src/text_transform.py	формування текстового подання
src/train_model.py	побудова і навчання моделі
src/evaluate.py	оцінювання якості класифікації
src/predict.py	прогнозування для нових записів
models/	збереження навченої моделі
results/	збереження метрик, графіків і матриці помилок
main.py	запуск повного сценарію роботи системи

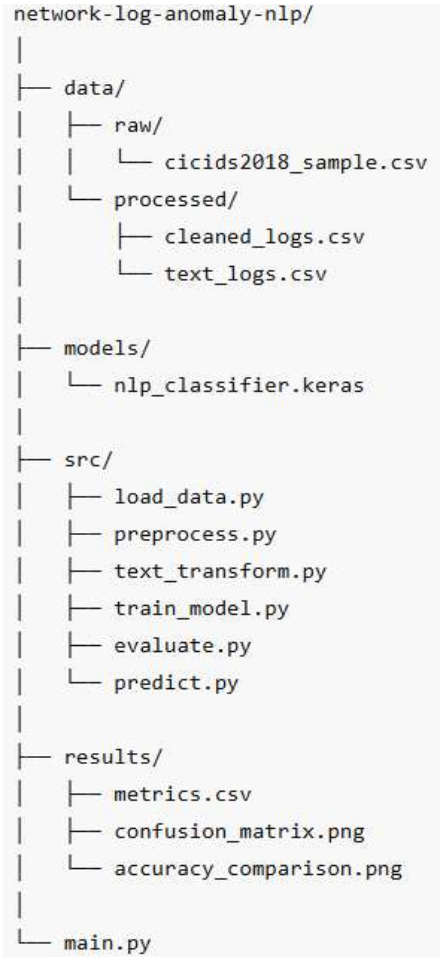

Рисунок 3.2 – Узагальнена структура файлів проєкту

Таблиця 3.2 – Основні компоненти програмної реалізації системи

Компонент	Реалізована функція	Результат
Модуль завантаження	Зчитування CSV-файлів і перевірка структури	Початковий набір логів
Модуль попереднього оброблення	Очищення, нормалізація, відбір полів	Підготовлені записи
Модуль текстового подання	Формування стандартизованого тексту	Текстові описи подій
Модуль підготовки до NLP	Токенізація, векторизація, вирівнювання довжини	Вхідні матриці ознак
Модуль класифікації	Навчання моделі та прогнозування	Передбачений клас
Модуль оцінювання	Обчислення метрик і формування звітних даних	Таблиці та графіки результатів

Під час реалізації особливу увагу приділено модульності. Кожен компонент системи виконує окрему функцію та може бути змінений незалежно від інших. Наприклад, можливо замінити шаблон текстового подання або модель класифікації без зміни логіки завантаження й очищення даних. Така побудова є доцільною для дослідницького прототипу, оскільки спрощує експерименти з різними варіантами оброблення логів і різними моделями.

У додатку А подано основні фрагменти програмної реалізації системи аналізу мережевих логів з використанням NLP-підходу для виявлення аномалій. Наведені лістинги відображають ключові етапи роботи системи: завантаження даних, попереднє оброблення, формування текстового подання, підготовку даних для моделі, навчання класифікатора та оцінювання результатів.

Отже, програмну систему реалізовано як послідовний модульний прототип, у якому поєднано завантаження мережевих логів, їх очищення, перетворення у текстове подання, NLP-класифікацію та оцінювання результатів. Включення узагальненої структури файлів дало змогу формалізувати програмну

організацію системи й показати, що реалізація є не набором окремих фрагментів коду, а цілісним програмним рішенням.

3.2 Формування набору даних і організація експерименту

Експериментальне дослідження виконано на реальному наборі мережевого трафіку CICIDS2018 [35], який містить нормальні та аномальні мережеві потоки. Для цієї роботи використано сценарій бінарної класифікації, у межах якого всі записи поділялися на два класи:

– нормальні події;
– аномальні події.

Такий підхід є доцільним для бакалаврської роботи, оскільки він дає змогу зосередитися не на деталізації всіх типів атак, а на перевірці працездатності самої системи виявлення відхилень.

На етапі підготовки даних виконано:

– завантаження мережевих записів;
– очищення та нормалізацію полів;
– відбір інформативних атрибутів;
– перетворення записів у текстове подання;
– поділ даних на навчальну і тестову частини.

У текстове представлення включалися ті атрибути, які найбільш повно характеризують мережеву подію: час, джерело, призначення, порти, протокол і службові параметри передавання. Така форма подання дала змогу використати модель класифікації тексту для задачі виявлення аномалій.

У таблиці 3.3 подано набір основних полів, використаних під час формування текстового опису логів.

Для наочності в роботі використано узагальнений приклад перетворення мережевого запису у текстовий формат (таблиця 3.4).

Таблиця 3.3 – Основні атрибути мережевого запису, використані в експерименті

Атрибут	Зміст
Timestamp	Час виникнення події
Src IP	IP-адреса джерела
Dst IP	IP-адреса призначення
Src Port	Порт джерела
Dst Port	Порт призначення
Protocol	Мережевий протокол
Bytes	Обсяг переданих даних
Label	Клас події
Атрибут	Зміст

Таблиця 3.4 – Приклад перетворення мережевого логу в текстове подання

Початковий запис	Текстове подання
Timestamp: 2025-03-10 14:20:10; Src IP: 192.168.1.15; Dst IP: 10.0.0.5; Src Port: 443; Dst Port: 5023; Protocol: TCP; Bytes: 2048	At 2:20 PM on March 10, 2025, a TCP packet of 2048 bytes was sent from IP 192.168.1.15 (port 443) to IP 10.0.0.5 (port 5023).

На рисунку 3.3 показано фрагмент роботи програмної системи на етапі попереднього оброблення та формування текстового подання. Скріншот демонструє виконання операцій очищення, оброблення пропущених значень, нормалізації протоколів, перетворення міток класів і формування текстового опису мережевої події.

Після виконання цих операцій формується підготовлений набір текстових записів, який передається до модуля токенізації та подальшої NLP-класифікації.

