Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Гребенчука Валентина Максимовича

на здобуття ступеня вищої освіти Бакалавра

Система виявлення атаки на відновлення паролю у мережі

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ.200103.20.01.05 ПЗ

Виконав студент 4 курсу група КБ-20-1 Валентин ГРЕБЕНЧУК
Керівник канд. техн. наук, доцент Вікторія ОРЛЕНКО
Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

12 06 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гребенчуку Валентину Максимовичу

1 Тема роботи Система виявлення атаки на відновлення паролю у мережі

Керівник роботи к.т.н. доцент Вікторія ОРЛЕНКО

Затверджено наказом ректора університету від 15 лютого 2024 № 8

2 Строк подання студентом кваліфікаційної роботи на кафедру 12.06.2024

3 Вихідні дані до роботи Проаналізувати особливості реалізації атак на відновлення паролю у мережі. Обрати тип нейронної мережі для ідентифікації зловмисного трафіку. Розробити алгоритм виявлення атаки. Підготувати набори даних для навчання нейронної мережі. Здійснити навчання нейронної мережі. Реалізувати систему виявлення атак. Розробити тестове середовище. Оцінити ефективність розробленої системи.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Вступ. Огляд предметної області. Засоби реалізації атак. Засоби захисту від атак на паролі. Розробка алгоритмів реалізації та підготовки даних. Вибір нейронної мережі. Алгоритм виявлення атаки у мережі. Система виявлення атаки на відновлення паролю у мережі. Опис реалізації. Розгортання тестового середовища. Оцінка ефективності.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Архітектура нейронної мережі та підготовка наборів даних, Алгоритми навчання нейронної мережі, підготовки даних та аналізу мережевого трафіку. Оцінка ефективності системи виявлення атаки на відновлення паролю у мережі.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В., старший викладач кафедри кібербезпеки		

7 Дата видачі завдання 16 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студент Валентин ГРЕБЕНЧУК

Керівник кваліфікаційної роботи Вікторія ОРЛЕНКО

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система виявлення атаки на відновлення паролю у мережі»

Автор роботи: студент групи КБ–20–1 Гребенчук В.М.

Керівник роботи: к.т.н. доц. Орленко В.С.

Пояснювальна записка: 63с., 24 рисунки, 8 таблиць, 45 джерел, 3 креслення.

ПЕРЕЛІК КЛЮЧОВИХ СЛІВ: НЕЙРОННА МЕРЕЖА, АТАКА НА ПАРОЛЬ, GRU, НАБІР ДАНИХ, НАВЧАННЯ НЕЙРОМЕРЕЖІ.

У кваліфікаційній роботі розроблено систему виявлення атаки на відновлення паролю в мережі, використовуючи рекурентну нейронну мережу для аналізу мережевого трафіку. Було вивчено структуру та особливості трьох наборів даних: KDDCup99, UNSW-NB15 та CICIDS2017, які використовувалися для тренування і тестування моделі. Це дозволило забезпечити надійну оцінку продуктивності розробленої системи. Розроблено алгоритм, що використовує RNN для виявлення атак на відновлення паролю. Використання TensorFlow та інших інструментів машинного навчання забезпечило створення гнучкої та потужної моделі, здатної ефективно аналізувати великі обсяги мережевого трафіку. Модель пройшла навчання на обраних наборах даних та тестування у змодельованих і реальних умовах. Порівняння результатів тестування на різних наборах даних показало стабільність та надійність системи, що робить її придатною для використання у реальних мережах.

10.06.2024

ABSTRACT

Topic of the qualification work: "Network password recovery attack detection system"

Author: student of group KB-20-1, Grebenchuk V. M.

Supervisor: Ph.D., Associate Professor Orlenko V. S.

Explanatory note: 63 pages, 24 figures, 8 tables, 45 sources, 3 diagrams.

LIST OF KEYWORDS: NEURAL NETWORK, PASSWORD ATTACK, GRU, DATASET, NEURAL NETWORK TRAINING.

This qualification work develops a system for detecting password recovery attacks in a network using a recurrent neural network to analyze network traffic. The structure and characteristics of three datasets—KDDCup99, UNSW-NB15, and CICIDS2017—were studied for training and testing the model, providing a reliable performance evaluation of the developed system. An algorithm using an RNN for detecting password recovery attacks was developed. Utilizing TensorFlow and other machine learning tools ensured the creation of a flexible and powerful model capable of efficiently analyzing large volumes of network traffic. The model was trained on selected datasets and tested in both simulated and real-world conditions. Comparing test results across different datasets demonstrated the system's stability and reliability, making it suitable for use in real networks.

10.06.2024

ЗМІСТ

ВСТУП

У сучасному світі, зростання цифрових технологій супроводжується збільшенням кількості та складності кіберзагроз, зокрема атак на відновлення паролів. Атаки на відновлення паролів є однією з найпоширеніших форм кіберзлочинів, які загрожують конфіденційності, цілісності та доступності інформаційних систем. Такі атаки можуть мати серйозні наслідки для організацій, включаючи фінансові втрати, втрату довіри клієнтів та порушення норм регуляторних вимог. Виявлення та запобігання цим атакам стає важливим завданням для забезпечення безпеки інформаційних ресурсів.

Основна проблема, яка потребує вирішення, полягає у високій складності ідентифікації спроб підбору чи злому паролів у реальному часі. Атакуючі постійно вдосконалюють свої методи, використовуючи автоматизовані інструменти та боти, що ускладнює завдання для традиційних систем захисту. Сучасні методи виявлення часто базуються на статичних правилах або простих евристиках, які не можуть ефективно виявляти нові типи атак або адаптуватися до змін у поведінці зловмисників. Це створює нагальну потребу у розробці нових, більш ефективних методів виявлення атак на відновлення паролів.

Метою даної кваліфікаційної роботи є розробка системи виявлення атак на відновлення паролів у мережі, яка використовує нейронні мережі для аналізу мережевого трафіку.

До основних завдань роботи можна віднести: аналіз існуючих методів виявлення атак на паролі та обґрунтування вибору методу для вирішення поставленої проблеми; вивчення структури та особливостей наборів даних, які використовуватимуться для тренування та тестування моделі системи; розробка алгоритму для виявлення атак на відновлення паролів; проведення навчання та тестування розробленої моделі на обраних наборах даних; доведення ефективності розробленої системи.

АТАКИ НА ВІДНОВЛЕННЯ ПАРОЛЮ

1.1 Типи атак на паролі

Атаки на паролі - це спроби несанкціонованого доступу до систем за допомогою вгадування, відновлення або обходу паролів користувачів [1]. До найпоширеніших типів атак на паролі відносять [2-5]:

- підбір (Brute-Force Attack);
- атаки за словником (Dictionary Attack);
- атаки з використанням досягнень (Credential Stuffing);
- атаки за райдужними таблицями (Rainbow Table);
- атаки з використанням соціальної інженерії;
- кейлогінг (Keylogging);
- фішинг (Phishing);
- "підглядання через плече" (Shoulder Surfing);
- підбір за допомогою масок (Mask Attack);
- «передача хешу» (Pass the Hash).

Метод підбору пароля полягає у вгадуванні пароля шляхом автоматизованого перебору всіх можливих комбінацій символів до тих пір, поки не буде знайдено правильного пароля. Цей метод використовує значні обчислювальні ресурси і може бути дуже ефективним, якщо паролі є простими або короткими. Однак, він стає менш ефективним з ростом довжини та складності пароля.

Тип атаки за словником включає в себе використання списку поширених слів або фраз (як правило, зібраних у «словнику» за певними критеріями) для спроб вгадування пароля. Цей метод базується на припущенні, що багато користувачів встановлюють свої паролі з використанням легко запам'ятовуваних слів або популярних фраз, таких як «password», «123456» або імена власних дітей чи домашніх тварин [6-7]. Словникові атаки значно швидші, ніж методи brute

force, оскільки вони обмежуються більш імовірними варіантами замість перебору всіх можливих комбінацій символів. Для підвищення ефективності, словники можуть включати варіації типу leetspeak (наприклад, заміна літери "a" на "4", "e" на "3"). Основні методи захисту від словникових атак включають використання довгих і складних паролів, які містять великі і малі літери, цифри і спеціальні символи, а також застосування двофакторної аутентифікації і обмеження кількості спроб введення пароля.

При атаках з використанням досягнень зловмисники використовують викрадені імена користувачів і паролі з одного джерела і пробують використати їх на інших сайтах, сподіваючись, що люди повторно використовують однакові дані для входу. Цей тип атаки відомий як "credential stuffing" і базується на припущенні, що багато користувачів використовують один і той самий пароль для різних облікових записів. Зловмисники часто отримують ці дані з попередніх зломів або витоків баз даних, що містять облікові дані мільйонів користувачів. Атаки можуть бути дуже ефективними, оскільки багато людей мають схильність до використання одного й того ж пароля для кількох сервісів, щоб уникнути необхідності запам'ятовувати численні паролі. Такий підхід знижує безпеку облікових записів, оскільки злам одного сервісу може призвести до компрометації облікових записів на інших платформах.

Rainbow table - це велика таблиця відповідностей між хешем пароля і його можливими значеннями. Зловмисники можуть швидко перевірити хеш, щоб знайти відповідний пароль, якщо хешування виконано без "солі" (додаткові випадкові дані). Rainbow tables значно знижують час, необхідний для зламування паролів, оскільки дозволяють уникнути перебору всіх можливих паролів. Замість цього зловмисник просто перевіряє хеш проти готової таблиці. Цей тип атаки ефективний проти хешованих паролів, де не використовуються додаткові заходи безпеки, такі як сіль. Сіль (salt) - це випадкові дані, що додаються до пароля перед його хешуванням. Вона забезпечує унікальність хешів навіть для однакових паролів і значно ускладнює використання rainbow tables, оскільки кожен унікальний хеш вимагає окремої таблиці відповідностей. Rainbow table атаки є

особливо небезпечними для систем, де не застосовуються сучасні методи безпеки для збереження паролів [8-9].

Метод атаки з використанням соціальної інженерії включає в себе маніпуляцію людьми для викриття їх паролів. Це може включати фішингові атаки, в яких користувачів обманом спонукають до надання своїх облікових даних. Зловмисники можуть надсилати електронні листи або створювати підроблені вебсайти, що виглядають як офіційні ресурси, з метою зібрати особисту інформацію [10-12]. Окрім фішингових атак, метод соціальної інженерії також включає:

– прямий обман, коли зловмисники можуть особисто звертатися до користувачів, видаючи себе за співробітників служби підтримки або інших авторитетних осіб, і просити їх надати свої паролі або іншу конфіденційну інформацію;

– телефонні шахрайства під час яких зловмисники можуть телефонувати користувачам і, представляючись представниками компаній, просити їх надати свої облікові дані або іншу інформацію, що може бути використана для доступу до системи;

– соціальні мережі, де можна отримати та в подальшому використовувати інформацію, доступну в соціальних мережах, для створення довірливих відносин з метою вивідати паролі або відповіді на секретні питання;

– використання фізичних пристроїв, таких як заражені USB-накопичувачі, які можуть залишатися в загальнодоступних місцях навмисне. Користувач, знайшовши такий пристрій і підключивши його до свого комп'ютера, може не помітити, що він завантажує шкідливе програмне забезпечення, що передає його паролі зловмиснику.

Соціальна інженерія часто використовує психологічні прийоми, такі як виклик довіри, страху або терміновості, щоб маніпулювати жертвами. Наприклад, у фішингових атаках зловмисники можуть використовувати повідомлення, які попереджають про термінову проблему з обліковим записом, змушуючи користувача негайно ввести свої дані.

Keylogger - це тип шпигунського програмного забезпечення, яке відстежує та записує натискання клавіш користувача, часто без їх знання, дозволяючи атакуючим збирати паролі та іншу конфіденційну інформацію [13-16].

Фішинг - це вид соціальної інженерії, де користувачам надсилаються підроблені електронні листи чи повідомлення, які імітують легітимні запити від надійних організацій для збору паролів. Фішинг також може включати спроби використання підроблених веб-сайтів, які схожі на офіційні веб-сайти банків, інтернет-магазинів або інших організацій, з метою викликати в користувачів довіру і отримати їхні особисті дані, такі як номери кредитних карток або адреси електронної пошти [17-20]. Ці атаки можуть також включати в себе використання зловмисних вкладень у електронних листах, спрямованих на вразливості в програмному забезпеченні, щоб отримати несанкціонований доступ до комп'ютера чи мережі користувача.

Метод "підглядання через плече" (shoulder surfing) є формою непублічного вторгнення, де зловмисник отримує доступ до конфіденційної інформації, спостерігаючи за діями жертви без її відома. Цей метод може використовуватися для отримання паролів, ПІН-кодів, номерів кредитних карток та інших чутливих даних. Зловмисник може фізично присутнім у тому ж приміщенні, що і жертва, або використовувати відеоспостереження або інші технічні засоби для здійснення спостереження. Цей метод атаки може бути особливо ефективним у громадських місцях, де жертва вводить чутливу інформацію у свої пристрої чи системи.

Підбір за допомогою масок (mask attack) є методом атак на паролі, що використовує попередньо відомі шаблони або правила для скорочення кількості варіантів, які потрібно перевірити. Зловмисник може знати, що паролі часто дотримуються певних стандартів, таких як: починаються з великої літери; містять певну кількість символів; закінчуються числом; містять спеціальні символи на певних позиціях. Використовуючи ці правила, зловмисник створює маску, яка включає всі можливі комбінації символів, що відповідають цим критеріям. Наприклад, маска "Aaaaaaaa9" може означати пароль, що починається з великої літери, за якою слідують вісім маленьких літер, а закінчується цифрою. Це

дозволяє значно скоротити кількість можливих паролів, які потрібно перевірити, що робить атаку більш ефективною. Атака за допомогою масок є потужною технікою, особливо коли зловмисник має певну інформацію про жертву або її поведінкові звички при створенні паролів, що дозволяє ще точніше налаштувати маски для підбору.

Атака «передача хешу» полягає у використанні хешу пароля замість самого пароля для отримання несанкціонованого доступу до системи. Цей метод дозволяє зловмиснику аутентифікуватися в системі, не знаючи фактичного пароля, а лише хеш, який його представляє. Зловмисник отримує доступ до хешів паролів, які зберігаються на комп'ютері або сервері. Це може бути здійснено через різні методи, такі як витягнення хешів з пам'яті, файл системи або через інші форми доступу до даних. Отримавши хеш, зловмисник використовує його для аутентифікації на інших системах, де використовується той самий хеш для доступу до облікового запису. Це робиться за допомогою спеціальних інструментів, які дозволяють передавати хеш в процесі автентифікації замість реального пароля. Система приймає хеш як дійсний пароль, що дозволяє зловмиснику отримати доступ до ресурсів, які захищені цим обліковим записом [21-23]. Цей тип атаки особливо ефективний у середовищах Windows, де хеші паролів можуть бути легко доступними у відносно незахищених місцях.

Захист від таких атак включає в себе використання сильних, унікальних паролів, двофакторної автентифікації, шифрування паролів з використанням "солі", а також інформування користувачів щодо безпечних практик використання паролів і загроз соціальної інженерії.

Атаки на відновлення паролю (password recovery attacks) — це методи, які зловмисники використовують для обходу звичайного процесу відновлення або скидання паролів з метою незаконного доступу до чужих облікових записів.

1.2 Засоби реалізації атак

Для отримання доступу до вікна скидання паролю зловмисник може

застосовувати наступні техніки:

– соціальної інженерії, щоб обманом змусити жертву розкрити відповіді на свої питання безпеки або іншу особисту інформацію, яку можна використовувати для скидання пароля. Отримавши ці дані, зловмисник може отримати доступ до облікового запису жертви і, можливо, розширити свої права, змінивши налаштування або отримавши доступ до інших облікових записів, пов'язаних з обліковим записом жертви;

– перехопити активний сеанс користувача на вебсайті або в додатку та використати його для скидання пароля користувача. Якщо у користувача є вищий рівень прав на вебсайті або в додатку, зловмисник також може отримати такі права;

– якщо вебсайт або додаток уразливий до SQL-ін'єкції, зловмисник може використовувати цю техніку для обходу аутентифікації та отримання доступу до системи або облікового запису без необхідності дійсного пароля. Отримавши доступ, він може спробувати розширити свої права, змінивши налаштування користувача або отримавши доступ до конфіденційної інформації;

– використовувати вразливості у механізмах скидання пароля для доступу до чужих облікових записів. Наприклад, вони можуть перехопити електронні листи або SMS-повідомлення для скидання пароля, або можуть використати слабкі місця у безпекових питаннях, що використовуються для скидання пароля;

– вразливості у веб-додатках для отримання доступу до облікових записів користувачів. Наприклад, вони можуть застосувати атаку з міжсайтовим скриптингом (XSS) для введення шкідливого коду на вебсайт і крадіжки користувацьких облікових даних, або вони можуть використовувати уразливості завантаження файлів для завантаження зворотного шелу, що дозволить отримати доступ до сервера.

Для підбору паролю зловмисники можуть застосовувати різні техніки, як ручні, так і автоматизовані. Слід зазначити, що більшість інструментів призначені для навчальних цілей, проте активно використовуються зловмисниками при реалізації атак. Програмні продукти відрізняються особливостями реалізації,

типами жертв та підходами до здійснення атаки.

