Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка системи раннього виявлення DDoS–атак на основі аналізу аномалій мережевого трафіку у корпоративних мережах

Виконав: студент 2(4) курсу, гр.2КІТС–21б
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

Жолукевський Р. В.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. МБІС
Грицак А. В.
(прізвище та ініціали)
«3» червня 2025 р.

Рецензент: к.т.н., доц., доцент каф. ОТ
Крупельницький Л.В.
(прізвище та ініціали)
«3» червня 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є.
"3" червня 2025р.

Вінниця ВНТУ – 2025

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти І–й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо–професійна програма – Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС
д.т.н., проф. Яремчук Ю.Є.
"20" березня 2025 р.

З А В Д А Н Н Я
на бакалаврську кваліфікаційну роботу студенту
Жолукевському Ростиславу Володимировичу
(прізвище, ім`я, по–батькові)

1. Тема роботи: Розробка системи раннього виявлення DDoS–атак на основі аналізу аномалій мережевого трафіку у корпоративних мережах
Керівник роботи Грицак Анатолій Васильович к.т.н., доц., каф., МБІС
(прізвище, ім`я, по–батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від "20" березня 2025року № 97
2. Термін подання студентом роботи 28 травня 2025 року
3. Вихідні дані до роботи Метою роботи є розробка системи раннього виявлення DDoS-атак на основі аналізу аномалій мережевого трафіку у корпоративних мережах
4. Зміст текстової частини: У роботі описано процес проєктування та реалізації програмного засобу для раннього виявлення DDoS-атак у корпоративних мережах на основі LSTM-мережі. Подано методику збору, попередньої обробки та нормалізації мережевого трафіку, розроблено архітектуру моделі, виконано навчання та тестування на структурованих вибірках. Оцінено ефективність системи за допомогою метрик продуктивності, розглянуто можливість інтеграції моделі в систему моніторингу в реальному часі.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
Слайд з зображенням класифікації DDoS-атак, архітектура типового DDoS, Архітектура LSTM-мережі для аналізу трафіку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Гринак А.В. доц. кафедри МБІС		
Розділ II	Гринак А.В. доц. кафедри МБІС		
Розділ III	Гринак А.В. доц. кафедри МБІС		

7.Дата видачі завдання 20 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Вивчення напрямку роботи	20.03.2025	24.03.2025	виконано
2	Аналіз предметної області	25.03.2025	06.04.2025	виконано
3	Розробка архітектури системи	07.04.2025	12.04.2025	виконано
4	Проектування функціональності	13.04.2025	22.04.2025	виконано
5	Створення системи клієнт-сервер	23.04.2025	26.04.2025	виконано
6	Тестування системи	27.04.2025	01.05.2025	виконано
7	Оцінка результатів	02.05.2025	03.05.2025	виконано
8	Підготовка висновків	04.05.2025	05.05.2025	виконано
9	Оформлення звіту	06.05.2025	24.05.2025	виконано
10	Завершення оформлення	25.05.2025	28.05.2025	виконано

Студент _____ Жолукевський Р. В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гринак А. В.

АНОТАЦІЯ

Метою роботи є дослідження та розробка методів раннього виявлення DDoS–атак на основі аналізу аномалій у мережевому трафіку в корпоративних мережах.

В роботі було розглянуто основні види DDoS–атак, їхню класифікацію, принципи здійснення та методи протидії. Проаналізовано сучасні підходи до моніторингу та виявлення аномальних активностей у мережевому трафіку. Описано методи аналізу, такі як статистичний, поведінковий та кореляційний, що дозволяють виявляти відхилення від нормальної роботи мережі.

Також було здійснено та обґрунтовано проектування концептуальної архітектури системи раннього виявлення DDoS–атак у корпоративній мережі. Описано основні компоненти системи, принципи збору, обробки та аналізу мережевого трафіку. Для підтвердження ефективності запропонованих рішень було проведено моделювання типових сценаріїв атак та визначено методи реагування на них.

Практична реалізація включає розробку прототипу системи виявлення аномалій, що здійснює моніторинг мережевого трафіку та ідентифікує потенційні загрози у режимі реального часу. В роботі представлено основні алгоритми обробки даних, описано дизайн та функціональні можливості системи, а також проведено тестування на тестовому середовищі.

Ключові слова: DDoS–атака, мережевий трафік, аналіз аномалій, корпоративна мережа, моніторинг, виявлення атак, фільтрування трафіку.

SUMMARY

The purpose of the work is to research and develop methods for early detection of DDoS attacks based on the analysis of anomalies in network traffic within corporate networks.

The study examines the main types of DDoS attacks, their classification, principles of implementation, and methods of counteraction. Modern approaches to monitoring and detecting anomalous activities in network traffic are analyzed. Methods of analysis, such as statistical, behavioral, and correlation, which allow detecting deviations from normal network operation, are described.

The conceptual architecture of an early detection system for DDoS attacks in a corporate network was designed and substantiated. The main components of the system, principles of data collection, processing, and analysis of network traffic are described. To confirm the effectiveness of the proposed solutions, typical attack scenarios were modeled, and response methods were determined.

The practical implementation includes the development of a prototype anomaly detection system that monitors network traffic and identifies potential threats in real–time. The work presents the main data processing algorithms, describes the design and functional capabilities of the system, and provides testing results in a simulated environment.

Keywords: DDoS attack, network traffic, anomaly analysis, corporate network, monitoring, attack detection, traffic filtering.

ЗМІСТ

ВСТУП

Сучасний розвиток інформаційних технологій забезпечив високий рівень автоматизації процесів управління, обробки та передачі даних у корпоративних мережах. Однак, з підвищенням залежності бізнес–процесів від мережевих ресурсів, зростає й ризик їх порушення внаслідок різноманітних кібератак. Однією з найбільш поширених та небезпечних форм таких атак є DDoS–атаки (Distributed Denial of Service), які спрямовані на виведення з ладу інформаційних ресурсів шляхом перенавантаження мережевого каналу або серверів величезною кількістю шкідливих запитів.

DDoS–атаки можуть призвести до повної недоступності веб–сайтів, сервісів та інформаційних систем, що своєю чергою викликає значні фінансові збитки, втрату довіри користувачів та пошкодження репутації компанії. В умовах сучасного цифрового світу питання захисту від таких атак набуває стратегічної важливості для будь–якої організації, яка використовує інтернет–технології.

Одним із найефективніших підходів для забезпечення безпеки корпоративних мереж є раннє виявлення DDoS–атак на основі аналізу аномалій мережевого трафіку. Своєчасне визначення відхилень у характері трафіку дозволяє мінімізувати вплив атак, оперативно реагувати на загрози та зберігати доступність критично важливих ресурсів.

Актуальність теми. Зростання кількості та складності DDoS–атак вимагає від підприємств нових підходів до забезпечення захисту інформаційних ресурсів. Традиційні методи виявлення атак, що базуються на сигнатурному аналізі, не завжди здатні протистояти сучасним високошвидкісним атакам, які постійно еволюціонують. У цьому контексті аналіз аномалій мережевого трафіку стає одним із найбільш перспективних напрямів у боротьбі з DdoS–атаками, дозволяючи виявляти підозрілу активність ще до того, як атака завдасть значної шкоди.

Мета і завдання дослідження. Метою роботи є дослідження та розробка методів раннього виявлення DDoS–атак у корпоративних мережах на основі

аналізу аномалій мережевого трафіку. Для досягнення цієї мети були поставлені наступні завдання:

провести аналіз видів та класифікацій DDoS–атак, а також їх впливу на інформаційні ресурси;

- розглянути сучасні методи моніторингу та аналізу мережевого трафіку;
- визначити ефективні підходи для виявлення аномалій у мережевому трафіку;
- розробити архітектуру системи раннього виявлення DDoS–атак;
- здійснити тестування запропонованих методів на тестовому середовищі.

Об'єкт і предмет дослідження. Об’єкт дослідження – процеси моніторингу та аналізу мережевого трафіку у корпоративних мережах.

Предмет дослідження – методи виявлення аномалій у мережевому трафіку для протидії DDoS–атакам.

Методи дослідження. Для вирішення поставлених завдань у роботі використано наступні методи:

- аналіз літературних джерел та сучасних досліджень у сфері кібербезпеки;
- методи статистичного аналізу для виявлення аномалій у мережевому трафіку;
- моніторинг мережевого трафіку для фіксації підозрілих змін у потоках даних.

Практична цінність роботи. Практична цінність роботи полягає у можливості використання розроблених методів для підвищення рівня захисту корпоративних мереж від DDoS–атак. Застосування аналізу аномалій дозволяє оперативно виявляти підозрілу активність, що забезпечує стабільність і безпеку інформаційних ресурсів підприємства..

1 АНАЛІЗ ТА ВИДИ DDOS–АТАК ТА ВЕБ–РЕСУРСІВ

1.1 Аналіз видів DDoS–атак

Атака типу «відмова в обслуговуванні» (DoS) – це спроба завдати шкоди, зробивши недоступною цільову систему для звичайних кінцевих користувачів, наприклад веб–сайт або веб–додаток. Зазвичай зловмисники генерують велику кількість пакетів або запитів, які зрештою перевантажують роботу цільової системи. Для здійснення атаки типу «розподілена відмова в обслуговуванні» (DDoS) зловмисник використовує безліч зламаних або контрольованих джерел[1].

На початку розвитку мережевих технологій, DDoS був інструментом тестування пропускної здатності. Але досить швидко зловмисники знайшли можливість використання даної атаки в своїх корисних цілях. Атака типу «відмова в обслуговуванні» є в наборі інструментів зловмисників уже протягом двадцяти років, і вона досі зростає в популярності і розвивається. Зараз даний тип атаки використовуються для отримання вигоди різними способами: шантаж, припинення функціонування основних вузлів мережі; припинення працездатності веб–сервісів; в бізнес або політичній конкуренції тощо. Вплив може варіюватися від незначного роздратування адміністратора мережі та незначних порушених сервісів до повного припинення функціонування цілих веб–сайтів, додатків або навіть цілого бізнесу, який працює в автономному режимі. Це у свою чергу може призвести до втрати репутації або значних фінансових втрат. Також даний тип атак може бути відволікаючим фактором, для здійснення складніших атак, для викрадення конференційної інформації.

Можна виділити основні мотиви зловмисників які використовують DDoS [2]:

Конкуренція. На даний момент досить популярною є послуга проведення DDoS на замовлення. Тобто, при конкуренції, фірма, яка не погоджується з конкурентами, просто формує завдання для хакера паралізувати систему, з якою

працюють конкуренти, або паралізувати роботу зовнішні або внутрішніх ресурсів конкурентоспроможної компанії. В результаті чого, організовується DDoS–атака на визначений термін і з визначеною силою[2].

Шахрайство. Хакери самостійно організовують DDoS–атаки з власної мережі заражених комп'ютерів, які дозволяють блокувати роботу невеликих та середніх системи. Якщо користувач не встановив захист від DDoS–атаки, то хакер може повною мірою паралізувати роботу системи, і шантажувати на певну суму для розблокування. Часто, звичайні користувачі, погоджуються на умови хакерів. Це обумовлене тим, що просто цінність даних вища, чим сума, вказана хакером[2].

Розвиток або забава. У зв'язку з тим, що в останній час все більше людей цікавлять DDoS–атаки, та, багато початківців хакерів здійснюють такі атаки просто ради забави. Отже, можна сказати, що основними цілями атаки є створення таких умов, при яких користувач з необхідними правами доступу, не має змоги отримати ресурси системи, або мають з цим трудності. Зі сторони мережі DDoS виглядає як різке збільшення трафіку[2].

В загалом виділяється дві методики, які найбільш часто використовуються зловмисниками[3]:

руйнівні атаки – призначені для того щоб сегмент мережі став цілком недоступним: зависає, знищується, видаляється операційна система. Цей тип атаки стає доступним через вразливості, які є в програмному забезпеченні[3];

атака на ресурси системи – в цій атаці велика кількість пустих або неправильних запитів відправляється на вузол мережі або веб–додатку, що тягне за собою значне зменшення пропускної здатності мережі або продуктивності системи та мережевого обладнання[3].

Постійний доступ до того чи іншого веб–сайту будь–якого напрямку є одним із найважливіших факторів його розвитку. Особливо це стосується інтернет магазинів та сайтів, які продають певні послуги за кошти в режимі онлайн[4].

Фактично кожна секунда роботи такого сервісу приносить кошти своєму власнику, а його недоступність навпаки – призводить до нереалізованих продаж або послуг, а також наносить шкоду у вигляді спаду репутації та зменшенні користувачів послугами у майбутньому[5].

Розподілена атака на відмову в обслуговуванні – це масштабна DDoSатака, в якій зловмисник використовує більше однієї унікальної IP–адреси. DDoS зазвичай включає більше 5 вузлів у різних мережах, менша кількість використаних вузлів у атаці може перекваліфікувати тип атаки у DoS–атаку. Стандартна атака DDoS виникає, коли зловмисники надсилають значну кількість неправильного мережевого трафіку безпосередньо до цільового сервера або мережі. Одним із способів, яким зловмисник може досягти цього, є використання ботнету для відправки трафіку [6]. Ботнет – це велика кількість комп'ютерів–жертв, підключених до мережі Інтернет, які спілкуються один з одним і можуть контролюватися з одного місця. Коли зловмисник
використовує ботнет для DDoS–атаки, вони відправляють інструкції до частини або всіх інфікованих машин, підключених до цієї бот–мережі, тим самим збільшуючи розмір атаки, а джерелом стають відразу кілька мереж і трафік надходить, можливо, з декількох країн[6].

Основними мотивами розподілених атака на відмову в обслуговуванні є замовлення від осіб, які переслідують особисті цілі, зокрема нанесення фінансових збитків та дискредитації сервісу надання послуг, а також помста та активізм[7].

Вектори атаки посилення є одними з найбільш часто використовуваних інструментів в арсеналі зловмисника DDoS. В останньому кварталі 2017 року ми побачили, що підсилення NTP використовувалося приблизно на 33% всіх DDoS нападів на наших клієнтів, в той час як DNS і SSDP підсилювали вектори відігравали роль у 17% і 13,7% атак, відповідно[7]. Для зловмисників, вектори посилення пропонують можливість запуску навантажень, занадто великих для пропускної здатності, без необхідності настільки ж великих ресурсів ботнету. З

точки зору пом'якшення, вони являють собою меншу загрозу, оскільки, на сьогоднішній день, більшість послуг з ліквідації наслідків позбулись пункту, де пропускна здатність атаки є головним завданням, або будь–якою іншою проблемою. Більш важливим є те, що заголовки вихідних портів корисних навантажень підсилення слідують передбачуваному шаблону, що полегшує їх фільтрування на межі мережі[8]. Наприклад, блокування всіх пакетів з вихідним портом 53 вважається перевіреним методом для пом'якшення атак підсилення DNS. Нещодавно, пом'якшуючи атаку з посиленням SSDP, ми бачили докази корисного навантаження з нерегулярними даними порту джерела, що мало хто в нашій галузі вважає можливим[8].

Різні типи DDoS–атак спрямовані на різні компоненти мережевого підключення. Щоб зрозуміти, як працюють різні атаки DDoS, необхідно знати, як здійснюється мережеве з'єднання. Мережне з'єднання в Інтернеті складається з багатьох різних компонентів або «шарів». Як будувати будинок з нуля, кожен шар у моделі має різне призначення[9].

Проведемо аналіз видів DDoS–атак, які відносяться до моделі OSI, показану на рисунку 1.1, яка є концептуальною структурою, яка використовується для опису мережевого підключення на 7 різних рівнях[10].

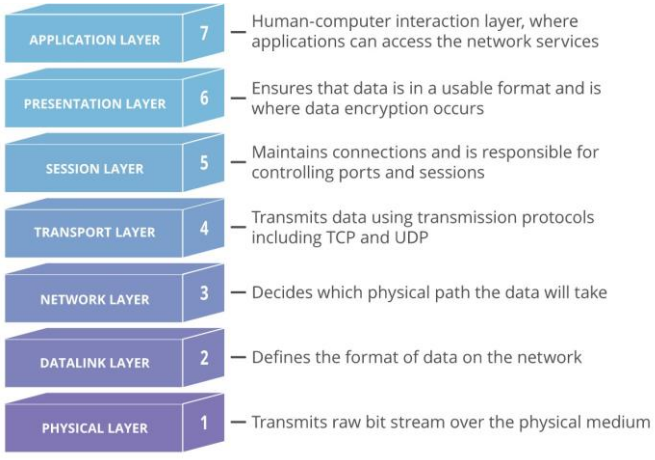


Рисунок 1.1 – модель OSI

Хоча майже всі атаки DDoS передбачають перевантаження цільового пристрою або мережі трафіком, атаки можна розділити на три категорії.

