Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації
(повна назва)
Кафедра _Радіотехнологій інформаційно-комунікаційних систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження методів Data Science у системах кібербезпеки
(тема)

Виконав: студент 2 курсу, групи АПСм-22-1

_____Прокіпець. В. О._____
(прізвище, ініціали

Спеціальність _126 Інформаційні системи та технології
(код і повна назва спеціальності)

Тип програми освітньо-професійна_
Освітня програма _Архітектурне проєктування інформаційних систем
(повна назва освітньої програми)

Керівник проф. Кузьомін О.Я.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри _____ ___Олександр ЗАРУДНИЙ_____
(підпис) (прізвище, ініціали)

2024р.

Не містить відомостей заборонених до відкритого публікування

Керівник _____ Кузьомін О.Я.

Студент _____ Прокіпець В. О.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Архітектурне проєктування інформаційних систем
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 20 ___ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Прокіпцю Валентину Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів Data Science у системах кібербезпеки

затверджена наказом університету від 03 жовтня 2023 р. № 1295 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 січня 2024 р.

3. Вихідні дані до роботи Провести аналіз практик застосування методологій data science у кібербезпеці, створити доданок для практичного дослідження питання.

4. Перелік питань, що потрібно опрацювати в роботі

ВСТУП

1. Дослідження існуючих рішень застосування методів науки про дані у кібербезпеці
2. Постановка задачі кваліфікаційної роботи
3. Реалізація проектного завдання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
Слайди у форматі Power Point (назва та мета роботи, вступ, поширення пандемії; аналіз технологій, вхідні дані, постановка задачі, використані технології, глибоке навчання детально, попередня обробка зображень, інструменти розробки, датасет, написання програмного коду, звітність з навчання моделі, приклад роботи програми)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Кузьомін О. Я.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ.	01.09.2023	вик.
2	Підбір літератури за темою роботи	02.09-20.09.2023	вик.
3	Огляд взаємодії data science та кібербезпеки	21.09-23.10.2023	вик.
4	Вивчення сучасних досягнень	24.10-24.11.2023	вик.
	обох напрямків		
5	Вибір необхідних технологій	25.11-25.12.2023	вик.
	для виконання проекту		
6	Написання та тестування програмного коду	26.12.2023-06.01.2024	вик.
7	Оформлення презентаційного матеріалу,	07.01-15.01.2024	вик.
	підготовка до захисту у ЕК		

Дата видачі завдання __01 вересня_____2023 р.

Студент _____
(підпис)

Керівник роботи _____ __проф. Кузьомін О.Я._____
(підпис) (посада, прізвище, ініціали)

АНОТАЦІЯ

Пояснювальна записка: 60 с., 13 рис., 2 табл., 13 джерел, 2 додатки.

ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, КІБЕРБЕЗПЕКА, DATA SCIENCE

Робота спрямована на дослідження та аналіз використання методологій Data Science в сфері кібербезпеки. У роботі розглядаються основні аспекти використання аналітичних і статистичних методів для виявлення та протидії кіберзагрозам. Розглядаються різноманітні підходи до обробки великих обсягів даних, використовуючи алгоритми машинного навчання та штучного інтелекту для виявлення аномалій та передбачення можливих атак.

У роботі детально аналізуються інструменти та техніки, які використовуються в Data Science для забезпечення кібербезпеки, такі як класифікація, кластеризація, аналіз вимірювань та важливість вибору ознак. Досліджуються важливі питання збору та обробки данних, а також забезпечення конфіденційності та інтеграції з існуючими системами безпеки.

ANNOTATION

Explanatory note: 60 p., 13 figures, 2 tables, 13 sources, 2 appendices.

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, CYBER SECURITY, DATA SCIENCE

The work is aimed at research and analysis of the use of Data Science methodologies in the field of cyber security. The work considers the main aspects of using analytical and statistical methods to detect and counter cyber threats. Various approaches to processing large volumes of data are considered, using machine learning and artificial intelligence algorithms to detect anomalies and predict possible attacks.

The paper analyzes in detail the tools and techniques used in Data Science for cybersecurity, such as classification, clustering, measurement analysis, and the importance of feature selection. Important issues of data collection and processing, as well as ensuring confidentiality and integration with existing security systems are explored.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАК ТА СКОРОЧЕНЬ

ШІ – штучний інтелект

МІС – медична інформаційна система

RF – Random forrest (випадковий ліс)

AdaBoost – Adaptive boosting

GAN – generative adversarial network (генеративно-змагальна мережа)

ВСТУП

Загрози кібербезпеці стають дедалі складнішими, що потребує інноваційних підходів до виявлення загроз і їх пом’якшення. Наука про дані (Data science) з її можливостями для аналізу даних, розпізнавання образів і прогнозного моделювання знайшла широке застосування в посиленні заходів кібербезпеки.

У більш загальному плані існує потреба в засобах захисту кібербезпеки [1], які виходять за рамки використання реактивних моделей, наприклад, заснованих на виявленні сигнатур або рішень на основі правил, до прогнозних моделей. Прогнозні моделі використовуватимуть статистичні принципи виявлення аномалій для виявлення та контролю нових шкідливих програм і нових стратегій використання програмного забезпечення. Такі моделі використовуватимуть дані, зібрані всередині та на периферії організацій, для постійного оновлення та покращення захисту. Найважливіше, що вони відповідатимуть поведінці атаки, яка називається «ланцюгом знищення», яка починається з початкового проникнення в мережу та прогресує до ексфільтрації, коли викрадені дані передаються назад противнику. Побудова ефективного захисту включатиме механізми, керовані даними, що поєднують стохастичне моделювання, динамічні графіки та концепції статистичного контролю.

В цій роботі розглядається застосування основних методів та технологій, притаманних Data Science, у кібербезпеці.