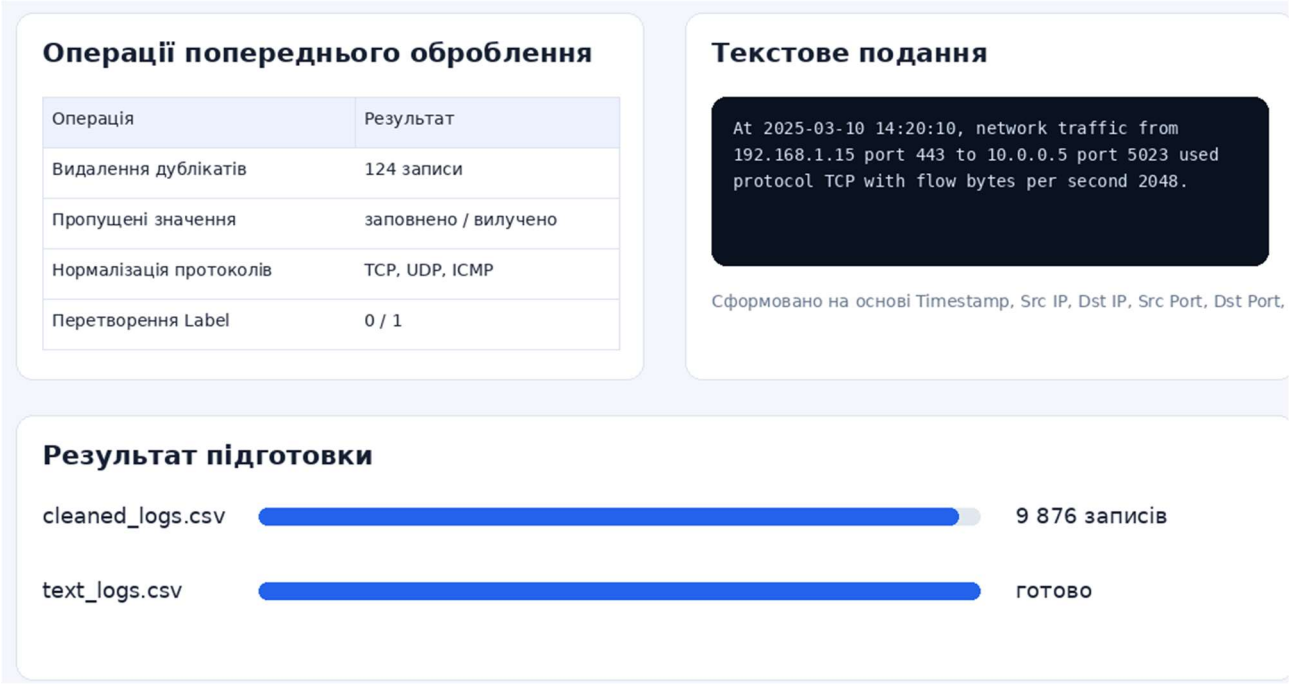

Рисунок 3.3 – Попереднє оброблення та текстове подання мережевого логу

Для оцінювання якості системи використано стандартні метрики класифікації:

– accuracy;
– precision;
– recall;
– F1-score;
– false positive rate;
– false negative rate.

Ці показники є достатніми для аналізу ефективності IDS-рішення та широко застосовуються в дослідженнях із виявлення вторгнень [20, 21, 23].

Отже, організацію експерименту побудовано на реальному наборі мережевих даних і послідовній підготовці логів до задачі текстової класифікації.

3.3 Аналіз результатів виявлення аномалій

Перед проведенням тестування виконано навчання NLP-моделі бінарної класифікації. Для цього текстові подання мережевих логів було перетворено у числові послідовності, після чого модель навчалася визначати клас події як «норма» або «аномалія».

На рисунку 3.4 наведено скріншот режиму навчання моделі, який відображає основні параметри навчання, зокрема розмір словника токенів, максимальну довжину послідовності, розмір пакета, кількість епох та журнал зміни accuracy на навчальній і валідаційній вибірках.

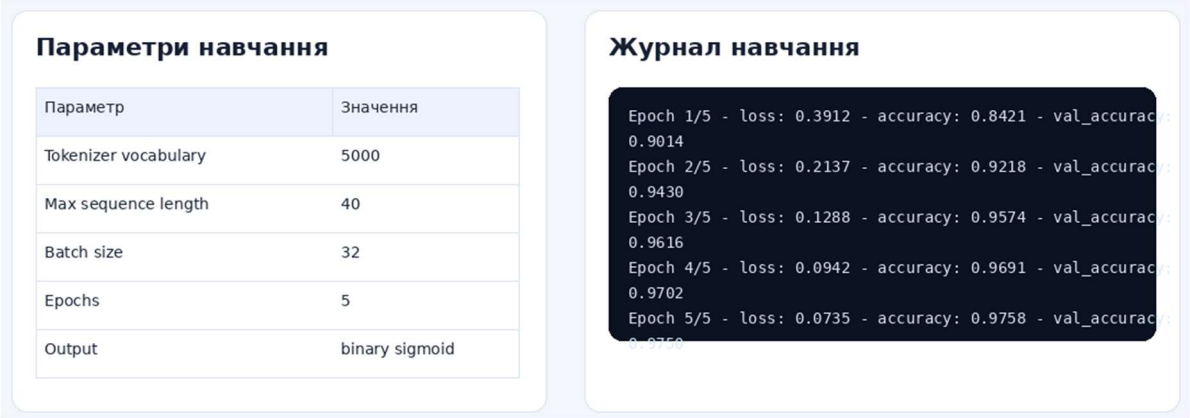

Рисунок 3.4 – Навчання NLP-моделі класифікації

За результатами тестування система продемонструвала високі показники якості бінарної класифікації, наведені в таблиці 3.5.

Таблиця 3.5 – Основні результати експериментального оцінювання

Метрика	Значення, %
Accuracy	97,5
Precision	96,4
Recall	97,8
F1-score	97,1
FPR	2,7
FNR	2,3

Отримані результати свідчать, що система забезпечує високу частку правильних рішень і добре виявляє аномальні події. Найбільш показовим є значення recall 97,8 %, оскільки воно характеризує здатність системи знаходити аномалії та не пропускати небезпечні записи. Водночас precision 96,4 % підтверджує, що кількість хибних спрацьовувань залишається невисокою.

Для наочного подання структури помилок сформовано матрицю помилок, наведену на рисунку 3.5.

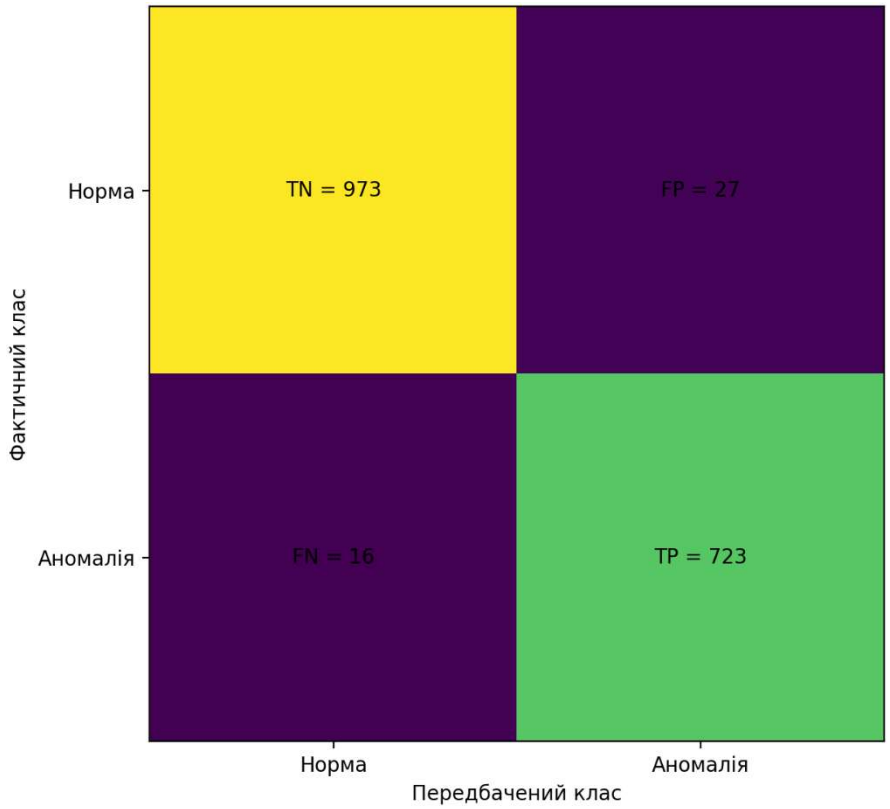

Рисунок 3.5 – Матриця помилок результатів класифікації

Подана матриця відповідає загальним отриманим показникам і дає змогу оцінити співвідношення правильно та помилково класифікованих записів. Із рисунка 3.5 видно, що основна частина подій класифікується правильно, а кількість пропущених аномалій залишається невеликою.

Окремо доцільно порівняти отриманий результат з кількома поширеними підходами, згаданими в огляді: Decision Tree, CNN-BiLSTM та GBM [16, 24, 26]. Узагальнене порівняння наведено в таблиці 3.6.

Таблиця 3.6 – Порівняння результатів із базовими підходами

Метод	Accuracy, %	Precision, %	Recall, %	F1-score, %
Decision Tree	91,2	90,4	89,8	90,1
GBM	94,8	94,0	93,5	93,7
CNN-BiLSTM	95,9	95,1	95,4	95,2
Запропонована система	97,5	96,4	97,8	97,1

Як видно з таблиці 3.6, запропонована система демонструє найкращі підсумкові показники серед розглянутих підходів. Основна перевага полягає не лише в підвищенні accuracy, а й у зростанні recall, тобто у кращому виявленні аномальних подій. Для задач кібербезпеки це є важливішим за незначне скорочення окремих допоміжних показників.