SET — це інструмент, розроблений, щоб допомогти тестувальникам проникнення та фахівцям із безпеки тестувати атаки соціальної інженерії. Він включає низку векторів атак, у тому числі збирання облікових даних і фішингові атаки, а також імітацію атак з використанням зловмисного програмного забезпечення та аудит безпеки.

Burp Suite — це інтегрований пакет інструментів для тестування безпеки веб-додатків, який містить ряд функцій для виявлення та використання вразливостей, включаючи атаки для відновлення пароля, перехоплення трафіку і аналіз вразливостей сесій [24].

Hydra — це потужний інструмент злому паролів, який може виконувати атаки грубою силою на сторінки входу, захищені паролем каталоги та інші ресурси, підтримуючи численні протоколи, такі як SSH, FTP, HTTP, Telnet [25].

John the Ripper — це інструмент злому паролів, який може виконувати атаки грубою силою на хеші паролів, підтримуючи різноманітні методи хешування, включаючи MD5, SHA-1 та інші [26].

Medusa — це інструмент злому паролів, який може виконувати атаки грубою силою на сторінки входу, захищені паролем каталоги та інші ресурси, з використанням модульної структури для легкого додавання нових модулів [27].

Medusa-gui — графічний інтерфейс для Medusa, який полегшує виконання атак грубою силою, забезпечуючи більш інтуїтивно зрозумілий інтерфейс користувача.

Password Cracking Toolkit: PCT — це набір інструментів для злому паролів, який включає John the Ripper, THC-Hydra та інші, забезпечуючи інтегрований підхід до виявлення слабких паролів.

RainbowCrack — це інструмент для злому паролів, який може зламувати хеші паролів за допомогою попередньо обчислених веселкових таблиць, значно прискорюючи процес злому порівняно з традиційними методами грубої сили [28].

Cain and Abel — це інструмент для злому паролів, який може виконувати низку атак для відновлення пароля, включаючи атаки грубої сили, атаки за

словником і атаки з перехопленням, а також має засоби для аналізу протоколів мережі.

L0phtCrack — це інструмент для злому паролів, який може виконувати атаки грубою силою, атаки за словником та інші атаки на паролі Windows, надаючи оцінку міцності паролів та можливість автоматизації злому [29].

Maltego — це інструмент видобутку даних, який можна використовувати для збору інформації та розвідки під час атак для відновлення пароля, дозволяючи аналізувати зв'язки між елементами даних на великій кількості джерел [30].

Aircrack-ng — цей комплексний інструмент призначений для аудиту бездротових мереж, включаючи можливості злому парольних фраз WPA і WPA2 через методи грубої сили або використання слабкостей протоколу [31].

Brutus — це відомий інструмент для злому паролів, який ефективно виконує атаки грубою силою на веб-інтерфейси та інші захищені паролем ресурси.

THC-Hydra — потужний інструмент для злому паролів, що підтримує численні протоколи і дозволяє виконувати швидкі атаки грубою силою на захищені паролем об'єкти.

THC-Hydra-gtk — графічний інтерфейс для THC-Hydra, що робить процес злому паролів більш доступним і зручним для користувачів.

Metasploit Framework — це розширюваний фреймворк для проведення тестування на проникнення, що містить модулі для виявлення та використання вразливостей, у тому числі модулі для відновлення паролів через вразливості систем [32].

Ncrack — цей інструмент злому паролів спрямований на аудит парольних політик мережевих служб, таких як SSH, RDP, і FTP, за допомогою атак грубою силою.

SQLMap — автоматизований інструмент, який дозволяє ідентифікувати та експлуатувати вразливості SQL-ін'єкцій, які можуть бути використані для обходу аутентифікації та відновлення паролів.

Wfuzz — інструмент для фаззінгу веб-додатків, що дозволяє здійснювати перебір паролів і параметрів на веб-сторінках для ідентифікації вразливостей.

1.3 Шаблони атак у базі CAPEC

CAPEC, або Common Attack Pattern Enumeration and Classification, це загальнодоступна база даних, що збирає різні шаблони кібератак. Щодо Password recovery attacks, тобто атак на відновлення або злом паролів, деякі з зазначених CAPEC ідентифікаторів мають прямий зв'язок з методами викрадення або відновлення паролів [33]:

- CAPEC-102 (Brute Force);
- CAPEC-474 (Password Recovery Exploits);
- CAPEC-50 (Password Brute Forcing);
- CAPEC-509 (Brute Force Password Guessing);
- CAPEC-551 (Testing for Default Passwords);
- CAPEC-555 (Testing for Weak Passwords);
- CAPEC-560 (Password Brute Force with User Enumeration);
- CAPEC-600 (Password Cracking);
- CAPEC-644 (Forceful Browsing);
- CAPEC-645 (Reverse Brute Force Attack);
- CAPEC-652 (Password Spraying);
- CAPEC-653 (Credential Stuffing).

CAPEC-102 включає спроби всіх можливих комбінацій символів для злому пароля. Це може бути виконано автоматично за допомогою програмного забезпечення, що швидко генерує і перевіряє кожен можливий пароль до того, як правильний пароль буде знайдено.

Атака CAPEC-474 зосереджена на використанні слабкостей у системах відновлення паролів. Наприклад, атакування механізмів безпеки, які використовуються для захисту процесів відновлення або скидання паролів, таких як слабкі контрольні питання або вразливості в процедурі скидання пароля.

CAPEC-50 схоже на CAPEC-102, описує атаки злому паролів шляхом грубої сили, де атакувач перебирає всі можливі комбінації символів.

CAPEC-509 є прикладом атаки грубою силою, де зловмисник пробує багато різних паролів з надією знайти правильний.

Атака CAPEC-551 заснована на спробах входу в систему з використанням відомих стандартних паролів, які часто встановлені за замовчуванням і не змінюються користувачами.

CAPEC-555 показує аналіз того, чи можна зламати пароль через його слабкість, наприклад, якщо він короткий, складається лише з цифр або з слів, які легко підібрати.

CAPEC-560 поєднує атаки грубою силою з переліком користувачів, щоб збільшити шанси успіху, визначивши спершу дійсних користувачів системи.

CAPEC-600 слугує для опису злому паролів за допомогою різних методик.

CAPEC-644 не прямо пов'язано з відновленням паролів, але може бути використано для обходу аутентифікації або авторизації.

CAPEC-645 використовує відомий пароль, щоб визначити, чи можна його використати на інших акаунтах.

CAPEC-652 описує спроби підбору одного пароля проти багатьох користувачів для визначення, чи використовує хто-небудь з них цей загальний пароль.

CAPEC-653 базується на використанні раніше викрадених паролів та логінів для спроби входу в інші акаунти, розраховуючи на те, що люди часто використовують однакові паролі на різних платформах.

Ці шаблони атак допомагають розуміти, як зловмисники можуть намагатися отримати несанкціонований доступ до паролів і яким чином можна захистити системи від таких видів загроз.

1.4 Типові засоби захисту від атак на паролі

Щоб захистити паролі від взлому, зазвичай рекомендують комбінацію із кількох методів захисту, оскільки комбіновані стратегії забезпечують найкращий результат. Нижче наведено методи, які переважно використовуються з метою

захисту даних:

– використання складних та довгих паролів (мінімум 12 символів), які включають великі та маленькі літери, цифри та спеціальні символи, уникаючи загальних слів і фраз;

– використання різних паролів для різних облікових записів, що гарантуватиме безпеку інших профілів при компрометації одного із них;

– використання менеджерів паролів.;

– використання двофакторної аутентифікації (2FA);

– налаштування обмеження на кількість невдалих спроб входу дозволяє запобігти брутфорс-атакам;

– розгортання систем виявлення вторгнень у корпоративних сегментах;

– систематична зміна паролів;

– навчання користувачів основам безпечної поведінки в інтернеті, щоб уникнути фішингових атак і шкідливого програмного забезпечення.

– використання HTTPS;

– використання біометричної аутентифікації там, де це можливо.

Менеджери паролів — це програмні інструменти, що допомагають користувачам створювати, зберігати та управляти складними паролями для різних облікових записів. Вони зберігають паролі у зашифрованому вигляді, що забезпечує високий рівень безпеки. До переваг використання менеджерів паролів можна віднести: генерацію складних та унікальних паролів для кожного облікового запису, автоматичне заповнення полів для входу, зберігання паролів у зашифрованому вигляді. Серед популярних менеджерів паролів можна відмітити LastPass, 1Password та Dashlane. Менеджери паролів забезпечують не лише безпеку, але й зручність використання, оскільки користувачам не потрібно запам'ятовувати численні складні паролі. Менеджери паролів є потужними інструментами для зберігання та управління паролями, забезпечуючи зручність та безпеку для користувачів. Проте, як і будь-який інший інструмент, вони мають свої недоліки. Менеджер паролів зберігає всі ваші паролі в одному місці. Якщо зловмисник отримає доступ до вашого менеджера паролів, він зможе отримати

доступ до всіх ваших облікових записів. Це робить менеджер паролів єдиною точкою відмови. Хоча більшість менеджерів паролів використовують сильне шифрування для захисту паролів, вони все одно можуть бути вразливі до зламів. Наприклад, можуть бути знайдені вразливості в програмному забезпеченні або в хмарних сервісах, що зберігають зашифровані паролі. Менеджери паролів зазвичай захищені головним паролем. Якщо ви забули головний пароль, ви можете втратити доступ до всіх своїх паролів. Деякі менеджери паролів надають можливість відновлення доступу, але цей процес може бути складним і небезпечним. Менеджери паролів можуть бути обдурені фішинговими сайтами, які виглядають як легітимні сайти. Якщо ви введете свої облікові дані на такому сайті, менеджер паролів може зберегти ці дані, вважаючи їх легітимними. Зловмисники можуть використовувати методи соціальної інженерії, щоб змусити вас розкрити свій головний пароль або іншу інформацію, необхідну для доступу до менеджера паролів. Багато менеджерів паролів, особливо ті, що працюють у хмарі, потребують підключення до інтернету для синхронізації даних між різними пристроями. Це може бути проблемою, якщо у вас немає доступу до інтернету або якщо сервери менеджера паролів тимчасово недоступні. Будь-яке програмне забезпечення може містити помилки або вразливості. Менеджери паролів не є винятком. Нещодавні приклади показали, що навіть популярні менеджери паролів можуть мати вразливості, які можуть бути використані зловмисниками для отримання доступу до ваших паролів. Менеджери паролів можуть не завжди правильно інтегруватися з усіма веб-сайтами або додатками. Це може призвести до проблем з автозаповненням або збереженням паролів. Деякі веб-сайти можуть використовувати спеціальні методи захисту, які ускладнюють роботу менеджерів паролів. Менеджери паролів є корисними інструментами, які можуть значно підвищити безпеку та зручність використання паролів. Однак, вони також мають свої недоліки, про які користувачі повинні знати. Важливо використовувати менеджери паролів з обережністю, вибираючи надійні та перевірені рішення, та дотримуватися найкращих практик безпеки для захисту своїх даних.

Двофакторна аутентифікація додає додатковий рівень безпеки, вимагаючи

другу форму автентифікації на додаток до пароля. Це може бути одноразовий пароль (OTP), біометричні дані або апаратний токен. Методи 2FA: одноразовий пароль (OTP), який генерується додатком на телефоні або надсилається через SMS; біометрія у вигляді використання відбитків пальців, розпізнавання обличчя або голосу; апаратні токени, тобто застосування спеціальних пристроїв, які генерують коди для входу. До переваг 2FA можна віднести підвищену безпеку, навіть якщо пароль було зламано, та складність підробки другої форми аутентифікації. Проте існують і недоліки 2FA: необхідність додаткових дій від користувача, проблеми з аутентифікацією у деяких ситуаціях, наприклад, при відсутності мобільного сигналу.

У 2012 році LinkedIn повідомила про витік даних, що торкнувся понад 117 мільйонів користувачів. Атака включала використання викрадених хешів паролів, які зловмисники зламували за допомогою rainbow tables. Це показало важливість використання надійного хешування та "солі" для захисту паролів. У 2013 році Yahoo стала жертвою однієї з найбільших атак, коли було викрадено дані понад 3 мільярдів користувачів. Зловмисники скористалися слабкими місцями у процесі відновлення паролів, використовуючи методи соціальної інженерії для отримання доступу до облікових записів. Цей інцидент підкреслив необхідність удосконалення безпекових заходів. У червні 2018 року Reddit повідомила про злам, що стався через атаку на систему двофакторної аутентифікації (2FA). Зловмисники використали метод соціальної інженерії, щоб обманом змусити одного з працівників Reddit розкрити свої облікові дані. Використовуючи ці дані, вони отримали доступ до внутрішніх систем компанії та копій старих баз даних користувачів, які містили захешовані паролі. Цей випадок підкреслює важливість безпеки не лише паролів, але й механізмів двофакторної аутентифікації. У липні 2020 року стався великий злам Twitter, в результаті якого зловмисники отримали доступ до облікових записів багатьох відомих осіб та компаній, включаючи Білла Гейтса, Ілона Маска та Apple. Зловмисники використали методи соціальної інженерії, щоб отримати доступ до внутрішніх інструментів підтримки Twitter. Потім вони скинули паролі та ввімкнули двофакторну аутентифікацію для деяких

облікових записів, щоб ускладнити відновлення доступу для законних власників. Це злам підкреслює важливість захисту внутрішніх інструментів та навчання співробітників безпеці. У листопаді 2018 року Marriott International повідомила про злам, що стався через компрометацію бази даних Starwood. Зловмисники отримали доступ до облікових записів понад 500 мільйонів гостей, використовуючи викрадені облікові дані. Вони скористалися вразливістю у процесі відновлення паролів, що дозволило їм отримати доступ до облікових записів без необхідності знати справжні паролі. Цей інцидент підкреслив важливість безпеки систем відновлення паролів та регулярного аудиту безпеки. У квітні 2020 року, під час пандемії COVID-19, платформа для відеоконференцій Zoom зазнала численних атак, відомих як "Zoom-bombing". Зловмисники використовували викрадені або слабкі паролі для доступу до приватних конференцій, що призвело до витоку конфіденційної інформації. Багато користувачів повідомили, що зловмисники отримали доступ до їх облікових записів, скориставшись вразливостями у процесі відновлення паролів. Це змусило Zoom вдосконалити свої механізми безпеки та впровадити додаткові заходи захисту. У червні 2021 року стало відомо про злам даних LinkedIn, в результаті якого понад 700 мільйонів облікових записів користувачів були виставлені на продаж у даркнеті. Зловмисники скористалися вразливостями у системі відновлення паролів та використали викрадені дані для доступу до облікових записів користувачів. Цей інцидент показав важливість використання надійних паролів та регулярного моніторингу систем безпеки.

1.5 Постановка задачі

Завданням даної кваліфікаційної роботи є розробка системи виявлення атаки на відновлення паролю у мережі, яка використовує RNN нейронні мережі для аналізу мережевого трафіку, з метою ідентифікації спроб підбору чи злому пароля в середині локальної мережі.

До основних завдань під час виконання роботи можна віднести:

– аналіз існуючих методів виявлення атак на паролі та обґрунтувати вибір методу для для вирішення поставленої проблеми;

– вивчити структуру та особливості трьох наборів даних: KDDCup99, UNSW-NB15 та CICIDS2017, які використовуватимуться для тренування та тестування моделі системи;

– розробка алгоритму для виявлення атак на відновлення;

– проведення навчання та тестування розробленої моделі на обраних наборах даних;

– проведення порівняння достовірності роботи розробленої системи на різних наборах даних.

У роботі об'єктом дослідження будуть виступати процеси виявлення атак на відновлення паролю у локальних мережах.

До методів дослідження, які будуть використовуватися, можна віднести:

– методи аналізу та синтезу інформаційних систем;

– методи машинного навчання, зокрема рекурентні нейронні мережі;

– експериментальні методи для порівняння ефективності моделі на різних наборах даних.

Передбачається, що розроблена система виявлення атак на відновлення паролю буде демонструвати високу точність і надійність на основі тестування з використанням наборів даних KDDCup99, UNSW-NB15 та CICIDS2017. А результати роботи можуть бути використані для підвищення рівня інформаційної безпеки в організаціях, що використовують мережеві ресурси, шляхом впровадження ефективної системи виявлення атак на відновлення паролю.

РОЗРОБКА АЛГОРИТМІВ РЕАЛІЗАЦІЇ ТА ПІДГОТОВКА НАВЧАЛЬНИХ ДАНИХ

2.1 Вибір нейронної мережі для аналізу мережевого трафіку

Для виявлення атак на відновлення паролів, які зазвичай включають аналіз поведінкових патернів і аномалій, може бути ефективним застосування кількох типів нейронних мереж та їх комбінацій. Ось декілька підходів, які переважно використовуються:

– рекурентні нейронні мережі (Recurrent Neural Network, RNN): довготривала короткочасна пам'ять (Long Short-Term Memory, LSTM), вентильний рекурентний вузол (Gated recurrent units, GRU);

– глибокі нейронні мережі (Deep Neural Networks, DNN);

– згорткові нейронні мережі (Convolutional Neural Networks, CNN);

– автокодувальник;

– генеративно-змагальні мережі (Generative Adversarial Networks, GAN).