1.ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ ЗАСТОСУВАННЯ МЕТОДІВ НАУКИ ПРО ДАНІ У КІБЕРБЕЗПЕЦІ

1.1 Загальні положення

Кібербезпека для мережевих систем була предметом досліджень більше трьох десятиліть і актуальна для широкого спектру мережевих програм. Такі системи вразливі до атак, спрямованих на порушення або пошкодження функціональності системи, і атак, спрямованих на порушення інформаційної безпеки. Розподілена відмова в обслуговуванні та блокування каналів є прикладами атак, які порушують функціональність системи. Вторгнення та підслуховування є прикладами атак, які порушують інформаційну безпеку, отримуючи несанкціонований доступ до приватних системних даних або приватних даних користувача та порушуючи конфіденційність, цілісність і доступність. Такі атаки можуть відбуватися послідовно або паралельно, наприклад, атака з вимогою викупу, яка викрадає конфіденційну інформацію для проникнення в пристрій, викрадення операційної системи та використання її для здійснення атак на інші мережеві пристрої. Два типи мереж представляють особливий інтерес і мають різні вразливості: централізовано керовані корпоративні мережі та нецентралізовано адміністровані мережі, такі як Інтернет Речей (Internet of things). У той час як корпоративні та спеціальні системи Інтернету речей можуть створювати різні виклики безпеці та відрізнятися з точки зору застосовних рішень безпеки, вони мають деякі спільні риси, зокрема неоднорідність мережі, їх великий масштаб і безліч уразливостей для атак. Вони обидва можуть мати складні взаємозв’язки різноманітних датчиків, виконавчих механізмів і пристроїв, починаючи від хмарних і периферійних ресурсів до пристроїв кінцевого користувача (камери, мікрофони, прилади тощо), які мають різні характеристики та можливості.

Окрім різноманітності пристроїв, ці системи є багаторівневими як з точки зору функціональності, так і з точки зору адміністрування мережі. Основна відмінність між корпоративними мережами та спеціальними мережами полягає в тому, що перші зазвичай мають централізоване адміністрування, тоді як у других адміністрування часто децентралізоване, наприклад, однорангові протоколи. Велика різноманітність і розподіл точок доступу в обох типах мереж створює проблеми безпеки, оскільки відкриває систему для великої кількості потенційних точок атаки, кожна з яких захищена на різних рівнях складності та ресурсів, наприклад паролів, автентифікації або брандмауерів. . Винахідливий зловмисник може застосувати різні інструменти для здійснення різних типів атак, наприклад, скоординованого вторгнення або розподіленої атаки на відмову в обслуговуванні (DDOS) на інфраструктуру підприємства, на додаток до атак типу "людина посередині" (MITM) і фішингових атак на кінцеві користувачі. Останні часто включають третю сторону, яка перехоплює конфіденційні комунікації між організацією та її клієнтами; Порушення в DigiNotar і Equifax є реальними прикладами. Ще інші зловмисники можуть мати на меті пошкодження даних, що стосуються конкретної програми, наприклад, атаки з впровадженням помилкових даних (FDI) на датчики та виконавчі механізми. Такі атаки часто спрямовані на промислові системи, щоб підробити базові процеси керування та автоматизації; реальні приклади включають такі атаки, як інциденти системи контролю стічних вод Stuxnet та Maroochy Shire.

Використовуючи вразливі місця в мережевих об’єктах і кінцевих точках з обмеженими ресурсами, складні зловмисники можуть подолати класичні механізми безпеки, які часто мають обмежені можливості для адаптації до мінливих сценаріїв агресивних атак. Щоб протистояти таким підступним атакам, майбутні системи повинні будуть включати проактивні підходи до кібербезпеки, що керуються даними, зокрема ті, що використовують підходи на основі статистики та штучного інтелекту (ШІ).

Критичні кіберфізичні інфраструктури, такі як інтелектуальна мережа, фінансові мережі, мережі моніторингу працездатності та транспортні мережі, повинні ретельно контролюватися на рівні пристроїв, транспорту та додатків як у внутрішніх, так і в крайових точках мережі. Таким чином, існує потреба в підходах до аналізу складних даних, створених під час багаторівневого моніторингу мережі, на основі статистичних даних. Деякі з атрибутів безпеки корпоративної системи заслуговують на згадку, оскільки вони визначають тип даних, які є найбільш цінними для виявлення атак.

Методи науки про дані, зокрема машинне навчання, виявлення аномалій і аналітика великих даних, використовуються для підвищення кібербезпеки кількома способами:

- Виявлення та запобігання загрозам. Моделі машинного навчання, навчені на історичних даних, використовуються для виявлення моделей, що вказують на кіберзагрози, забезпечуючи раннє виявлення та механізми проактивного захисту.

- Аналіз поведінки: Наука про дані допомагає аналізувати поведінку користувачів, щоб виявити відхилення від нормальних моделей, які можуть сигналізувати про внутрішні загрози або скомпрометовані облікові записи.

- Аналіз великих даних: Обробка величезних обсягів даних у режимі реального часу дозволяє швидко оцінювати загрози та реагувати на них, що є ключовим аспектом у сучасному кібернетичному середовищі.

- Прогнозне моделювання: Удосконалені алгоритми передбачають потенційні вразливості та атаки, що дозволяє організаціям зміцнювати захист в очікуванні загроз.

1.2 Огляд основних технологій

Розглянемо застосування різних методологічних процесів, притаманних науці про дані, у сфері кібербезпеки

1.2.1 Machine learning (машинне навчання)

Використання методів Machine Learning в кібербезпеці дозволяє підвищити ефективність виявлення загроз, вразливостей та атак, а також зменшити час реакції на події. Ось деякі з основних способів, які демонструють важливість Machine Learning у кібербезпеці:

- Виявлення аномалій (Anomaly Detection): Методи Machine Learning можуть аналізувати трафік мережі та системи, щоб виявити аномалії, які можуть бути ознаками атак або вразливостей. Для цього можна використовувати алгоритми класифікації, кластеризації та навчання без учителя. [2]

- Виявлення вразливостей (Vulnerability Detection): Машинне навчання допомагає виявити потенційні вразливості в програмному забезпеченні шляхом аналізу коду та тестування на предмет вразливостей.

- Прогнозування загроз (Threat Prediction): Моделі Machine Learning можуть прогнозувати можливі кібератаки, а також ідентифікувати потенційні цілі та способи атаки на основі аналізу історичних даних.