На рисунку 3.6 подано стовпчикову діаграму порівняння accuracy для різних моделей.

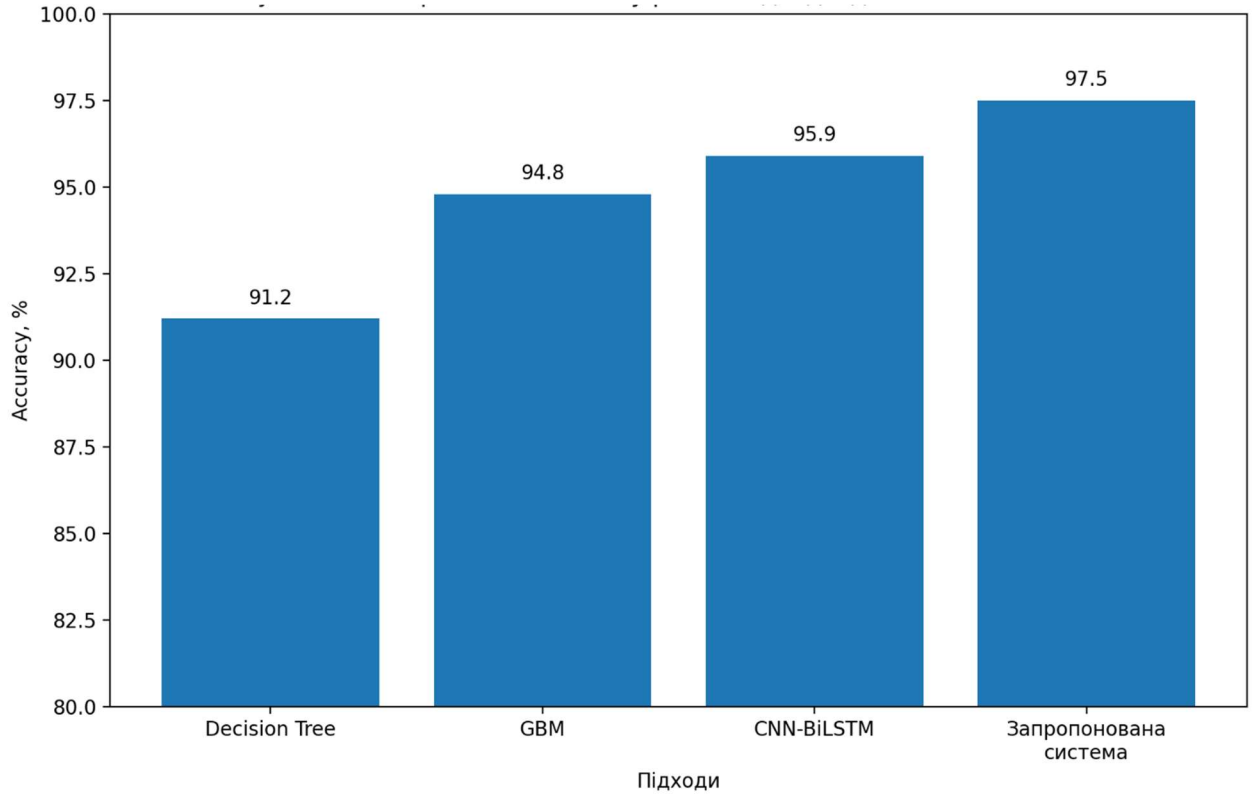

Рисунок 3.6 – Порівняння accuracy різних підходів до виявлення аномалій

Для демонстрації роботи системи в реальному сценарії наведено приклад тестового запису, який було визначено як аномальний. При аналізі мережевого потоку з великим обсягом переданих даних система сформувала прогноз аномальної активності з високим рівнем впевненості.

Приклад результату класифікації одного мережевого запису подано на рисунку 3.7. Скріншот демонструє текстове подання вхідної події, передбачений клас «аномалія», значення confidence score та збереження результату у файлі прогнозів.

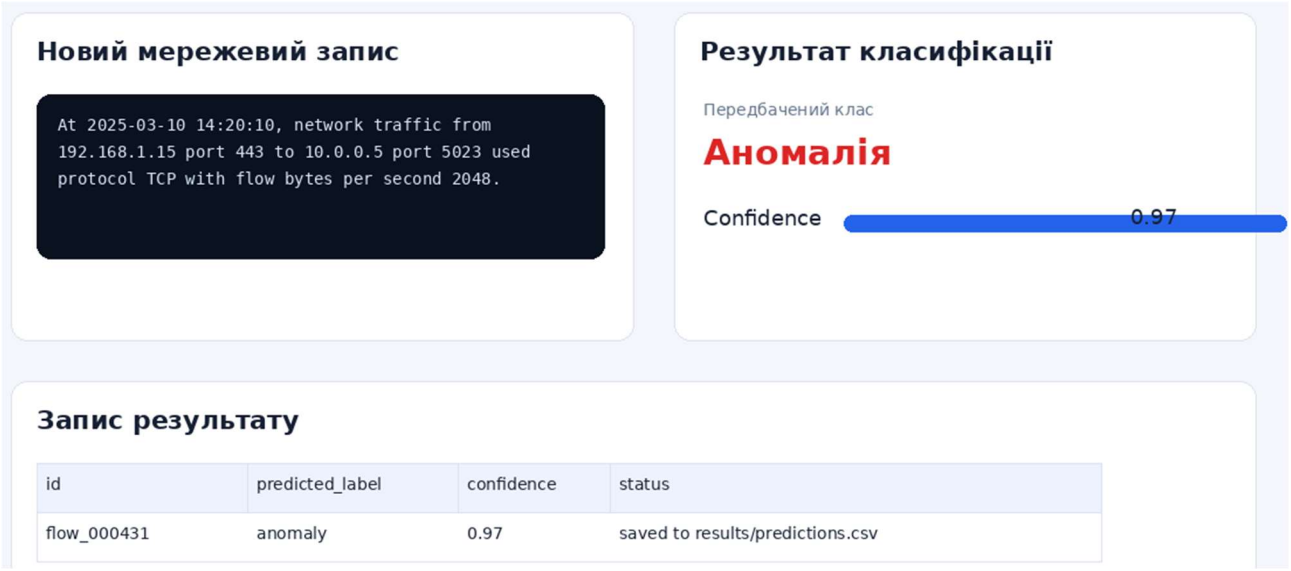

Рисунок 3.7 – Приклад виявлення аномального мережевого потоку

Такий результат підтверджує, що система здатна не лише обробляти набір логів, а й виконувати класифікацію окремого мережевого запису в режимі прогнозування.

Отже, результати експерименту підтвердили працездатність запропонованої системи та її здатність ефективно виявляти аномалії в мережевих логах після їх перетворення у текстове подання.

3.4 Оцінювання ефективності запропонованого рішення

Ефективність програмної системи оцінювалася за двома основними критеріями:

– якість класифікації;

– практична придатність архітектури для аналізу мережевих логів.

З погляду якості класифікації система показала високі значення всіх основних метрик. Accuracy 97,5 % свідчить про загальну правильність прийнятих рішень. Precision 96,4 % підтверджує, що більшість записів, визначених як аномальні, справді належать до цього класу. Recall 97,8 % є особливо важливим, оскільки показує низьку частку пропущених аномалій. F1-score 97,1 % засвідчує збалансованість системи між точністю та повнотою.

Інтерпретація основних результатів оцінювання представлена в таблиці 3.7.