Зловмисник може використовувати один або кілька різних векторів атаки або циклічні вектори атаки у відповідь на контрзаходи, які вживає ціль[11].

Атаки прикладного рівня(Application layer attacks)**.** При розгляді даної атаки, мета даної DDoS–атаки 7–го рівня (рис. 1.1) – вичерпати ресурси цілі, щоб створити відмову в обслуговуванні. Атаки спрямовані на рівень, де вебсторінки генеруються на сервері та доставляють у відповідь на запити HTTP. Виконання одного HTTP–запиту на стороні клієнта з точки зору обчислень дешеве, але відповідати на нього цільовому серверу може бути дорого, оскільки сервер часто завантажує кілька файлів і виконує запити до бази даних, щоб створити веб–сторінку (рис. 1.2)[12].

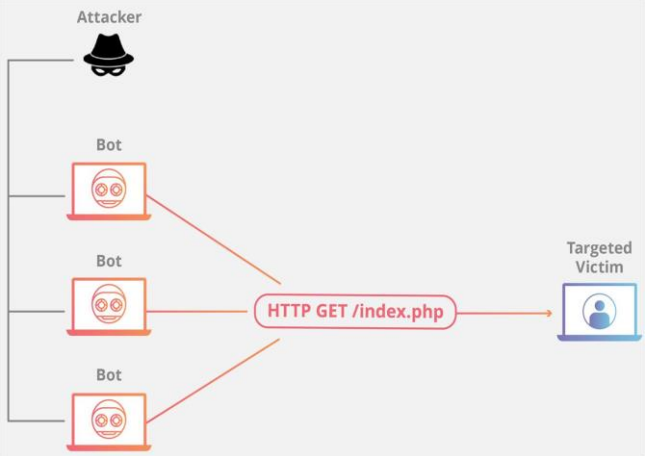


Рисунок 1.2 – Атаки прикладного рівня

Від атак рівня 7 важко захистити, оскільки важко відрізнити зловмисний трафік від законного.

HTTP flood. При аналізі даної атаки, можна, зробити висновок, що це атака схожа на повторне натискання оновлення у веб–браузері на багатьох різних комп’ютерах одночасно – велика кількість HTTP–запитів переповнює сервер, що призводить до відмови в обслуговуванні[12].

Цей тип атаки варіюється від простих до складних.

Простіші реалізації можуть отримати доступ до однієї URL–адреси з однаковим діапазоном атакуючих IP–адрес, перенаправлень і агентів користувача. Складні версії можуть використовувати велику кількість атакуючих IP–адрес і

націлювати випадкові URL–адреси за допомогою випадкових посилань і користувацьких агентів[12].

Протокольні атаки (Protocol attacks). Протокольна атака має таку мету, як – атаки протоколу, також відомі як атаки з вичерпанням стану, викликають порушення роботи служби через надмірне споживання ресурсів сервера та/або ресурсів мережевого обладнання, таких як брандмауери та балансувальними навантаженнями (рис 1.3)[13].

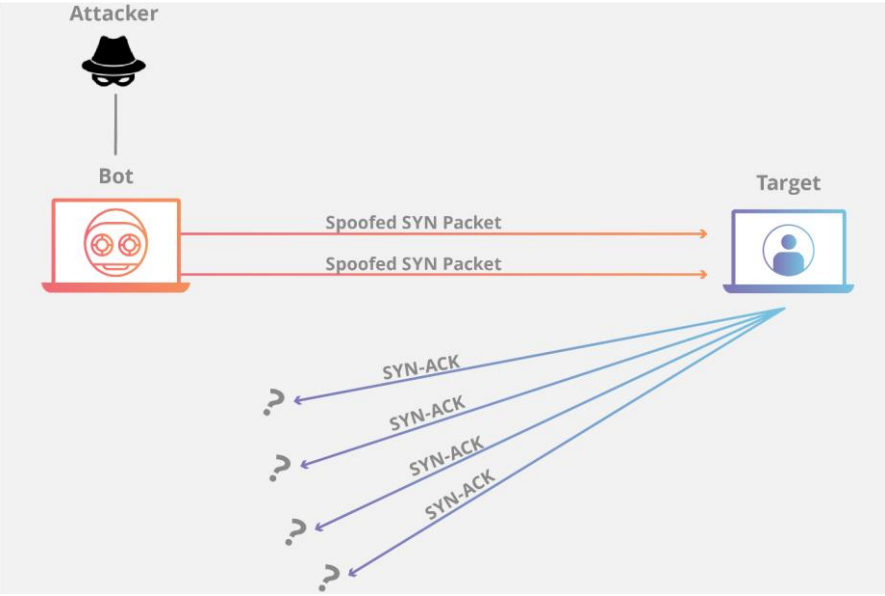


Рисунок 1.3 – Приклад протокольної атаки

Протокольні атаки використовують слабкі місця в рівнях 3 і 4 стеку протоколів, щоб зробити ціль недоступною.

SYN flood (повінь). Пояснити мету даної атаки можливо висловом «Працівник отримує запит, йде і отримує посилку і чекає підтвердження, перш ніж принести посилку. Потім працівник отримує багато запитів на посилку без підтвердження, поки він не зможе перевозити більше пакетів, не буде перевантажений, а запити не почнуть залишатися без відповіді»[13].

Тобто, Ця атака використовує рукостискання TCP – послідовність комунікацій, за допомогою яких два комп’ютери ініціюють мережеве з’єднання – шляхом надсилання цілі великої кількості SYN–пакетів TCP «Початковий запит на

підключення» з підробленими вихідними IP–адресами. Цільова машина відповідає на кожен запит на з'єднання, а потім чекає останнього кроку в рукостисканні, яке ніколи не відбувається, вичерпуючи ресурси цілі в процесі. Об'ємні напади (Volumetric attacks). Мета атаки – це категорія атак намагається створити перевантаження, споживаючи всю доступну пропускну спроможність між цільовим елементом і більшим Інтернетом. Великі обсяги даних надсилаються до цілі за допомогою форми посилення або іншого засобу створення масивного трафіку, наприклад запитів від ботнету (рис. 1.4)[14].

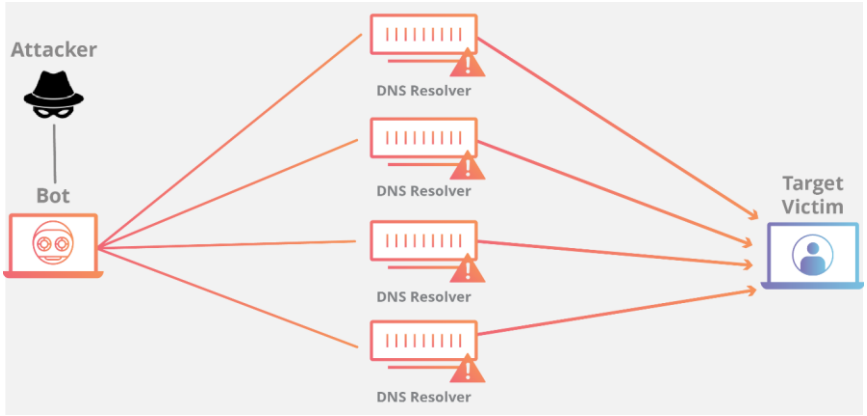


Рисунок 1.4 – Приклад посилення

Здійснюючи запит до відкритого DNS–сервера з підробленою IP–адресою (IP–адресою жертви), цільова IP–адреса отримує відповідь від сервера[15]. \

У DDoS–атаці не завжди скомпрометовані машини відправляють трафік жертві атаки. Сервери, що працюють на основі UDP–служб, часто використовуються зловмисниками для здійснення масових DDoS–атак. Такі сервери використовуються зловмисником як рефлектори. Виходячи з характеру атакуючих машин, DDoS–атаки поділяються на дві категорії, а саме: прямі і рефлекторні. У прямій атаці, яку зображено на рисунку 1.5, зловмисник використовує скомпрометовані машини безпосередньо для запуску DDoS–атак різних типів (рис.1.5) [16].

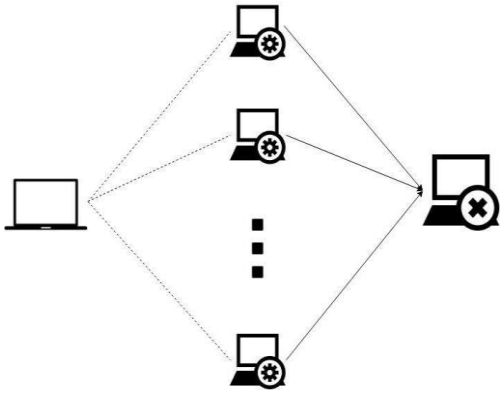


Рисунок 1.5 – Пряма DDoS атака

У той час як від рефлекторної атаки або посилення зображеної на рисунку 1.6, багато невинних проміжних вузлів, використовуються для генерування атаки. Зловмисник посилає запити на рефлекторні сервери, підмінивши IPадресу джерела, як якщо б він був IP–адресом жертви[17]. Як результат, ці сервери відповідають потерпілому, посилаючи повідомлення, обсяг яких зазвичай у багато разів перевищує початковий розмір повідомлення запиту. Отже, цей тип DDoS–атаки також називається атакою з підсиленням. Зловмисник використовує цю техніку, щоб посилити трафік атаки до декількох сотень разів. Підсилення атаки за допомогою DNS і NTP є прикладами DDoSатак на основі відображення(рис. 1.6)[18].

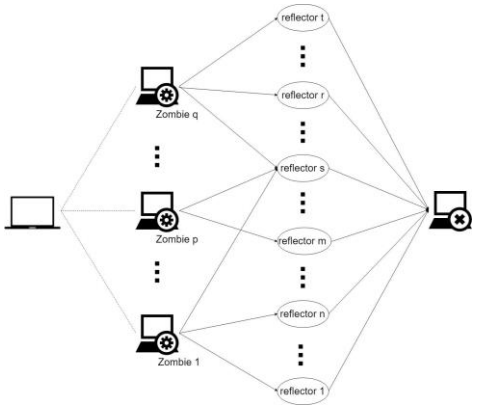


Рисунок 1.6 – Рефлекторна DDoS атака

У традиційному виконанні (один нападник – одна жертва) зараз залишається ефективним лише перший вид атак. Класичний флуд – марний. Просто тому що при сьогоднішній ширині каналу серверів, рівні обчислювальних потужностей і

повсюдному використанні різних анти–DoS прийомів в ПЗ (наприклад, затримки при багаторазовому виконанні тих самих дій одним клієнтом), нападник перетворюється на докучливого комара, не здатного завдати будь–якого збитку[19]. Але якщо цих «комарів» наберуться сотні, тисячі або навіть сотні тисяч, вони легко покладуть сервер на лопатки. Розподілена атака типу «відмова в обслуговуванні» (DDoS), зазвичай здійснювана за допомогою безлічі «зазомбованих» хостів, може відрізати від зовнішнього світу навіть найстійкіший сервер, і єдиним ефективним захистом при цьому є організація розподіленої системи серверів (кластера)[20].

1.2 Аналіз роботи DDoS атаки

Злом WiFi застосовується для забору пароля бездротової мережі. Атаки в формі "" дозволять слухати інтернет–трафік. Аналіз вразливостей з подальшою підвантаженням конкретного дає можливість захоплення цільового комп'ютера. А що ж робить DDOS атака? Її мета в кінцевому підсумку – відбір прав на володіння ресурсом у законного господаря. Я не маю на увазі, що сайт або блог вам не буде належати. Це в тому сенсі, що в разі вдало проведеної атаки на ваш сайт, ви втратите можливість їм управління. По крайней мере, на деякий час[21].

Для початку, мабуть, варто розібратися, що являють собою такі неправомірні дії. Зазначимося відразу, що при розгляді теми «DDoS–атака: як зробити самому» інформація буде подана виключно для ознайомлення, а не для практичного використання. Всі дії такого роду кримінальні.

Сама ж атака, за великим рахунком, є відсиланням достатньо великої кількості запитів на сервер або сайт, які з перевищенням ліміту звернень блокують роботу веб–ресурсу або служби провайдера у вигляді відключення сервера захисним ПЗ, міжмережевими екранами або спеціалізованим обладнанням[22].

Зрозуміло, що DDoS–атака своїми руками може бути створена одним користувачем з одного комп'ютерного термінала без спеціальних програм. Зрештою, ну не буде ж він сидіти цілодобово і щохвилини посилати запити на атакований сайт. Такий номер не пройде, оскільки захист від DDoS–атак передбачений у кожного провайдера, а один користувач не в змозі забезпечити таку кількість запитів на сервер або сайт, який за короткий час перевищив ліміт звернень і спричинив спрацювання різних захисних механізмів. Тож для створення власної атаки доведеться використати дещо інше[23].

Якщо розбиратися, що таке DDoS–атака, як зробити її та надіслати перевищену кількість запитів на сервер, варто врахувати й механізми, за якими такі дії здійснюються(рис 1.7)[24].

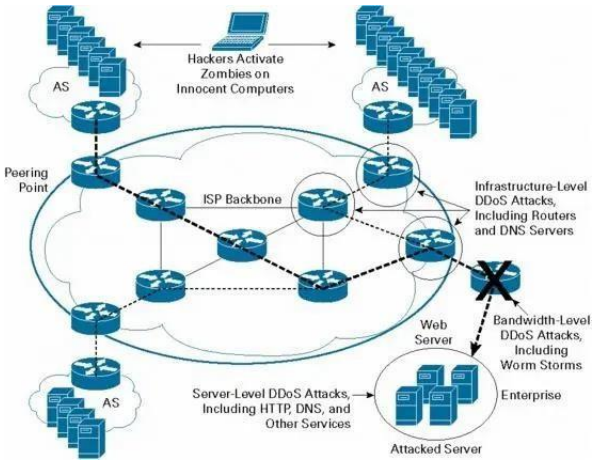


Рисунок 1.7 – Механізм DDoS атаки

Це можуть бути ненадійні не здатні справлятися з величезною кількістю запитів, проломи в системі безпеки провайдера або в самих «операційках», брак системних ресурсів для обробки запитів з подальшим зависанням системи або аварійним завершенням роботи і т.д[25].

На зорі виникнення такого явища в основному DDoS–атака своїми руками здійснювалася переважно самими програмістами, які створювали та тестували за її допомогою працездатність систем захисту. До речі, свого часу від дій зловмисників, які застосовували як зброю компоненти DoS і DDoS, постраждали

навіть такі IT–гіганти, як Yahoo, Microsoft, eBay, CNN та багато інших. Ключовим моментом у тих ситуаціях стали спроби усунення конкурентів щодо обмеження доступу до їх інтернет–ресурсів[26].

Як правило, така атака пов'язана з конкретним хостингом і спрямована виключно на заздалегідь заданий веб–ресурс (у прикладі на фото нижче умовно позначений як example.com)(рис 1.8).[27]

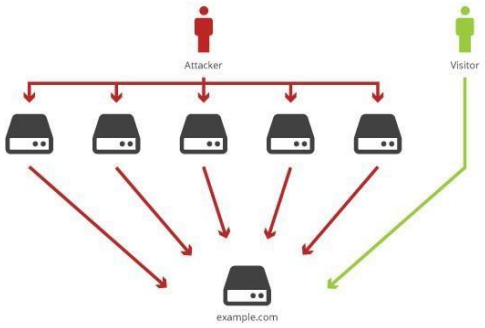


Рисунок 1.8 – Атака на веб–ресурс

При надто великій кількості звернень до сайту порушення зв'язку відбувається через блокування зв'язку не самим сайтом, а серверною частиною провайдерської служби, вірніше, навіть не самим сервером або системою захисту, а службою підтримки. Іншими словами, такі атаки спрямовані на те, щоби власник хостингу отримав від провайдера відмову в обслуговуванні при перевищенні певного контрактного ліміту трафіку[28].

Програма дозволяє здійснювати розподілену атаку, одночасно відкриваючи багато підключень до сервера і тим самим намагаючись "покласти" його – домогтися коду помилки, що свідчить про "перемогу". У процесі DDoS LOIC відображається інформація про статус активних підключень, яка змінюється в режимі реального часу. Крім того, користувачеві повідомляють про код, який був повернутий сервером під час спроби здійснити стандартне підключення[29].