RNN мережі ефективні для аналізу послідовних даних, таких як часові ряди логів входу або послідовності аутентифікаційних спроб. LSTM може враховувати як довгострокові, так і короткострокові залежності в даних, що дозволяє ідентифікувати складні патерни поведінки, які можуть вказувати на атаки [34-35].

Використання DNN дозволяє класифікувати і кластеризувати великі обсяги даних і знаходити нетипові шаблони, які можуть свідчити про спроби зламу. Вони можуть бути використані для визначення, чи є певна спроба входу нормальною або підозрілою [36].

Хоча CNN зазвичай асоціюють із обробкою зображень, вони також можуть бути ефективні для аналізу матриць, які відображають поведінкові патерни в даних аутентифікації, особливо коли дані можна представити у формі візуальних патернів [37-38].

Автокодувальники навчаються відтворювати нормальні вхідні дані, і коли

вхід відрізняється від того, що мережа очікує (наприклад, вхід, який відображає атаку), помилка відтворення може вказувати на наявність аномалії.

GAN можуть бути використані для покращення моделей виявлення аномалій, де одна мережа намагається генерувати патерни атак, а інша — відрізняти їх від нормальної поведінки. Це дозволяє системі постійно покращувати свою здатність виявляти нові типи атак [39-40].

Ефективність будь-якої нейронної мережі значною мірою залежить від якості навчальних даних і ретельного налаштування параметрів мережі. Крім того, інтеграція декількох типів мереж може допомогти створити більш надійну та адаптивну систему, здатну ефективно виявляти складні атаки на відновлення паролів у реальному часі.

RNN є типом нейронних мереж, який ефективно обробляє послідовні дані або часові ряди, загальний вигляд архітектури показано на рисунку 2.1.

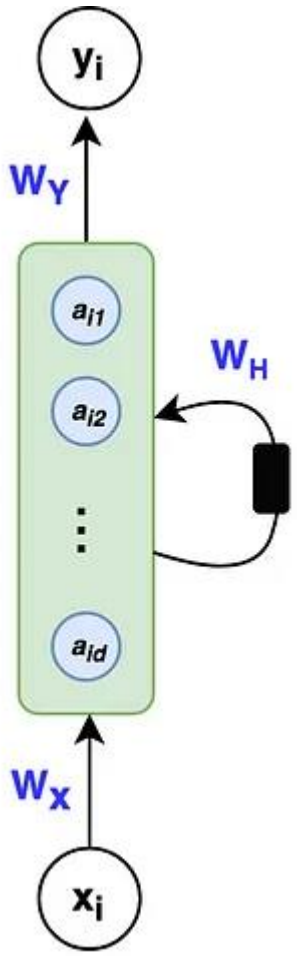


Рисунок 2.1 – Узагальнена архітектура RNN

На відміну від звичайних нейронних мереж, які обробляють вхідні дані незалежно одне від одного, RNN здатна зберігати інформацію про попередні входи в своєму внутрішньому стані, що дозволяє їй виявляти залежності в часі або послідовностях. RNN має внутрішній стан (або приховані шари), який зберігає інформацію про попередньо оброблені дані. Це дозволяє мережі передавати інформацію від одного кроку обробки до наступного, як показано на рисунку 2.2.

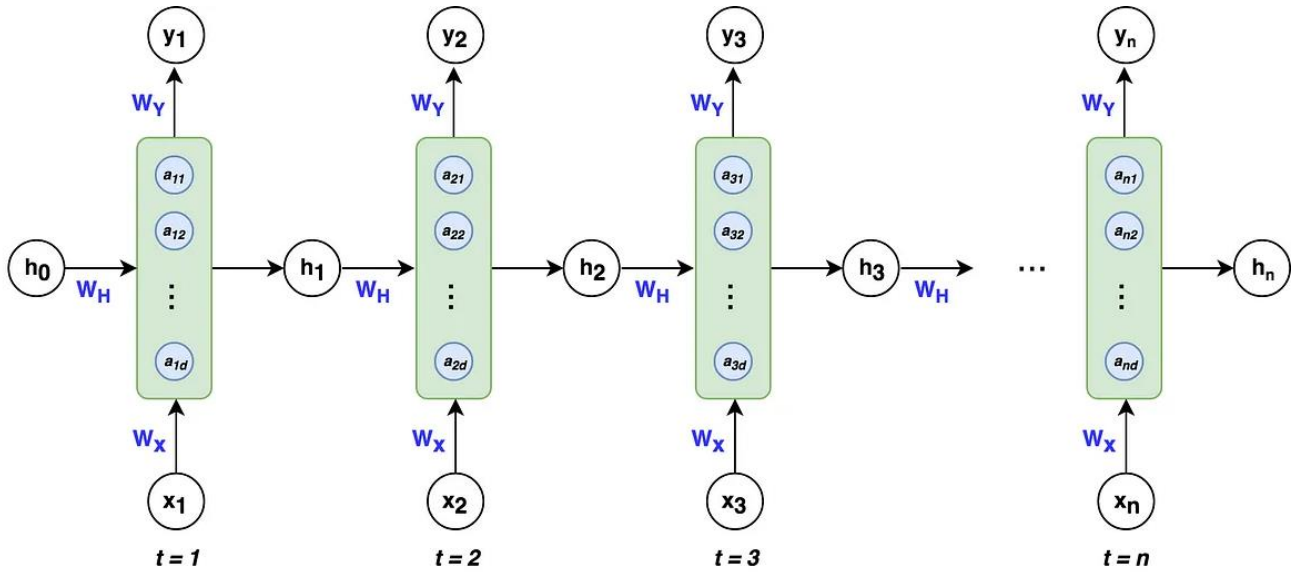


Рисунок 2.2 – Деталізована архітектура RNN

Вектор h є виходом прихованого стану після застосування функції активації до прихованих вузлів. У момент часу t архітектура враховує події, що сталися в момент часу t-1, включаючи h з попереднього прихованого стану, а також вхідні дані x у момент часу t, як показано на рисунку 2.3. Це дає змогу мережі зберігати та враховувати інформацію з попередніх вхідних даних, що передують поточним. Варто зазначити, що початковий вектор h завжди починається з нульових значень, оскільки алгоритм не має інформації про події до першого елемента в послідовності. У RNN один і той же набір параметрів використовується на кожному кроці обробки. Це значно скорочує кількість вільних параметрів та спрощує модель. RNN може обробляти дані різної довжини, що робить їх ідеальними для завдань, де входи та/або виходи можуть бути послідовностями різної довжини (наприклад, машинний переклад, розпізнавання мовлення).

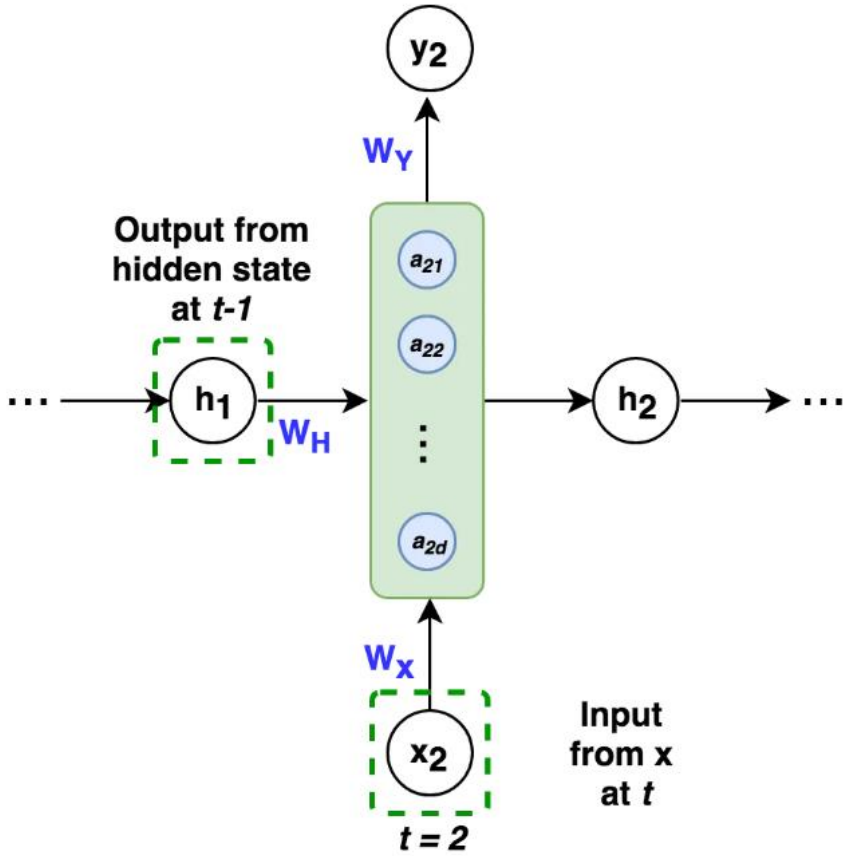


Рисунок 2.3 - Прихований стан при t=2 приймає як вхідні дані від t-1 і x при t

RNN використовуються в різноманітних застосуваннях, де важлива обробка послідовностей даних, в тому числі:

– перетворення аудіо в текст;

– машинний переклад тексту з однієї мови на іншу;

– автоматичне створення текстового вмісту;

– прогнозування майбутніх значень на основі попередніх спостережень.

RNN надає потужні інструменти для обробки і аналізу послідовних даних, що робить їх ключовим компонентом у багатьох сучасних системах штучного інтелекту. RNN краще працює з великими обсягами даних, а використання зворотного поширення покращує кінцевий результат. У даній роботі буде використано RNN, оскільки набір даних великий і містить послідовності атак.

До переваг використання нейронної мережі можна віднести:

– швидкість, а саме змогу обробляти великі обсяги даних в реальному часі;

– вміння точно визначати складні шаблони поведінки;

– розроблені системи можуть адаптуватися до нових загроз та змін у поведінці користувачів.

2.2 Опис використаних засобів для реалізації

TensorFlow — це потужна відкрита бібліотека для чисельних обчислень, яка зазвичай використовується для розробки, тренування та впровадження машинного навчання (ML). Вона була розроблена командою Google Brain і вперше випущена у 2015 році. Основна мета TensorFlow полягає у забезпеченні зручної та гнучкої платформи для роботи з алгоритмами глибокого навчання, але вона також ефективна для більш широкого спектру наукових обчислень [41].

TensorFlow може працювати на багатьох пристроях, включаючи комп'ютери, сервери та мобільні пристрої, завдяки своїй здатності до запуску на різних платформах, таких як CPUs (центральні процесорні одиниці), GPUs (графічні процесорні одиниці) та TPUs (Tensor Processing Units). TensorFlow підтримує широкий спектр можливостей, включаючи дослідження, розробку прототипів та виробництво. TensorFlow підтримує кілька мов програмування, найпоширенішою з яких є Python. Крім того, існують інтерфейси для C++, Java та інших мов, що дозволяє інтегрувати TensorFlow у більш широкий спектр додатків. Як одна з найпопулярніших платформ для машинного навчання, TensorFlow має велику і активну спільноту розробників, які постійно допомагають удосконалювати бібліотеку і надають підтримку. Включає набір високорівневих інструментів, таких як TensorFlow Serving для розгортання моделей у виробництво, TensorFlow Lite для мобільних та вбудованих пристроїв, та TensorBoard для візуалізації тренувань мереж і метрик. Також включає допоміжні бібліотеки, які розширюють його функціональність. Наприклад, TensorFlow Extended (TFX) для створення та обслуговування виробничих пайплайнів машинного навчання, і TensorFlow.js для виконання моделей ML безпосередньо в браузері або на Node.js.

NumPy (Numerical Python) — це бібліотека для наукових обчислень у

Python. Вона забезпечує підтримку великих, багатовимірних масивів і матриць, разом з великою колекцією високорівневих математичних функцій для ефективної роботи з цими даними. NumPy дозволяє створювати і маніпулювати масивами з даними, які можуть мати багато вимірів [42]. Масиви NumPy набагато швидші за стандартні списки Python завдяки використанню оптимізованих бібліотек на C та Fortran. NumPy включає багато векторизованих функцій, які дозволяють виконувати елементарні операції з даними в масивах без необхідності писати цикли. Це значно підвищує продуктивність обчислень. NumPy пропонує інструменти для основних статистичних обчислень, лінійної алгебри, трансформації Фур'є та багато іншого. Він є основою для багатьох інших наукових бібліотек, таких як SciPy, Pandas, Matplotlib, scikit-learn, scikit-image та багато інших. Велика частина функціональності цих бібліотек залежить від можливостей NumPy. NumPy може взаємодіяти з даними, які зберігаються в форматах, згенерованих іншими мовами програмування, такими як C, C++ і Fortran, що робить його важливим інструментом для інтеграції систем. Зазвичай, NumPy використовується для підтримки складних математичних операцій і великих масивів, що робить NumPy хорошим рішенням для наукових досліджень. NumPy є критично важливим інструментом у світі Python для будь-яких операцій, що вимагають ефективних обчислень, і є однією з основних причин популярності Python у наукових та інженерних застосуваннях.

Pandas — це високорівнева бібліотека Python, яка надає широкі можливості для аналізу даних. Це одна з найпопулярніших і найважливіших бібліотек у світі Python для аналізу даних та наукових обчислень. Бібліотека була створена Весом Маккінні у 2008 році і з того часу стала невід'ємною частиною екосистеми наукового програмування в Python [43]. Pandas вводить дві нові структури даних, засновані на NumPy arrays:

– DataFrame;

– Series.

DataFrame представляє двовимірну таблицю з змінними розмірами, яка може містити різні типи даних. DataFrame має функціонал, подібний до таблиць в

SQL або Excel. Він дозволяє виконувати злиття, групування, агрегацію, резюмування та багато іншого.

Series представляє одновимірний масив з етикетками, здатний містити будь-який тип даних (цілі числа, рядки, числа з плаваючою комою тощо). Series дуже схожий на один стовпець у DataFrame.

Pandas підтримує безліч форматів для читання та запису даних, включаючи CSV, Excel, JSON, HTML і SQL. Це дозволяє легко імпортувати та експортувати дані з різних джерел. Він має розширені можливості для обробки та перетворення даних, такі як зміна форми даних, сортування, фільтрація, додавання/видалення стовпців/рядків, групування, об'єднання та інші операції з даними. Pandas пропонує великі можливості для аналізу часових рядів, включаючи можливості для створення діапазонів дат, конвертації часових зон, зсувів часу та індексації по датах. Pandas надає багато інструментів для обробки пропущених даних, включаючи можливості для виявлення, видалення та заміни пропущених значень. Pandas оптимізований для високої продуктивності завдяки використанню векторизованих операцій і Cython.

Pandas широко використовується аналітиками для розвідки та аналізу даних, розробки моделей машинного навчання, статистичного аналізу тощо. Можливість легко обробляти пропущені дані, фільтрувати та перетворювати дані робить Pandas ідеальним інструментом для підготовки даних перед подальшим аналізом. Створення звітів із даних, включаючи табличні підсумки і базові графіки, є ще однією сильною стороною Pandas.

Keras є високорівневим API для глибокого навчання, який спочатку був розроблений Франсуа Шолле як незалежний проект, але згодом став частиною TensorFlow як його офіційний API для спрощення розробки моделей машинного навчання. З TensorFlow 2.0 та вище, Keras тісно інтегрований у TensorFlow, ставши основним інструментом для створення і тренування нейронних мереж [44].

Keras зроблений з акцентом на швидкість і простоту розробки, пропонуючи чисті та прості абстракції, що дозволяє швидко прототипувати нейронні мережі.

Моделі в Keras будуються шляхом з'єднання конфігурованих модулів, таких як шари, активатори, оптимізатори. Це дає можливість легко експериментувати з різними архітектурами. Початково Keras дозволяв використовувати кілька бекендів обчислень, таких як TensorFlow, Theano або Microsoft CNTK. Проте, з інтеграцією в TensorFlow, він тепер використовує можливості TensorFlow для обчислень та автоматичного диференціювання. Keras тісно інтегрований з TensorFlow, що дозволяє легко використовувати потужності GPU для тренування моделей, значно прискорюючи процес навчання. Моделі, створені в Keras, легко можуть бути вивантажені в реалізацію, використовуючи TensorFlow Serving або інші платформи, які підтримують TensorFlow моделі. Цей API відрізняється своєю легкістю використання та здатністю швидко переходити від ідеї до результату, що робить його популярним серед дослідників і розробників у сфері штучного інтелекту.