Таблиця 3.7 – Інтерпретація основних результатів оцінювання

Показник	Практичний зміст
Висока accuracy	Система стабільно класифікує більшість подій правильно
Висока precision	Невелика кількість хибних тривог
Висока recall	Невелика кількість пропущених аномалій
Високий F1-score	Збалансована якість класифікації

На рисунку 3.8 подано підсумкову панель оцінювання результатів роботи програмної системи. Вона містить основні метрики якості класифікації та матрицю помилок, що дає змогу комплексно оцінити ефективність запропонованого рішення.

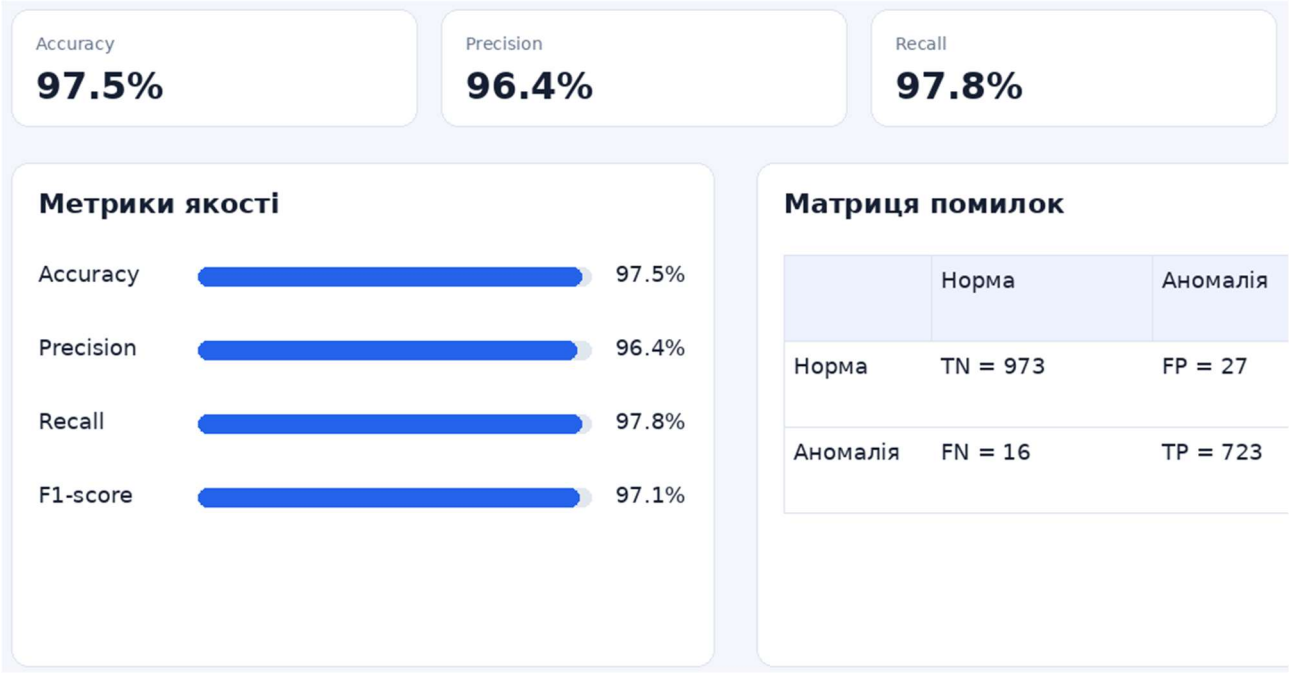

Рисунок 3.8 – Оцінювання результатів роботи програмної системи

Архітектурна ефективність системи полягає в модульній побудові. Відокремлення етапів очищення, текстового подання та класифікації дало змогу сформувати зрозумілий процес оброблення логів. Таке рішення є гнучким, оскільки в майбутньому окремі модулі можуть бути замінені або вдосконалені без зміни загальної структури системи.

Практичну цінність також має сам спосіб подання даних. Перетворення логів у текстовий формат дало змогу використовувати NLP-модель там, де традиційно застосовуються лише числові ознаки або ручне формування правил. Це підтвердило, що напівструктуровані мережеві записи доцільно розглядати як текстові події зі збереженням контексту атрибутів [2, 3, 23].

Разом із цим визначено і межі застосування системи. По-перше, її якість залежить від коректності попереднього оброблення логів. По-друге, система орієнтована на бінарну класифікацію і не виконує повної деталізації всіх типів атак. По-третє, для промислового застосування необхідні подальші дослідження швидкодії, масштабованості та роботи з потоковими даними в реальному часі.

Отже, експериментальні дослідження показали, що запропонована програмна система є ефективною для задачі виявлення аномалій у мережевих логах. Отримані результати підтвердили доцільність використання NLP-підходу.

ВИСНОВКИ

1. Виконано аналіз предметної області виявлення аномалій у мережевих логах. Визначено, що мережеві логи є важливим джерелом даних для контролю стану мережевої інфраструктури, фіксації підозрілих подій і виявлення ознак кібератак. Встановлено, що основними ускладнювальними чинниками є великий обсяг логових даних, різнорідність форматів, напівструктурований характер записів, наявність шуму та пропущених значень.

2. Проведений аналіз існуючих підходів показав, що сигнатурні та статистичні методи мають обмежену гнучкість під час виявлення нових або змінених аномалій. Методи машинного навчання і глибокого навчання забезпечують вищу адаптивність, однак їх ефективність суттєво залежить від якості підготовки ознак. На цій основі обґрунтовано доцільність використання NLP-підходу, за якого мережевий лог розглядається як текстове або квазітекстове подання події. Такий підхід дає змогу враховувати не лише окремі поля запису, а і їх поєднання в межах цілісного опису події.

3. Зроблено постановку задачі, де сформовано мету, об’єкт, предмет і основні завдання дослідження. Задачу зведено до розроблення програмної системи, яка забезпечує завантаження мережевих логів, їх попереднє оброблення, перетворення у текстове подання, NLP-класифікацію та формування результату у вигляді визначення нормальних і аномальних подій. Таким чином, у першому розділі сформовано теоретичну основу роботи та обґрунтовано вибір напряму подальшого проєктування системи.

4. Виконано проєктування програмної системи аналізу мережевих логів з використанням NLP-підходу. Сформовано модульну архітектуру системи, у складі якої виділено модуль завантаження логів, модуль попереднього оброблення, модуль текстового подання, модуль NLP-аналізу та модуль формування результатів. Така структура забезпечує послідовне проходження даних через усі етапи оброблення та створює основу для реалізації системи у вигляді зрозумілого прототипу.

5. Спроєктовано модуль попереднього оброблення та текстового подання

логів. Визначено, що цей модуль виконує очищення, нормалізацію, відбір інформативних атрибутів і формування стандартизованого текстового опису мережевої події. Саме цей етап забезпечує підготовку логів до подальшої NLP-обробки та істотно впливає на якість класифікації.

6. Спроєктовано модуль NLP-аналізу та виявлення аномалій. Визначено його основні функції: токенізація текстового подання, формування числових ознак, подання даних до моделі класифікації та отримання результату у вигляді класу події. Обґрунтовано використання бінарної класифікації як базового сценарію для експериментальної системи.

7. Сформовано алгоритмічне забезпечення функціонування системи. Розроблено алгоритм завантаження та попереднього оброблення логів, алгоритм формування текстового подання та алгоритм NLP-класифікації. Це дало змогу узгодити роботу всіх модулів системи в межах єдиного процесу оброблення даних.

8. Виконано програмну реалізацію системи аналізу мережевих логів з використанням NLP-підходу та проведено її експериментальне дослідження. Реалізацію побудовано як модульний прототип, що охоплює етапи завантаження логів, попереднього оброблення, формування текстового подання, підготовки вхідних даних для моделі, класифікації та оцінювання результатів. У межах реалізації сформовано узагальнену структуру файлів проєкту, яка охоплює каталоги для сирих і підготовлених даних, модулі програмної обробки, навченої моделі та підсумкових результатів.