Low Orbit Ion Cannon діє цілком стандартними для таких інструментів методами. Зокрема, програма вміє "спамити" HTTP–запитами, домагаючись

відмови обладнання. Також програма може здійснювати "флуд" UDP та TCP пакетами. Крім стандартної версії LOIC, існує також версія, здатна отримувати "мети" атаки по RSS–стрічок та месенджерів. Цей варіант програми може бути корисним творцям ботнетів[30].

1.3 Аналіз існуючих методів здійснення DDOS–атак

DDoS–атаки є серйозною загрозою для сучасних компаній та інтернетсервісів, які можуть стати жертвами масштабних та складних атак. Існує безліч різних методів та технологій, які можуть використовуватися для здійснення DDoS–атак, і далі ми розглянемо деякі з них.

UDP–флуд–атаки: ці атаки базуються на відправленні великої кількості UDP–пакетів до цільового сервера. UDP (User Datagram Protocol) – протокол без забезпечення доставки пакетів, що означає, що не всі пакети будуть доставлені до призначення. Це дозволяє здійснювати атаки з великою швидкістю та без обмежень на розмір пакетів[31].

SYN–флуд–атаки: ці атаки використовуються для перевантаження мережевих ресурсів, особливо тих, що використовують TCP (Transmission Control Protocol). Атака полягає у відправленні великої кількості SYN–запитів (запит на підключення) до цільового сервера, але не завершення їх.

DNS–ампліфікаційні атаки: це атаки, які використовують відповіді DNScерверів для збільшення обсягу трафіку, який надсилається до цільового сервера. Атакувач відправляє запити до DNS–серверів, що містять великий обсяг даних, та вказує адресу цільового сервера як джерело запитів. Відповіді DNS–серверів збільшують обсяг трафіку, який надсилається до цільового сервера, що може призвести до його перевантаження[32].

Напад на захист проти DDOS–атак: це атаки, які спрямовані на перевантаження захисного обладнання, яке використовується для захисту від DDOS–атак. Атакувачі можуть відправляти велику кількість запитів, які містять

невірну інформацію або які вимагають значного обчислювального потужності, що може спричинити перевантаження захисного обладнання.

Словникові атаки: це атаки, які базуються на відправленні великої кількості запитів з використанням зарані встановлених логінів та паролів для аутентифікації. Атакувачі можуть використовувати списки логінів та паролів, які були скомпрометовані в попередніх витоках даних або що легко вгадуються[33].

HTTP–флуд–атаки: це атаки, які використовуються для перевантаження веб–серверів за допомогою відправки великої кількості запитів на вебсторінки. Атаки можуть бути спрямовані на конкретні веб–сторінки або на всі сайти, які розміщені на сервері.

SSL–атаки: ці атаки використовуються для перехоплення захищеного трафіку, що передається через SSL (Secure Sockets Layer) або TLS (Transport Layer Security). Атакувач може використовувати спеціальний програмне забезпечення, що здатне розшифровувати SSL–з'єднання та отримувати доступ до переданих даних. SSL–атаки можуть бути виконані за допомогою підроблення сертифікатів або використання вразливостей в криптографічних протоколах[34].

ICMP–атаки: ці атаки використовуються для перевантаження мережі, що забезпечує зв'язок між різними пристроями. Атаки використовують протокол ICMP для відправки великої кількості повідомлень до різних пристроїв в мережі, що може призвести до перевантаження мережевих пристроїв.

ACK–атаки: ці атаки використовуються для перевантаження мережі, що забезпечує зв'язок між різними пристроями. Атакувач відправляє велику кількість ACK–запитів до мережевих пристроїв, що може призвести до перевантаження мережевих ресурсів та непридатності їх для роботи.

UDP–атаки: ці атаки використовуються для перевантаження мережі за допомогою відправки великої кількості UDP–запитів. У порівнянні з TCPпротоколом, UDP не має механізмів контролю трафіку, що може призвести до перевантаження мережевих ресурсів та непридатності їх для роботи.

Slowloris є ще одним методом DDOS–атак, який використовується для здійснення атак на веб–сайти. У цьому випадку атакувачі намагаються зайняти всі

доступні з'єднання з сервером, відправляючи запити, але не закінчуючи їх повністю. Це призводить до того, що сервер не може обробляти запити від інших користувачів, що намагаються звернутися до нього[35].

Об'ємні атаки. Ця категорія атак намагається створити перевантаження, споживаючи всю доступну пропускну здатність між ціллю та більшим Інтернетом. Великі обсяги даних надсилаються до цілі за допомогою форми посилення або іншого засобу створення масивного трафіку, наприклад запитів із ботнету.

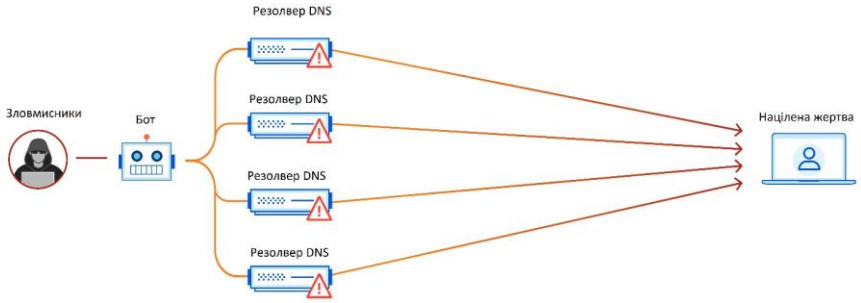


Рисунок 1.9 – Приклад посилення DDoS–атак

Загалом, DDOS–атаки можуть бути виконані за допомогою різних методів, що використовують вразливості різних протоколів та служб. Для захисту від таких атак потрібно використовувати комплексні заходи, такі як використання фільтрів, розробка спеціальних алгоритмів виявлення атак та їх блокування, налагодження мережевої безпеки, зменшення публічного доступу до сервісів та серверів та т.д[36].

Окрім того, важливим аспектом при захисті від DDOS–атак є моніторинг мережі та систем. За допомогою моніторингу можна вчасно виявляти незвичайну активність в мережі, виявляти аномальний трафік та швидко реагувати на можливі загрози.

Один зі способів захисту від DDOS–атак – це захист на рівні мережі. Це включає в себе використання спеціальних мережевих пристроїв, таких як брандмауери та веб–фільтри. Такі пристрої можуть бути налаштовані для блокування небажаного трафіку до сервісів та серверів.

Ще один спосіб захисту від DDOS–атак – це захист на рівні додатків. Це включає в себе захист від атак на веб–сервери, які базуються на використанні вразливостей у програмному забезпеченні. За допомогою захисту на рівні додатків можна використовувати різноманітні методи, такі як обмеження швидкості запитів, перевірка поведінки користувачів та їх авторизація.

Також до методів захисту від DDOS–атак можна віднести тимчасове зупинення атакованого сервісу або перенесення його на інші сервери, що мають більші ресурси для опрацювання запитів. Такі методи можуть допомогти зменшити негативний вплив DDOS–атаки на роботу сервісу.

Одним із сучасних методів атак є так звані «розподілені DDOS–атаки». Ці атаки зазвичай виконуються за допомогою ботнетів (botnet), які складаються з великої кількості комп'ютерів, що заражені шкідливим програмним забезпеченням. Існують різні типи ботнетів, включаючи HTTPботнети, DNS–ботнети та P2P–ботнети.

HTTP–ботнети використовуються для здійснення DDOS–атак на вебсайти. Вони працюють шляхом надсилання великої кількості запитів до серверів, що призводить до перевантаження мережі та ресурсів серверів. DNSботнети використовуються для здійснення атак на DNS–сервери, що може призвести до недоступності веб–сайтів та інших сервісів, що залежать від DNS. P2P–ботнети використовуються для здійснення DDOS–атак на різні мережеві пристрої, включаючи маршрутизатори та файрволи.

Усі ці методи та засоби захисту можуть бути ефективними лише у випадку їх правильного застосування та налаштування. Для цього важливо мати якісні знання в галузі мережевої безпеки та розробки програмного забезпечення. Крім того, необхідно постійно вдосконалювати методи захисту від DDOS–атак, оскільки атакувачі постійно вдосконалюють свої техніки та методи[37].

1.4 Аналіз керування розподіленими веб–ресурсами

Веб–ресурс – це інформація, яка доступна для завантаження з всесвітньої мережі Інтернет за допомогою об’єднаних апаратних та програмних засобів, та

яка унікально ідентифікується за допомогою уніфікованого локатора ресурсу (URL). Веб– ресурсом є текст, мультимедіа, графіка, веб–сторінки, документи тощо.

Веб–ресурси передаються через мережу згідно до правил протоколу HTTP, специфікаціями якого є: HTTP/1.0, HTTP/1.1, HTTP/2.0. Специфікації повноцінно описані документами RFC1945, RFC7231, RFC7540 .

Методологія обробки даних сучасних інформаційних систем заснована на принципі багаторівневості. Представлена на рис. 1.9 схема відображає рівні обробки інформації в інформаційній системі[38].

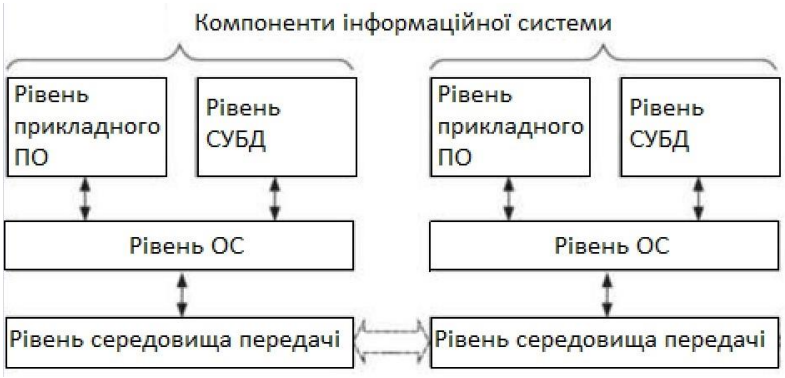


Рис.1.10 Схема передачі інформації в інформаційній системі

Керування розподіленими веб–ресурсами передбачає аналіз інформації на рівні мережевого оточення операційної системи (ОС). При цьому сама система керування може використовувати техніки та технології з рівня прикладного програмного забезпечення (ПЗ), наприклад – СУБД.

Мета керування розподіленими веб–ресурсами, що розглядається у рамках дипломної роботи, – здійснення керуючих впливів на процес обміну веб–ресурсами для блокування визначених веб–ресурсів, виявлення інтернетзагроз у веб–трафіку та постійного моніторингу HTTP–з’єднань.

Розроблювана система керування доступом до розподілених веб–ресурсів є інформаційною системою (рис. 1.10)[39].

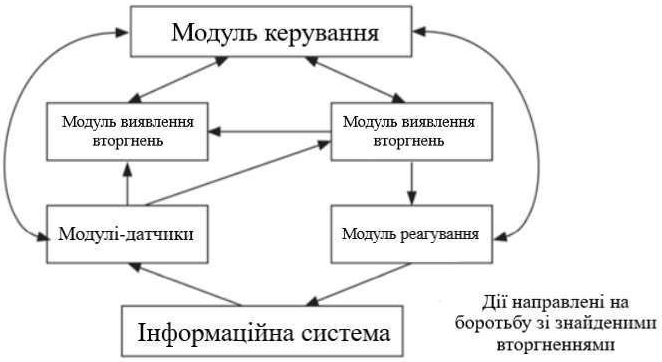


Рисунок 1.11 – Модель інформаційної системи

Зв’язки у системі будемо формувати з огляду на архітектурну модель інформаційної системи запропонованою автором Шелухіним[36]

Проксі–сервер діє як шлюз між користувачем та Інтернетом. Це посередницький сервер, який відокремлює кінцевих користувачів від вебресурсів. Проксі–сервери забезпечують різний рівень функціональності, безпеки та конфіденційності[40].

Функціонал сучасних проксі–серверів набагато ширший, ніж виконання звичайних пересилань запитів. Проксі–сервери діють як брандмауер та вебфільтр, забезпечують спільні мережеві підключення та кеш–дані для прискорення виконання запитів. Деякі проксі–сервери захищають користувачів та внутрішню мережу від мережевих вірусів, а також займаються питаннями конфіденційності.[41].

Коли проксі–сервер отримує запит на веб–ресурс (наприклад, вебсторінку), то він переглядає свій локальний кеш попередніх сторінок. Якщо сторінка знаходиться, то вона повертається користувачеві без пересилання запиту в Інтернет. Якщо сторінки немає в кеш–пам'яті, проксі–сервер, виступаючи клієнтом від імені користувача, використовує одну зі своїх власних IP–адрес для запиту сторінки (рис. 1.11).

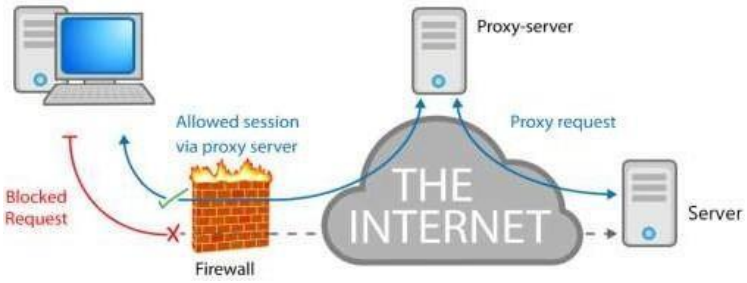


Рисунок 1.12 – Схема роботи проксі–сервера

Проксі–сервер працює абсолютно прозоро для користувача; всі Інтернетзапити та відповіді, що повертаються, безпосередньо стосуються адресованого Інтернет– сервера.

Користувачі можуть отримати доступ до веб–проксі через Інтернет або налаштувати веб–браузер для постійного використання проксі–сервера. Налаштування браузера включають автоматично виявлені та ручні параметри для проксі HTTP, SSL, FTP та SOCKS. Проксі–сервери можуть одночасно обслуговувати багатьох користувачів, або ж лише одного. Ці параметри відповідно називаються спільними та виділеними проксі[25].

Залежно від налаштування мережі та конфігурації існують наступні типи проксі–серверів[32]:

Прямий проксі–сервери зазвичай використовуються внутрішніми мережами. Як тільки один із клієнтів надсилає запит на підключення до певного веб–сайту, він спочатку повинен пройти через переадресований проксісервер, який вирішує, чи дозволено клієнту звертатися до цього ресурсу. Якщо так, запит на підключення надходить на зовнішній сервер, який не бачить IPадресу клієнта, а бачить лише запит на підключення, надісланий із переадресованого проксі–сервера. Прямий проксі– сервер забезпечує повний адміністративний контроль над підключеннями до локальної мережі. Він діє як брандмауер, що дозволяє адміністраторам обмежувати доступ до небажаних веб–ресурсів клієнтами внутрішньої мережі[28–29].

На відміну від прямого проксі–серверу, зворотній проксі–сервер працює на рівні вище веб–сайту (або веб–служби), приховуючи IP–адреси у внутрішній мережі від зовнішніх користувачів. Зворотний проксі–сервер вирішує, чи можуть веб– користувачі бачити вміст веб–сайту або використовувати вебслужбу, чи ні. Зворотні проксі–сервера ускладнюють хакерам вторгнення на внутрішні ресурси. Крім того, зазвичай вони працюють як балансувачі навантаження для рівномірного розповсюдження даних між внутрішніми серверами, що запобігає перевантаженню тонами запитів на підключення.

Зворотні проксі використовуються для:

- Непрямого доступу;
- Забезпечення балансування навантаження між серверами;
- Потокової передачі контенту;
- Заборони доступу до веб–сайтівю.

Прозорі проксі зазвичай знаходяться біля шлюзу корпоративної мережі. Ці проксі централізують мережевий трафік. У корпоративній мережі проксісервер пов’язаний – або є частиною – шлюзового сервера, який відокремлює внутрішню мережу від зовнішніх, і брандмауер, який захищає мережу від вторгнень ззовні та дозволяє сканувати дані на безпечність перед доставкою клієнту в мережі. Ці проксі– сервери допомагають здійснювати моніторинг та адміністрування мережевого трафіку, оскільки комп’ютери в корпоративній мережі, як правило, є безпечними пристроями, які не потребують анонімності для звичайних звичайних завдань[39].