Google Colab, або Colaboratory, — це безкоштовний сервіс від Google, що забезпечує онлайн-середовище для написання та виконання коду Python. Він базується на Jupyter Notebook, популярному інструменті в науковому програмуванні, який дозволяє комбінувати виконуваний код, текст, математику, графіки та інтерактивний вміст у єдиних документах. Colab широко використовується для машинного навчання, аналізу даних та освіти. Google Colab надає безкоштовний доступ до обчислювальних ресурсів, включаючи графічні процесори (GPU) та тензорні процесори (TPU), що робить його ідеальним для тренування складних моделей машинного навчання. Google Colab повністю сумісний з Jupyter, що означає можливість відкривати, виконувати та розповсюджувати Jupyter notebooks безпосередньо. Подібно до Google Docs, Colab дозволяє легко спільно працювати над проєктами, надаючи можливість кільком користувачам одночасно працювати над одним і тим же документом. Colab тісно інтегрований з Google Drive, що дозволяє легко зберігати, ділитися та доступати до notebooks. Також можлива інтеграція з GitHub. Підтримка інтерактивних віджетів дозволяє створювати інтерактивні звіти та інструменти аналізу, що робить Colab потужним інструментом для наукових досліджень та освіти. Завдяки

доступу до GPU та TPU, користувачі можуть ефективно тренувати складні моделі машинного навчання. Colab надає потужні інструменти для аналізу та візуалізації даних, що допомагає в наукових дослідженнях та прийнятті рішень.

2.3 Алгоритм виявлення атаки у мережі

Класифікатори RNN включають використання двох основних типів архітектури: LSTM та GRU. Ці класифікатори можуть ефективно обробляти послідовні дані, такі як потоки даних, і можуть виявляти складні залежності в даних, що робить їх корисними для задач виявлення атак. LSTM — це спеціальний тип рекурентної нейронної мережі, яка здатна вчитися на довгострокових залежностях. Вона розроблена для вирішення проблеми зникаючих градієнтів, яка заважає традиційним RNN вчитися на довгих послідовностях. LSTM складається з осередків, які зберігають значення на довгий час і мають три гейти (ворота чи вентилі) (рисунок 2.4): вхідний, вихідний і забуваючий.

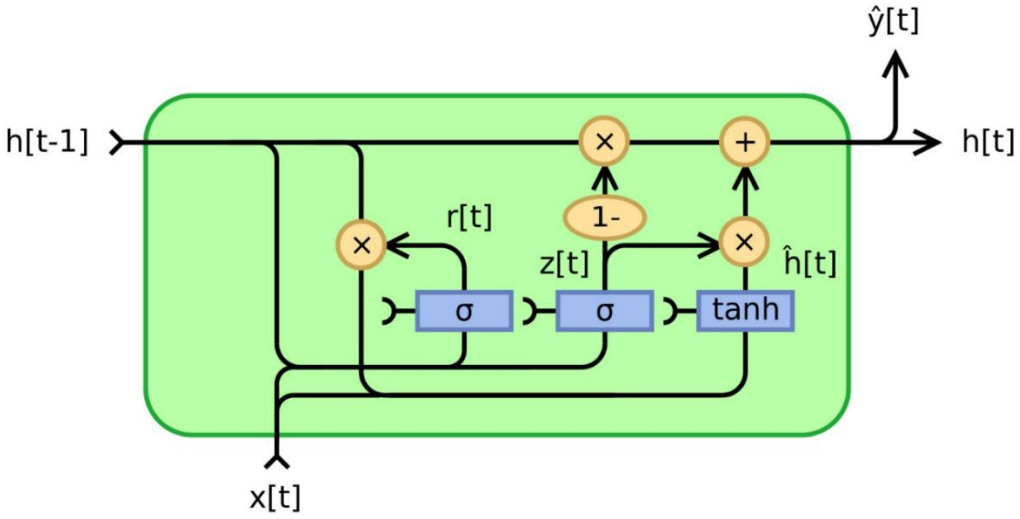


Рисунок 2.4 - Повний рекурентний вузол

Повний рекурентний вузол функціонує наступним чином. Вхідними даними є значення вектора входу x_t та значення виходу h_{t-1} (для t=0, вихідний вектор h_0=0). На основі цих даних обчислюється кандидат на нове значення виходу — вектор скидання r_t, який визначається через активаційну функцію від матричного виразу

з параметрами W, U та b. Подібним чином, обчислюється вектор уточнення z_t, який визначає, чи слід зберегти значення з попереднього вектора, чи взяти нове значення. Це фактично набір "воріт", які вирішують, пропускати старе значення або замінити його новим. Далі обчислюється вектор виходу h_t, де з ймовірністю z_t береться старе значення з вектора h_{t-1}, а з ймовірністю $1-z_t$ обчислюється нове значення.

GRU є вдосконаленою версією стандартної рекурентної нейронної мережі (RNN). Вона була запропонована Кюнгхюн Чо та іншими дослідниками у 2014 році. Разом з тим це спрощена версія LSTM, яка має менше параметрів, але зберігає подібну функціональність. GRU об'єднує вхідний і забуваючий гейти в один і використовує менше пам'яті та обчислювальних ресурсів, що може бути перевагою при роботі з великими наборами даних або обмеженими ресурсами. Саме тому буде використовуватися у даній роботі.

Як і LSTM, GRU використовує гейти для контролю потоку інформації. Однак вони є відносно новими в порівнянні з LSTM. Саме тому вони пропонують деякі покращення в порівнянні з LSTM та мають простішу архітектуру (рисунок 2.5).

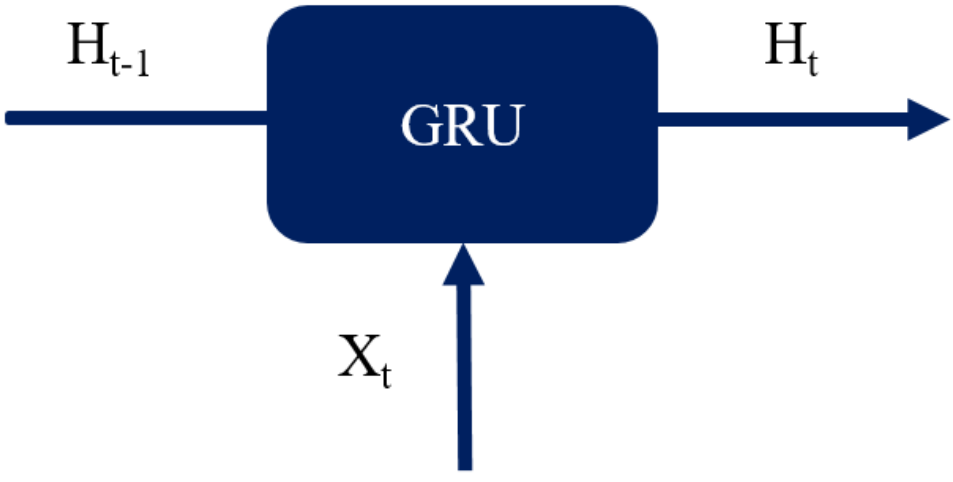


Рисунок 2.5 – Архітектура GRU

На відміну від стандартних RNN, GRU використовує спеціальні гейти

(шлюз оновлення та шлюз скидання), щоб контролювати потік інформації в мережі. Ці гейти діють як фільтри, вирішуючи, яку інформацію з минулого зберегти, забути чи оновити. Вибірково пропускаючи відповідну інформацію через гейти, GRU запобігає повному зникненню градієнтів. Це дозволяє мережі вивчати довготривалі залежності навіть у довгих послідовностях. Механізм гейтів дозволяє GRU ефективно керувати потоком інформації. Шлюз Reset може відкидати нерелевантну минулу інформацію, а шлюз Update контролює баланс між збереженням минулої інформації та включенням нової інформації. Це покращує здатність мережі запам'ятовувати важливі деталі протягом більш тривалого часу. Завдяки ефективним механізмам гейтів GRU часто можна навчити швидше, ніж стандартні RNN, для виконання завдань, що включають довгі послідовності. Гейти допомагають мережі навчатися ефективніше, зменшуючи кількість необхідних ітерацій навчання.

GRU використовує комірки, схожі на комірки LSTM або RNN. Кожен часовий крок t приймає вхідні дані X_t і прихований стан H_{t-1} з попереднього часового кроку t-1. Пізніше він виводить новий прихований стан H_t, який передається до наступного часового кроку.

У GRU є два гейти, на відміну від трьох гейтів у LSTM. Перший гейт — це Reset Gate (шлюз скидання), а інший — Update Gate (шлюз оновлення). Шлюз Reset відповідає за короткочасну пам'ять мережі, тобто прихований стан H_t. Формула шлюзу Reset виглядає наступним чином:

$$r_t = \sigma(x_t * U_r + H_{t-1} * W_r), \quad (2.1)$$

де значення r_t варіюватиметься від 0 до 1 через сигмоїдну функцію; W_r та U_r - вагові матриці для вентиля скидання.

Шлюз Update відповідає за довготривалу пам'ять, і формулу цього гейту показано нижче:

$$u_t = \sigma(x_t * U_u + H_{t-1} * W_u), \quad (2.2)$$

Послідовність роботи GRU наступна.

Першочергово відбувається підготовка вхідних даних, де GRU приймає два входи як вектори: поточний вхід X_t і попередній прихований стан h_{t-1}.

Наступним кроком відбуваються розрахунки гейтів. У GRU є два гейти: Reset Gate і Update Gate. Слід обчислити значення для кожного гейту окремо. Для цього виконується поелементне множення між поточним вхідним і попереднім векторами прихованого стану. Це робиться окремо для кожного гейту, по суті створюючи «параметризовані» версії входів, специфічних для кожного гейту. Нарешті, застосовується функція активації поелементно до кожного елемента в цих параметризованих векторах. Ця функція активації зазвичай виводить значення від 0 до 1, які використовуватимуться гейтами для керування потоком інформації.

Щоб знайти прихований стан H_t у GRU, необхідно виконати двоетапний процес. Першим кроком є створення так званого прихованого стану кандидата. Як показано нижче:

$$\widehat{H}_t = tanh\left(x_t * U_g + (r_t \circ H_{t-1)} * W_g\right) \quad (2.3)$$

Він приймає поточний вхід і прихований стан із попередньої мітки часу t-1, яка множиться на вихідний сигнал скидання r_t. Пізніше всю цю інформацію передають функції tanh, результуюче значення — прихований стан кандидата.

Найважливіша частина цього рівняння полягає в тому, як використовується значення вентиля скидання, щоб контролювати, наскільки може впливати попередній прихований стан на стан-кандидат. Якщо значення r_t дорівнює 1, це означає, що розглядається вся інформація з попереднього прихованого стану h_{t-1}. Так само, якщо значення r_t дорівнює 0, це означає, що інформація з попереднього прихованого стану повністю ігнорується.

Коли є стан-кандидат, то він використовується для створення поточного прихованого стану H_t. Саме тут з’являються ворота оновлення. Замість того, щоб використовувати окремий шлюз, як у LSTM, у GRU використовується єдиний

шлюз оновлення, щоб контролювати як історичну інформацію, якою є h_{t-1}, так і нову інформацію, яка надходить із стану-кандидата:

$$H_t = u_t \circ H_{t-1} + (1 - u_t) \circ \widehat{H_t} \quad (2.4)$$

Припустімо, що значення u_t дорівнює приблизно 0, тоді перший член у рівнянні буде нульовим, що означає, що новий прихований стан не матиме багато інформації з попереднього прихованого стану. З іншого боку, друга частина стає майже такою, що по суті означає, що прихований стан у поточній мітці часу складатиметься лише з інформації з стану-кандидата.

Подібним чином, якщо значення u_t є другим членом, стає повністю 0, а поточний прихований стан повністю залежатиме від першого члена, тобто інформація з прихованого стану в попередній мітці часу t-1. Значення u_t є критичним у цьому рівнянні та може коливатися від 0 до 1.

Використання нейронних мереж для виявлення атак на відновлення паролів включає застосування штучного інтелекту для аналізу та розпізнавання підозрілих патернів активності, які можуть свідчити про спроби зламу. Такі системи спрямовані на збір та аналіз даних про спроби аутентифікації, щоб вчасно ідентифікувати потенційно шкідливі дії.

Основні кроки роботи RNN нейронної мережі для виявлення атак на відновлення паролів (рисунок 2.6):

– алгоритм навчається визначати аномалії в поведінці користувачів, такі як нехарактерні час та місце входу, надмірна кількість невдалих спроб входу або використання різних IP-адрес. Слід відмітити, що нейронні мережі можуть використовувати під наглядом або без нагляду навчання для вдосконалення своїх алгоритмів на основі нових даних, що дозволяє їм адаптуватися до нових методів атак;

– нейронна мережа аналізує логи входу та спроби аутентифікації, збираючи інформацію про частоту та характер спроб входу до системи;

– після аналізу даних мережа ідентифікує потенційні атаки на основі

відхилень від звичайної поведінки. Наприклад, якщо мережа помітить, що користувач, який зазвичай входить до системи з одного місця та в один і той же час, раптом починає здійснювати багато невдалих спроб входу з різних IP-адрес, це може бути сигналом про атаку;

– якщо нейронна мережа виявляє підозрілу активність, вона інформує систему безпеки або адміністраторів для подальших дій, що може включати надсилання сповіщень або звітів про виявлені аномалії;

– автоматизована система може блокувати спроби входу або вимагати додаткової автентифікації в разі виявлення підозрілої активності.

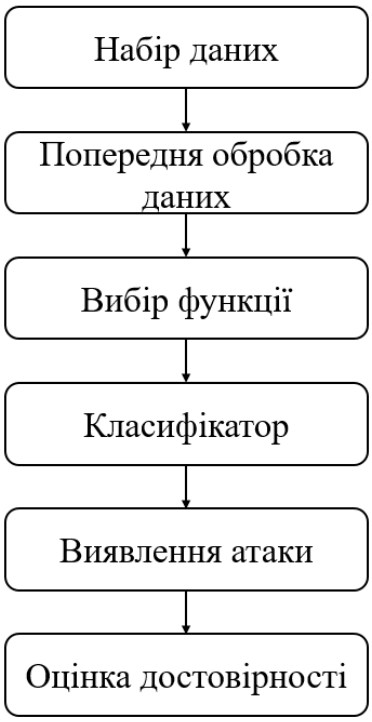


Рисунок 2.6 – Алгоритм RNN нейронної мережі для виявлення атак на відновлення паролів

Для навчання та тестування нейронної мережі буде використано набір даних UNSW-NB15. Етап попередньої обробки даних умовно можна розділити на шість кроків (рисунок 2.7):

– вилучення змінних;

– обробка відсутніх і нульових значень;

– перетворення даних;

– нормалізація значень;

– вибір найбільш значущих змінних;

– поділу набору даних.

Рисунок 2.7 – Етап попередньої обробки даних

При вилучення змінних відповідні змінні витягуються з необроблених даних. До типів змінних доцільно додати IP-адресу джерела та призначення, протокол, розмір пакетів, час між пакетами, кількість з'єднань за певний проміжок часу, частоту пакетів за секунду, прапори TCP та інші параметри для того, щоб побудувати детальний опис трафіку і визначити, чи є він нормальним чи зловмисним, тобто чи присутні ознаки кібератаки. Обрані змінні повинні максимально охоплювати всі аспекти трафіку, щоб забезпечити високу точність моделі виявлення атак.

Будь-які відсутні або нульові значення в наборі даних обробляються належним чином. Зокрема не критично важливі стовпці чи рядки із великою

кількість відсутніх значень будуть видалені, позначити спеціальним маркером пусті комірки (щоб явно вказувало на відсутність даних при неможливості видалити рядок чи стовпець). Заповнення комірок середніми значеннями, за медіаною чи модою, методом k-найближчих сусідів може призвести до хибного прийняття рішення, тому дані методи використовуватися не будуть. Зміни у наборі даних за допомогою бібліотеки pandas реалізуються із використанням функцій як показано на рисунку 2.8.

```
df = pd.DataFrame(data)
# Видалення рядків з відсутніми значеннями
df_dropped_rows = df.dropna()
# Видалення стовпців з відсутніми значеннями
df_dropped_columns = df.dropna(axis=1)
# Використання спеціального маркера
df_filled_marker = df.fillna(-1)

print("Original DataFrame:\n", df)
print("\nRows with NaNs dropped:\n", df_dropped_rows)
print("\nColumns with NaNs dropped:\n", df_dropped_columns)
print("\nNaNs filled with marker:\n", df_filled_marker)
```

Рисунок 2.8 – Обробка відсутніх та нульових значень набору даних

Наступним є перетворення дані, які мають категоріальний характер (тобто які належать до одного з обмеженого набору категорій або міток), у числові значення, щоб їх можна було використовувати в алгоритмах машинного навчання. Більшість алгоритмів машинного навчання працюють з числовими даними, тому категоріальні дані потрібно перетворити на числові, щоб їх можна було використати для моделювання, як показано на рисунку 2.9.

```
target_mean = df.groupby('Category')['Target'].mean()
df['Category_Target_Encoded'] = df['Category'].map(target_mean)

print(df)
```

Рисунок 2.9 – Перетворення даних

Проведення нормалізації значень, тобто реалізація процесу перетворення числових даних до єдиної шкали без зміни відносних відмінностей між значеннями. У даному випадку буде використано метод мінімаксної нормалізації (рисунок 2.10), де значення перетворюються до діапазону [0, 1], використовуючи формулу 2.1:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

```
scaler = MinMaxScaler()
df['Value_Normalized'] = scaler.fit_transform(df[['Value']])
```

Рисунок 2.10 – Застосування методу мінімаксної нормалізації

Метою вибору найбільш значущих змінних є зменшення кількості вхідних змінних, щоб зменшити обчислювальну складність моделі, зменшити ризик перенавчання і покращити продуктивність моделі.