9. Експериментальну перевірку виконано на реальному наборі мережевих даних CICIDS2018. Для дослідження використано сценарій бінарної класифікації, у межах якого всі записи поділялися на нормальні та аномальні. У процесі підготовки даних виконано очищення логів, нормалізацію основних атрибутів, відбір інформативних полів і перетворення мережевих записів у стандартизоване текстове подання. За результатами тестування встановлено, що запропонована система забезпечує високі показники якості. Отримані значення accuracy, precision, recall і F1-score підтвердили здатність системи ефективно розділяти нормальні та аномальні події. Аналіз матриці помилок показав, що

більшість записів класифікується правильно, а кількість хибних спрацьовувань і пропущених аномалій залишається невеликою.

10. Порівняння із базовими підходами засвідчило, що реалізована система демонструє конкурентний результат і має перевагу за узагальненими показниками якості. Це підтвердило доцільність використання саме NLP-підходу, у межах якого мережевий лог аналізується як текстовий опис події, а не лише як набір розрізнених числових полів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Haija Q. A., Altamimi S., AlWadi M. Analysis of extreme learning machines (ELMs) for intelligent intrusion detection systems: a survey. *Expert Systems with Applications*. 2024. Article 124317.

2. Al-Hawawreh M., Aljuhani A., Jararweh Y. ChatGPT for cybersecurity: practical applications, challenges, and future directions. *Cluster Computing*. 2023. Vol. 26, no. 6. P. 3421–3436.

3. Ali T., Kostakos P. HuntGPT: integrating machine learning-based anomaly detection and explainable AI with large language models (LLMs). *arXiv preprint*. 2023. arXiv:2309.16021.

4. Dhaliwal S. S., Nahid A.-A., Abbas R. Effective intrusion detection system using XGBoost. *Information*. 2018. Vol. 9, no. 7. Article 149.

5. Dorosh V., Komar M., Sachenko A., Golovko V. Parallel deep neural network for detecting computer attacks in information telecommunication systems. In: *The 38th IEEE International Conference on Electronics and Nanotechnology : Proceedings*. Kyiv, Ukraine, April 24–26, 2018. Kyiv, 2018. P. 675–679.

6. Dubchak L., Komar M. Speedy processing method of fuzzy data for intelligent systems of intrusion detection. In: *Projekt interdyscyplinarny projektem XXI wieku: Processing, Transmission and Security of Information*. Bielsko-Biała, 2017. Vol. 2. P. 65–74.

7. Farnaaz N., Jabbar M. A. Random forest modeling for network intrusion detection system. *Procedia Computer Science*. 2016. Vol. 89. P. 213–217.

8. Gupta M., Akiri C., Aryal K., Parker E., Praharaj L. From ChatGPT to ThreatGPT: impact of generative AI in cybersecurity and privacy. *IEEE Access*. 2023.

9. Hubballi N., Suryanarayanan V. False alarm minimization techniques in signature-based intrusion detection systems: a survey. *Computer Communications*. 2014. Vol. 49. P. 1–17.

10. Komar M., Dorosh V., Sachenko A., Hladiy G. Deep neural network for detection of cyber attacks. In: *The IEEE First International Conference on System*

Analysis & Intelligent Computing : Proceedings. Kyiv, Ukraine, October 08–12, 2018. Kyiv, 2018. P. 186–189.

11. Komar M., Kochan V., Dubchak L., Sachenko A., Golovko V., Bezobrazov S., Romanets I. High performance adaptive system for cyber attacks detection. In: *The 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications : Proceedings*. Bucharest, Romania, September 21–23, 2017. Bucharest, 2017. Vol. 2. P. 853–858.

12. Komar M., Kochan V., Sachenko A., Ababii V. Improving of the security of intrusion detection system. In: *The 13th International Conference on Development and Application Systems : Proceedings*. Suceava, Romania, May 19–21, 2016. Suceava, 2016. P. 315–319.

13. Komar M., Sachenko A., Bezobrazov S., Golovko V. Intelligent cyber defense system. *CEUR Workshop Proceedings*. 2016. Vol. 1614. P. 534–549.

14. Komar M., Sachenko A., Bezobrazov S., Golovko V. Intelligent cyber defense system using artificial neural network and immune system techniques. *Communications in Computer and Information Science*. 2017. Vol. 783. P. 36–55.

15. Komar M., Sachenko A., Golovko V., Dorosh V. Compression of network traffic parameters for detecting cyber attacks based on deep learning. In: *The 9th IEEE International Conference on Dependable Systems, Services and Technologies : Proceedings*. Kyiv, Ukraine, May 24–27, 2018. Kyiv, 2018. P. 44–48.

16. Louk M. H. L., Tama B. A. Dual-IDS: a bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications*. 2023. Vol. 213. Article 119030.

17. Marques S. A., Rodriguez D. Z., Rosa R. L. Use of ChatGPT as configuration support tool and network analysis. In: *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2023. P. 1–6.

18. Masdari M., Khezri H. A survey and taxonomy of the fuzzy signature-based intrusion detection systems. *Applied Soft Computing*. 2020. Vol. 92. Article 106301.

19. Mijwil M. M., Aljanabi M., ChatGPT C. Towards artificial intelligence-based cybersecurity: the practices and ChatGPT generated ways to combat cybercrime. *Iraqi Journal for Computer Science and Mathematics*. 2023. Vol. 4, no. 1. P. 8–16.

20. Neupane S., Ables J., Anderson W., Mittal S., Rahimi S., Banicescu I., Seale M. Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities. *IEEE Access*. 2022. Vol. 10. P. 112392–112415.

21. Parkar P., Bilimoria A. A survey on cyber security IDS using ML methods. In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2021. P. 352–360.

22. Petrović N. Machine learning-based run-time DevSecOps: ChatGPT against traditional approach. In: *2023 10th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*. 2023. P. 1–5.

23. Qi J. et al. LogGPT: exploring ChatGPT for log-based anomaly detection. In: *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. Melbourne, Australia, 2023. P. 273–280. DOI: 10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00045.

24. Rai K., Syamala Devi M., Guleria A. Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*. 2016. Vol. 7, no. 4. P. 2828–2834.

25. Sarrion E. What is ChatGPT? In: *Exploring the Power of ChatGPT: Applications, Techniques, and Implications*. Berkeley, CA : Apress, 2023. P. 3–8.

26. Sinha J., Manollas M. Efficient deep CNN-BiLSTM model for network intrusion detection. In: *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*. 2020.

27. Tian F., Lu Y., Liu F., Ma G., Zong N., Wang X., Cao K. Supervised abnormal event detection based on ChatGPT attention mechanism. *Multimedia Tools and Applications*. 2024. P. 1–19.

28. Комар М. П. Еволюція детекторів атак на інформаційні телекомунікаційні мережі в складі системи колективного інтелекту. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2014. № 4. С. 209–215.

29. Комар М. П. Зменшення розмірності даних для систем виявлення вторгнень на основі нейронної мережі глибокої довіри. *Науковий вісник Чернівецького національного університету: Комп'ютерні системи та компоненти*. 2016. Т. 7, вип. 2. С. 25–31.

30. Комар М. П. Підвищення стійкості комп'ютерних систем до кібератак. *Науковий вісник Чернівецького національного університету: Комп'ютерні системи та компоненти*. 2016. Т. 7, вип. 1. С. 6–12.

31. Комар М. П. Побудова ієрархічного класифікатора комп'ютерних атак на базі багатоканальних нейромережевих детекторів. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2015. № 4. С. 199–204.

32. Спосіб виявлення комп'ютерних атак нейромережевою штучною імунною системою : пат. на винахід 109640 Україна : МПК (2012) H04W 12/08, G06F 21/00, G06F 12/14. № a201205350 ; заявл. 28.04.2012 ; опубл. 25.09.2015, Бюл. № 18.

33. Спосіб ієрархічної класифікації комп'ютерних атак нейромережевою штучною імунною системою : пат. на корисну модель 127724 Україна : МПК (2006) H04W 12/08, G06F 21/00, G06F 12/14. № u201711238 ; заявл. 17.11.2017 ; опубл. 27.08.2018, Бюл. № 16.