- Фільтрація спаму та відсіювання шкідливого ПЗ: Машинне навчання може бути використане для автоматичної фільтрації спаму в електронній пошті та виявлення шкідливого програмного забезпечення на комп'ютерах та серверах.

- Аналіз журналів подій (Log Analysis): Методи Machine Learning дозволяють аналізувати великі обсяги журналів подій для виявлення підозрілих активностей та атак[3].

1.2.2 Natural Language Processing (Обробка живого мовлення)

Використання Natural Language Processing (NLP), або обробки природної мови, у кібербезпеці є дуже важливим, оскільки ця технологія дозволяє

аналізувати та розуміти мовлення, текстові дані та комунікації, що може бути корисним для виявлення кібератак, вразливостей та інших загроз. Ось кілька способів, які демонструють важливість NLP в кібербезпеці:

а) Аналіз електронної пошти та комунікацій:

1) Виявлення фішингу: Моделі NLP можуть аналізувати текстові дані в електронній пошті для виявлення спроб фішингу та атаки шляхом імітації легітимних комунікацій[4].

2) Виявлення загроз в тексті: NLP може виявляти загрози та агресію в текстах, що допомагає у виявленні зловмисних повідомлень та коментарів.

б) Моніторинг соціальних медіа та форумів: Виявлення дискусій щодо атак: Аналіз текстів на соціальних медіа та форумах може допомогти виявити обговорення кібератак або викриття вразливостей.

в) Аналіз журналів подій:

1) Виявлення аномалій в системних журналах: Моделі NLP можуть аналізувати текстові журнали подій та виявляти аномалії, які можуть свідчити про атаку або вразливість.

г) Моніторинг новин та публікацій:

1)Виявлення нових загроз: Системи NLP можуть сканувати новини та статті для виявлення нових загроз або інформації про кібератаки.

д) Класифікація текстів:

1) Виявлення загроз у тексті: Моделі NLP можуть класифікувати текстові дані, щоб визначити, чи містить текст загрози або підозрілі вирази[5].

е) Аналіз внутрішніх комунікацій в організаціях:

1) Виявлення витоків інформації: NLP може виявляти витоки конфіденційної інформації або підозрілу активність серед працівників.

1.2.3 Time Series Analysis

Використання аналізу часових рядів (Time Series Analysis) в кібербезпеці є важливим інструментом для виявлення та відстеження кібератак, моніторингу даних про безпеку та прогнозування майбутніх загроз. Ця технологія дозволяє аналізувати дані в часових рядах, виявляти аномалії та ідентифікувати патерни атак. Ось деякі способи використання аналізу часових рядів у кібербезпеці:

- Часові ряди можуть бути використані для аналізу трафіку мережі та виявлення аномалій, таких як несподіваний збільшений обсяг даних, відмови в обслуговуванні (DoS) або Distributed Denial of Service (DDoS) атаки[6].

- Аналіз часових рядів може допомагати виявляти зміни у системних журналах та журналах подій, що може свідчити про незвичайну активність або вразливості[7].

- З використанням часових рядів можна розробляти моделі для прогнозування майбутніх кібератак або загроз.

- IDS можуть використовувати аналіз часових рядів для виявлення атак, які здатні змінюватися з часом або мати певні змінні патерни.

- Аналіз часових рядів допомагає виявляти аномалії у логах додатків, які можуть свідчити про кібератаки або використання вразливостей[8].

- Використання аналізу часових рядів може допомагати виявляти незвичайну активність у системах ідентифікації та автентифікації, таку як спроби несанкціонованого доступу.

1.2.4 Reinforcement Learning

Використання Reinforcement Learning (RL), або навчання з підсиленням, в кібербезпеці є цікавим і перспективним напрямком. RL може бути використано для оптимізації прийняття рішень в умовах кібербезпеки, а також для

вдосконалення систем контролю та реакції на загрози. Ось деякі способи використання Reinforcement Learning у кібербезпеці:

- Агенти RL можуть бути навчені виявляти та реагувати на кібератаки у реальному часі, виконуючи дії для блокування або виявлення аномальної активності.

- RL може бути використаний для автоматизації моніторингу мережі та виявлення атак, визначаючи оптимальні стратегії для оборони.

- RL може допомагати в оптимізації політик безпеки, визначаючи найкращі налаштування та правила для систем безпеки.

- RL може бути використаний для пошуку вразливостей у програмному забезпеченні та мережевих системах, шукаючи аномалії та тестируючи на вразливості.

- Системи управління доступом можуть бути оптимізовані за допомогою RL для надійного та ефективного контролю доступу до ресурсів.

- RL може допомогти виробляти оптимальні стратегії реакції на кіберінциденти, включаючи автоматичне вимкнення атакованих систем або відновлення роботи.

- RL може бути використаний для оптимізації розподілу бюджету для кібербезпеки та вибору найкращих заходів захисту.

- RL може аналізувати та передбачати ризики в сфері кібербезпеки, що допомагає приймати більш обґрунтовані рішення з питань безпеки.

1.2.5 Bayesian Networks

Використання байєсівських мереж (Bayesian Networks) в кібербезпеці є важливою стратегією для моделювання та управління ризиками, виявлення загроз та вразливостей, а також для прийняття найкращих рішень у сфері кібербезпеки. Байєсівські мережі дозволяють моделювати ймовірнісні зв'язки

між подіями та оцінювати ризики на основі доступних даних. Ось деякі способи використання байєсівських мереж у кібербезпеці:

- Байєсівські мережі можуть бути використані для моделювання структури загроз та вразливостей, що допомагає визначити потенційні ризики.

- Вони дозволяють кібербезпеці аналізувати та квантифікувати ризики на основі ймовірностей та впливу різних подій.

- Байєсівські мережі можуть використовуватися для виявлення аномалій у мережевому трафіку або системних журналах подій.

- Вони можуть допомагати у прийнятті рішень щодо вибору оптимальних заходів забезпечення кібербезпеки відповідно до поточних умов.

- Байєсівські мережі можуть бути навчені виявляти атаки на основі аналізу системних журналів та інших джерел інформації.