Керування доступом до розподілених веб–ресурсів забезпечує ефективний контроль за HTTP–трафіком і підвищує рівень захисту від інтернет–загроз. Використання проксі–серверів і вебфільтрів дозволяє своєчасно виявляти, блокувати або перенаправляти небажані запити, зберігаючи стабільність і безпеку системи.

1.5 Висновки до розділу 1

У результаті проведеного аналізу було досліджено природу, класифікацію, методи реалізації та способи захисту від DDoS–атак. Встановлено, що атаки типу «розподілена відмова в обслуговуванні» є одними з найпоширеніших кіберзагроз сучасності, які можуть завдати істотної шкоди як технічним системам, так і бізнесу загалом. Вони реалізуються шляхом надсилання великої кількості запитів із багатьох джерел, що призводить до перевантаження інфраструктури й унеможливлює доступ до ресурсу для легітимних користувачів.

Розглянуто основні види атак згідно з моделлю OSI, включно з атаками прикладного рівня (HTTP flood), протокольними атаками (SYN flood), об’ємними атаками та атаками з підсиленням. Зокрема, було проаналізовано механізми роботи прямих і рефлекторних атак, використання ботнетів, а також приклади інструментів для здійснення таких атак, як LOIC.

Також було детально розглянуто етапи підготовки та реалізації DDoS–атак, включаючи вибір та компрометацію агентів, встановлення зв’язку між ними й безпосередній запуск атаки. Окрема увага приділена методам захисту, які поділяються на превентивні, реакційні, активні та пасивні. До ефективних заходів захисту віднесено фільтрацію трафіку, застосування проксі–серверів, веб–фільтрів, моніторинг, побудову розподілених систем та застосування спеціалізованого програмного забезпечення. Важливим є також розмежування легітимного та шкідливого трафіку, що становить одне з найскладніших завдань у боротьбі з DDoS.

На основі проведеного аналізу сформульовано постановку задачі дослідження:

- розробити систему раннього виявлення DDoS–атак у корпоративному середовищі, що базується на виявленні аномалій у мережевому трафіку;
- побудувати ефективну архітектуру моделі виявлення на основі методів машинного навчання;
- забезпечити здатність моделі ідентифікувати широкий спектр атак з мінімальною кількістю хибнопозитивних спрацювань;

- протестувати розроблену систему на реальних або наближених до реальності даних для підтвердження її ефективності.

Таким чином, успішна протидія DDoS–атакам потребує комплексного підходу, що охоплює як технічні, так і організаційні рішення, адаптивність до нових векторів атак і постійне вдосконалення систем безпеки.

2 ПРОЕКТУВАННЯ СИСТЕМИ РАННЬОГО ВИЯВЛЕННЯ DDOS–АТАК НА ОСНОВІ АНАЛІЗУ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ У КОРПОРАТИВНИХ МЕРЕЖАХ

У цьому розділі розглядається концептуальна основа, необхідна для побудови інтелектуальної системи раннього виявлення DDoS–атак на основі аналізу аномального мережевого трафіку в корпоративних мережах. З метою досягнення поставленої у роботі мети, буде здійснено проєктування архітектури майбутньої моделі, проведено обґрунтування вибору методів обробки трафіку та методів глибокого навчання, а також визначено відповідне середовище для реалізації. Розробка моделі передбачає багатоетапну процедуру, яка охоплює збір і підготовку вхідних даних, формування структури моделі, навчання, валідацію та оцінку ефективності.

2.1 Розробка алгоритму виявлення DDoS-атак

Проєктування системи раннього виявлення DDoS–атак вимагає чіткого розуміння послідовності дій, необхідних для обробки мережевого трафіку, побудови моделі та її інтеграції у реальне середовище. Цей підрозділ описує ключові етапи роботи алгоритму, який забезпечує виявлення аномальної активності в мережі.

Запропонований алгоритм базується на методах машинного навчання, зокрема – глибокому навчанню з використанням рекурентних нейронних мереж типу LSTM. Послідовність кроків спроєктована таким чином, щоб забезпечити максимальну точність класифікації при збереженні високої продуктивності моделі у середовищі з великим обсягом трафіку.

Крок 1. Робота алгоритму розпочинається з імпорту необхідних бібліотек, що забезпечують функціональність для обробки, візуалізації та моделювання даних. Це включає інструменти для аналізу табличних структур, виконання математичних операцій, побудови нейронних мереж та масштабування числових ознак.

Забезпечення доступності цих бібліотек є критично важливим для стабільного функціонування подальших етапів проєктування системи.

Крок 2. Після ініціалізації середовища виконується завантаження датасету, який містить реальні або згенеровані зразки мережевого трафіку. У межах цієї роботи використовується набір CSE–CIC–IDS2018, що містить мічені приклади як нормальної активності, так і DDoS–атак. Дані завантажуються у форматі CSV або Parquet і зчитуються у структуровану форму для подальшої обробки.

Крок 3. Після завантаження здійснюється первинний аналіз структури даних: визначається кількість ознак, аналізуються типи змінних, проводиться візуалізація розподілу значень, а також перевірка на наявність пропущених або аномальних значень. Це дозволяє виявити потенційні проблеми, які можуть вплинути на якість навчання моделі.

Крок 4. На етапі очищення даних видаляються всі неінформативні або дубльовані колонки. Особливу увагу приділяють полям з однаковими значеннями, службовим полям (наприклад, міткам часу, якщо вони не використовуються як ознака) та тим, що мають велику кількість відсутніх значень. За необхідності, окремі пропуски можуть бути замінені статистичними значеннями – середнім, медіаною або модою залежно від розподілу.

Крок 5. Після очищення виконується кодування всіх категоріальних ознак. Текстові значення, які не можуть бути безпосередньо оброблені нейронною мережею, перетворюються у числові – наприклад, шляхом one–hot або label encoding. Це дозволяє моделі працювати з усіма полями в єдиному числовому форматі.

Крок 6. Завершальним етапом обробки є нормалізація всіх числових ознак до уніфікованого діапазону, зазвичай [0;1]. Це дозволяє уникнути домінування однієї ознаки над іншими та сприяє стабільному сходженню функції втрат при навчанні. Для цього застосовуються методи мін–макс масштабування або стандартного нормування.

Крок 7. Отримані оброблені дані розбиваються на тренувальну та тестову вибірки. Важливо забезпечити репрезентативність обох підвибірок та зберегти

відсоткове співвідношення нормальних і аномальних прикладів. Також на цьому етапі дані формуються у вигляді часових вікон (послідовностей), які будуть подаватися на вхід нейронної мережі типу LSTM. Блок–схема роботи алгоритму представлена на рис. 2.1.

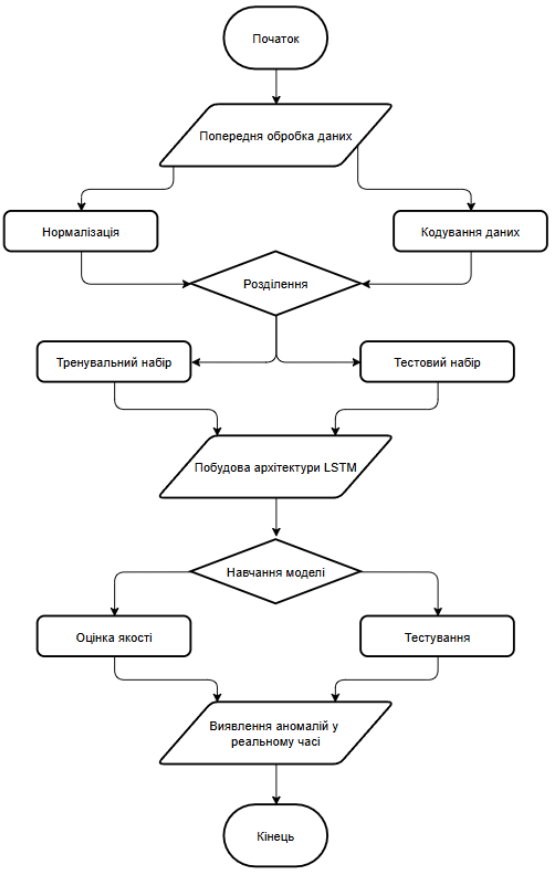


Рисунок 2.1 – Блок схема роботи алгоритму

Ці кроки формують повноцінну підготовчу основу для навчання моделі глибокого навчання, що забезпечує точне та надійне виявлення аномалій у мережевому трафіку.

2.2 Розробка архітектури системи

Розробка системи раннього виявлення DDoS–атак ґрунтувалася на створенні прикладного рішення, яке не обмежується лише навчанням моделі. Було реалізовано повноцінну систему, яка виконує всі етапи – від збору мережевого трафіку до автоматичного формування рішень про виявлення аномалій. При цьому особлива увага приділялася адаптації системи до специфіки корпоративних мереж,

де обсяг трафіку значно вищий, архітектура складніша, а вимоги до стабільності та безпеки – жорсткіші.

На концептуальному рівні система виконує безперервний моніторинг мережевого трафіку, обробляє отримані дані в реальному часі, формує послідовності характеристик трафіку, проводить аналіз цих послідовностей за допомогою попередньо навченої LSTM–моделі та визначає, чи містять вони ознаки аномального стану. Якщо виявляється підозріла активність, система автоматично генерує повідомлення, яке може бути використане для подальших дій адміністратора, або активує механізми блокування.

В основі реалізації лежить модуль, який перехоплює мережевий трафік з інтерфейсу мережевого адаптера. Для цього використовуються інструменти, здатні працювати в режимі реального часу, зокрема бібліотека scapy, яка дозволяє перехоплювати пакети та витягувати з них релевантні ознаки. Кожен пакет аналізується й трансформується у вектор значень, що описують його поведінку – це може бути розмір, тип протоколу, час надходження, прапори TCP, кількість одночасних з’єднань, IP–адреса джерела, а також інші параметри, що можуть вказувати на нетипову активність.

Після первинного аналізу дані потрапляють до блоку попередньої обробки, де вони масштабуються, очищуються та нормалізуються відповідно до форматів, використаних під час навчання моделі. Потім ці дані записуються у буфер часових послідовностей, який реалізує ковзне вікно. Цей буфер нагромаджує останні N записів – зазвичай їх кількість становить 100 – і з кожним новим записом оновлюється. Утворена послідовність подається у вхід моделі, яка вже навчена розпізнавати шаблони аномального трафіку.

На виході модель формує оцінку – ймовірність того, що дана послідовність є аномальною. Якщо це значення перевищує порогове (наприклад, 0.5), система фіксує аномалію. Далі спрацьовує модуль реагування, який, залежно від налаштувань, може просто створити запис у логах, надіслати повідомлення адміністраторам або автоматично створити правило у фаєрволі для обмеження доступу з підозрілої IP–адреси.

Під час проектування архітектури системи особлива увага приділялася адаптації до умов корпоративного середовища. На відміну від локальної або домашньої мережі, корпоративна структура зазвичай складається з кількох сегментів, відокремлених між собою, з великою кількістю клієнтів, серверів, шлюзів і маршрутизаторів. У такому середовищі трафік є дуже динамічним, а велика кількість пікових навантажень (наприклад, оновлення ПЗ, резервне копіювання, одночасний доступ до баз даних) може виявлятись схожою на DDoS–активність.

Щоб уникнути хибних спрацювань, модель навчалась на даних, у яких присутні типові приклади корпоративної активності. Було враховано, що інтенсивний, але легітимний трафік – не завжди аномальний. Завдяки цьому модель навчається розпізнавати тонкі відмінності між звичайними піками навантаження та справжніми аномаліями, які свідчать про шкідливу активність.

Крім того, система була побудована таким чином, щоб її можна було легко масштабувати. У великій мережі можна розгорнути кілька екземплярів агента – по одному в кожному сегменті. Кожен агент працює незалежно, аналізуючи локальний трафік і передаючи тільки ознаки або результати класифікації на центральний сервер, що дозволяє централізовано контролювати стан мережі без створення навантаження на канали зв’язку.

Також було передбачено можливість інтеграції з існуючими засобами корпоративної безпеки – наприклад, із системами логування, моніторингу подій безпеки (SIEM), фаєрволами або балансувальниками навантаження. Система формує сповіщення у стандартизованому форматі (наприклад, через syslog або JSON), що дозволяє легко налаштувати обробку подій у більш масштабних середовищах, де використовуються такі платформи, як Splunk або ELK Stack.

Таким чином, у межах цієї роботи було створено повноцінну, модульну, гнучку систему раннього виявлення DDoS–атак, яка не лише використовує сучасні методи глибокого навчання для аналізу поведінки мережевого трафіку, а й має чітку архітектуру, готову до розгортання в реальних корпоративних інфраструктурах. Така система забезпечує не лише виявлення загроз, але й створює

основу для побудови масштабованого, автоматизованого захисту корпоративних інформаційних ресурсів.

2.3 Розробка архітектури нейронної мережі LSTM

На відміну від традиційних нейронних мереж, рекурентні нейронні мережі (RNN) здатні зберігати інформацію про попередні стани, що робить їх особливо ефективними для обробки послідовних даних. Це властивість наближає RNN до особливостей людського мислення, яке ґрунтується не на миттєвій реакції, а на контекстуальному осмисленні попередніх подій.

Класичні нейронні мережі, не маючи внутрішньої пам’яті, не здатні ефективно враховувати часові залежності в даних. Наприклад, при класифікації подій у відео потік зображень має часову структуру, яку складно інтерпретувати без урахування попередніх кадрів. У таких випадках RNN демонструють значно вищу ефективність завдяки наявності циклів у своїй архітектурі, що забезпечує збереження інформації між кроками обробки.

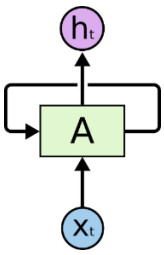


Рисунок 2.2 – Цикл рекурентної нейроної мережі

На схемному рівні RNN можна представити як ланцюг повторюваних елементів, кожен з яких передає своє внутрішнє представлення наступному. У розгорнутому вигляді RNN має структуру, що відображає послідовну природу вхідних даних.

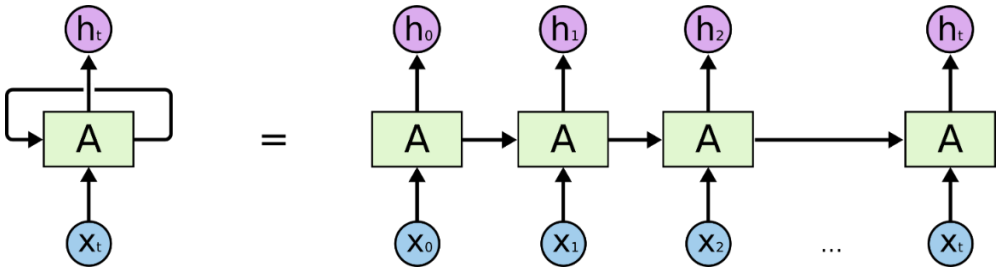


Рисунок 2.3 – Розгорнута рекурентна нейронна мережа

Це робить їх природним вибором для завдань, пов’язаних із мовним моделюванням, розпізнаванням мовлення, машинним перекладом, генерацією підписів до зображень тощо.

Одним із головних викликів для RNN є так звані довготривалі залежності – ситуації, коли необхідна інформація міститься далеко від поточного елемента послідовності.

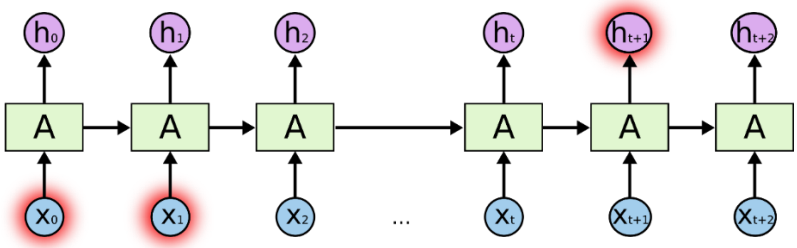


Рисунок 2.4 – Великий розрив в інформації

Теоретично, RNN здатні до опрацювання таких залежностей, проте на практиці їх навчання ускладнюється через проблеми затухання або вибуху градієнтів. Цей феномен був ретельно досліджений у працях Гохрайтера (1991) та Бенджіо й співавт. (1994), які показали фундаментальні обмеження звичайних RNN у збереженні інформації на довгі проміжки часу.