Поділ набору даних здійснюється на дві частини: навчальний набір (training set) і тестовий набір (testing set), як показано на рисунку 2.11.

```
# Розподіл даних на ознаки та цільову змінну
X = df.drop(columns=['target'])
y = df['target']

# Розподіл даних на навчальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Навчальний набір (X_train):")
print(X_train)
print("Тестовий набір (X_test):")
print(X_test)
print("Навчальні мітки (y_train):")
print(y_train)
print("Тестові мітки (y_test):")
print(y_test)

target_mean = df.groupby('Category')['Target'].mean()
df['Category_Target_Encoded'] = df['Category'].map(target_mean)
```

Рисунок 2.11 – Поділ набору даних

Мета цього поділу полягає в тому, щоб навчити модель на одному наборі даних (навчальний набір) і оцінити її продуктивність на іншому, невідомому для моделі наборі (тестовий набір). Це дозволяє оцінити здатність моделі до узагальнення на нові дані. З використанням навчального набору модель вивчає залежності між вхідними даними та цільовою змінною. На тестовому наборі модель перевіряється на предмет узагальнюючої здатності, тобто як добре вона передбачає цільові значення для нових, невідомих даних.

Вибір значень є важливим кроком у підготовці даних для моделювання. Використання методів на основі дерев рішень, таких як Extra Trees Classifier, дозволяє автоматично визначити найбільш значущі функції для подальшого аналізу і моделювання. Адже це допоможе зменшити складність моделі, покращити продуктивність і знизити ризик перенавчання. Що є ефективним підходом для роботи з великими наборами даних, такими як UNSW-NB15.

Виявлення атак (Attack Detection) — це процес, в якому класифікатори визначають, чи є спостережувана діяльність нормальною або підозрілою, що може свідчити про атаку. У даному випадку класифікатор RNN аналізує вхідні дані та надає результати у вигляді двох категорій (рисунок 2.12): нормальний трафік або зловмисний трафік.

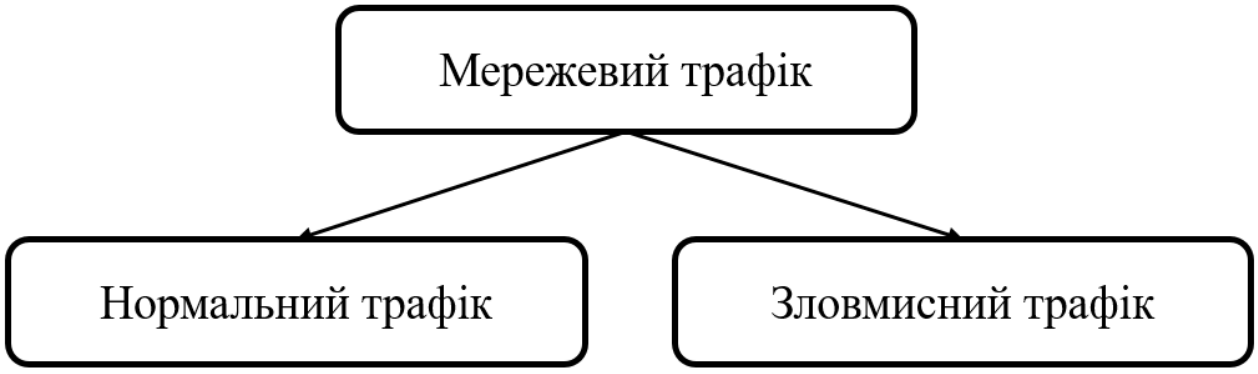


Рисунок 2.12 – Класифікація трафіку при виявленні атаки

Продуктивність моделі оцінюється за кількома показниками (рисунок 2.13): повнота, точність, акуратність, помилка, F-міра. Для розрахунку показників слід використати матрицю плутанини - це таблиця, яка дозволяє візуалізувати роботу

алгоритму класифікації, що показує кількість правильних і неправильних передбачень для кожного класу та містить наступні значення:

– True Positives (TP) - правильно передбачені позитивні випадки;
– True Negatives (TN) - правильно передбачені негативні випадки;
– False Positives (FP) - неправильно передбачені позитивні випадки;
– False Negatives (FN) - неправильно передбачені негативні випадки.

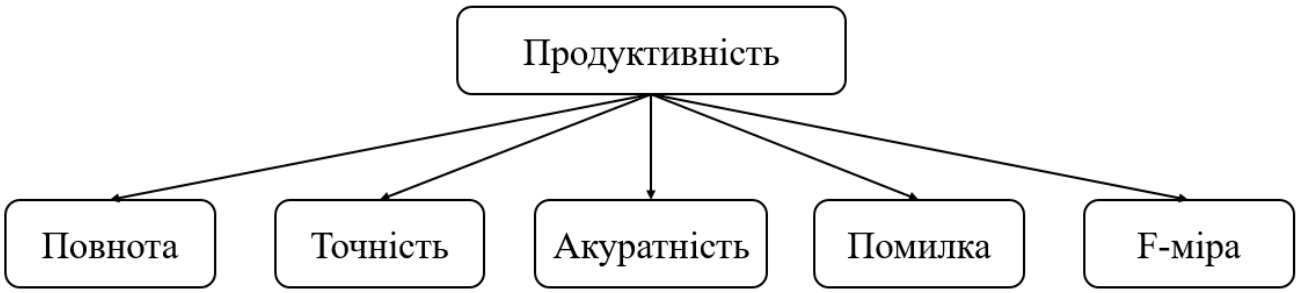


Рисунок 2.13 – Параметри продуктивності

Кожен етап процесу переходить у наступний, починаючи від попередньої обробки даних і закінчуючи оцінкою продуктивності, щоб забезпечити надійність і точність моделі виявлення атак.

2.4 Навчання нейронної мережі

Для навчання нейронної мережі при моделюванні засобами Google Colab було використано три набори даних: KDDCup99, UNSW-NB15 та CICIDS2017. Ці набори даних використовуються для дослідження та виявлення різних типів мережевих аномалій, включаючи атаки на пароль. KDDCup99 є класичним набором даних для виявлення мережевих вторгнень. Він містить велику кількість записів, які класифікуються як нормальний трафік або різні типи атак. У нашому випадку цей набір містить 98,456 записів, з яких 97,278 є нормальним трафіком, а 1,178 - атаки на пароль. UNSW-NB15 є сучасним та більш різноманітним набором даних для виявлення вторгнень, який містить 2,221,090 записів. З них 2,218,761 записів відображають нормальний трафік, а 2,329 записів - атаки на пароль.

CICIDS2017 є ще одним популярним набором даних, який використовується для дослідження безпеки мереж. Він містить 169,767 записів, з яких 168,187 записів є нормальним трафіком, а 1,580 записів - атаки на пароль. Загальну кількість записів наборів даних показано у таблиці 2.1.

Таблиця 2.1 – Вміст наборів даних

Набір даних	Всього записів	Нормальний трафік	Атаки на пароль
KDDCup99	98456	97278	1178
UNSW-NB15	2221090	2218761	2329
CICIDS2017	169767	168187	1580

Для нормалізації набору частину записів нормального трафіку було видалено. Це зроблено для того, щоб збалансувати кількість записів нормального трафіку та атак на пароль. Кількість записів за типами наборів даних та категоріями записів показано в таблиці 2.2.

Таблиця 2.2 - Вміст наборів даних після нормалізації

Набір даних	Всього записів	Нормальний трафік	Атаки на пароль
KDDCup99	2443	1265	1178
UNSW-NB15	4710	2381	2329
CICIDS2017	3272	1692	1580

Після нормалізації набори даних були розділені у співвідношенні 70% для навчання нейронної мережі та 30% для тестування. Розподіл даних на 70% для навчання і 30% для тестування є стандартною практикою в машинному навчанні та аналізі даних. Використання 70% даних для навчання гарантує, що нейронна мережа отримує достатню кількість інформації для вивчення та виявлення закономірностей. Це важливо для складних моделей, які потребують великих

обсягів даних для точного навчання. Залишаючи 30% даних для тестування, залишається достатній обсяг даних для оцінки продуктивності моделі. Це дозволяє перевірити, наскільки добре модель може узагальнюватися на нових, невідомих їй даних, що є важливим для забезпечення її здатності працювати з реальними даними. Співвідношення 70:30 забезпечує гарний баланс між кількістю даних для навчання і тестування. Використання більшого обсягу даних для навчання зменшує ризик перенавчання моделі, оскільки модель навчається на більш різноманітних даних. Це допомагає моделі узагальнювати краще і запобігає ситуації, коли модель надто сильно підлаштовується під навчальні дані, втрачаючи здатність правильно обробляти нові дані. Кількість записів, яка буде використана для навчання, показано в таблиці 2.3.

Таблиця 2.3 – Поділ наборів даних для навчання та тестування

Набір даних	Всього записів	Нормальний трафік	Атаки на пароль
KDDCup99	1709	885	824
UNSW-NB15	3298	1668	1630
CICIDS2017	2290	1184	1106

Таким чином, дані були підготовлені для подальшого використання в навчанні нейронної мережі. Процес нормалізації і поділу даних на навчальну та тестову частини забезпечує збалансоване представлення класів і сприяє більш точній оцінці продуктивності моделі.

2.5 Висновки до розділу

У цьому розділі було розглянуто вибір і реалізацію нейронної мережі для аналізу мережевого трафіку з метою виявлення атак на відновлення паролів. Вибір нейронної мережі є ключовим етапом у розробці системи, що забезпечує ефективний захист від кіберзагроз. RNN були обрані як основний інструмент для

обробки послідовних даних, зокрема часових рядів логів входу або послідовностей аутентифікаційних спроб. Це особливо важливо для виявлення аномалій у поведінці користувачів, які можуть свідчити про спроби несанкціонованого доступу. Було розглянуто загальну архітектуру RNN та її модифікацій, що дозволяє ефективно обробляти дані різної довжини. Було детально описано основні засоби та інструменти, що використовуються для реалізації системи. Серед них TensorFlow, потужна відкрита бібліотека для чисельних обчислень, яка забезпечує зручну та гнучку платформу для роботи з алгоритмами глибокого навчання. TensorFlow підтримує різні платформи, такі як CPU, GPU та TPU, що дозволяє ефективно використовувати обчислювальні ресурси. Крім того, було використано бібліотеки NumPy та Pandas, що надають широкі можливості для наукових обчислень та аналізу даних. NumPy забезпечує підтримку багатовимірних масивів і матриць, разом з високорівневими математичними функціями, що значно підвищує продуктивність обчислень. Pandas, у свою чергу, дозволяє легко маніпулювати даними, виконувати злиття, групування та інші операції. Описано процес попередньої обробки даних, який включає кілька ключових етапів: вилучення змінних, обробка відсутніх значень, перетворення категоріальних даних у числові, нормалізація значень, вибір найбільш значущих змінних та поділ на навчальний і тестовий набори даних. Такий підхід забезпечує високу точність моделі та дозволяє ефективно використовувати обчислювальні ресурси. Використання нейронних мереж для виявлення атак на відновлення паролів включає застосування штучного інтелекту для аналізу та розпізнавання підозрілих патернів активності. Нейронна мережа аналізує логи входу та спроби аутентифікації, збираючи інформацію про частоту та характер спроб входу до системи. Після аналізу даних мережа ідентифікує потенційні атаки на основі відхилень від звичайної поведінки та інформує систему безпеки або адміністраторів для подальших дій. Автоматизована система може блокувати спроби входу або вимагати додаткової автентифікації в разі виявлення підозрілої активності.

СИСТЕМА ВИЯВЛЕННЯ АТАКИ НА ВІДНОВЛЕННЯ ПАРОЛЮ У МЕРЕЖІ

3.1 Опис реалізації

Для реалізації системи першочергово потрібно підключити бібліотеки для: роботи з багатовимірними масивами і математичних операцій, обробки і аналізу даних, машинного навчання і глибокого навчання, як показано на рисунку 3.1.

```
import numpy as np
import pandas as pd
import tensorflow as tf
```

Рисунок 3.1 – Підключення бібліотек

Далі створено модель (рисунок 3.2), що складається із послідовності шарів, та безпосередньо підключаються шари.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, SimpleRNN, Dropout
```

Рисунок 3.2 – Створення моделі

Наступним кроком (рисунок 3.3) буде підключення метрик для оцінки моделі

```
from tensorflow.keras.metrics
import Precision, Recall, Accuracy, Specificity, Fscore
```

Рисунок 3.3 – Підключення метрик

Після чого потрібно вказати шлях до файлу із набором даних та виконати

завантаження даних, підготувати набір даних до використання відповідно до раніше описаної послідовності дій, як показано на рисунку 3.4

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
file_id = '1bum1KLtkCNbMrm2faLVOQ5FiY-pBUUIR'
url=f'https://drive.google.com/uc?export=download&id={file_id}'
data = pd.read_csv(url)
file_id = '1bum1KLtkCNbMrm2faLVOQ5FiY-pBUUIR'
url=f'https://drive.google.com/uc?export=download&id={file_id}'
data = pd.read_csv(url)
features = data.drop('label', axis=1)
labels = data['label']
```

Рисунок 3.4 – Підготовка набору даних

Потім виконати розділення даних на тренувальні та тестові, зміна форми представлення даних для сумісності з вимогами RNN (рисунок 3.5).

```
features_train, features_test, labels_train, labels_test =
train_test_split(features, labels, test_size=0.2, random_state=42)
features_train = np.reshape(features_train,
 (features_train.shape[0], 1, features_train.shape[1]))
```

Рисунок 3.5 – Поділ набору даних

Останнім кроком буде навчання та тестування моделі (рисунок 3.6) із раніше підготовленим набором даних.

```
model.fit(features_train, labels_train, epochs=10,
          batch_size=64, validation_data=(features_test, labels_test))
loss, accuracy, precision, recall = model.evaluate(features_test, labels_test)
```

Рисунок 3.6 – Навчання та тестування моделі

Команда model.fit() в TensorFlow/Keras використовується для навчання

моделі на вказаних даних. Пояснення для кожного з параметрів знаходиться нижче.

features_train - це дані для навчання моделі, які використовуються для входу в нейронну мережу, в даній реалізації вони вказуватимуть на ознаки, які модель буде використовувати для прогнозування міток.

labels_train - це мітки, або цільові значення, що відповідають features_train, використовуються моделлю для вивчення та порівняння з її прогнозами, щоб виміряти точність і здійснити коригування вагів під час тренування.

Параметр epochs визначає кількість повних проходів через весь тренувальний набір даних, тобто якщо вказано 100 епох, то це означає, що весь набір даних буде пропущено через модель 100 разів.

Параметр batch_size визначає кількість прикладів, що обробляються перед оновленням внутрішніх параметрів моделі; при batch_size дорівнює 64, це означає, що кожні 64 приклади з features_train будуть передані через мережу, після чого відбудеться одне оновлення вагів моделі.

Validation_data=(features_test, labels_test) вказує на дані, які використовуються для перевірки моделі після кожної епохи тренування, а features_test та labels_test — це тестові набори даних, які не використовуються для тренування моделі, а служать для перевірки її загальної здатності до прогнозування на нових даних. Це важливо для виявлення проблем, таких як перенавчання (overfitting), коли модель добре працює на тренувальних даних, але погано на даних, які вона не бачила.

Вказані параметри взаємодіють для контролю процесу навчання моделі, оптимізації її продуктивності та запобігання перенавчанню, забезпечуючи достатнє узагальнення.

Створення набору даних у форматі CSV для навчання RNN нейронної мережі з метою виявлення bruteforce атак на системи може включати різні ознаки (features), які допомагають ідентифікувати аномалії в мережевому трафіку та поведінці користувачів. Нижче наведено параметри набору даних (поля dataset), які використано для реалізації поставленого завдання:

– Timestamp - відмітка часу події;

– Source_IP - IP-адреса, з якої відбувається підключення;

– Destination_IP - цільова IP-адреса;

– Login_Attempt - кількість спроб входу за одиницю часу, що може бути високим при bruteforce атаках;

– Login_Success - успішний вхід до системи чи ні (1 або 0);

– User_Agent - інформація про клієнтське програмне забезпечення, що може допомогти виявити підозрілі або нестандартні клієнтські програми;

– Session_Duration - тривалість сесії в секундах;

– Bytes_Sent - кількість байтів, відправлених за сесію;

– Bytes_Received - кількість байтів, отриманих за сесію;

– Label - мітка класу (1 для bruteforce атаки, 0 для нормальної активності).