- Вони можуть використовуватися для моніторингу даних про безпеку та виявлення порушень безпеки.

- Байєсівські мережі можуть допомагати моделювати внутрішні загрози та ідентифікувати джерела потенційних атак.

- Вони можуть покращити ефективність систем виявлення вторгнень, використовуючи ймовірнісні моделі.

1.2.6 Graph Analytics

Використання аналізу графів (Graph Analytics) в кібербезпеці є надзвичайно важливим інструментом для виявлення та аналізу кібератак, а також для побудови зв'язків між різними сутностями в інформаційних системах. Аналіз графів дозволяє моделювати та аналізувати взаємозв'язки між об'єктами, такі як користувачі, комп'ютери, програми та інші складові інфраструктури. Ось деякі способи використання аналізу графів у кібербезпеці:

- Графовий аналіз може використовуватися для виявлення аномалій у мережевому трафіку та взаємозв'язків, які можуть вказувати на кібератаки або вторгнення.

- В аналізі графів можна створити моделі поведінки користувачів і виявляти виходи за межі звичайних сценаріїв, що може свідчити про несанкціонований доступ.

- Графовий аналіз допомагає виявляти зв'язки між ботнетами, спамерами та іншими зловмисниками.

- Аналіз графів може допомагати визначати зв'язки між ресурсами, такі як файли, бази даних і сервери, що сприяє виявленню ризиків та точок вразливості.

- Графовий аналіз може виявляти відомості про те, як зломники отримують доступ до системи, і допомагати у їх ідентифікації.

- Він дозволяє аналізувати ефективність правил безпеки та вдосконалювати їх на основі виявлених зв'язків і ризиків.

- Графовий аналіз може створювати карту зв'язків між різними об'єктами в мережі, що допомагає візуалізувати структуру та потенційні загрози.

1.2.7 Симуляція та емуляція

Використання симуляції (simulation) і емуляції (emulation) в кібербезпеці є важливими підходами для вивчення та тестування безпеки інформаційних систем, мереж та додатків. Ці методи дозволяють моделювати різні сценарії кібератак, тести безпеки і вразливостей, а також проводити тренування персоналу з питань кібербезпеки. Ось деякі способи використання симуляції та емуляції в кібербезпеці:

- За допомогою симуляції та емуляції можна моделювати сценарії атак і тестувати системи на вразливості, не завдаючи реальних пошкоджень.

- Ці методи дозволяють тренувати кібербезпечних фахівців та інших працівників з питань реагування на кібератаки та виконання безпеки.

- За допомогою симуляції можна оцінити ефективність заходів захисту та виявити можливі недоліки або слабкі місця.

- Симуляція може бути використана для аналізу та передбачення розвитку кіберпогроз та атак.

- За допомогою симуляції можна підтвердити, що система відповідає вимогам щодо безпеки перед її впровадженням.

- Симуляція може бути використана для вивчення нових типів загроз та їх можливого впливу на системи безпеки.

- Емуляція дозволяє запускати та тестувати додатки в умовах, що імітують реальне середовище, для виявлення можливих вразливостей.

- Симуляція допомагає вивчати та вдосконалювати процедури реагування на кіберінциденти та вибору оптимальних стратегій.

1.2.8 Human Behavioral Models

Використання моделей людської поведінки (Human Behavioral Models) в кібербезпеці є важливим інструментом для розуміння та прогнозування дій користувачів, зокрема їхнього поведінки у віртуальних та фізичних середовищах. Ці моделі допомагають виявляти аномалії та загрози, а також виробляти рекомендації для забезпечення безпеки. Ось деякі способи використання моделей людської поведінки в кібербезпеці:

- Моделі людської поведінки дозволяють виявляти аномальні дії користувачів, такі як надмірні запити, несподівані патерни дій, надмірний обсяг даних, який завантажується, тощо.

- Вони допомагають ідентифікувати можливі загрози, такі як фішинг, соціальний інжиніринг, шкідливі програми, шпигунське програмне забезпечення тощо.

- Моделі поведінки можуть бути використані для керування доступом, визначаючи, коли та які ресурси можуть бути доступні користувачам на основі їхньої типової поведінки.

- Вони допомагають аналізувати журнали подій та інші дані про діяльність користувачів для виявлення ненормальних та потенційно загрожуючих дій.

- Моделі поведінки можуть бути використані для оцінки ризиків, пов'язаних з конкретними користувачами або діями.

- Вони дозволяють системам кібербезпеки адаптувати рівень захисту в залежності від змінюючої поведінки користувачів.

- Моделі поведінки можуть навчатися на основі історичних даних та прогнозувати майбутню поведінку користувачів для забезпечення безпеки.

- Вони можуть виявляти спроби ідентифікаційних атак, коли зловмисники намагаються видачі себе за легітимних користувачів.

2. ПОСТАНОВКА ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

2.1 Вибір предметної області для дослідження

Для більш предметного розгляду застосування методологій Data Science у сфері кібербезпеки необхідно вибрати предметну область та поставити конкретну задачу, притаманну цій сфері. Було вирішено обрати сферу охорони здоров’я як предметну область для розгляду.

Найбільш цінна інформація у медичній сфері – це особисті данні пацієнтів. Ця інформація підпадає під визначення медичної таємниці, її оприлюднення може нанести шкоду репутації, особистому та професійному життю постраждалого, тож ця інформація є суттєвою ціллю для зловмисників та має бути захищена якомога краще.