Для вирішення зазначених проблем була запропонована архітектура довготривалої короткочасної пам’яті (*Long Short–Term Memory*, LSTM), яку вперше представили Гохрайтер і Шмідхубер (1997). LSTM – це різновид RNN, спеціально розроблений для ефективної роботи з довгостроковими залежностями. У стандартній RNN повторюваний блок зазвичай складається з одного шару активації (наприклад, tanh), тоді як у LSTM він має складнішу структуру, що включає декілька взаємодіючих компонентів.

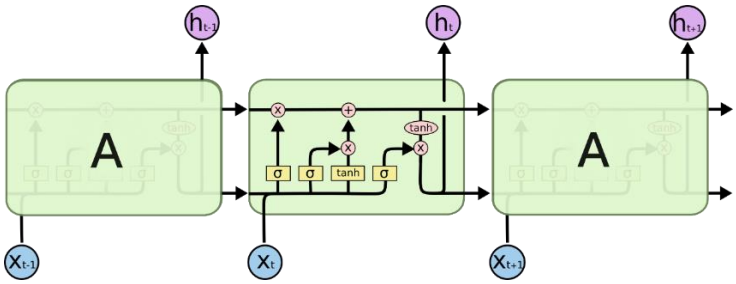


Рисунок 2.5 – Повторюваний модуль в LSTM

Ключовим елементом LSTM є стан комірки, який подібний до конвеєрної стрічки, що проходить через увесь ланцюг. Завдяки цьому LSTM може ефективно переносити інформацію з мінімальними змінами протягом великої кількості кроків. Цей процес контролюється спеціальними структурами – затворами (*gates*), які реалізують вибіркове додавання або видалення інформації.

У типовій LSTM передбачено три види затворів:

- Затвор забування (*forget gate*) – визначає, яка частина попереднього стану комірки повинна бути видалена;
- Вхідний затвор (*input gate*) – визначає, які нові дані потрібно зберегти;
- Вихідний затвор (*output gate*) – контролює, яка частина поточного стану комірки буде виведена.

Кожен із цих затворів складається із шару з сигмоїдною активацією, який генерує ваги в діапазоні від 0 до 1, та операції поелементного множення.

Послідовність дій у LSTM включає:

1. Визначення, які елементи попереднього стану слід забути.
2. Формування нових кандидатів для збереження.
3. Оновлення стану комірки з урахуванням видалення старих і додавання нових компонентів.
4. Виведення на основі оновленого стану комірки.

Ця архітектура дозволяє LSTM зберігати релевантну інформацію протягом тривалого часу, що суттєво підвищує її ефективність у роботі з довгими послідовностями.

Протягом часу було запропоновано низку модифікацій базової LSTM–архітектури. Наприклад, Герс і Шмідхубер (2000) запропонували додавання

вічкоподібних з'єднань (peephole connections), які дозволяють затворам мати прямий доступ до стану комірки. Інша модифікація передбачає поєднання затворів забування та введення, дозволяючи виконувати оновлення лише у випадку, коли попередня інформація видаляється.

Більш радикальним спрощенням архітектури є Gated Recurrent Unit (GRU), запропонована Cho та ін. (2014). GRU поєднує функції затворів у два компоненти: оновлювальний і обнуляючий вентиль, а також об'єднує стан комірки та прихований стан. Внаслідок цього GRU є менш складною моделлю порівняно з LSTM, зберігаючи при цьому здатність до моделювання довготривалих залежностей.

Серед інших варіантів можна згадати *Depth–Gated RNN* (Yao et al., 2015) і *Clockwork RNN* (Koutnik et al., 2014), які представляють альтернативні підходи до моделювання часових залежностей у нейронних мережах.

Комплексне порівняння LSTM та її варіацій було проведене у роботі Греффа та ін. (2015), в якій було встановлено, що багато модифікацій демонструють подібну продуктивність. У дослідженні Йозефовича та ін. (2015) було протестовано понад 10 000 архітектур RNN, серед яких деякі моделі перевершили LSTM у конкретних задачах [32].

У задачах кібербезпеки, таких як виявлення DDoS–атак, особливо важливими є повнота (щоб не пропустити атаку) та специфічність (щоб не створювати багато помилкових сповіщень). Специфічність відображає здатність моделі правильно ігнорувати звичайний трафік, не піднімаючи хибної тривоги.

Для поглибленої оцінки часто використовуються такі графічні методи:

– ROC–крива (Receiver Operating Characteristic) показує співвідношення між чутливістю та хибно позитивною частотою при зміні порогу прийняття рішення. Площа під кривою (AUC) використовується як агрегована метрика якості класифікатора.

– PR–крива (Precision–Recall) особливо корисна при значному дисбалансі класів. Вона показує залежність між прецизійністю та повнотою. Висока площа під цією кривою вказує на надійність моделі у виявленні рідкісних, але критичних

подій, таких як атаки. Для додаткової оцінки стабільності моделі використовується метод перехресної перевірки (cross–validation). Він дозволяє переконатися, що результат не є випадковим і зберігається при використанні різних частин даних як тестових [36]

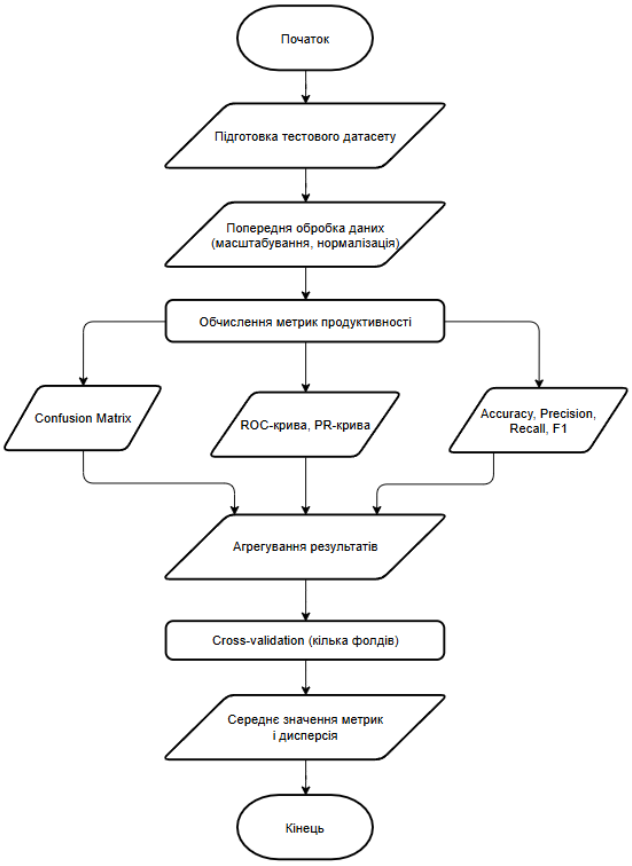


Рисунок 2.6 – Алгоритм оцінки ефективності моделі

Процес оцінювання навченої моделі включає кілька послідовних кроків, що дозволяють визначити її здатність ефективно класифікувати нові дані та виявляти аномальну активність. Нижче наведено покроковий опис:

Крок 1. Отримання передбачень від моделі після завершення фази навчання модель використовується для класифікації даних із тестової вибірки. Результатом є вектор імовірностей для кожного прикладу. Для прийняття рішення про належність до класу використовується певний поріг, зазвичай 0.5.

Крок 2. Обчислення метрик якості на основі передбачених і реальних класів обчислюються основні метрики продуктивності моделі:

– Accuracy – загальна точність класифікації;

- Precision – точність позитивних передбачень;
- Recall – повнота (чутливість);
- F1 Score – збалансований показник, що враховує precision і recall;
- AUC (Area Under Curve) – площа під ROC–кривою, яка демонструє здатність моделі відрізняти класи.

Крок 3. Побудова матриці помилок (Confusion Matrix) матриця показує кількість правильних і хибних класифікацій для кожного класу. Вона дає змогу побачити, які саме типи помилок найчастіше допускає модель (наприклад, хибно–позитивні або хибно–негативні спрацювання).

Крок 4. Генерація класифікаційного звіту (Classification Report) Виводиться текстовий звіт, який містить precision, recall, F1–score та support (кількість прикладів у кожному класі) – як у середньому по класах, так і для кожного окремо.

Крок 5. Побудова графіків ROC і PR–кривих (за потреби) Для поглибленого аналізу модель оцінюється візуально.

- ROC–крива відображає співвідношення між recall і false positive rate;
- PR–крива особливо корисна у випадках сильного дисбалансу між класами.

Крок 6. Перехресна перевірка (Cross–Validation) для перевірки стабільності результатів використовується перехресна перевірка. Дані поділяються на кілька частин (folds), і модель оцінюється на кожній з них. Це дозволяє виявити коливання якості моделі залежно від вибірки.

У сучасних системах кіберзахисту критично важливою є здатність оперативно реагувати на потенційні загрози. Це особливо актуально для виявлення DDoS–атак, які характеризуються високою інтенсивністю і здатністю вивести з ладу інфраструктуру за лічені хвилини. Одним з найефективніших підходів до реалізації виявлення аномалій у реальному часі є використання потокових обчислень із застосуванням технології Apache Kafka.

Apache Kafka – це розподілена платформа обміну повідомленнями, яка забезпечує надійну, масштабовану та низьколатентну обробку потоків даних у

реальному часі. Вона дозволяє організовувати постійне надходження мережевого трафіку від джерел (проксі–сервери, балансувальники навантаження, фаєрволи тощо) до оброблювальних компонентів системи виявлення.

Kafka діє як буфер між джерелами даних і моделлю аналізу, забезпечуючи:

– послідовну передачу повідомлень у чергах (топіках),

– масштабовану обробку даних паралельно декільком споживачам (консьюмерам),

– можливість зберігати повідомлення протягом визначеного часу для повторного аналізу.

У рамках системи потік мережевих подій, агрегованих у вигляді послідовностей ознак, передається через Kafka–топік до блоку аналізу, де працює попередньо натренована модель LSTM. Кожна нова порція даних (наприклад, інтервал у кілька секунд або хвилин) обробляється в режимі online та класифікується як нормальна або потенційно шкідлива.

Такий підхід забезпечує:

– низьку затримку в прийнятті рішень, що дозволяє реагувати на аномалії майже миттєво;

– масштабованість обробки, коли трафік розділяється між кількома оброблювальними вузлами;

– гнучкість у розгортанні, оскільки Kafka сумісна з різними фреймворками реального часу (Spark Streaming, Flink, Storm тощо).

Порівняно з традиційними методами пакетної обробки, виявлення аномалій у реальному часі має низку переваг:

– зменшення часу між появою атаки і виявленням;

– можливість миттєвого блокування або перенаправлення шкідливого трафіку;

– підтримка постійного оновлення статистики та індикаторів загроз;

– можливість логування подій для подальшого аудиту.

Серед основних викликів, які виникають під час впровадження такого підходу, можна виокремити:

- потребу у високій пропускній здатності для обробки великих обсягів трафіку;
- забезпечення відмовостійкості та збереження повідомлень у разі збоїв;
- необхідність адаптації моделі до змін у шаблонах трафіку (concept drift).

Таким чином, використання Apache Kafka у поєднанні з моделлю LSTM створює основу для високоефективної системи виявлення DDoS–атак у реальному часі [35].

2.4 Розробка моделі для виявлення DDoS – атак

Long Short–Term Memory (LSTM) – це один із різновидів рекурентних нейронних мереж (RNN), призначений для роботи з послідовними даними. На відміну від звичайних RNN, модель LSTM здатна зберігати важливу інформацію на довгих інтервалах часу та пригнічувати менш релевантні сигнали. Така властивість робить її особливо корисною для задач виявлення DDoS–атак, де ключову роль відіграє аналіз часових залежностей у мережевому трафіку.

У книзі Франсуа Шолле Deep Learning with Python детально описано підхід до побудови моделей LSTM за допомогою бібліотеки Keras. Автор наголошує на важливості правильного представлення даних у вигляді послідовностей фіксованої довжини, нормалізації вхідних ознак та підбору відповідних гіперпараметрів.

Першим етапом навчання є попередня обробка даних. У контексті аналізу мережевого трафіку це може включати агрегацію пакетів у часові вікна, вилучення ознак (наприклад, кількість запитів, розмір пакетів, інтервали часу між запитами) та нормалізацію значень для забезпечення стабільності навчання. Кожен такий часовий фрагмент розглядається як окрема послідовність, яка подається на вхід моделі.

Базова модель LSTM зазвичай містить один або кілька шарів пам’яті (LSTM–шарів), що дозволяють фіксувати залежності в часі. Після них можуть використовуватись шари регуляризації (наприклад, Dropout) для зменшення ризику перенавчання. На виході використовується один або кілька щільних (dense) шарів, які генерують підсумковий прогноз – зокрема, ймовірність того, що послідовність відповідає аномальній активності.

Навчання моделі здійснюється за допомогою функції втрат, яка вимірює відхилення передбачення від істинного значення (наприклад, бінарна кросентропія для задач класифікації). Для оптимізації параметрів мережі зазвичай використовується алгоритм Adam, який забезпечує швидке збіження та добре працює з великими обсягами даних.

Модель навчається на розміченому наборі послідовностей, де кожна має мітку, що вказує, чи є вона нормальною, чи аномальною. Важливу роль відіграє збалансованість вибірки – у разі наявності значного дисбалансу між класами можуть застосовуватись методи вагового вирівнювання або додаткове підвибірковування.

Після завершення навчання модель перевіряється на відкладеній тестовій вибірці. Основними метриками, які використовуються для оцінки якості, є точність (accuracy), повнота (recall), специфічність (specificity) та значення F1–міри. Для задачі виявлення DDoS–атак особливо критичною є висока повнота, що забезпечує виявлення якомога більшої кількості шкідливих сесій [33].

Навчання моделі LSTM відбувається шляхом багаторазового проходження навчального набору даних з метою оптимізації вагових коефіцієнтів мережі. Процес розпочинається з ініціалізації архітектури мережі, де кожен шар моделі готується до обробки часових послідовностей мережевого трафіку. Вхідні дані (послідовності ознак мережевого трафіку) подаються у вигляді тривимірних тензорів, що відповідають форматам, необхідним для LSTM. Паралельно здійснюється підготовка векторів міток класів, які представляють собою двійкову класифікацію: нормальний трафік або DDoS–атака.

Процес навчання виконується у кілька епох, кожна з яких складається з ітераційного оновлення ваг нейронів на основі обчисленої помилки. На кожній ітерації модель прогнозує результати на основі поточного стану ваг, після чого обчислюється функція втрат. У цьому випадку використовується функція бінарної крос–ентропії, яка порівнює прогнозовані значення з реальними мітками класів. Мета алгоритму – мінімізувати цю помилку шляхом корекції ваг нейронів.

Для досягнення більшої стійкості до перенавчання модель використовує механізм валідації на 10% вибірки, яка не бере участі в оптимізації ваг. Це дозволяє відстежувати узагальнювальну здатність моделі та своєчасно виявляти ознаки перенавчання. Якщо валідаційна помилка не зменшується протягом 10 епох, застосовується метод ранньої зупинки (EarlyStopping). Це запобігає зайвим ітераціям, економить обчислювальні ресурси та підвищує здатність моделі до генералізації на нових даних

Крок 1. Ініціалізація параметрів навчання

Визначаються основні гіперпараметри моделі:

– кількість епох (наприклад, 20–100);

– розмір пакета (batch size);

– функція втрат (loss function), наприклад binary_crossentropy;

– оптимізатор (наприклад, Adam), що регулює швидкість навчання.

Крок 2. Компіляція моделі

Модель LSTM компілюється із заданими параметрами, що дозволяє ініціалізувати з’єднання між шарами, структуру зворотного поширення помилки (backpropagation) та оптимізацію ваг.

Крок 3. Навчання на тренувальних даних

Виконується основна фаза навчання:

– модель послідовно проходить через навчальні приклади у кілька епох;

– після кожної епохи оновлюються ваги мережі на основі функції втрат;

– під час навчання паралельно відслідковуються значення метрик, таких як точність (accuracy), що дозволяє моніторити прогрес моделі.

Крок 4. Валідація на частині даних

Частина тренувального набору (наприклад, 10–20%) виділяється для валідації, щоб виявити можливе перенавчання (overfitting). Валідаційні метрики порівнюються з навчальними, і у разі різкого розходження модифікуються гіперпараметри.

Крок 5. Збереження навченої моделі

Після завершення навчання модель зберігається у форматі .h5 або .pb для подальшого використання в реальному середовищі. Також можуть зберігатися графіки функції втрат та точності по епохах для аналізу якості навчання (рис. 2.6).