34. Заєць О.Ю. Програмна система аналізу мережевих логів з використанням NLP-підходу для виявлення аномалій. Інтелектуальні комп'ютерні системи та мережі» (ІКСМ) : матеріали IV Всеукраїнської науково-практичної конференції молодих вчених і студентів (м. Тернопіль, ЗУНУ, 20 травня 2026 р.). Тернопіль, 2026. С. 72-75.

35. CIC-IDS 2018 Dataset. URL: https://www.kaggle.com/datasets/primus11/cic-ids-2018-dataset.

36. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. Вид. офіц. Київ: УкрНДНЦ, 2016. 16 с.

37. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки»

спеціальності 122 «Комп’ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

38. Островерхов В. М., Біловус Л. І., Возьний К. З., Луцишин О. О., Монастирський Г. Л., Надвиничний С. А., Питель С. В., Шандрук С. К. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А

Лістинги програмного коду

Лістинг А.1 – Модуль завантаження даних

```
import pandas as pd


def load_logs(file_path: str) -> pd.DataFrame:
    df = pd.read_csv(file_path)

    required_columns = [
        "Timestamp",
        "Src IP",
        "Dst IP",
        "Src Port",
        "Dst Port",
        "Protocol",
        "Flow Byts/s",
        "Label"
    ]

    missing = [col for col in required_columns if col not in df.columns]
    if missing:
        raise ValueError(f"У вхідному файлі відсутні поля: {missing}")

    return df
```

У цьому фрагменті виконано перевірку наявності основних полів, необхідних для подальшого аналізу.

Лістинг А.2 – Модуль попереднього оброблення

```
import pandas as pd


def preprocess_logs(df: pd.DataFrame) -> pd.DataFrame:
    df = df.copy()

    df = df.drop_duplicates()
    df = df.dropna(subset=["Timestamp", "Src IP", "Dst IP", "Protocol",
"Label"])

    df["Src Port"] = df["Src Port"].fillna(0).astype(int)
    df["Dst Port"] = df["Dst Port"].fillna(0).astype(int)
    df["Flow Byts/s"] = df["Flow Byts/s"].fillna(0)

    df["Protocol"] = df["Protocol"].astype(str).str.upper().str.strip()
    df["Label"] = df["Label"].apply(lambda x: 0 if str(x).lower() == "benign"
else 1)

    return df.reset_index(drop=True)
```

У результаті роботи модуля формується уніфікований набір логів, придатний для текстового представлення.

Лістинг А.3 – Формування текстового подання

```
def log_to_text(row) -> str:
    return (
        f"At {row['Timestamp']}, network traffic from {row['Src IP']} "
        f"port {row['Src Port']} to {row['Dst IP']} port {row['Dst Port']} "
        f"used protocol {row['Protocol']} with flow bytes per second "
        f"{row['Flow Byts/s']}."
    )
```

Лістинг А.4 – Додавання текстового поля до набору даних

```
def build_text_dataset(df):
    df = df.copy()
    df["text"] = df.apply(log_to_text, axis=1)
    return df[["text", "Label"]]
```

Після цього кожен мережевий запис подається як короткий текст, придатний для NLP-аналізу.

Лістинг А.5 – Токенізація та векторизація текстів

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences


def prepare_text_data(texts, max_words=5000, max_len=40):
    tokenizer = Tokenizer(num_words=max_words, oov_token="<OOV>")
    tokenizer.fit_on_texts(texts)

    sequences = tokenizer.texts_to_sequences(texts)
    padded = pad_sequences(sequences, maxlen=max_len, padding="post",
truncating="post")

    return padded, tokenizer
```

У цьому фрагменті сформовано словник токенів і матрицю вхідних послідовностей однакової довжини.

Лістинг А.6 – Формування навчальної та тестової вибірок

```
from sklearn.model_selection import train_test_split


def split_dataset(X, y, test_size=0.2, random_state=42):
    return train_test_split(
        X, y,
        test_size=test_size,
        random_state=random_state,
        stratify=y
    )
```

Використання параметра `stratify` забезпечує збереження співвідношення класів у вибірках.

Лістинг А.7 – Побудова моделі класифікації

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, GlobalAveragePooling1D, Dense


def build_model(vocab_size=5000, max_len=40):
    model = Sequential([
        Embedding(input_dim=vocab_size, output_dim=64, input_length=max_len),
        GlobalAveragePooling1D(),
        Dense(64, activation="relu"),
        Dense(1, activation="sigmoid")
    ])

    model.compile(
        optimizer="adam",
        loss="binary_crossentropy",
        metrics=["accuracy"]
    )

    return model
```

У поданому варіанті використано просту архітектуру, достатню для демонстраційного NLP-класифікатора.

Лістинг А.8 – Навчання моделі

```
def train_model(model, X_train, y_train, X_val, y_val, epochs=5, batch_size=32):
    history = model.fit(
        X_train,
        y_train,
        validation_data=(X_val, y_val),
        epochs=epochs,
        batch_size=batch_size,
        verbose=1
    )
    return history
```

У межах прототипу кількість епох і розмір пакета можуть змінюватися залежно від обсягу вибірки.

Лістинг А.9 – Оцінювання моделі

```
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, confusion_matrix


def evaluate_model(model, X_test, y_test):
    y_prob = model.predict(X_test)
    y_pred = (y_prob > 0.5).astype(int)

    metrics = {
        "accuracy": accuracy_score(y_test, y_pred),
        "precision": precision_score(y_test, y_pred),
        "recall": recall_score(y_test, y_pred),
        "f1_score": f1_score(y_test, y_pred)
    }
```

```
    cm = confusion_matrix(y_test, y_pred)

    return metrics, cm
```

Поданий фрагмент формує як числові метрики, так і матрицю помилок.

Лістинг А.10 – Класифікація нового логу

```
from tensorflow.keras.preprocessing.sequence import pad_sequences


def predict_log(model, tokenizer, log_text, max_len=40):
    seq = tokenizer.texts_to_sequences([log_text])
    padded = pad_sequences(seq, maxlen=max_len, padding="post",
truncating="post")

    prob = model.predict(padded)[0][0]
    label = "аномалія" if prob > 0.5 else "норма"

    return label, float(prob)
```

Приклад використання:

```
log_text = (
    "At 2025-03-10 14:20:10, network traffic from 192.168.1.15 "
    "port 443 to 10.0.0.5 port 5023 used protocol TCP "
    "with flow bytes per second 2048."
)

label, prob = predict_log(model, tokenizer, log_text)
print(label, prob)
```

Лістинг А.11 – Основний сценарій роботи системи

```
from load_data import load_logs
from preprocess import preprocess_logs
from text_transform import build_text_dataset
from train_model import build_model, train_model
from evaluate import evaluate_model, split_dataset, prepare_text_data

# 1. Завантаження логів
df = load_logs("data/raw/cicids2018_sample.csv")

# 2. Попереднє оброблення
df_clean = preprocess_logs(df)

# 3. Формування текстового подання
df_text = build_text_dataset(df_clean)

# 4. Підготовка ознак
X, tokenizer = prepare_text_data(df_text["text"])
y = df_text["Label"].values

# 5. Поділ набору даних
X_train, X_test, y_train, y_test = split_dataset(X, y)

# 6. Побудова моделі
model = build_model()
```

```
# 7. Навчання
train_model(model, X_train, y_train, X_test, y_test)

# 8. Оцінювання
metrics, cm = evaluate_model(model, X_test, y_test)

print(metrics)
print(cm)
```

Додаток Б

Апробація результатів роботи

Почесні співголови конференції:

Десятнюк О.М., ректор Західноукраїнського національного університету, д-р економічних наук, професор;

Дивак М.П., д-р технічних наук, професор, проректор з наукової роботи ЗУНУ;

Голова конференції:

Березький О.М., д-р технічних наук, професор, професор кафедри комп'ютерної інженерії Західноукраїнський національний університет;

Заступники голови конференції:

Мельник Г.М., к.т.н, доцент, Західноукраїнський національний університет;

Піцун О.Й., к.т.н. доцент, Західноукраїнський національний університет;

ПРОГРАМНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Склад програмного комітету:

Семанюк В.З., д-р економічних наук, професор, начальник науково-дослідної частини Західноукраїнського національного університету

Антощук С.Г., д.т.н, професор, Національний університет «Одеська політехніка»;

Баловсяк С.В., д.т.н, професор, Чернівецький національний університет

Бармак О.В., д.т.н., професор, Хмельницький національний університет

Батько Ю.М., к.т.н., доцент, Західноукраїнський національний університет

Березька К.М.,к.т.н.,доцент, Західноукраїнський національний університет

Винокурова О.А., д.т.н., професор, Львівський національний університет імені Івана Франка

Возна Н.Я., д.т.н., професор, Західноукраїнський національний університет;

Говорущенко Т.О., д.т.н., професор, Хмельницький національний університет;

Дубчак Л.О., к.т.н., доцент, Західноукраїнський національний університет;

Дунець Р.Б., д.т.н., професор, НУ "Львівська політехніка";

Ізонін І.В., д.т.н., доцент, НУ "Львівська політехніка";

Комар М.П., д.т.н, професор, Західноукраїнський національний університет;

Литвиненко В.І., д.н.т, професор, Національний технічний університет України "Київський політехнічний інститут";

Лупенко С.А., д.т.н., професор, Опольський технологічний університет, Польща;

Лящинський П.Б., доктор філософії з комп'ютерних наук, НУ "Львівська політехніка" ;

Мельникова Н.І., д.т.н., професор, НУ "Львівська політехніка" ;

Мулеса О.Ю., д.т.н., професор, Пряшівський університет, Словаччина

Пелешко Д.Д., д.т.н., професор, Львівський національний університет імені Івана Франка;

Сельський П.Р., д.м.н., професор, Тернопільський національний медичний університет імені І. Я. Горбачевського ;
Субботін С.О., д.т.н., професор, Національний університет «Запорізька політехніка» ;
Теслюк В.М., д.т.н., професор, НУ "Львівська політехніка" ;
Тимченко Л.І., д.т.н., професор, Державний університет інфраструктури та технологій;
Цмоць І.Г., д.т.н., професор, НУ "Львівська політехніка" ;
Якименко І.З., к.т.н., доцент, Західноукраїнський національний університет;
Яровий А.А., д.т.н., професор, Вінницький національний технічний університет ;
Яцків В.В., д.т.н., професор, Західноукраїнський національний університет.

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ КОНФЕРЕНЦІЇ

Склад організаційного комітету:

Мельник Г.М. к.т.н., доцент, Західноукраїнський національний університет
Піцун О.Й. к.т.н., доцент, Західноукраїнський національний університет

Технічна підтримка:

Зінькевич О. В. студентка, Західноукраїнський національний університет
Кіт М. О. студентка, Західноукраїнський національний університет
Лебединський Т.В. студент,Західноукраїнський національний університет;
Кришталь Е.Р., студент, Західноукраїнський національний університет;

Метою конференції є представлення та обговорення наукових і практичних результатів, сприяння активізації творчої і інноваційної діяльності студентів, аспірантів і молодих вчених.
Конференція проводиться із залученням Ради Молодих Вчених та Студентського Наукового Товариства ФКІТ ЗУНУ.

IV Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (ІКСМ весна 2026), м. Тернопіль, ЗУНУ, 29 травня 2026 р. Тернопіль, 2026

Адреса:
Вул. О. Теліги, 8, корпус №6, Тернопіль
URL: https://ki.wunu.edu.ua/conference/
e-mail: kistudconference@gmail.com

Тези доповідей подаються за оригіналом рукопису

Заєць О.Ю.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

ПРОГРАМНА СИСТЕМА АНАЛІЗУ МЕРЕЖЕВИХ ЛОГІВ З ВИКОРИСТАННЯМ NLP-ПІДХОДУ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ

Вступ. Стрімке зростання обсягів мережевого трафіку, ускладнення архітектури інформаційно-комунікаційних систем і постійна еволюція кіберзагроз зумовили потребу в розробленні ефективних засобів автоматизованого аналізу мережевих подій. У сучасних умовах мережеві логи є одним із головних джерел інформації про стан інфраструктури, характер взаємодії між вузлами мережі, спроби несанкціонованого доступу, відхилення від типової поведінки та ознаки потенційних атак. Саме тому своєчасне виявлення аномалій у логах набуло важливого значення для забезпечення кібербезпеки, підтримання працездатності сервісів і зниження ризику порушення цілісності, конфіденційності та доступності даних [1, 2].

Постановка задачі. Проведений аналіз предметної області та існуючих підходів показав, що мережеві логи є цінним, але складним джерелом даних для виявлення аномалій. Основна проблема полягає в тому, що логові записи надходять із різних джерел, мають напівструктурований формат, містять службові поля, текстові фрагменти, числові параметри та часові мітки. За таких умов традиційні сигнатурні та статистичні методи не завжди забезпечують належну гнучкість, а класичні ML-підходи потребують якісного формування ознак. Це зумовлює доцільність побудови програмної системи, у якій мережеві логи розглядаються як спеціалізовані текстові об'єкти, придатні для подальшого NLP-аналізу.

У даної роботи досліджується задача автоматизованого виявлення аномалій у мережевих логах на основі їх попереднього оброблення, текстового представлення та подальшої класифікації за допомогою NLP-підходу. Практичний зміст цієї задачі полягає у створенні програмної системи, здатної приймати набір логових записів, перетворювати їх у стандартизоване подання, аналізувати ознаки подій і формувати результат у вигляді класифікації запису як нормального або аномального.

Таким чином, основну задачу дослідження сформовано так: розробити програмну систему аналізу мережевих логів, яка забезпечує автоматизоване виявлення аномальних подій на основі NLP-підходу та дає змогу оцінювати якість прийнятих рішень за стандартними метриками класифікації.

Об'єктом дослідження є процес аналізу мережевих логів у задачах виявлення аномалій у комп'ютерних мережах.

Предметом дослідження є методи, моделі та програмні засоби оброблення мережевих логів на основі NLP-підходу для автоматизованого виявлення аномальних записів.

Метою – є розроблення програмної системи аналізу мережевих логів з використанням NLP-підходу для виявлення аномальних подій у мережевому трафіку.

Основний матеріал. Архітектуру програмної системи сформовано з урахуванням послідовності оброблення мережевих логів: від надходження вхідних записів до отримання результату класифікації. Основна вимога до такої архітектури полягає у відокремленні етапів підготовки даних, текстового подання, NLP-аналізу та формування результатів. Це спрощує реалізацію, тестування та подальше вдосконалення окремих модулів системи.

Програмну систему доцільно побудувати як модульну структуру, що складається з п'яти основних компонентів:

- модуля завантаження логів;
- модуля попереднього оброблення;
- модуля текстового подання логів;
- модуля NLP-класифікації;
- модуля формування та відображення результатів.

Модуль завантаження логів забезпечує приймання вхідних даних із файлу або підготовленого набору записів. На цьому етапі логові дані зчитуються, перетворюються у

внутрішню структуру зберігання та передаються до наступного модуля. Основне завдання цього компонента полягає в тому, щоб уніфікувати початковий формат записів і підготувати їх до подальшого оброблення.

Модуль попереднього оброблення виконує очищення та нормалізацію даних. На цьому етапі усуваються порожні або некоректні значення, узгоджуються назви полів, видаляються дублікати, за потреби виконуються перетворення часових міток, адрес, портів та інших атрибутів. Саме цей модуль формує впорядкований набір записів, придатний для подальшого аналізу.