2.2 Особливості кібербезпеки в медицині

Сектор охорони здоров’я дуже стурбований безпекою, особливо в Інтернеті, де кібератаки стають все більш поширеними та витонченими. Порушення контролю доступу, напади, які вводять і запускають зловмисне програмне забезпечення, а також атаки на відмову в обслуговуванні (DoS) є одними з найчастіших загроз безпеці охорони здоров’я. На відміну від DDoS-атак, які використовують численні хости для атаки на систему, DoS-атаки включають єдине джерело, яке переповнює цільову систему запитами. Це ускладнює визначення походження атаки. Пацієнти можуть постраждати внаслідок цих атак, а організації охорони здоров’я можуть постраждати від репутації. Ще однією серйозною загрозою для сектору охорони здоров’я є зловмисне програмне забезпечення, яке практично завжди з’являється в нових варіантах. Програми-вимагачі — це одне з сімейств шкідливих програм, про які

медичні установи починають турбуватися. Програми-вимагачі посіли друге місце в списку небезпек для кібербезпеки для медичних компаній в опитуванні Товариства інформації та систем управління охороною здоров'я (HIMSS). 17% респондентів повідомили, що були жертвами атак програм-вимагачів. За останні роки медичні підприємства стали об'єктом кількох резонансних кібератак. Наприклад, атака програми-вимагача, яка вразила Управління охорони здоров'я Ірландії (HSE) у 2021 році, серйозно порушила роботу медичних послуг. Подібно до цього, понад 150 країн постраждали від нападу програми-вимагача WannaCry у 2017 році, що змусило Національну службу охорони здоров'я Великобританії (NHS) перенести процедури та скасувати зустрічі. Заклади охорони здоров'я повинні запровадити надійні засоби кібербезпеки, щоб зупинити кібератаки та захистити конфіденційні дані пацієнтів. Проте багато закладів охорони здоров'я й надалі не мають адекватних протоколів безпеки, що робить їх відкритими для вторгнень. Згідно з дослідженням Ponemon Institute, лише 44% медичних підприємств мають ретельну політику безпеки. Базуючись на відповідях 167 спеціалістів з кібербезпеки охорони здоров'я, на рис. 1 показано рейтинг кібератак у 2021 році в опитуванні. Авторський метод передбачав проведення оцінки контенту наукових статей з 2014 по 2020 рік, які обговорювали зловмисне програмне забезпечення, DoS та атаки соціальної інженерії на лікарні. Наприклад, 20 березня 2014 року DDoS-атака була спрямована на бостонську лікарню, спричинивши збій мережі, який тривав два тижні, і негативно порушив роботу лікарні. У 2016 році програми-вимагачі, які використовували методи соціальної інженерії, вразили лікарню Lukas і Hollywood Presbyterian Medical, порушивши роботу систем і зробивши дані пацієнтів непридатними для використання. У 2020 році атаки програм-вимагачів торкнулися трьох лікарень: по одній у Чехії, США та Лондоні. Бостонська дитяча лікарня мала найдовший напад, який тривав 14 днів, а ChampaignUrbana Public Health District мав найкоротший, лише чотири дні. Таблиця показує, що наслідки є серйозними незалежно від

стратегій і тактик атак, які використовують кіберзлочинці. Таким чином, якби кібербезпеці було надано головний пріоритет у секторі охорони здоров'я, системні збої, шкода репутації та інші пов'язані з цим проблеми могли б зменшитися. Усі ці заклади, які були об'єктами різноманітних кібератак, у тому числі лікарні Бостона, лікарні Лукас, лікарні Брно та лікарні Хенкока, дотримувалися однакового курсу: вони вимкнули свої системи, щоб обмежити шкоду. Таблиця демонструє, що в лікарнях не було визначених стратегій чи резервних планів боротьби зі вторгненнями, що демонструє зневагу до кібербезпеки. Наприклад, лікарня Брно продовжувала використовувати Windows XP до 2020 року. Це підкреслює, наскільки важливо для медичних підприємств займатися кібербезпекою та впроваджувати профілактичні кроки, щоб зменшити та усунути онлайн-небезпеки. Інформація про викуп, сплачений лікарнями хакерам за відновлення доступу до їхніх систем, наведена в таблиці III. У порівнянні зі спробами відновити зламані системи інформаційних технологій без дешифрування ключ, необхідний для видалення інфекції, сплата викупу може бути менш шкідливою для операцій і прибутку. Бостонська лікарня витратила найбільше на відновлення своїх систем, близько 600 000 доларів, а Округ громадського здоров'я Шампейн-Урбана витратив друге місце, 350 000 доларів. Найменшу суму заплатила Hollywood Presbyterian Medical — 17 000 доларів. Хоча сплата викупу може призвести до фінансових витрат, це краще, ніж загрожувати життю, заплямувати імідж чи розкрити приватну інформацію. 1) Ін'єкційна атака: зловмисник може «впровадити» зловмисні дані у веб-програму, що вплине на її роботу, скеровуючи її на виконання певних команд. Ін'єкція є одним із ранніх різновидів веб-атак. Зловмисне програмне забезпечення є ілюстрацією ін'єкційної атаки. Відповідно до [6] шкідливе програмне забезпечення – це будь-який комп'ютерний код, написаний з метою отримання несанкціонованого доступу до цифрових пристроїв та IT-інфраструктури. Це робиться шляхом порушення заходів безпеки, які захищають їх, і використання недоліків безпеки. Було помітно три

окремі підтипи зловмисного програмного забезпечення: SamSam - зловмисне програмне забезпечення-вимагач, яке вперше з'явилося наприкінці 2015 року, націлене в основному на сектор охорони здоров'я. SamSam спеціалізується на використанні вразливостей RDP, FTP і веб-серверів на основі Java для доступу до машин жертв. Locky - це сімейство програм-вимагачів, яке використовує гібридну криптосистему та було запущено в 2016 році. Його механізм роботи передбачає сканування дисків жертви, таких як мережеві диски, на наявність певних типів файлів для їх шифрування за допомогою RSA та AES. Netwalker - також відомий як Mailto, це тип атаки, коли зловмисник використовує мережу жертви для шифрування всіх пристроїв на базі Windows. Для здійснення атаки зловмисник може використовувати або фішингові електронні листи, або виконувані файли, які переміщуються по мережах.

Статистику найбільш розповсюджених атака на медичні інформаційні системи зображено на рисунку 2.1 нижче.