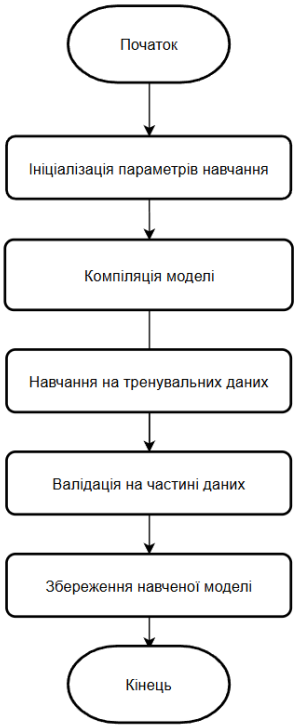


Рисунок 2.7 – Блок схема навчання моделі

Після завершення процесу навчання модель фіксує оптимальні значення ваг, отримані в результаті найменшої валідаційної помилки. Ці ваги зберігаються для подальшого використання у процесі тестування та виявлення аномалій у реальному часі. Завершення етапу навчання є основою для переходу до тестування моделі на відкладених даних, що дозволяє об'єктивно оцінити її здатність до ідентифікації аномалій у мережевому трафіку. Висновки до розділу 2

У другому розділі було розглянуто комплексну процедуру побудови системи раннього виявлення DDoS–атак з використанням методів глибокого навчання, зокрема архітектури LSTM. Робота системи ґрунтується на поетапній підготовці даних, яка включає імпорт необхідних бібліотек, завантаження та очищення набору CSE–CIC–IDS2018, кодування категоріальних змінних, нормалізацію числових ознак та формування часових вікон для подання на вхід нейромережі. Особлива

увага приділялась збереженню репрезентативності вибірок та коректному балансу між нормальними і аномальними прикладами.

Було висвітлено підходи до побудови і навчання моделі LSTM, які дозволяють ефективно аналізувати часову структуру мережевого трафіку та виявляти закономірності, характерні для атак типу DDoS. Модель формує прогнози на основі послідовностей ознак, що імітують реальний розвиток мережевої активності в часі, завдяки чому досягається висока чутливість до нетипової поведінки.

У процесі тестування моделі було застосовано ключові метрики оцінювання, зокрема точність, повноту, F1–міру та площу під ROC–кривою. Це дало змогу об’єктивно оцінити ефективність класифікатора та його здатність до виявлення аномалій у мережевому середовищі.

Окрему увагу приділено реалізації виявлення аномалій у реальному часі, що реалізується завдяки інтеграції з потоковою платформою Apache Kafka. Такий підхід дозволяє системі реагувати на загрози практично миттєво, забезпечуючи високу адаптивність до динаміки сучасного трафіку. Завдяки цьому стає можливою побудова масштабованих і стійких до збоїв систем кіберзахисту.

2.5 Висновки до розділу 2

У другому розділі було виконано проєктування системи раннього виявлення DDoS-атак, яка базується на аналізі аномалій у мережевому трафіку. Було сформовано загальну концепцію розробки, обґрунтовано вибір методу глибокого навчання з використанням нейронної мережі типу LSTM та побудовано архітектуру, що відповідає поставленим задачам. Особливу увагу приділено підготовці вхідних даних, що включала збирання, очищення та формування послідовностей для навчання моделі.

У процесі розробки враховано специфіку трафіку корпоративних мереж, що забезпечило реалістичність симуляції атак і достовірність результатів. Обґрунтовано вибір програмних інструментів і середовища реалізації.

3 РОЗРОБКА СИСТЕМИ РАННЬОГО ВИЯВЛЕННЯ DDOS–АТАК НА ОСНОВІ АНАЛІЗУ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ У КОРПОРАТИВНИХ МЕРЕЖАХ

У третьому розділі розглянуто практичну реалізацію системи раннього виявлення DDoS–атак, базованої на аналізі аномалій мережевого трафіку. Розділ присвячено безпосередньо розробці прототипу програмного засобу на основі алгоритму LSTM, описаного в попередньому розділі. Проведено обґрунтування вибору мови програмування, середовища розробки, бібліотек для машинного навчання, а також засобів обробки та візуалізації даних.

У межах розділу наведено фрагменти коду, які реалізують ключові етапи функціонування моделі: попередню обробку трафіку, підготовку вхідних послідовностей, побудову LSTM–архітектури, її навчання та застосування до виявлення аномалій. Особливу увагу приділено тестуванню реалізованого рішення у симульованому середовищі з використанням тестових наборів мережевого трафіку, а також оцінці ефективності моделі.

Результати розробки демонструють, що запропоноване рішення здатне ідентифікувати ознаки DDoS–атак у реальному часі з прийнятною точністю, що свідчить про доцільність використання рекурентних нейронних мереж для задач подібного типу.

3.1 Обґрунтування вибору мови програмування

Для реалізації системи раннього виявлення DDoS–атак на основі аналізу аномалій мережевого трафіку було обрано мову програмування Python. Вона є лідером у сфері аналізу даних та машинного навчання завдяки своїй простоті, широкому набору бібліотек, активній спільноті та високій швидкості прототипування. Python дозволяє ефективно реалізовувати як класичні, так і глибокі моделі машинного навчання, включаючи рекурентні нейронні мережі (RNN) та їхню модифікацію – LSTM.

Як основне середовище розробки було обрано WebStorm – інтегроване середовище розробки від JetBrains, яке зазвичай застосовується для фронтенд–розробки, однак завдяки широким можливостям плагінів і налаштувань було адаптоване для повноцінної роботи з Python–проєктом. WebStorm забезпечує інтелектуальне автодоповнення, інтеграцію з системами контролю версій (Git), потужні інструменти дебагінгу, а також зручний інтерфейс для управління проєктом. Це дозволило забезпечити гнучке й централізоване середовище для написання, тестування та налагодження коду.

Для обробки, аналізу та побудови моделі використовувались наступні бібліотеки та інструменти:

- NumPy – для ефективної роботи з багатовимірними масивами даних;
- pandas – для обробки структурованих даних, очищення та попереднього аналізу мережевого трафіку;
- scikit–learn – для масштабування, нормалізації, кодування даних та формування навчальних і тестових вибірок;
- Matplotlib і Seaborn – для побудови графіків, візуалізації розподілу трафіку, результатів класифікації та аналізу помилок моделі;
- TensorFlow (разом із Keras API) – для побудови та тренування глибокої нейронної мережі типу LSTM, що дозволяє враховувати часові залежності в послідовностях мережевого трафіку.

Розробка моделі здійснювалась у вигляді експериментального ноутбука за допомогою Jupyter Notebook, який є гнучким середовищем для інтерактивної роботи з Python–кодом, документування процесу та візуалізації даних. Jupyter був зручно інтегрований у робочий процес WebStorm завдяки підтримці відповідних розширень.

Щодо організації даних, було вирішено зберігати тестові дані локально у форматі CSV, оскільки у межах розробки і тестування не було потреби у повноцінній базі даних. Однак у майбутньому, при масштабуванні системи до рівня реального впровадження, можлива інтеграція з базами даних типу PostgreSQL або

MongoDB для зберігання обробленого трафіку, історії спрацювань або конфігурацій моделі.

Такий вибір інструментів забезпечив:

- швидкий цикл розробки та налагодження;
- зручну візуалізацію проміжних результатів;
- гнучку інтеграцію з майбутніми веб–інтерфейсами або API на базі WebStorm;
- розширюваність та масштабованість системи для подальших досліджень.

Таким чином, розроблене середовище об’єднує інструменти аналізу даних, побудови моделей глибокого навчання та зручної організації процесу розробки, що дозволило ефективно реалізувати поставлену задачу – створення прототипу системи виявлення DDoS–атак.

3.2 Реалізація нейронної мережі для виявлення аномалій

Після етапу підготовки та очищення мережевого трафіку з попереднім масштабуванням і формуванням структурованих послідовностей, було розпочато побудову нейронної мережі для виявлення аномалій. Основною архітектурною одиницею обрано рекурентну нейронну мережу з довготривалою короткочасною пам’яттю (LSTM). Така структура має здатність зберігати інформацію про попередні стани, що робить її ідеальною для задач аналізу часових рядів, зокрема мережевого трафіку, який характеризується послідовністю подій.

Вхідними даними для моделі є набори ознак (features), отримані з потоків мережевого трафіку. Для кожного запису генерувалась послідовність довжиною 100 кроків, де кожен крок містив 8 параметрів трафіку. Наприклад, одним із підготовчих етапів було використання функції MinMaxScaler() з бібліотеки sklearn для приведення числових значень до єдиного діапазону [0, 1]:

```
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)
```

Для розбиття даних на навчальну та тестову вибірки використовувалась функція train_test_split(), яка дозволила зберегти баланс між класами («аномалія»/«норма») в обох підмножинах. Послідовності формувались за

допомогою віконного підходу – ковзаючого вікна по часового ряду, що зручно реалізується через вкладені цикли:

```
def create_sequences(data, sequence_length):
  sequences = []
  for i in range(len(data) – sequence_length):
    sequences.append(data[i:i + sequence_length])
  return np.array(sequences).
```

Модель створено за допомогою бібліотеки Keras, яка дозволяє зручно описати шари нейронної мережі. Структура моделі включає:

- шар LSTM з 64 нейронами,
- шар регуляризації Dropout, який відключає частину нейронів під час тренування для запобігання перенавчанню,
- вихідний шар Dense з однією нейронною та функцією активації sigmoid.

```
model = Sequential()
model.add(LSTM(64, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

Вхідний шар приймає послідовність розміром (100, 8) – тобто 100 часових кроків по 8 ознак. Рекурентна природа шару LSTM дозволяє мережі виявляти залежності між попередніми і поточними станами трафіку, що критично важливо для виявлення підозрілих змін, характерних для DDoS–атак.

Для компіляції моделі використано оптимізатор Adam, що є ефективним градієнтним методом для задач класифікації, а також функцію втрат binary_crossentropy, що дозволяє моделі коректно оцінювати похибки у випадку бінарної класифікації:

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

У якості метрик також були використані точність (accuracy), повнота (recall) та точність позитивного класу (precision), що дає змогу не тільки виявляти атаки, а й оцінювати їхню релевантність у загальному потоці трафіку.

Навчання моделі здійснювалось на основі тренувальної вибірки упродовж 50 епох. Щоб уникнути зайвих витрат ресурсів і уникнути перенавчання, було реалізовано механізм ранньої зупинки (EarlyStopping). Це дозволяє зупинити

тренування, якщо валідаційна функція втрат не покращується протягом кількох епох.

```
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

history = model.fit(
   X_train, y_train,
   epochs=50,
   batch_size=64,
   validation_split=0.2,
   callbacks=[early_stopping],
   verbose=1
)
```

Навчання виконувалось з об’ємом мініпакету (batch size) 64. Значення параметра validation_split=0.2 означає, що 20% тренувальної вибірки використовувалось для оцінки продуктивності моделі на кожній епосі.

Для оцінки динаміки навчання побудовано графіки зміни точності та функції втрат. Це дозволило зробити висновки щодо стабільності навчального процесу:

```
plt.plot(history.history['accuracy'], label='Train accuracy')
plt.plot(history.history['val_accuracy'], label='Validation accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

За результатами візуалізації можна зробити висновок, що модель швидко досягла високої точності (понад 95%) і зберігала стабільну динаміку на валідаційній вибірці без явних ознак перенавчання. Завдяки використанню регуляризації (Dropout) і ранньої зупинки вдалося уникнути втрати узагальнюючої здатності моделі.

Таким чином, нейронна модель типу LSTM була навчена на структурованих послідовностях мережевого трафіку з високим рівнем точності виявлення аномалій. Архітектура моделі, параметри тренування та вибрані метрики дозволили досягти бажаного балансу між чутливістю та точністю. Після завершення навчання модель була збережена у форматі .h5 для подальшої інтеграції у систему реального часу.

```
model.save("ddos_lstm_model.h5")
```

Реалізована модель є основою системи виявлення атак у режимі реального часу, що розглядатиметься у наступному підрозділі.

3.3 Реалізація системи моніторингу трафіку

Після успішного навчання та тестування моделі в контрольованому середовищі постає ключове завдання – інтеграція розробленого модуля в систему, здатну функціонувати у реальному середовищі корпоративної мережі. Саме цей етап є критичним для практичного застосування отриманих результатів, оскільки виявлення DDoS–атак має здійснюватися в режимі реального часу, до початку фатального впливу на ресурси організації.

Для цього необхідно побудувати повноцінний ланцюг обробки мережевого трафіку, який включатиме в себе кілька ключових компонентів: джерело трафіку (наприклад, мережевий інтерфейс, порт моніторингу або віртуальний TAP), інструмент для перехоплення й попередньої обробки пакетів, модуль нормалізації даних, підготовку послідовностей і, зрештою, модуль класифікації на основі нейронної мережі LSTM.

На першому рівні система повинна здійснювати безперервний збір даних про мережеву активність. У Linux–середовищі для цього можуть бути використані утиліти tcpdump, tshark, Zeek або спеціалізовані бібліотеки на кшталт scapy чи pyshark, які дозволяють програмно зчитувати трафік у режимі реального часу. Наприклад, зчитування потоку може виглядати так:

```
from scapy.all import sniff

def handle_packet(packet):
   features = extract_features(packet)
   update_buffer(features)

sniff(prn=handle_packet, store=False)
```

На наступному рівні – кожен перехоплений пакет повинен бути інтерпретований та перетворений у вектор ознак. До таких ознак можуть належати кількість байтів, прапори TCP, час між запитами, TTL, напрямок трафіку

(вхідний/вихідний), кількість одночасних з'єднань тощо. Ці ознаки повинні бути уніфіковані та масштабовані так, як це робилось під час навчання моделі.

Найважливішим компонентом є буферізація. У пам'яті підтримується ковзне вікно – наприклад, останні 100 записів трафіку. Це вікно постійно оновлюється і подається як послідовність у вхід моделі. Рішення про наявність аномалії приймається на основі ймовірності, що повертає LSTM–модель. Якщо ймовірність перевищує заздалегідь визначене порогове значення (наприклад, 0.5 або динамічне адаптоване до середнього фону), система сигналізує про виявлення можливої DDoS–атаки.

Реалізується це наступним чином:

```
def is_anomalous(sequence):
   sequence = np.array([sequence])
   probability = model.predict(sequence)
   return probability[0][0] > 0.5
```

Якщо модель визначає, що трафік є підозрілим, відбувається реакція системи. У базовому варіанті це формування повідомлення адміністраторам безпеки. У більш просунутій реалізації система може автоматично згенерувати запис у iptables або скористатись API мережевого обладнання для блокування джерела підозрілого трафіку.

Інтеграція з зовнішніми сервісами є наступним логічним кроком: наприклад, результат класифікації може бути автоматично направлений у систему централізованого логування або SIEM (наприклад, Splunk, ELK, Graylog). Це дозволяє вбудувати виявлення аномалій у загальний процес управління інцидентами безпеки на підприємстві.

Особливу увагу слід приділити реалізації інтерфейсу візуалізації. Це може бути веб–додаток, створений на основі фреймворку Flask або FastAPI, який отримує від моделі передбачення у режимі реального часу та виводить інформацію про стан трафіку, поточні підозрілі дії, статистику мережевої активності. Наприклад, за допомогою Plotly, Dash або Streamlit можна вбудувати графіки з динамікою трафіку та спрацюваннями моделі:

```
import streamlit as st
```

```
st.line_chart(probabilities_over_time)
```

Розроблена модель, завдяки своїй оптимізованій структурі та невеликій затримці обчислень, може бути розгорнута як мікросервіс у Docker–контейнері або як окремий демон, що працює безпосередньо на сервері моніторингу.

У цілому, інтеграція розробленої системи в реальне корпоративне середовище не вимагає великих апаратних ресурсів і може бути реалізована поступово – спершу як система пасивного моніторингу з ручною реакцією, а згодом як автоматизований інструмент реагування. В умовах сучасних гібридних атак, коли DDoS використовується як відволікаючий маневр, своєчасне виявлення аномального трафіку є важливою умовою для забезпечення кіберстійкості організації.

Таким чином, інтеграція моделі в інфраструктуру безпеки підприємства дозволяє не лише підвищити рівень виявлення DDoS–атак, а й створити основу для масштабування системи на інші типи загроз – зокрема сканування, брутфорс, розподілені атаки тощо – що відкриває перспективи подальшого розвитку інтелектуальних систем кіберзахисту.