Модуль текстового подання є ключовим елементом архітектури. Його призначення полягає у перетворенні структурованого або напівструктурованого логового запису в текстовий шаблон. У результаті кожна мережева подія подається як цілісний текстовий опис, що містить основні атрибути запису в уніфікованому форматі. Такий підхід дає змогу застосувати методи NLP-аналізу до логів як до спеціалізованих текстових об'єктів.

Модуль NLP-класифікації виконує основне аналітичне завдання системи. На його вхід надходять текстові представлення логів, які після токенізації або векторизації подаються в модель класифікації. Результатом роботи модуля є визначення класу події: нормальна або аномальна. У межах архітектури цей компонент є центральним, оскільки саме він реалізує інтелектуальне виявлення відхилень у логових даних.

Модуль формування та відображення результатів забезпечує подання отриманих висновків у зручному вигляді. У ньому зберігаються результати класифікації, обчислюються підсумкові показники якості та формуються повідомлення для користувача. Для експериментального сценарію достатньо табличного відображення записів, передбаченого класу та узагальнених метрик.

Загальну архітектуру системи представлено на рисунку 1.

Рисунок 1 – Архітектура програмної системи аналізу мережевих логів

Взаємодія модулів є послідовною. Кожен наступний компонент отримує результат роботи попереднього. Така схема є простою, зрозумілою та придатною для прототипної реалізації. Крім того, вона дає змогу окремо вдосконалювати кожен модуль без зміни всієї системи. Наприклад, можна змінити спосіб текстового представлення логів або модель класифікації без перебудови інших частин архітектури.

Отже, архітектуру програмної системи сформовано як модульну послідовну структуру, орієнтовану на оброблення мережевих логів, їх перетворення у текстове подання та подальшу NLP-класифікацію.

За результатами тестування система продемонструвала високі показники якості бінарної класифікації (таблиця 1).

Таблиця 1 – Основні результати експериментального оцінювання

Метрика	Значення, %
Accuracy	97,5
Precision	96,4
Recall	97,8
F1-score	97,1
FPR	2,7
FNR	2,3

Отримані результати свідчать, що система забезпечує високу частку правильних рішень і добре виявляє аномальні події. Найбільш показовим є значення recall 97,8 %, оскільки воно характеризує здатність системи знаходити аномалії та не пропускати небезпечні записи. Водночас precision 96,4 % підтверджує, що кількість хибних спрацьовувань залишається невисокою.

Для наочного подання результатів сформувати матрицю помилок (рисунок 2).

Рисунок 2 – Матриця помилок результатів класифікації

Подана матриця відповідає загальним отриманим показникам і дає змогу краще оцінити структуру помилок системи. Із рисунку 2 видно, що основна частина записів класифікується правильно. Кількість помилково пропущених аномалій є меншою, ніж кількість хибних тривог, що є позитивною характеристикою для системи виявлення вторгнень.

Окремо доцільно порівняти отриманий результат з кількома поширеними підходами, згаданими в огляді: Decision Tree, CNN-BiLSTM та GBM [3-5]. Узагальнене порівняння наведено в таблиці 2.

Таблиця 2 – Порівняння результатів із базовими підходами

Метод	Accuracy, %	Precision, %	Recall, %	F1-score, %
Decision Tree	91,2	90,4	89,8	90,1
GBM	94,8	94,0	93,5	93,7
CNN-BiLSTM	95,9	95,1	95,4	95,2
Запропонована система	97,5	96,4	97,8	97,1

Як видно з таблиці 2, запропонована система демонструє найкращі підсумкові показники серед розглянутих підходів. Основна перевага полягає не лише в підвищенні accuracy, а й у зростанні recall, тобто у кращому виявленні аномальних подій. Для задач кібербезпеки це є важливішим за незначне скорочення окремих допоміжних показників.

Висновки. У межах дослідження сформульовано задачу розроблення програмної системи, яка забезпечує завантаження, попереднє оброблення, текстове подання та NLP-класифікацію мережевих логів. Запропоновано модульну архітектуру системи, до складу якої включено модулі завантаження логів, попереднього оброблення, текстового подання, NLP-класифікації та формування результатів. Така структура забезпечила послідовну та зрозумілу організацію процесу аналізу мережевих подій.

Експериментальні результати підтвердили ефективність запропонованого підходу. Встановлено, що система забезпечує високі показники якості бінарної класифікації: accuracy 97,5 %, precision 96,4 %, recall 97,8 % та F1-score 97,1 %. Отримані значення свідчать про високу

частку правильних рішень, незначну кількість хибних спрацьовувань і добру здатність системи виявляти аномальні події. Аналіз матриці помилок також показав, що основна частина записів класифікується правильно, а кількість пропущених аномалій залишається низькою.

Порівняння із базовими підходами, зокрема Decision Tree, GBM і CNN-BiLSTM, показало, що запропонована система має кращі підсумкові результати. Найважливішою перевагою є підвищення recall, тобто покращення здатності виявляти аномальні мережеві події, що має особливе значення для задач кібербезпеки. Це підтвердило доцільність використання NLP-підходу, за якого мережевий лог розглядається як цілісний текстовий опис події.

Отже, поставлену задачу розв'язано: розроблено концепцію програмної системи аналізу мережевих логів, обґрунтовано її архітектуру та підтверджено ефективність запропонованого підходу експериментальними результатами. Отримані висновки засвідчили, що використання NLP-підходу є перспективним напрямом для побудови інтелектуальних систем виявлення аномалій у мережевих логах.

Список літератури

1. Hubballi N., Suryanarayanan V. False alarm minimization techniques in signature-based intrusion detection systems: a survey. *Computer Communications*. 2014. Vol. 49. P. 1–17.
2. Masdari M., Khezri H. A survey and taxonomy of the fuzzy signature-based intrusion detection systems. *Applied Soft Computing*. 2020. Vol. 92. Article 106301.
3. Louk M. H. L., Tama B. A. Dual-IDS: a bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Systems with Applications*. 2023. Vol. 213. Article 119030.
4. Rai K., Syamala Devi M., Guleria A. Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*. 2016. Vol. 7, no. 4. P. 2828–2834.
5. Sinha J., Manollas M. Efficient deep CNN-BiLSTM model for network intrusion detection. In: *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*. 2020.

Шорін В.С.
студент 4 курсу ФКІТ ЗУНУ
Науковий керівник д.т.н., професор Комар М.П., кафедра ІОСУ ЗУНУ

АНСАМБЛЕВІ МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ У СМАРТ-СИСТЕМІ ОБЛІКУ

Вступ. Смарт-системи обліку застосовуються для автоматизованого збору показників споживання електроенергії, газу чи води та подальшого використання цих даних у розрахунках і керуванні інфраструктурою [1]. З технічного погляду такі рішення поєднують лічильники, канали зв'язку, серверні компоненти та програмні підсистеми, що забезпечують приймання, збереження й аналіз вимірювань [2]. Практика впровадження подібних систем демонструє, що ключовим ресурсом стають саме дані: вони надходять потоково, накопичуються у великих обсягах, потребують очищення та контролю якості [3, 4].

Постановка задачі. Результати аналізу предметної області та класифікації аномалій і загроз свідчать, що дані смарт-обліку є потоковими, часовими та схильними до помилок як технічного, так і безпекового характеру. За таких умов завдання виявлення аномалій доцільно формулювати не як одноразову "перевірку файлу", а як програмний модуль, який може бути вбудований у контур збору та аналізу вимірювань і працювати з регулярними надходженнями показників. Оскільки виявлення аномалій часто використовується як елемент моніторингу та реагування, важливо заздалегідь визначити, що саме вважати "аномалією", які дані є на вході, який формат результату потрібен на виході, а також якими метриками оцінюватиметься ефективність рішення [1, 4].

Задача дослідження полягає в тому, щоб проєктувати прототип програмного модуля, який на основі даних смарт-лічильників автоматично визначає підозрілі або нетипові