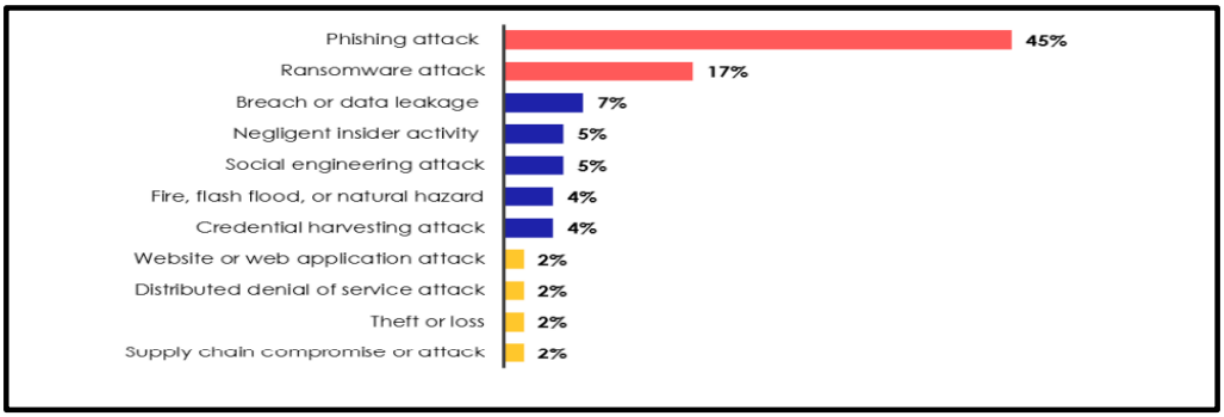


Рис. 2.1 – Найбільш розповсюджені атаки на медичні інформаційні системи

Соціальна інженерія - це метод, при якому зловмисник використовує міжособистісні взаємодії, щоб виловити психологічні недоліки жертви, щоб переконати її розкрити критичну інформацію зловмиснику. Фішинг — це тип соціальної інженерії, який використовують хакери, щоб обманом змусити своїх жертв розкрити конфіденційну інформацію, як-от імена користувачів, паролі,

реквізити банківських рахунків тощо. Це досягається шляхом обману, змушуючи користувача натиснути посилання на фальшивий веб-сайт або завантажити шкідливу програму.

Атака на відмову в обслуговуванні - це тип кібератаки, яка здебільшого зосереджена на споживанні ресурсів, включаючи пам’ять або обчислювальну потужність. Для здійснення цього нападу можна використовувати як бездротові, так і кабельні з’єднання. Особливий вид DoS-атаки, спрямованої на веб-сайти, відомий як розподілена атака на відмову в обслуговуванні. Щоб напасти на одну жертву, зловмисник використовує шкідливий сценарій, розміщений на кількох інших комп’ютерах. Веб-сайт планується припинити працювати. Ризик — це потенційна можливість втрати або шкоди, якщо зловмисник використає дірку в безпеці. Операційний ризик, пов’язаний з діяльністю в Інтернеті, яка загрожує інформаційним активам, ресурсам для інформаційно-комунікаційних технологій і технологічним активам і може завдати матеріальної шкоди матеріальним і нематеріальним активам організації, призупинити бізнес або завдати шкоди репутації, є іншим повним визначенням ризику кібербезпеки. Зменшення ризику та уникнення ризику є двома альтернативами, які пропонуються стратегіями зменшення ризику. Запобіжні заходи використовуються в стратегіях пом’якшення, щоб зменшити можливість або вплив кібератаки. Ці тактики зосереджені на виявленні та усуненні будь-яких слабких місць і ризиків безпеки в правилах та інформації організації. Заходи з пом’якшення ризиків можуть включати встановлення систем виявлення проникнення та захисних бар’єрів, а також часте оновлення програмного та апаратного забезпечення та навчання персоналу найкращим практикам кібербезпеки. A. Проактивне реагування на інцидент (IR) Планування та підготовка, виявлення, аналіз та оцінка, стримування та викорінення, відновлення та дії після аварії – це шість кроків, які складають цю процедуру. Фірма повинна спочатку встановити свою політику безпеки та можливість реагування на інциденти. Це передбачає створення команди для

управління інцидентами та придбання необхідних інструментів і витратних матеріалів. На другому етапі подія автоматично виявляється за допомогою таких інструментів, як мережеві або хост-системи виявлення вторгнень, або вручну за допомогою ручних вимог, як-от попередження користувачів про проблеми. На третьому етапі після інциденту група реагування на інцидент аналізує та перевіряє інцидент. Реалізація стратегій стримування, таких як пісочниця, відбувається на четвертому етапі. На п’ятому етапі адміністратор перевірить, чи системи працюють нормально, і виправить будь-які проблеми, щоб запобігти повторенню в майбутньому. Після інциденту слід провести зустріч як останній крок. Метою цієї зустрічі є розвиток технологій та отримання знань. В. Захищена архітектура на основі технології блокчейн і штучного інтелекту Пропонована архітектура безпечної системи на основі штучного інтелекту і технології блокчейн складається з п’яти рівнів. Перший рівень, який називають «рівнем даних», збирає інформацію від датчиків пацієнта, включаючи температуру та серцебиття. Крім того, на цьому рівні збираються зразки шкідливого програмного забезпечення та надсилаються на рівень аналізу шкідливого програмного забезпечення. Такі інструменти, як Pestudio та Process Explorer, використовуються в другий рівень, відомий як аналіз зловмисного програмного забезпечення, для перевірки зловмисного програмного забезпечення. Нешкідливі зразки другого рівня включаються в розвідку третього рівня, яка перевіряє їх на наявність недоліків у безпеці за допомогою методів штучного інтелекту, таких як машини підтримки векторів (SVM) і випадкові ліси (RF). Дані, передані з рівня 3, безпечно зберігаються на рівні 4, рівні Blockchain. Лікарні, аптеки, лабораторії та машини швидкої допомоги є прикладами одержувачів медичних даних на прикладному рівні (третій рівень).

2.3 Формулювання проектного завдання

Підсумовуючи вищесказане, можна помітити, що найбільш розповсюдженими видами атак на медичні інформаційні системи є фішинг та атака програм-вимагачів.

Тож в якості проектного завдання було вирішено обрати моделювання атаки на медичну інформаційну систему із застосуванням фішингу, застосування класичних методів захисту та визначення їхньої ефективності, застосування методів Data Science для атаки та захисту інформації та визначити ефективність їхнього впливу. Визначити способи практичної імплементації методів Data Science в систему кібербезпеки.