Фрагмент даних у форматі CSV приведено на рисунку 3.7.

```
Timestamp,Source_IP,Destination_IP,Login_Attempt,Login_Success,User_Agent,Session_Duration,Bytes_Sent,Bytes_Received,Label
2024-04-12T00:00:00Z,192.168.1.100,10.20.30.40,5,0,"Mozilla/5.0",30,300,500,1
2024-04-12T00:05:00Z,192.168.1.101,10.20.30.40,1,1,"Mozilla/5.0",300,1500,1500,0
2024-04-12T00:10:00Z,192.168.1.102,10.20.30.40,3,0,"Mozilla/5.0",60,500,700,1
2024-04-12T00:15:00Z,192.168.1.103,10.20.30.40,1,1,"Mozilla/5.0",200,1000,1200,0
2024-04-12T00:20:00Z,192.168.1.104,10.20.30.40,20,0,"Mozilla/5.0",10,100,150,1
2024-04-12T00:10:00Z,192.168.1.102,10.20.30.40,3,0,"Mozilla/5.0",60,400,800,1
2024-04-12T00:15:00Z,192.168.1.103,10.20.30.40,1,1,"Mozilla/5.0",20,1000,1100,0
2024-04-12T00:20:00Z,192.168.1.104,10.20.30.40,20,0,"Mozilla/5.0",40,800,150,0
```

Рисунок 3.7 – Фрагмент даних у форматі CSV

Параметр швидкості навчання визначає крок для кожного повторення, коли він рухається до функції мінімальних втрат. Для пошуку найкращої швидкості навчання, було проведено експеримент з використанням кількох швидкостей навчання. У цій роботі використовувався метод адаптивної оцінки моменту (Адам), щоб знайти швидкість навчання для моделі GRU. Моделі дали найкращу

оптимізацію за швидкості навчання 0,001.

Таблиця 3.1 - Параметри моделі GRU

Параметри	GRU
Активатор	Relu, Softmax (multiclass), Sigmoid (binary class)
Оптимізатор	Adam
Швидкість навчання	0.001
Функція втрат	Categorical cross entropy (multiclass), Binary cross entropy (binary-class)
Кількість шарів GRU	2
Приховані шари	2
Нейронів на шар	8
Нейронів на прихований шар	16, 8 (1-й шар, 2-й шар)
Розмір пакету	1000
Кількість епох	100

У даній роботі використовувалася активація випрямленої лінійної функції активації (ReLU). Застосувавши функцію ReLU, модель дізналася про складні особливості прихованих шарів мережі. У порівнянні з іншими функціями активації, такими як sigmoid і tanh, результати ReLU більш ефективні.

Рання зупинка визначається як техніка, при якій навчання моделі припиняється через деякий час, коли продуктивність моделі не покращується після фіксованої кількості епох. Зворотний виклик дострокового припинення відстежує втрати перевірки з мінімальною зміною 0,001. Навчання завершиться раніше, якщо втрати підтвердження не зменшаться принаймні на 0,001 протягом п'яти послідовних епох.

Оптимізатор Adam — це алгоритм оптимізації, який використовує методи RMSprop і AdaGrad. Модифікація їх відповідно до першого та другого моментів градієнтів зберігає швидкість навчання попередніх параметрів. Оптимізатор Adam

динамічно змінює швидкість навчання для кожного параметра під час навчання для ефективного оновлення вагових коефіцієнтів моделей LSTM і GRU.

3.2 Тестування реалізованої системи

Тестування системи проводилося у два етапи, щоб забезпечити всебічну оцінку її продуктивності та здатності виявляти атаки на паролі.

Перший етап тестування був проведений на змодельованій системі на платформі Google Colab із використанням заздалегідь підготовлених наборів даних. На цьому етапі основна увага приділялася: оцінці продуктивності системи, де вимірювалися показники точності, повноти та швидкості обробки даних; перевірці коректності алгоритмів шляхом аналізу правильності виконання алгоритмів виявлення атак; виявленню потенційно слабких місць в логіці роботи системи. Використання платформи Google Colab дозволило швидко налаштувати середовище для тестування та забезпечити доступ до потужних обчислювальних ресурсів. На цьому етапі використовувалися дані із таблиці 3.2.

Таблиця 3.2 – Записи для першого тестування

Набір даних	Всього записів	Нормальний трафік	Атаки на пароль
KDDCup99	734	380	354
UNSW-NB15	1412	713	699
CICIDS2017	982	508	474

Результати тестування першого етапу відображено в таблиці 3.3.

Таблиця 3.3 – Результати тестування першого етапу

Набір даних	TP	TN	FP	FN
KDDCup99	372	349	7	6
UNSW-NB15	697	691	10	14
CICIDS2017	496	468	6	12

Під час першого етапу тестування було виявлено, що система демонструє високі показники точності виявлення атак. Зокрема, кількість помилково негативних і помилково позитивних випадків була мінімальною, що свідчить про ефективність алгоритмів. Проте були виявлені певні недоліки, які потребували доопрацювання перед другим етапом тестування.

Другий етап тестування проводився в умовах реального середовища в ізольованій локальній комп'ютерній мережі. На цьому етапі здійснювалося запуск реальних атак на сервер із різних пристроїв, комбінуючи трафік із нормальним трафіком роботи користувач-сервер та адміністратор-сервер протягом 3 днів по 6 годин на добу. Для реалізації реальних атак використовувалося програмне забезпечення, описане у параграфі 1.2.

Цей етап був спрямований на перевірку здатності системи виявляти та протидіяти реальним атакам на паролі. Після кожного етапу тестування здійснювався детальний аналіз зібраних даних, включаючи логування всіх подій та їх ретельне вивчення. Це дозволило виявити слабкі місця в системі та внести зміни задля покращення роботоздатності системи.

Таблиця 3.4 - Записи для другого тестування

Всього записів	Нормальний трафік	Атаки на пароль
4882	2846	2036

Результати тестування другого етапу відображено в таблиці 3.5:

Таблиця 3.5 - Результати тестування другого етапу

TP	TN	FP	FN
2773	1968	68	73

На другому етапі тестування було виявлено, що система здатна ефективно виявляти реальні атаки в умовах, наближених до реального використання. Хоча кількість помилково негативних та помилково позитивних випадків трохи

збільшилася порівняно з першим етапом, загальна продуктивність системи залишилася високою.

За результатами обох етапів тестування можна зробити висновок про високу достовірність та ефективність роботи розробленої системи. Система продемонструвала здатність ефективно виявляти атаки на паролі як в умовах змодельованого середовища, так і в реальних умовах. Подальші покращення можуть бути спрямовані на зниження кількості помилково позитивних та помилково негативних випадків для ще більшої точності та надійності системи.

3.3 Оцінка ефективності

На основі отриманих даних можна розрахувати наступні показники продуктивності:

- повнота (recall) – співвідношення правильно класифікованих позитивних зразків до загальної кількості позитивних зразків:

$$Recall = \frac{TP}{TP + FN} \tag{3.1}$$

- точність (precision) – частка правильно визначених зловмисних подій серед усіх подій, які система визначила як зловмисні:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

- акуратність (accuracy) – частка правильно виявлених та правильно не виявлених подій серед усіх подій:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3}$$

- помилка (specificity) – вимірює здатність системи виявлення

зловмисників правильно ідентифікувати незловмисні об'єкти або події як незловмисні:

$$Specificity = \frac{FP + FN}{TP + FP + TN + FN} \quad (3.4)$$

- F-міра (F-score) – гармонійне середнє значення точності та повноти, яке забезпечує єдину міру, що збалансовує обидва аспекти. Значення варіюється від 0 до 100, де 100 вказує на ідеальну точність і повноту, а 0 вказує на найгіршу продуктивність:

$$F\ score = \frac{Recall + Precision}{2} \quad (3.5)$$

Повнота під час першого тестування для набору даних KDDCup99:

$$Recall = \frac{372}{372 + 6} * 100\% = 98{,}41\% \quad (3.6)$$

Точність під час першого тестування для набору даних KDDCup99:

$$Precision = \frac{372}{372 + 7} * 100\% = 98{,}15\% \quad (3.7)$$

Акуратність під час першого тестування для набору даних KDDCup99:

$$Accuracy = \frac{372 + 349}{372 + 7 + 6 + 349} * 100\% = 98{,}23\% \quad (3.8)$$

Помилка під час першого тестування для набору даних KDDCup99:

$$Specificity = \frac{7 + 6}{372 + 7 + 349 + 6} * 100\% = 1{,}77\% \quad (3.9)$$

F-міра під час першого тестування для набору даних KDDCup99:

$$F1\ score = \frac{98{,}41\% + 98{,}15\%}{2} = 98{,}28\% \quad (3.10)$$

Повнота під час першого тестування для набору даних UNSW-NB15:

$$Recall = \frac{697}{697 + 14} * 100\% = 98{,}03\% \quad (3.11)$$

Точність під час першого тестування для набору даних UNSW-NB15:

$$Precision = \frac{697}{697 + 10} * 100\% = 98{,}59\% \quad (3.12)$$

Акуратність під час першого тестування для набору даних UNSW-NB15:

$$Accuracy = \frac{697 + 691}{697 + 10 + 14 + 691} * 100\% = 98{,}3\% \quad (3.13)$$

Помилка під час першого тестування для набору даних UNSW-NB15:

$$Specificity = \frac{10 + 14}{697 + 10 + 691 + 14} * 100\% = 1{,}7\% \quad (3.14)$$

F-міра під час першого тестування для набору даних UNSW-NB15:

$$F1\ score = \frac{98{,}03\% + 98{,}59\%}{2} = 98{,}31\% \quad (3.15)$$

Повнота під час першого тестування для набору даних CICIDS2017:

$$Recall = \frac{496}{496 + 12} * 100\% = 97{,}64\% \quad (3.16)$$

Точність під час першого тестування для набору даних CICIDS2017:

$$Precision = \frac{496}{496 + 6} * 100\% = 98{,}8\% \quad (3.17)$$

Акуратність під час першого тестування для набору даних CICIDS2017:

$$Accuracy = \frac{496 + 468}{496 + 6 + 12 + 468} * 100\% = 98{,}17\% \quad (3.18)$$

Помилка під час першого тестування для набору даних CICIDS2017:

$$Specificity = \frac{6 + 12}{496 + 6 + 468 + 12} * 100\% = 1{,}83\% \quad (3.19)$$

F-міра під час першого тестування для набору даних CICIDS2017:

$$F1\ score = \frac{97{,}64\% + 98{,}8\%}{2} = 98{,}22\% \quad (3.20)$$

Порівняння результатів роботи системи виявлення під час тестування із різними наборами даних представлено на рисунках 3.8 та 3.9.

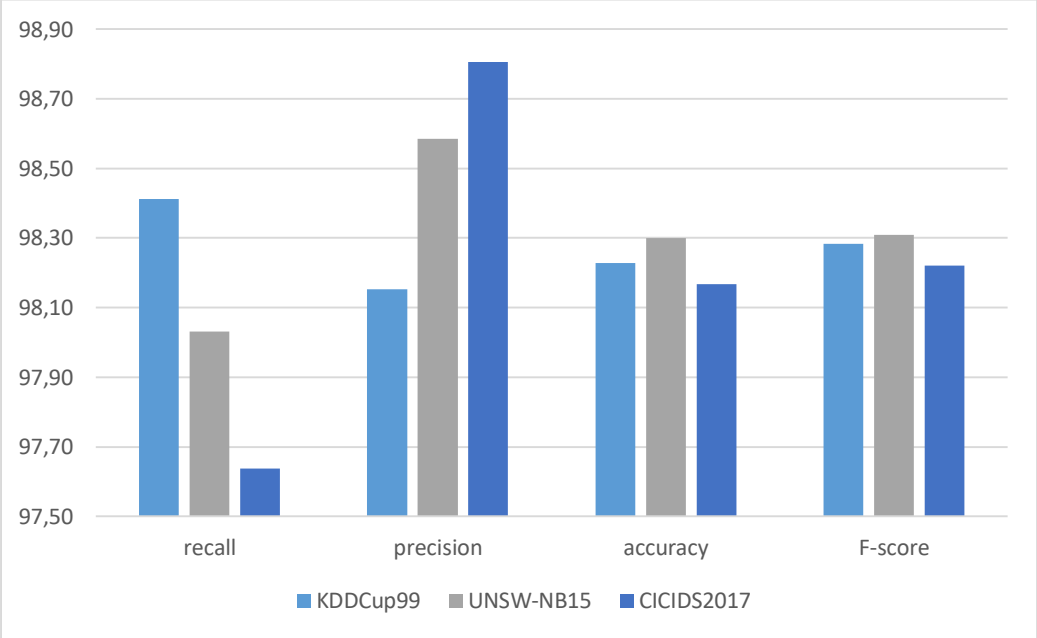


Рисунок 3.8 – Результати тестування з використанням різних наборів даних для метрик повноти, точності, акуратності та F-міри

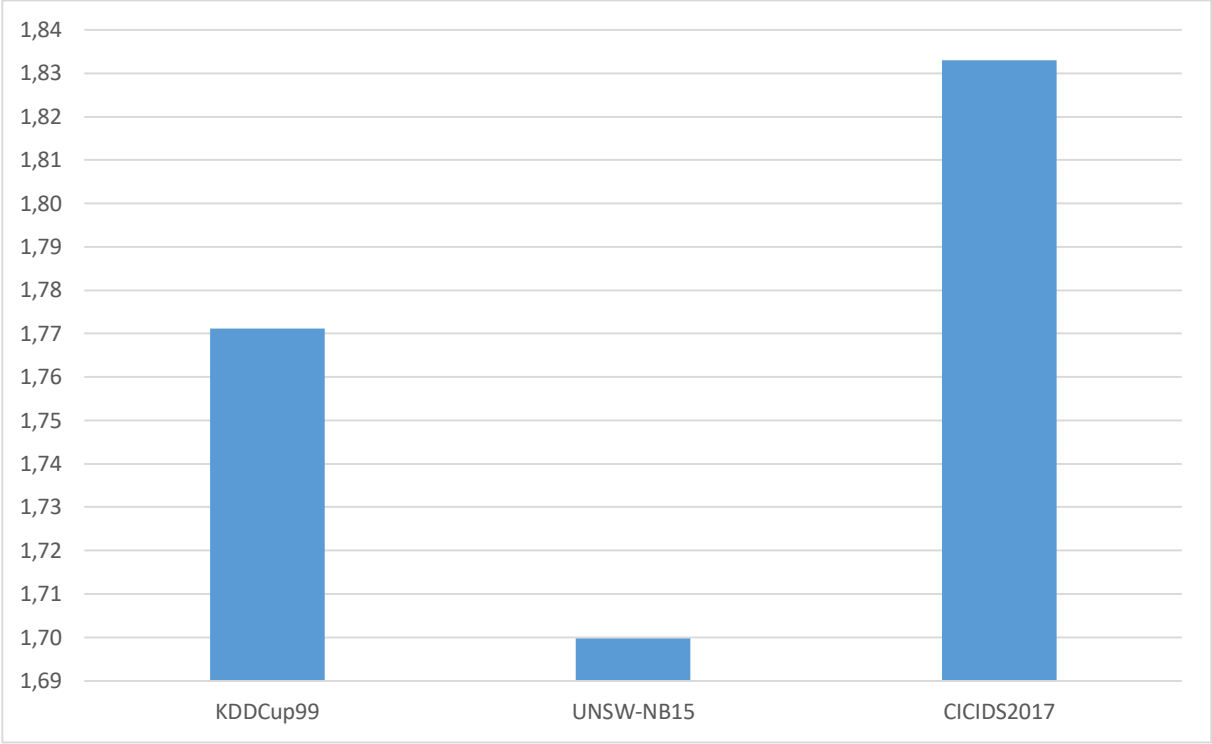


Рисунок 3.9 – Результати тестування з використанням різних наборів даних для метрики помилки

Повнота під час другого тестування:

$$Recall = \frac{2773}{2773 + 73} * 100\% = 97{,}43\% \quad (3.21)$$

Точність під час другого тестування:

$$Precision = \frac{2773}{2773 + 68} * 100\% = 97{,}61\% \quad (3.22)$$

Акуратність під час другого тестування:

$$Accuracy = \frac{2773 + 1968}{2773 + 68 + 73 + 1968} * 100\% = 97{,}11\% \quad (3.23)$$

Помилка під час другого тестування:

$$Specificity = \frac{68 + 73}{2773 + 68 + 1968 + 73} * 100\% = 2{,}89\% \quad (3.24)$$

F-міра під час другого тестування:

$$F1\ score = \frac{97{,}43\% + 97{,}61\%}{2} = 97{,}52\% \tag{3.25}$$

Порівнюючи середні значення результатів тестування системи з використанням наборів даних та тестування у фізичному тестовому середовищі (рисунки 3.10-3.11) можна зробити наступні висновки:

– значення повноти у реальному середовищі на 0,59% менше за середнє значення, яке було отримано при використанні заздалегідь підготовлених наборів даних;

– значення точності на 0,91% менше при тестуванні із імітацією атак;