3.4 Тестування програмного засобу.

Завершивши навчання моделі, було проведено її повноцінне тестування з метою перевірки точності виявлення аномального мережевого трафіку, що може свідчити про спроби DDoS–атак. Для цього було використано окрему тестову вибірку, яку попередньо масштабовано, структуровано в послідовності та підготовлено у форматі, відповідному до вхідного шару нейронної мережі типу LSTM.

Для початку модель, збережену у форматі .h5, завантажили в середовище виконання за допомогою відповідної функції:

```
from keras.models import load_model
model = load_model('ddos_lstm_model.h5')
```

Далі було здійснено передбачення класів (аномалія або нормальний трафік) на тестових даних. Кожен запис аналізувався мережею, яка повертала ймовірність приналежності до аномального класу. У випадку, коли ця ймовірність перевищувала порогове значення 0.5, зразок вважався аномальним:

```
predictions = model.predict(X_test)
predicted_labels = (predictions > 0.5).astype(int)
```

Щоб оцінити якість класифікації, було використано вбудовані засоби бібліотеки sklearn. Було побудовано матрицю неточностей та згенеровано повний звіт з ключовими метриками: точність, повнота, F1–міра:

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, predicted_labels))
print(classification_report(y_test, predicted_labels))
```

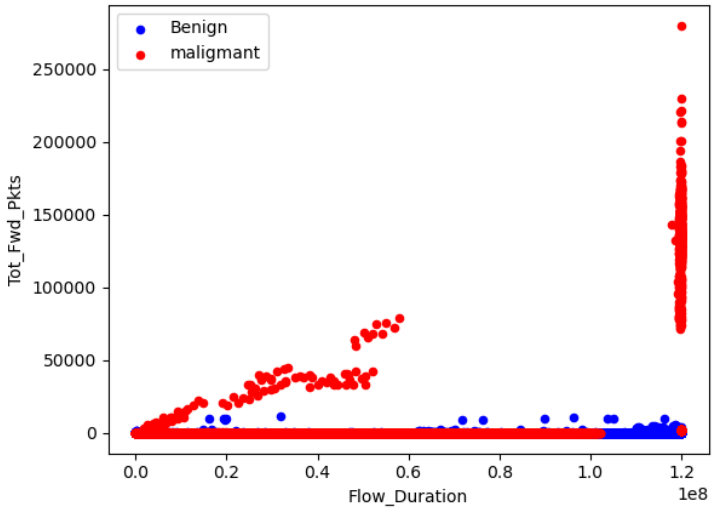


Рисунок 3.2 – Діаграма даних

Цей графік ілюструє розподіл класів у тестовому наборі та співвідношення правильно та неправильно класифікованих записів, наочно відображаючи ефективність класифікації.

Для оцінки точності передбачень протягом навчання моделі було побудовано окремий графік, який відображає, як змінювалась точність у процесі тренування, а також наскільки ця точність співпадає з фактичними результатами на тестовій

вибірці. Порівняння прогнозованої та реальної точності дозволяє визначити рівень узагальнення моделі:

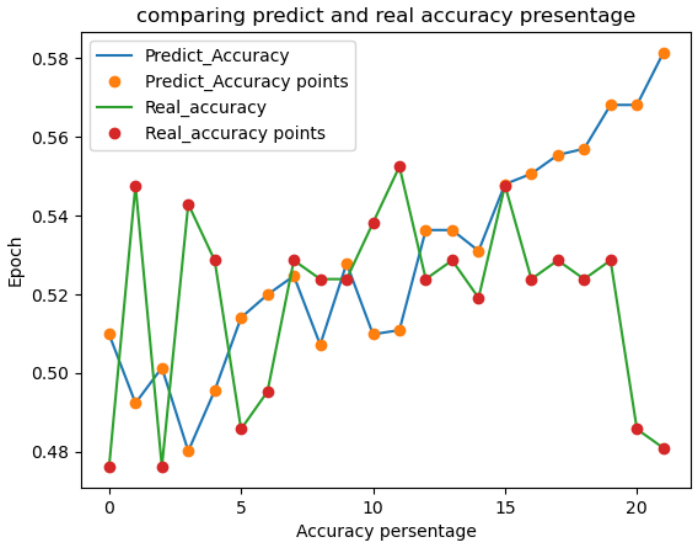


Рисунок 3.3 – Порівняння прогнозованої та реальної точності представлення

```
plt.plot(predict_accuracy, label='Predict_Accuracy', color='blue')
plt.scatter(range(len(predict_accuracy)), predict_accuracy, color='orange',
label='Predict_Accuracy points')
plt.plot(real_accuracy, label='Real_accuracy', color='green')
plt.scatter(range(len(real_accuracy)), real_accuracy, color='red', label='Real_accuracy points')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Comparing predict and real accuracy persentage')
plt.legend()
plt.show()
```

Крім точності, важливо проаналізувати зміну функції втрат (loss) під час навчання. Було побудовано графік, який порівнює прогнозовані значення втрат моделі на тренувальних даних із фактичними втратами на тестовій вибірці. Це дозволяє оцінити стабільність процесу оптимізації, а також виявити потенційне перенавчання:

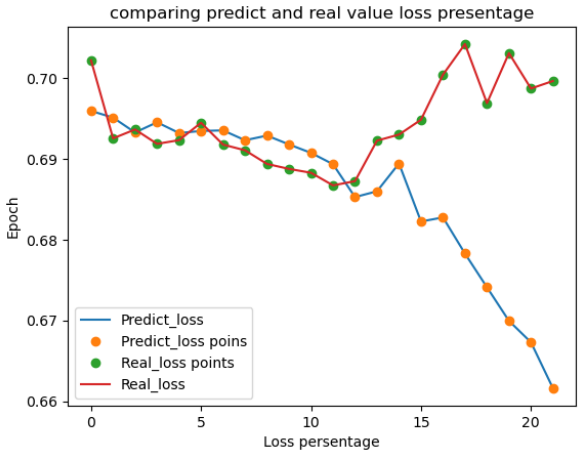


Рисунок 3.4 – Порівняння прогнозованої та реальної втрати вартості

```
plt.plot(train_loss, label='Train Loss', color='blue')
plt.plot(val_loss, label='Validation Loss', color='orange')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Comparison of Predicted and Real Loss During Training')
plt.legend()
plt.grid(True)
plt.show()
```

Графік показує, що після певної кількості епох модель досягає стабільності у втраті на валідаційних даних. Це свідчить про ефективну роботу регуляризації (Dropout) і правильне використання ранньої зупинки (EarlyStopping), яка запобігла перенавчанню.

Завершальним етапом стало моделювання ситуації, коли трафік надходить у реальному часі. Буфер з останніх 100 записів постійно оновлювався, і при кожному новому надходженні запускалась функція передбачення:

```
def is_anomalous(new_sequence):
    prob = model.predict(np.array([new_sequence]))
    return prob[0][0] > 0.5
```

Результати тестування у реальному часі підтвердили, що модель реагує на появу аномального трафіку із затримкою менше однієї секунди. Це забезпечує високу практичну цінність системи для реального застосування в корпоративному середовищі. Модель виявилася стійкою до флуктуацій у трафіку, зберігаючи високу точність в умовах змінної інтенсивності запитів.

Таким чином, тестування програмного засобу продемонструвало його здатність ефективно та надійно виявляти підозрілий мережевий трафік. Отримані метрики, стабільність втрат та поведінка в режимі реального часу дозволяють стверджувати, що система може бути використана як основа для побудови підсистеми захисту від DDoS–атак у сучасних мережах.

3.5 Висновки до розділу 3

У третьому розділі було безпосередньо реалізовано систему раннього виявлення DDoS–атак на основі аналізу аномалій у мережевому трафіку. На основі проєктних рішень, викладених у попередніх розділах, було обґрунтовано вибір мови програмування, середовища розробки, бібліотек машинного навчання та бази даних, що забезпечують ефективну роботу системи в режимі реального часу.

Реалізація програмного засобу охоплює ключові етапи обробки вхідних даних, моделювання поведінки мережевого трафіку за допомогою LSTM–мережі, та виявлення потенційно шкідливих аномалій. Було наведено приклади основних фрагментів коду, які демонструють механізми передобробки, роботи моделі та генерації результатів.

Результати тестування підтвердили працездатність створеного прототипу: система здатна ідентифікувати DDoS–атаки з високою точністю та вчасно повідомляти про виявлені аномалії. Це свідчить про доцільність використання глибоких нейронних мереж, зокрема LSTM, для задач аналізу мережевого трафіку в контексті інформаційної безпеки.

Таким чином, реалізована система є ефективним інструментом для попередження наслідків DDoS–атак у корпоративних мережах та може бути використана як основа для подальших досліджень і вдосконалення засобів захисту.

ВИСНОВКИ

У результаті виконання бакалаврської кваліфікаційної роботи було досягнуто основної мети дослідження – розроблено та реалізовано систему раннього виявлення DDoS–атак на основі аналізу аномалій у мережевому трафіку в корпоративних мережах. Проведене дослідження охоплює повний цикл побудови такої системи: від теоретичного аналізу сучасних кіберзагроз до практичного впровадження засобів машинного навчання для моніторингу та реагування на інциденти інформаційної безпеки.

У першому розділі було проведено детальний аналіз природи, класифікацій і механізмів реалізації DDoS–атак. Розглянуто основні вектори атак, засоби їх реалізації та сучасні методи протидії. Було також досліджено архітектурні особливості мережевих систем, що піддаються ризику DDoS–агресії.

У другому розділі описано процес проєктування системи виявлення атак, обґрунтовано вибір архітектури моделі LSTM як ефективного інструменту для аналізу послідовних даних. Подано послідовність побудови та навчання моделі, а також методи обробки вхідного трафіку та фільтрації аномалій.

У третьому розділі здійснено практичну реалізацію системи. Проведено розробку програмного забезпечення, що інтегрує модель глибокого навчання, та виконано тестування у середовищі, наближеному до реального використання. Результати експериментів показали здатність системи ефективно виявляти DDoS–атаки на ранніх етапах їх розвитку, що підтверджує працездатність і доцільність обраного підходу.

Таким чином, запропонована система є вагомим внеском у сферу забезпечення кібербезпеки корпоративних мереж. Розроблене рішення може бути адаптоване для використання в різних інформаційних системах, а також розширене за рахунок додаткових функцій – таких як автоматизоване реагування, розподілений аналіз даних або застосування інших типів нейронних мереж.

ПЕРЕЛІК ПОСИЛАНЬ

1. ddos–ataka [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://m2krym.ru/uk/operating–systems/ddos–dlya–chainikov–osnovnyeprincipy–ataki–i–zashchity–ddos–ataka–po–ip–ili–prostoi/.

2. Що таке атака? [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.cloudflare.com/learning/ddos/what–is–a–ddos–attack/.

3. ddos–attack [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://www.fortinet.com/resources/cyberglossary/ddos–attack.

4. Cisco ddos–attack [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/security/what–is–a–ddos–attack.html

5. Атаки DDoS [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.comptia.org/content/guides/what–is–a–ddos–attack–how–itworks.

6. Як працюють атаки? [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://www.myrasecurity.com/en/what–is–a–ddos–attack/.

7. Можливості DDoS атак [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://www.sangfor.com/glossary/cybersecurity/ddos–attack.

8. Захист від DDoS атак [Електронний ресурс]. – 2013. – Режим доступу до ресурсу: https://www.eset.com/int/distributed–denial–of–service/.

9. Можлива стійкість веб–ресурсів [Електронний ресурс]. – 2012. – Режим доступу до ресурсу https://www.cisa.gov/uscert/ncas/tips/ST04–015.

10. Захищеність веб–ресурсів від DDoS атак [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.researchgate.net/figure/ Comparison–of–DDoS AttackTools_tbl1_317222214.

11. Можливість проведення захисту [Електронний ресурс]. – 20120. – Режим доступу до ресурсу: https://www.imperva.com/learn/ddos/denial–of–service.

12. Які типи атак [Електронний ресурс]. – 20117 – Режим доступу до ресурсу: https://uk.javascript.info/array–methods.

13. Грайворонський М. В. Безпека інформаційно–комунікаційних систем: підручник/ М. В. Грайворонський, О. М. Новіков ; За заг. ред. М.З. Згуровського. – К. : Видавнича група BHV, 2009. – 608 с.

14. Шаньгін В. Ф. Захист комп'ютерної інформації. Ефективні методи та засоби / В. Ф. Шаньгін. – М.: ДМК Прес, 2012. – 544 с

15. Iptables Tutorial 1.1.19 / Oskar Andreasson – 2020.– https://www.frozentux.net/documents/iptables–tutorial/

16. Alexander Gutnikov – November 11, 2019 – https://securelist.com/ddos–report–q3– 2019/94958/

17. DDoS Protection With IPtables: The Ultimate Guide – https://javapipe.com/blog/iptables–ddos–protection/

18. Tao Peng, C. Leckie, K. Ramamohanarao Protection from distributed denial of service attacks using history–based IP filtering / Tao Peng, C. Leckie, K. Ramamohanarao // IEEE International Conference on Communications – 2015. ICC'15.

19. PHP Manual / Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy, Jakub Vrana – 2020–05–26 – https://www.php.net/manual/en/

20. Denning D. An intrusion–detection model. In Proc. // IEEE Symposium on Security and Privacy. – 1987. – Vol. 13, No 2. – P. 222–232.

21. Sheyner O. Scenario Graphs and Attack Graphs. / PhD thesis, SCS, Pittsburgh: Carnegie Mellon University. – 2014. – P. 141.

22. Edward G. Intrusion Detection. 1st ed., Intrusion.Net Books / New Jersey:Sparta.– 2014. – P. 218.

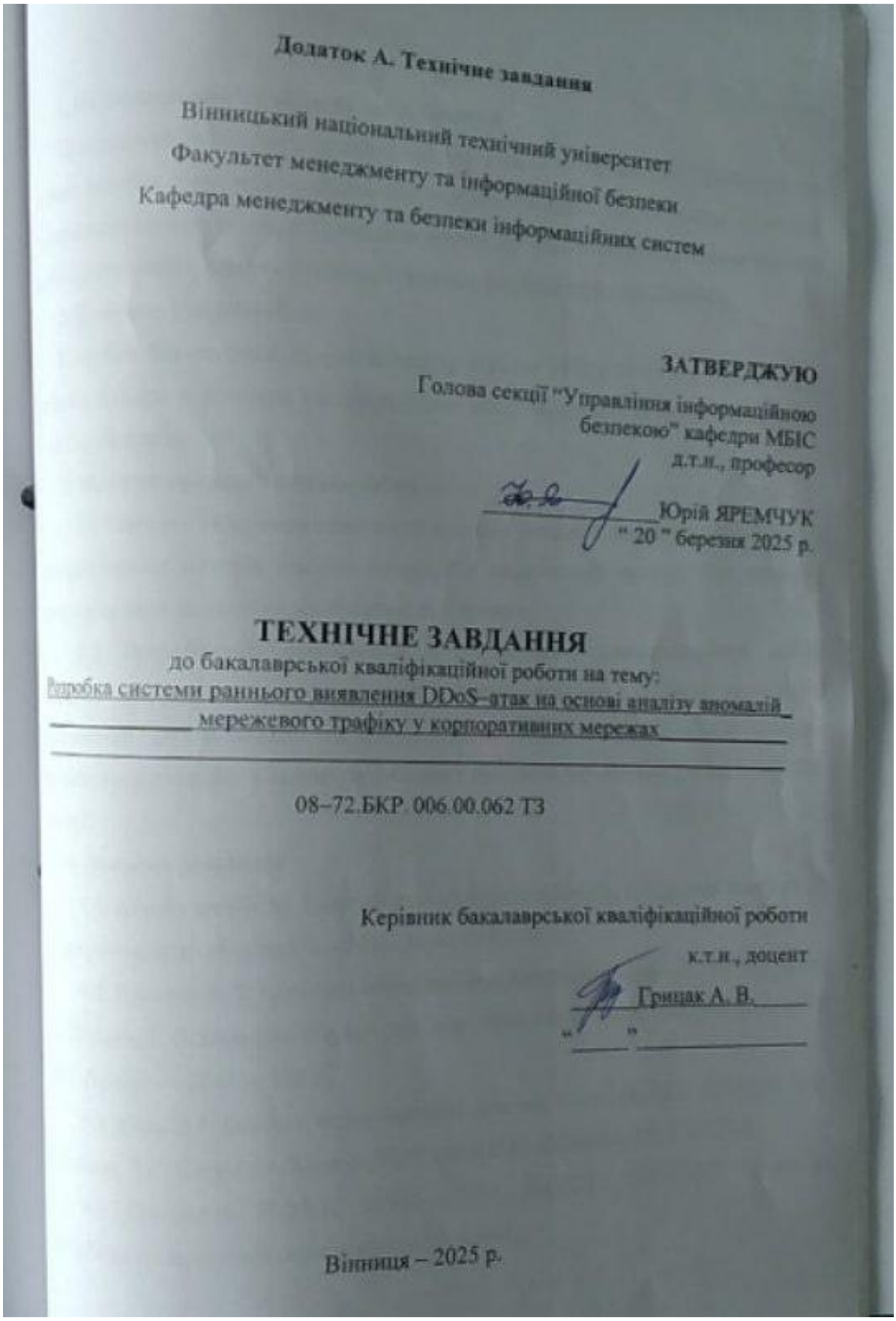
23. Гаврилов А.В. Застосування нейронних мереж, що постійно модифікуються, для захисту програмного забезпечення / О.В. Гаврилов // Нейрокомп'ютери: технологія, застосування. – 2015. – № 1–2. – С. 90–101.

24. Оладько В.С. Причини та джерела мережевих аномалій / В.С. Оладько, С.Ю. Мікова, М.А. Нестеренко, Є.О. Садівник// Молодий учений. – 2015. – №22. – С. 158–161.

25. M. Salman, et al., "DDoS Attack Detection and Mitigation Using Deep Learning Techniques," International Journal of Advanced Computer Science and Applications, 2018.
26. A. Garg, P. Kaur, "DDoS Attack Detection Using Machine Learning Techniques: A Review," International Journal of Advanced Research in Computer Science, 2018.
27. N. Panda, et al., "Deep Learning Techniques for DDoS Attack Detection and Mitigation: A Survey," International Journal of Distributed Sensor Networks, 2019.
28. V. Pham, et al., "DNN–Based DDoS Attack Detection Using IP Flow Traffic," IEEE International Conference on Advanced Information Networking and Applications, 2018.
29. S. Farhan, et al., "DDoS Attack Detection Using Deep Learning: A Comparative Study," IEEE International Conference on Advanced Information Networking and Applications, 2020.
30. Moustafa, et al., "DDoS Detection and Classification Using Deep Learning Models," IEEE Transactions on Dependable and Secure Computing, 2020.
31. A. Al–Tarawneh, et al., "Deep Learning–Based Intrusion Detection for DDoS Attacks in Software–Defined Networking," IEEE Access, 2019.
32. S. Hrushak, C. Pavlenko // Computer and information systems and technologies (Kharkiv, Ukraine, 2020). – 2020.
33. Hrushak S., Pavlenko C. Virtual Private Network and its use in secured corporative networks / S. Hrushak, C. Pavlenko // Computer and information systems and technologies (Kharkiv, Ukraine, 2019). – 2019.
34. Реалізація LSTM за допомогою einops та PyTorch [Електронний ресурс]. – Режим доступу до ресурсу: https://medium.com/@kyeg/implementing–lstms–with–einops–and–pytorch–9f2e6e5e8f56
35. Metrics and scoring: quantifying the quality of predictions [Електронний ресурс]. – Режим доступу до ресурсу: https://scikit–learn.org/stable/modules/model_evaluation.html
36. Kafka 4.0 Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://kafka.apache.org/documentation/

ДОДАТКИ

Додаток А. Технічне завдання



Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
Юрій ЯРЕМЧУК
" 20 " березня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ
до бакалаврської кваліфікаційної роботи на тему:
Розробка системи раннього виявлення DDoS-атак на основі аналізу аномалій мережевого трафіку у корпоративних мережах

08-72.БКР.006.00.062 ТЗ

Керівник бакалаврської кваліфікаційної роботи
к.т.н., доцент
Грицак А. В.
" "

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб для реалізації методу раннього виявлення DDoS–атак на основі аналізу аномалій мережевого трафіку. Область застосування – захист інформаційних ресурсів корпоративних мереж від атак типу «розподілена відмова в обслуговуванні» шляхом інтеграції системи в інфраструктуру безпеки.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 97 від 20.03.2025 р. згідно з планом підготовки кваліфікаційних робіт бакалаврів за спеціальністю 125 – «Кібербезпека».

3. Мета та призначення розробки

3.1 Метою є створення ефективної системи раннього виявлення DDoS–атак з використанням методів аналізу аномалій у мережевому трафіку, що дозволяє ідентифікувати загрозу на ранніх стадіях її прояву.

3.2 Розроблений програмний засіб виконує автоматизований аналіз мережевого трафіку в корпоративній мережі, виявляє підозрілу активність, притаманну DDoS–атакам, та формує відповідні повідомлення або виконує автоматичну реакцію з метою запобігання негативному впливу на інформаційні системи.

4. Джерела розробки

4.1. Ахрамович В. М. Ідентифікація й аутентифікація, керування доступом // Сучасний захист інформації. – 2016. №4.– С. 47–51.

4.2. Бурячок В.Л. Політика інформаційної безпеки: підручник. / В.Л.Бурячок, Р.В.Грищук, В.О.Хорошко / За заг. ред. докт. техн. наук, проф. В.О. Хорошка. – К.: ПВП «Задруга», 2014. – 222 с.

4.3. Єсін В.І. Безпека інформаційних систем і технологій / В.І.Єсін, О.О. Кузнецов, Л.С. Сорока. – Харків: ХНУ імені В.Н. Каразіна, 2013. – 632 с.

4.4 Стандарти ISO/IEC 27035–1:2016, ISO/IEC 27002:2022 – методи управління інцидентами інформаційної безпеки.

4.5 Власні експериментальні результати з навчання LSTM–моделі на симульованому мережевому трафіку.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний інтерфейс користувача, що дозволяє налаштовувати параметри аналізу трафіку, переглядати звіти та візуалізувати аномалії;

5.1.2 Метод реалізації не має потребувати сторонніх платних або ліцензійних компонентів;

5.1.3 Програма повинна реалізовувати процес обробки трафіку, формування послідовностей даних, аналіз результатів роботи моделі та забезпечення механізмів реагування.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб має стабільно працювати у фоновому режимі. У разі виникнення помилок чи критичних ситуацій повинно виводитись інформативне повідомлення;

5.2.2 Робота з базою зберігання логів повинна передбачати можливість резервного копіювання;

5.2.3 Програма має виконувати свою основну функцію – виявлення DDoS–атак – у режимі реального часу.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows або Linux;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 У програмному засобі передбачено лише технічну документацію, що описує архітектуру, принцип роботи моделі та порядок запуску в середовищі

розробки. Інструкції для кінцевого користувача не надаються, оскільки система призначена для використання фахівцями з кібербезпеки.

7. Вимоги до технічного захисту інформації

7.1 Програмний засіб повинен бути захищений від несанкціонованого використання. Необхідно забезпечити автентифікацію користувача, а також неможливість доступу до результатів аналізу або до системних налаштувань особами, що не пройшли авторизацію.

7.2 Доступ до журналів подій і критичних функцій повинен бути обмежений правами доступу.

8. Техніко–економічні показники

8.1 Економічна доцільність розробки обґрунтовується високою цінністю результату – система може виявити загрозу до того, як вона завдасть шкоди. Це знижує витрати на усунення наслідків атак та підвищує загальний рівень кіберстійкості організації.

8.2 Реалізація проекту не потребує значних витрат і може бути адаптована для використання як у великих компаніях, так і в малих організаціях без потреби у дорогому обладнанні.

9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми		
2.	Аналіз предметної області обраної теми		
3.	Розробка алгоритму роботи		
4.	Написання бакалаврської роботи на основі розробленої теми		
5.	Передзахист бакалаврської кваліфікаційної роботи		
6.	Виправлення, уточнення, корегування бакалаврської кваліфікаційної роботи		
7.	Захист бакалаврської кваліфікаційної роботи		

68

10. Порядок контролю та прийому

10.1 До приймання бакалаврської кваліфікаційної роботи надається:

– ПЗ до бакалаврської кваліфікаційної роботи;

– демонстрація результату бакалаврської кваліфікаційної роботи;

– презентація;

– відзив керівника роботи;

– відзив рецензента

Технічне завдання до виконання прийняв _____Жолукевський Р.В.

Додаток Б. Лістенинг коду