3. РЕАЛІЗАЦІЯ ПРОЕКТНОГО ЗАВДАННЯ

3.1 Моделювання атаки на медичну інформаційну систему

Після аналізу особливостей кібербезпеки в медицині та визначення основних напрямків хакерських атак, було прийнято рішення змоделювати атаку зловмисників на медичну інформаційну систему, застосувати методи Data science задля покращення методів захисту та проаналізувати їхній вплив на якість захисту.

Моделювання атаки буде проведено за допомогою симуляції у вигляді типового для кібербезпеки змагання «Синя команда проти червоної».

У моделюванні кібербезпеки червоної команди/синьої команди червона команда діє як супротивник, намагаючись виявити та використати потенційні слабкі місця в системі кіберзахисту організації за допомогою складних методів атак. Ці наступальні команди зазвичай складаються з досвідчених професіоналів із безпеки або незалежних етичних хакерів, які зосереджуються на тестуванні на проникнення, імітуючи техніки та методи атак у реальному світі.

За умовами експерименту, червона команда отримує доступ до інформаційної системи через крадіжку облікових даних користувачів за допомогою фішингу. Потрапивши в мережу, червона команда підвищує свої привілеї та переміщається по системах з метою просування якомога глибше в мережу, викрадаючи дані, уникаючи виявлення. у цьому проекті червона команда намагається атакувати систему, маскуючись під активність звичайного користувача системи.

Синя команда в свою чергу виступає на захисті інформаційної системи. ЇЇ завданням є виявлення зловмисників шляхом аналізу активності облікових записів користувачів та виявлення в них нетипової активності для того, щоб

заблокувати зловмисників, які користуються обліковими записами користувачів.

Метою цього проекту є створення інтелектуальних підроблених завдань GAN. Водночас вам потрібно оцінити точність виявлення оригінальних підроблених завдань і створених GAN підроблених завдань. Механізми виявлення містять класичні моделі машинного навчання (ML) (наприклад, Random Forest (RF) і Adaboost) і структуру каскадного виявлення на основі GAN. Ще одна мета — порівняти ефективність виявлення традиційних ML і системи каскадного виявлення.

3.2 Перелік технологічного стеку

Виходячи з усього вище сказаного, для розробки проекту було вирішено застосовувати мову програмування Python 3 через наявність відповідного інструментарію, що відповідає всім потребам та знаходиться в активній розробці, та середовище розробки jupyter.

Для розробки проекту знадобляться наступні бібліотеки:

Tensorflow – бібліотека для розробки моделей ШІ із технологією Deep Learning;

Keras – фреймворк бібліотеки TensorFlow для спрощення обробки типових задач;

Scikit-learn – бібліотека для створення моделей машинного навчання, але у проекті вона використовується через вбудований інструментарій для оцінки роботи моделі;

OpenCV – бібліотека комп’ютерного зору для завантаження та попередньої обробки зображень з датасету;

NumPy – бібліотека математичних обчислень, використовується для обробки масивів;

Matplotlib – бібліотека візуалізації даних, використовується для створення графіків;

Також у симуляції застосовані такі алгоритми, як Random Forest Classifier та AdaBoost у якості традиційних методів виявлення атак на систему. Та також застосована модель машинного навчання GAN задля порівняння ефективності застосування її для відбиття атак на MIC у порівнянні з більш класичними методами. Ці алгоритми детально розглянуто нижче.

3.2.1 Random Forest Classifier

Random Forest (Випадковий ліс) - це алгоритм машинного навчання, який відноситься до класу ансамблевих методів. Він використовується для завдань класифікації та регресії. Random Forest є потужним і дуже популярним алгоритмом завдяки своїй ефективності і здатності працювати з різноманітними типами даних.

Основна ідея за алгоритмом Random Forest полягає в тому, щоб створити багато дерев рішень і об'єднати їх в ансамбль. Кожне дерево рішень тренується на випадковому підмножині навчальних даних, і для кожного вузла в дереві обирається випадковий піднабір ознак. Це допомагає забезпечити різноманітність та уникнути перенавчання.

Основні переваги Random Forest включають:

- Стійкість до перенавчання: Через використання випадкового вибору даних та ознак, Random Forest має схильність до меншого перенавчання порівняно з окремими деревами рішень.
- Висока точність: Random Forest часто показує високу точність класифікації та регресії через агрегацію рішень багатьох дерев.
- Можливість обробки великої кількості ознак: Алгоритм ефективно працює з великою кількістю ознак і великими наборами даних.

3.2.2 ADA Boost Classifier

AdaBoost (Adaptive Boosting) - це інший популярний алгоритм ансамблювання, що використовується для класифікації та регресії. Алгоритм створений з метою покращення точності базових моделей (наприклад, слабких класифікаторів), шляхом надання більшого вагомості невірно класифікованим екземплярам даних.

Основні принципи AdaBoost:

- Слабкі класифікатори: AdaBoost використовує слабкі класифікатори, які мають низьку точність порівняно з випадковим вгадуванням. Такі класифікатори можуть бути, наприклад, короткими деревами рішень або одним рішенням.
- Ваги для невірно класифікованих прикладів: Під час тренування, AdaBoost надає більш велику вагу прикладам, які були невірно класифіковані попередніми класифікаторами. Це дозволяє алгоритму фокусуватися на тих прикладах, які важче класифікувати.
- Ансамблювання: AdaBoost об'єднує результати багатьох слабких класифікаторів, присвоюючи їм ваги в залежності від їхньої точності. Основна ідея - об'єднання кількох слабких класифікаторів може створити сильний класифікатор.

Основні переваги AdaBoost:

- Висока точність: AdaBoost може досягати високої точності, особливо коли використовуються слабкі класифікатори, такі як короткі дерева рішень.
- Відсутність перенавчання: Завдяки підходу з вагами для невірно класифікованих екземплярів, AdaBoost має властивість не схильний до перенавчання.

- Здатність працювати з різними класифікаторами: AdaBoost може бути використаний з будь-яким базовим класифікатором, що робить його універсальним.