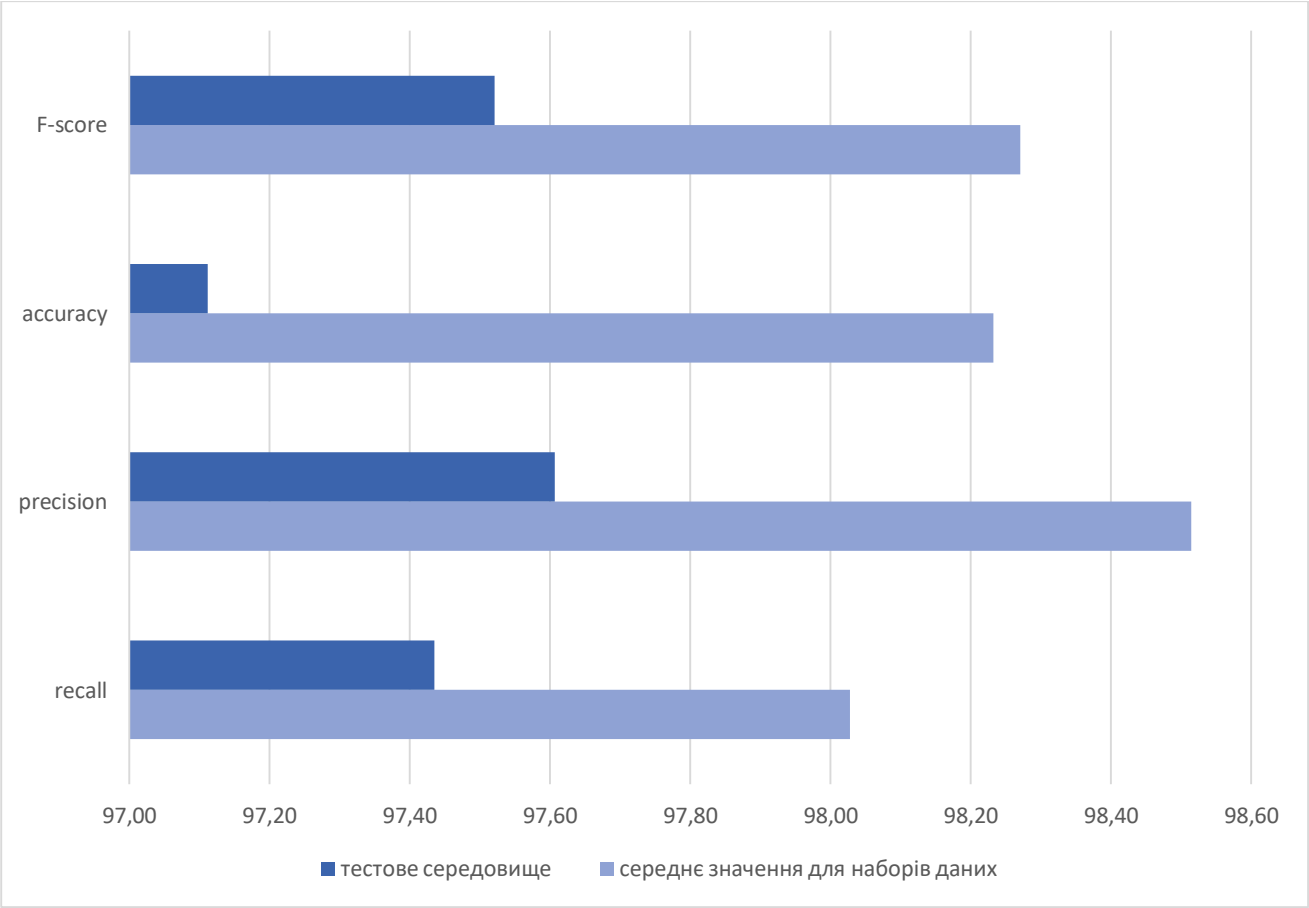


Рисунок 3.10 – Порівняння результатів тестування у різних середовищах для метрик повноти, точності, акуратності та F-міри

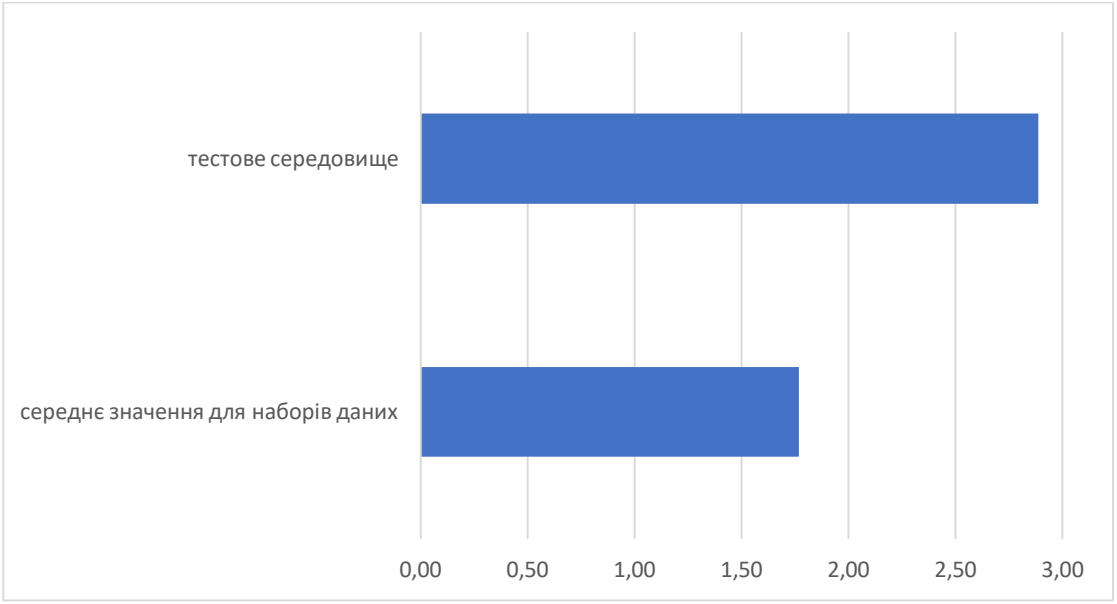


Рисунок 3.11 – Порівняння результатів тестування у різних середовищах для метрики помилки

– відсоток помилкових спрацювань лише на 1,12% більший у реальній системі у порівнянні із середнім значенням даних, що було отримано в результаті тестування наборами даних.

3.4 Висновки до розділу

Розробка та тестування системи виявлення атаки на відновлення паролю у мережі включала кілька ключових етапів.

Для реалізації системи були підключені необхідні бібліотеки для роботи з багатовимірними масивами, математичних операцій, обробки та аналізу даних, а також машинного та глибокого навчання. Це забезпечило основу для побудови моделі.

Створення моделі включало послідовність шарів, де кожен шар мав свою специфічну роль. Було вибрано оптимальні параметри для навчання моделі, що включають вибір функцій активації, оптимізаторів та метрик оцінки.

Виконано завантаження та підготовку набору даних, включаючи їх розділення на тренувальні та тестові частини. Важливою частиною процесу була

зміна форми даних для сумісності з вимогами RNN. Дані для навчання були зібрані та підготовлені у форматі CSV, де кожен запис включав різноманітні параметри, що допомагали виявити аномалії у мережевому трафіку та поведінці користувачів. Параметри навчання, такі як кількість епох та розмір пакетів, були обрані для забезпечення оптимальної продуктивності моделі та запобігання перенавчанню.

Тестування системи проводилося у два етапи: на змодельованій системі та в реальних умовах. Перший етап тестування на платформі Google Colab показав високу ефективність системи, зокрема точність (98,15%-98,8%) та повноту (97,64%-98,41%). Другий етап тестування в реальних умовах показав трохи нижчі, але все ж високі показники точності та повноти, що свідчить про здатність системи адекватно реагувати на реальні загрози.

Порівняння результатів тестування в різних середовищах виявило стабільність та надійність системи. Незначне зниження точності та повноти в реальному середовищі (0,59%-0,91%) вказує на те, що система здатна адаптуватися до різноманітних умов та типів атак. Помилкові спрацювання в реальному середовищі були лише на 1,12% більшими, що свідчить про високу стійкість системи до реальних загроз.

Загалом, результати свідчать про те, що розроблена система виявлення атаки на відновлення паролю у мережі є ефективним інструментом для забезпечення безпеки. Вона демонструє високу продуктивність та здатність до адаптації в різних умовах, що робить її надійним рішенням для виявлення кібератак. Подальші покращення можуть бути спрямовані на зниження кількості помилково позитивних та помилково негативних випадків, що ще більше підвищить точність та надійність системи.

ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено систему виявлення атаки на відновлення паролю у мережі, що базується на використанні рекурентних нейронних мереж для аналізу мережевого трафіку.

Проведено детальний аналіз існуючих методів та підходів до виявлення атак на паролі, що дозволило обґрунтувати вибір RNN як основного інструменту для вирішення поставленої задачі. Це вибір забезпечив ефективність в обробці послідовних даних, необхідних для виявлення аномалій у поведінці користувачів. Вивчено структуру та особливості трьох наборів даних: KDDCup99, UNSW-NB15 та CICIDS2017, які були використані для тренування та тестування моделі, що дозволило забезпечити надійну та всебічну оцінку продуктивності розробленої системи. Розроблено алгоритм, що використовує RNN для виявлення атак на відновлення паролю. Використання TensorFlow та інших інструментів для машинного навчання забезпечило створення гнучкої та потужної моделі, здатної ефективно аналізувати великі обсяги мережевого трафіку.

Проведено навчання моделі на обраних наборах даних, а також її тестування у змодельованих та реальних умовах. Модель продемонструвала високу точність (98,15%-98,8%) та повноту (97,64%-98,41%), що підтвердило її ефективність у виявленні атак на відновлення паролю. Порівняння результатів тестування на різних наборах даних показало, що система зберігає високі показники продуктивності незалежно від умов тестування. Виявлена стабільність та надійність системи, що робить її придатною для використання у реальних мережах.

Розроблена система виявлення атак на відновлення паролю демонструє високу точність та надійність, що робить її ефективним інструментом для підвищення рівня інформаційної безпеки у локальних мережах. Результати дослідження можуть бути використані для вдосконалення існуючих систем безпеки, що значно знизить ризик несанкціонованого доступу до мережевих ресурсів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Атаки на відновлення пароля. URL: https://cqr.company/ua/web-vulnerabilities/password-recovery-attacks/ (дата звернення 21.02.2024)

2. Ebu Yusuf Güven, Ali Boyaci, Muhammed Ali Aydin. A Novel Password Policy Focusing on Altering User Password Selection Habits: A Statistical Analysis on Breached Data. *Computers & Security*. 2022. Vol. 113. DOI: 10.1016/j.cose.2021.102560.

3. Xuerui Wang, Zheng Yan, Rui Zhang, Peng Zhang. Attacks and defenses in user authentication systems: A survey. *Journal of Network and Computer Applications.* 2021. Vol. 188. DOI: 10.1016/j.jnca.2021.103080.

4. Rafael Veras, Christopher Collins, and Julie Thorpe. A Large-Scale Analysis of the Semantic Password Model and Linguistic Patterns in Passwords. *ACM Trans*. 2021. Vol. 24, No. 3. DOI: 10.1145/3448608

5. Runhan Feng, Ziyang Yan, Shiyan Peng, and Yuanyuan Zhang. Automated detection of password leakage from public GitHub repositories. *In Proceedings of the 44th International Conference on Software Engineering (ICSE '22).* 2022. PP. 175–186. DOI: 10.1145/3510003.3510150

6. A. Kanta, I. Coisel and M. Scanlon. A Novel Dictionary Generation Methodology for Contextual-Based Password Cracking. *IEEE Access*. 2022. Vol. 10, PP. 59178-59188. DOI: 10.1109/ACCESS.2022.3179701.

7. S. Shourya, I. Venkatachalam, H. Patel, M. Mittal. Comparative and Preventive Analysis of Dictionary Attacks. *Recent Advances in Electrical and Electronic Engineering. ICSTE 2023. Lecture Notes in Electrical Engineering*. 2024. Vol. 1071. DOI: 10.1007/978-981-99-4713-3_42

8. CAPEC-55: Rainbow Table Password Cracking. URL: https://capec.mitre.org/data/definitions/55 (дата звернення 11.03.2024)

9. Understanding BruteForce — RainbowTables and Dictionary Attacks. URL: https://medium.com/@hsuyaji/bruteforce-rainbowtables-and-dictionary-attacks-98c24e20252f (дата звернення 12.03.2024)

10. R. Salama, F. Al-Turjman, S. Bhatla and S. P. Yadav. Social engineering attack types and prevention techniques- A survey. *International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*. 2023. PP. 817-820. DOI: 10.1109/CICTN57981.2023.10140957.

11. Z. Wang, H. Zhu and L. Sun. Social Engineering in Cybersecurity: Effect Mechanisms, Human Vulnerabilities and Attack Methods. *IEEE Access*. 2021. Vol. 9, PP. 11895-11910. DOI: 10.1109/ACCESS.2021.3051633.

12. W. Syafitri, Z. Shukur, U. A. Mokhtar, R. Sulaiman and M. A. Ibrahim. Social Engineering Attacks Prevention: A Systematic Literature Review. *IEEE Access*. 2022. Vol. 10, PP. 39325-39343. DOI: 10.1109/ACCESS.2022.3162594.

13. Arjun Singh, Pushpa Choudhary, Akhilesh kumar singh, Dheerendra kumar tyagi. Keylogger Detection and Prevention. *Journal of Physics: Conference Series*. 2021. Vol. 2007, No 1, P. 012005. DOI: 10.1088/1742-6596/2007/1/012005

14. A. B. Ruhani, M. Zolkipli. Keylogger: The Unsung Hacking Weapon. *Bij*. 2023. Vol. 6, No. 1, PP. 33-43.

15. Ekele Victoria C., Ayodele Ariyo Adebiyi. Keylogger Detection: A Systematic Review. *International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*. 2023. PP. 1-6. DOI: 10.1109/SEB-SDG57117.2023.10124477.

16. A. Kumar, K.K. Dubey, H. Gupta, M. Memoria, K. Joshi. Keylogger Awareness and Use in Cyber Forensics. *Rising Threats in Expert Applications and Solutions. Lecture Notes in Networks and Systems*. 2022. Vol. 434. DOI: 10.1007/978-981-19-1122-4_75

17. Z. Alkhalil, C. Hewage, L. Nawaf, I Khan I. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Front. Comput. Sci*. Vol. 3, P. 563060. DOI: 10.3389/fcomp.2021.563060

18. A. K. Jain, B.B. Gupta. A survey of phishing attack techniques, defence mechanisms and open research challenges. *Enterprise Information Systems*. 2021. Vol. 16, No 4, PP. 527–565. DOI: 10.1080/17517575.2021.1896786

19. A. Basit, M. Zafar, X. Liu. A comprehensive survey of AI-enabled phishing

attacks detection techniques. *Telecommun Syst*. 2021. Vol. 76, PP. 139–154. https://doi.org/10.1007/s11235-020-00733-2

20. J. Lee, Y. Lee, D. Lee, H. Kwon, D. Shin. Classification of Attack Types and Analysis of Attack Methods for Profiling Phishing Mail Attack Groups. *IEEE Access*. 2021. Vol. 9, PP. 80866-80872. DOI: 10.1109/ACCESS.2021.3084897.

21. N. Tihanyi, T. Bisztray, B. Borsosm, S. Raveau. Privacy-Preserving Password Cracking: How a Third Party Can Crack Our Password Hash Without Learning the Hash Value or the Cleartext. *IEEE Transactions on Information Forensics and Security*. 2024. Vol. 19, PP. 2981-2996. DOI: 10.1109/TIFS.2024.3356162.

22. Janmaya Kumar Mishra, Midhunchakkaravarthy Janarthanan. GPU-based security of password hashing in cloud computing. *Materials Today: Proceedings*. 2022. Vol. 60, Part 2, PP. 939-944. DOI: 10.1016/j.matpr.2021.11.077.

23. Z. Bao et al. Automatic Search of Meet-in-the-Middle Preimage Attacks on AES-like Hashing. *Advances in Cryptology*. *Lecture Notes in Computer Science*. 2021. Vol. 12696. DOI: 10.1007/978-3-030-77870-5_27

24. Burp Suite Enterprise Edition. URL: https://portswigger.net/burp/enterprise (дата звернення 25.02.2024)

25. hydra | Kali Linux Tools. URL: https://www.kali.org/tools/hydra/ (дата звернення 25.02.2024)

26. john | Kali Linux Tools. URL: https://www.kali.org/tools/john/ (дата звернення 25.02.2024)

27. medusa | Kali Linux Tools. URL: https://www.kali.org/tools/medusa/ (дата звернення 26.02.2024)

28. RainbowCrack - Crack Hashes with Rainbow Tables. URL: http://project-rainbowcrack.com/ (дата звернення 27.02.2024)

29. L0phtCrack. URL: https://l0phtcrack.gitlab.io/ (дата звернення 27.02.2024)

30. Maltego. URL: https://www.maltego.com/ (дата звернення 4.03.2024)

31. Aircrack-ng. URL: https://www.aircrack-ng.org/ (дата звернення 4.03.2024)

32. Metasploit | Penetration Testing Software, Pen Testing Security. URL: https://www.metasploit.com/ (дата звернення 5.03.2024)

33. CAPEC - Common Attack Pattern Enumeration and Classification. URL: https://capec.mitre.org/ (дата звернення 6.03.2024)

34. Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas. Chapter 5 - Recurrent neural networks, *Deep Learning for Robot Perception and Cognition, Academic Press*. 2022. PP. 101-115. DOI: 10.1016/B978-0-32-385787-1.00010-5.