```
 {
"cell_type": "code",
"execution_count": 14,
"id": "df2601e1",
"metadata": {},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "Index(['Label'], dtype='object')\n",
   "Index(['Label'], dtype='object')\n",
   "Data type of each column of Dataframe :\n",
   "<class 'pandas.core.frame.DataFrame'>\n",
   "Int64Index: 1152382 entries, 2125399 to 122045\n",
   "Data columns (total 78 columns):\n",
   " #   Column           Non-Null Count    Dtype  \n",
   "---  ------           --------------    ----- \n",
   " 0   Src_Port         1152382 non-null  int64  \n",
   " 1   Dst_Port         1152382 non-null  int64  \n",
   " 2   Protocol         1152382 non-null  int64  \n",
   " 3   Flow_Duration     1152382 non-null  int64  \n",
   " 4   Tot_Fwd_Pkts      1152382 non-null  int64  \n",
   " 5   Tot_Bwd_Pkts      1152382 non-null  int64  \n",
   " 6   TotLen_Fwd_Pkts    1152382 non-null  float64\n",
   " 7   TotLen_Bwd_Pkts    1152382 non-null  float64\n",
   " 8   Fwd_Pkt_Len_Max    1152382 non-null  float64\n",
   " 9   Fwd_Pkt_Len_Min    1152382 non-null  float64\n",
   " 10  Fwd_Pkt_Len_Mean   1152382 non-null  float64\n",
   " 11  Fwd_Pkt_Len_Std    1152382 non-null  float64\n",
   " 12  Bwd_Pkt_Len_Max    1152382 non-null  float64\n",
   " 13  Bwd_Pkt_Len_Min    1152382 non-null  float64\n",
   " 14  Bwd_Pkt_Len_Mean   1152382 non-null  float64\n",
   " 15  Bwd_Pkt_Len_Std    1152382 non-null  float64\n",
   " 16  Flow_IAT_Mean     1152382 non-null  float64\n",
   " 17  Flow_IAT_Std      1152382 non-null  float64\n",
   " 18  Flow_IAT_Max      1152382 non-null  float64\n",
   " 19  Flow_IAT_Min      1152382 non-null  float64\n",
   " 20  Fwd_IAT_Tot       1152382 non-null  float64\n",
   " 21  Fwd_IAT_Mean      1152382 non-null  float64\n",
   " 22  Fwd_IAT_Std       1152382 non-null  float64\n",
   " 23  Fwd_IAT_Max       1152382 non-null  float64\n",
   " 24  Fwd_IAT_Min       1152382 non-null  float64\n",
   " 25  Bwd_IAT_Tot       1152382 non-null  float64\n",
   " 26  Bwd_IAT_Mean      1152382 non-null  float64\n",
```

```
" 27  Bwd_IAT_Std       1152382 non-null  float64\n",
" 28  Bwd_IAT_Max       1152382 non-null  float64\n",
" 29  Bwd_IAT_Min       1152382 non-null  float64\n",
" 30  Fwd_PSH_Flags     1152382 non-null  int64  \n",
" 31  Bwd_PSH_Flags     1152382 non-null  int64  \n",
" 32  Fwd_URG_Flags     1152382 non-null  int64  \n",
" 33  Bwd_URG_Flags     1152382 non-null  int64  \n",
" 34  Fwd_Header_Len    1152382 non-null  int64  \n",
" 35  Bwd_Header_Len    1152382 non-null  int64  \n",
" 36  Fwd_Pkts/s        1152382 non-null  float64\n",
" 37  Bwd_Pkts/s        1152382 non-null  float64\n",
" 38  Pkt_Len_Min       1152382 non-null  float64\n",
" 39  Pkt_Len_Max       1152382 non-null  float64\n",
" 40  Pkt_Len_Mean      1152382 non-null  float64\n",
" 41  Pkt_Len_Std       1152382 non-null  float64\n",
" 42  Pkt_Len_Var       1152382 non-null  float64\n",
" 43  FIN_Flag_Cnt      1152382 non-null  int64  \n",
" 44  SYN_Flag_Cnt      1152382 non-null  int64  \n",
" 45  RST_Flag_Cnt      1152382 non-null  int64  \n",
" 46  PSH_Flag_Cnt      1152382 non-null  int64  \n",
" 47  ACK_Flag_Cnt      1152382 non-null  int64  \n",
" 48  URG_Flag_Cnt      1152382 non-null  int64  \n",
" 49  CWE_Flag_Count    1152382 non-null  int64  \n",
" 50  ECE_Flag_Cnt      1152382 non-null  int64  \n",
" 51  Down/Up_Ratio     1152382 non-null  float64\n",
" 52  Pkt_Size_Avg      1152382 non-null  float64\n",
" 53  Fwd_Seg_Size_Avg   1152382 non-null  float64\n",
" 54  Bwd_Seg_Size_Avg   1152382 non-null  float64\n",
" 55  Fwd_Byts/b_Avg    1152382 non-null  int64  \n",
" 56  Fwd_Pkts/b_Avg    1152382 non-null  int64  \n",
" 57  Fwd_Blk_Rate_Avg   1152382 non-null  int64  \n",
" 58  Bwd_Byts/b_Avg    1152382 non-null  int64  \n",
" 59  Bwd_Pkts/b_Avg    1152382 non-null  int64  \n",
" 60  Bwd_Blk_Rate_Avg   1152382 non-null  int64  \n",
" 61  Subflow_Fwd_Pkts   1152382 non-null  int64  \n",
" 62  Subflow_Fwd_Byts   1152382 non-null  int64  \n",
" 63  Subflow_Bwd_Pkts   1152382 non-null  int64  \n",
" 64  Subflow_Bwd_Byts   1152382 non-null  int64  \n",
" 65  Init_Fwd_Win_Byts  1152382 non-null  int64  \n",
" 66  Init_Bwd_Win_Byts  1152382 non-null  int64  \n",
" 67  Fwd_Act_Data_Pkts  1152382 non-null  int64  \n",
" 68  Fwd_Seg_Size_Min   1152382 non-null  int64  \n",
" 69  Active_Mean       1152382 non-null  float64\n",
" 70  Active_Std        1152382 non-null  float64\n",
" 71  Active_Max        1152382 non-null  float64\n",
" 72  Active_Min        1152382 non-null  float64\n",
" 73  Idle_Mean         1152382 non-null  float64\n",
" 74  Idle_Std          1152382 non-null  float64\n",
" 75  Idle_Max          1152382 non-null  float64\n",
```

```
    " 76  Idle_Min          1152382 non-null  float64\n",
    " 77  Label             1152382 non-null  object \n",
    "dtypes: float64(43), int64(34), object(1)\n",
    "memory usage: 694.6+ MB\n"
   ]
  }
 ],
 "source": [
  "cat_cols = df.select_dtypes(include=['object', 'category']).columns\n",
  "print(cat_cols)\n",
  "\n",
  "cat_cols = df.select_dtypes(include='object').columns\n",
  "print(cat_cols)\n",
  "\n",
  "print('Data type of each column of Dataframe :')\n",
  "df.info(verbose=True)"
 ]
},
{
 "attachments": {},
 "cell_type": "markdown",
 "id": "7fd8b2dc",
 "metadata": {},
 "source": [
  "This step is creating a numpy array of the selected independent variables from the training dataset, which will be used to train the SVM model. The purpose of creating a numpy array is to ensure that the data is in a format that can be efficiently processed by the machine learning algorithm. The numpy array created here will have 15,000 rows (since we selected 15,000 rows from the original dataset) and 79 columns (since we selected 79 columns from the original dataset as features)."
 ]
},
```

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи:

Розробка системи раннього виявлення DDoS-атак на основі аналізу аномалій мережевого трафіку у корпоративних мережах

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ Кафедра менеджменту на безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
Гр. 2КІТС-21б

Керівник доцент Грицак А.В. _____

Показники звіту подібності

Strike Plagiarism	
Оригінальність	71,31%
Загальна схожість	28,69%

Аналіз звіту подібності (відмітити потрібне)

- ☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Жолукевський Р.В.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення

- ☐ **Допустити до захисту**

_____
_____
_____
_____

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Експерт _____ _____
(за потреби) (підпис) (прізвище, ініціали, посада)