35. P. Liu, J. Wang, Z. Zeng. An Overview of the Stability Analysis of Recurrent Neural Networks With Multiple Equilibria. *IEEE Transactions on Neural Networks and Learning Systems*. 2023. Vol. 34, No. 3, PP. 1098-1111. DOI: 10.1109/TNNLS.2021.3105519.

36. W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, K. -R. Müller. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proceedings of the IEEE*. 2021. Vol. 109, No. 3, PP. 247-278. DOI: 10.1109/JPROC.2021.3060483.

37. Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*. 2022. Vol. 33, No. 12, PP. 6999-7019. DOI: 10.1109/TNNLS.2021.3084827.

38. N. Ketkar, J. Moolayil. Convolutional Neural Networks. In: Deep Learning with Python. *Apress*. 2021. DOI: https://doi.org/10.1007/978-1-4842-5364-9_6

39. Alankrita Aggarwal, Mamta Mittal, Gopi Battineni. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*. 2021. Vol. 1, Issue 1. DOI: 10.1016/j.jjimei.2020.100004.

40. J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering*. 2023. Vol. 35, No. 4, PP. 3313-3332. DOI: 10.1109/TKDE.2021.3130191.

41. TensorFlow. URL: https://www.tensorflow.org/ (дата звернення 11.03.2024)

42. NumPy. URL: https://numpy.org/ (дата звернення 12.03.2024)

43. pandas - Python Data Analysis Library. URL: https://pandas.pydata.org/ (дата звернення 14.03.2024)

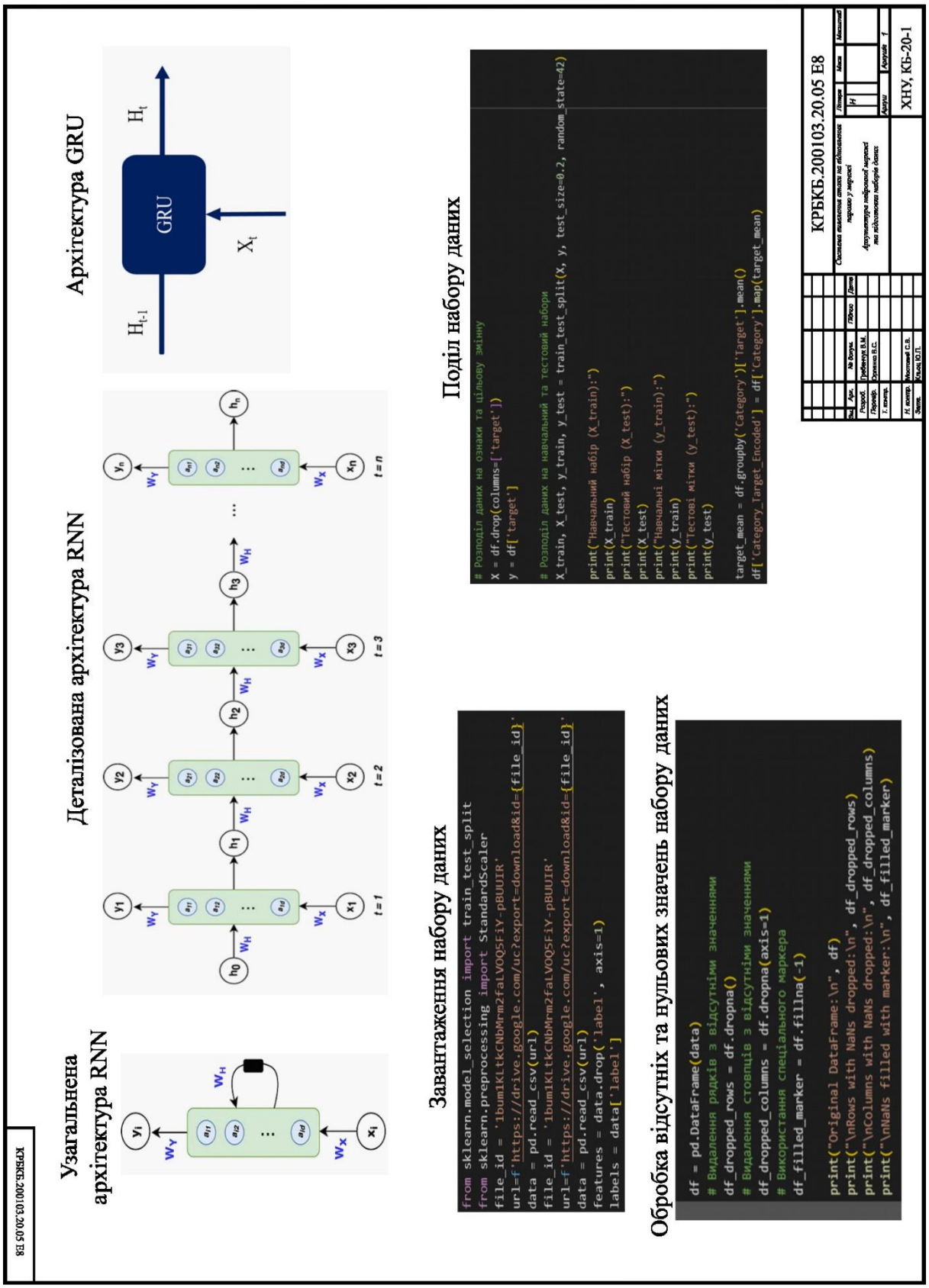
44. Keras: Deep Learning for humans. URL: https://keras.io/ (дата звернення 17.03.2024)

45. Google Colab. URL: https://colab.google/ (дата звернення 21.03.2024)

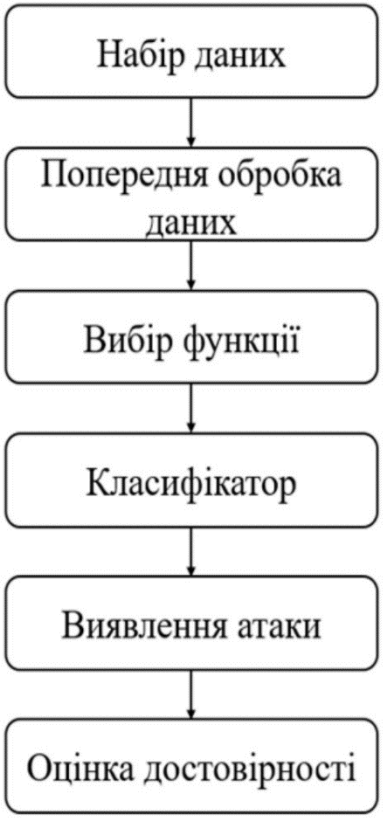
ДОДАТОК А

(обов'язковий)

Копія графічної частини



Алгоритм RNN нейронної мережі
для виявлення атак на відновлення паролів

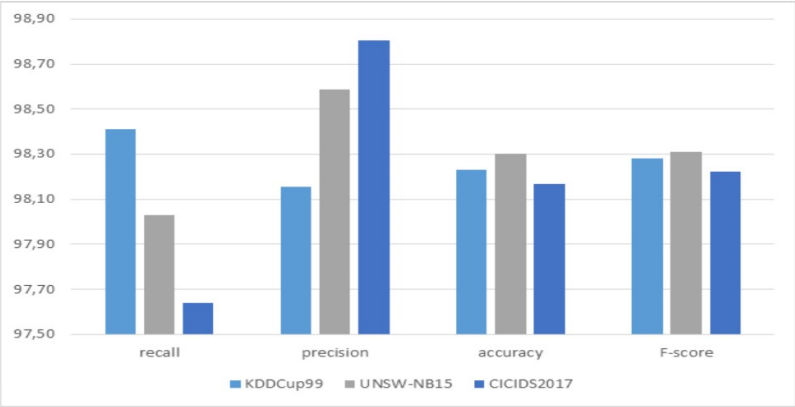


Етап попередньої обробки даних

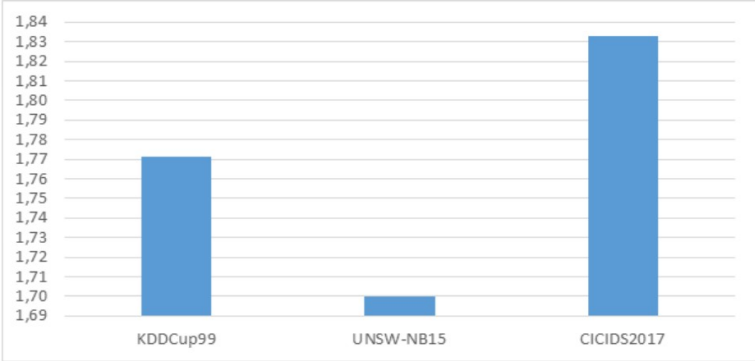
Класифікація трафіку

Зм.	Арк.	№ докум.	Підпис	Дата	КРБКБ.200103.20.05 Е8		
Розроб.		Гребенчук В.М.			Система виявлення атаки на відновлення паролю у мережі	Літера	Маса	Масштаб
Перевір.		Орленко В.С.			Алгоритми навчання нейронної мережі, підготовки даних та аналізу мережевого трафіку	Н		
Т. контр.						Аркуш	Аркушів 1	
Н. контр.		Мостовий С.В.				ХНУ, КБ-20-1		
Затв.		Кільоц Ю.П.						

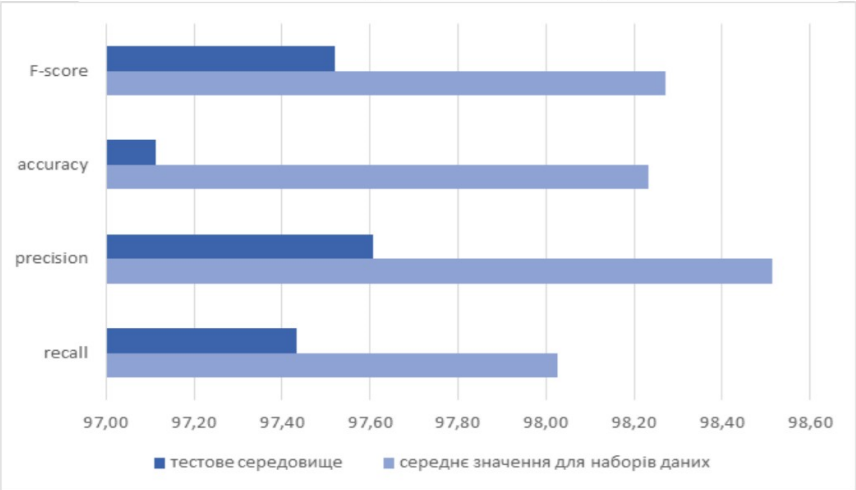
Результати тестування з використанням різних наборів даних для метрик повноти, точності, акуратності та F-міри



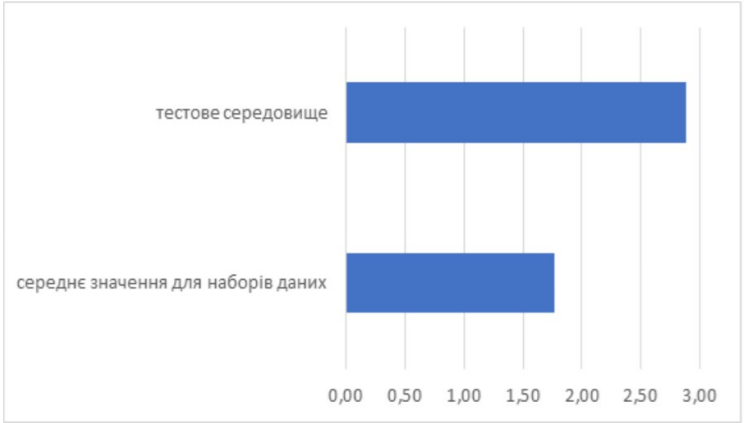
Результати тестування з використанням різних наборів даних для метрики помилки



Порівняння результатів тестування у різних середовищах для метрик повноти, точності, акуратності та F-міри



Порівняння результатів тестування у різних середовищах для метрики помилки



Зм.	Арк.	№ докум.	Підпис	Дата	КРБКБ.200103.20.05 Е8		
Розроб.		Гребенчук В.М.			Система виявлення атаки на відновлення пароля у мережі	Літера	Маса	Масштаб
Перевір.		Орленко В.С.			Оцінка ефективності системи виявлення атаки на відновлення паролю мережі	Н		
Т. контр.						Аркуш	Аркушів 1	
Н. контр.		Мостовий С.В.				ХНУ, КБ-20-1		
Затв.		Кільоц Ю.П.						

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Гребенчука Валентина Максимовича
ПІБ здобувача вищої освіти

Студента ФІТ, 4 курсу, групи КБ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10.06.2024
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 8%**

	Документ		Сумарний збіг по Базі Даних	
ID: 129627 Назва: Система виявлення атаки на відновлення паролю у мережі Додано в БД: 2024-06-11 Автора: Гребенчук В.М. Керівники: Орленко В.С. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	75288	604	1225 (2%)	14 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016347110

Дата перевірки:
11.06.2024 20:43:52 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
11.06.2024 22:35:43 EEST

ID користувача:
100008300

Назва документа: **Гребенчук_Плагіат**

Кількість сторінок: **57** **Кількість слів:** **11076** **Кількість символів:** **84952** **Розмір файлу:** **863.60 KB** ID **файлу:** **1016148834**

2.93%
Схожість

Найбільша схожість: **1.48% з джерелом з Бібліотеки** (ID **файлу:** **1016148836**)

1.54% Джерела з Інтернету	149	**Сторінка** 59
1.68% Джерела з Бібліотеки	42	**Сторінка** 60

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система виявлення атаки на відновлення паролю у мережі

Автор: Гребенчук Валентин Максимович

Спеціальність: 125 – Кібербезпека

Освітня програма: освітньо-професійна

Науковий керівник: Орленко Вікторія Сергіївна, канд. техн. наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 97,07%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 99%.

Згідно з Положенням про систему забезпечення академічної доброчесності у ХНУ (https://khmnu.edu.ua/wp-content/uploads/normatyvni-dokumenty/polozhennya/pro-systemu-zabezpechennya-akademichnoyi-dobrochesnosti.pdf, Додаток В) кваліфікаційна робота, виконана за , освітньо-професійною програмою, кількісні показники рівня унікальності тексту у відсотках до загального обсягу матеріалу в якій складає 75-100 %, визнається роботою з високою унікальністю тексту: «Текст вважається унікальним і не потребує додаткових дій щодо запобігання неправомірним запозиченням».

Керівник роботи _____ Вікторія ОРЛЕНКО

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Гребенчук Валентин Максимович

Тема Система виявлення атаки на відновлення паролю у мережі

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3; кількість сторінок записки 61.

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі була розроблена система виявлення атак на відновлення паролю в мережі. Для цього було обрано тип нейронної мережі, призначений для ідентифікації зловмисного трафіку. У процесі розробки було виконано такі завдання: розроблено алгоритм виявлення атаки, підготовлено набори даних для навчання нейронної мережі, здійснено навчання нейронної мережі та реалізовано систему виявлення атак. Крім того, було розроблено тестове середовище та проведено оцінку ефективності розробленої системи.

2. Висновок про відповідність кваліфікаційної роботи завданню У кваліфікаційній роботі було виконано поставлене завдання як у теоретичній, так і в практичній частині.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі роботи наведена загальна характеристика задачі, визначені об'єкт, предмет та методи дослідження, а також сформульована мета. Зазначені завдання, що потрібно виконати для досягнення поставленої мети, проведений аналіз досліджуваної проблеми та обґрунтований підхід до її вирішення. У першому розділі розглядаються атаки на відновлення паролів та типи атак на паролі. Наступні розділи присвячені розробці алгоритмів реалізації та підготовці навчальних даних, а також системи виявлення атак на відновлення паролю у мережі, включаючи вибір нейронної мережі для аналізу мережевого трафіку, навчання нейронної мережі та тестування реалізованої системи. Також була проведена оцінка ефективності розробленої системи

4. Позитивні сторони Кваліфікаційна робота має практичну цінність. Вона полягає у розробці системи виявлення атак на відновлення паролю в мережі, яка використовує нейронні мережі для аналізу мережевого трафіку. Це забезпечує своєчасне виявлення зловмисних дій та підвищує рівень безпеки інформаційних систем. Завдяки цьому організація може ефективно протидіяти кіберзагрозам та мінімізувати ризики фінансових втрат та втрати конфіденційної інформації.

5. Негативні сторони роботи Система також може потребувати значних обчислювальних ресурсів для аналізу мережевого трафіку в режимі реального часу, що може бути викликом для організацій з обмеженими ресурсами.

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення кваліфікаційної роботи відповідає темі роботи та виконане з дотриманням стандартів. В цілому, графічне оформлення є якісним, а пояснювальна записка відповідає нормам оформлення.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки, оскільки весь матеріал роботи є структурованим, чітким та послідовним. Усі розділи роботи мають логічну послідовність, що сприяє зрозумінню викладеного матеріалу в рамках теми роботи. Графічний матеріал допомагає наочно продемонструвати доцільність та ефективність прийнятих рішень для досягнення мети.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінки «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Мартинюк Валерій Володимирович,
завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки, доктор технічних наук, професор

« 12 » червня 2024 (підпис)