3.2.3 Генеративно-змагальні мережі (GAN)

Generative Adversarial Networks (GAN) [9] — це структура глибокого навчання, в якій дві моделі, генеративна модель і дискримінаційна модель, навчаються одночасно. Мета генеративної моделі — охопити розподіл деяких цільових даних (наприклад, розподіл інтенсивності пікселів на зображеннях). Дискримінаційна допомагає навчанню генеративної, перевіряючи дані, створені G, з посиланням на «реальні» дані, і таким чином допомагає G вивчити розподіл, який лежить в основі реальних даних. GAN конкретизовано в Goodfellow et al. (2014) як пару простих нейронних мереж. Однак на практиці моделями в принципі можуть бути будь-які генеративно-дискримінаційні пари2 В оригінальній роботі та в інших місцях GAN була аналогічна процесу виготовлення фальшивих грошей: G відіграє роль фальшивомонетник проходить підготовку, тоді як D банк прагне виявляти фальшиві купюри та в процесі (ненавмисно) допомагає G вдосконалювати свої навички виготовлення купюр. Більш конкретно, нехай x ~ pdata є характеристиками реальних купюр, а G(z) є ознаками, які G створює з деякого розподілу шуму z ~ pz. Крім того, нехай J — деяка кількісна метрика, яка вимірює ступінь реальності купюри. Потім робота D полягає в тому, щоб знизити J(G(z)) (оцінка фальшивої купюри) і збільшити оцінку J(x) (оцінка справжньої купюри) для більш успішної ідентифікації. G, з іншого боку, має на меті збільшити J(G(z)) (тобто покращити якість фальшивої купюри), вивчаючи «спостереження» за тим, як D робить диференціації. По мірі того, як триває гра у «викриття фальшивих купюр» і «вироблення фальшивих купюр», розподіл моделі pG наближається до pdata і врешті-решт досягає рівноваги, де D більше не може класифікуватися краще,

ніж випадковість (тобто D(x) = D(G(z)) = 12). Тепер ми кажемо, що G досяг оптимальної точки 3 підробки. GAN набув великої популярності в комп'ютерному баченні, представленні функцій, а нещодавно в задачах обробки природної мови (NLP): моделювання документів, генерація діалогів, аналіз настроїв та адаптація домену.

Структуру GAN зображено на рисунку 3.1 нижче.

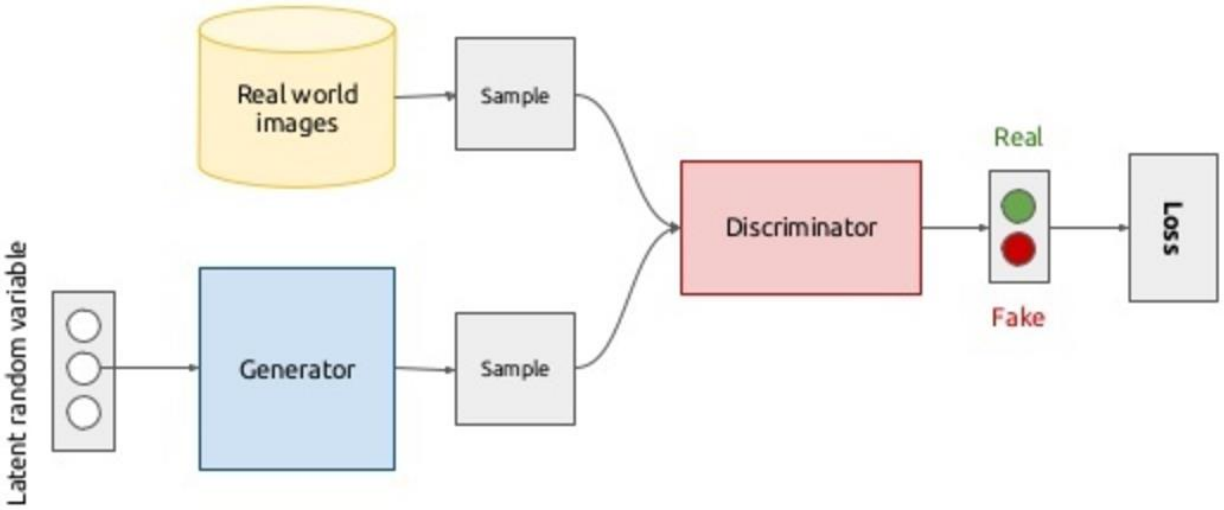


Рис. 3.1 – Структура GAN

3.2 Генерація даних запитів за допомогою CrowdSenSim

Задля генерації списку запитів користувачів до MIC скористаємося open source системою CrowdSenSim.

CrowdSenSim — це симулятор дискретних подій, що підтримує мобільність пішоходів, здатний імітувати генерацію та збір даних у сценаріях мобільного краудсенсінгу [13].

Основні особливості CrowdSenSim такі:

- Масштабованість: CrowdSenSim розроблений для розміщення великої кількості учасників, порядку десятків тисяч користувачів, які переміщуються у широкому географічному просторі.

- Реалістичне міське середовище: CrowdSenSim може використовувати реалістичне міське середовище.

- Мобільність користувача: CrowdSenSim використовує єдиний алгоритм мобільності. Кожен учасник розподіляється в певній точці міста та йде пішки протягом періоду часу, рівномірно розподіленого між 10 - 20 хвилинами, із середньою швидкістю, рівномірно розподіленою між 1 м с−1 і 1,5 м с−1. Учасники вносять дані під час ходьби. Коли період ходьби закінчується, вони перестають рухатися і брати участь у своїх пристроях.

- Комунікаційні технології: пристрої підтримують технології WiFi і стільникового зв'язку. Кожен з них по-різному впливає на енергоспоживання пристроїв користувача.

CrowdSenSim написано мовою C++ і випущено за загальною публічною ліцензією.

Задля генерації даних використаємо готову зборку додатку на Ubuntu, який розміщуємо на віртуальній машині Oracle VirtualBox.

Текстовий інтерфейс додатку у віртуальній машині зображено на рисунку 3.2 нижче.

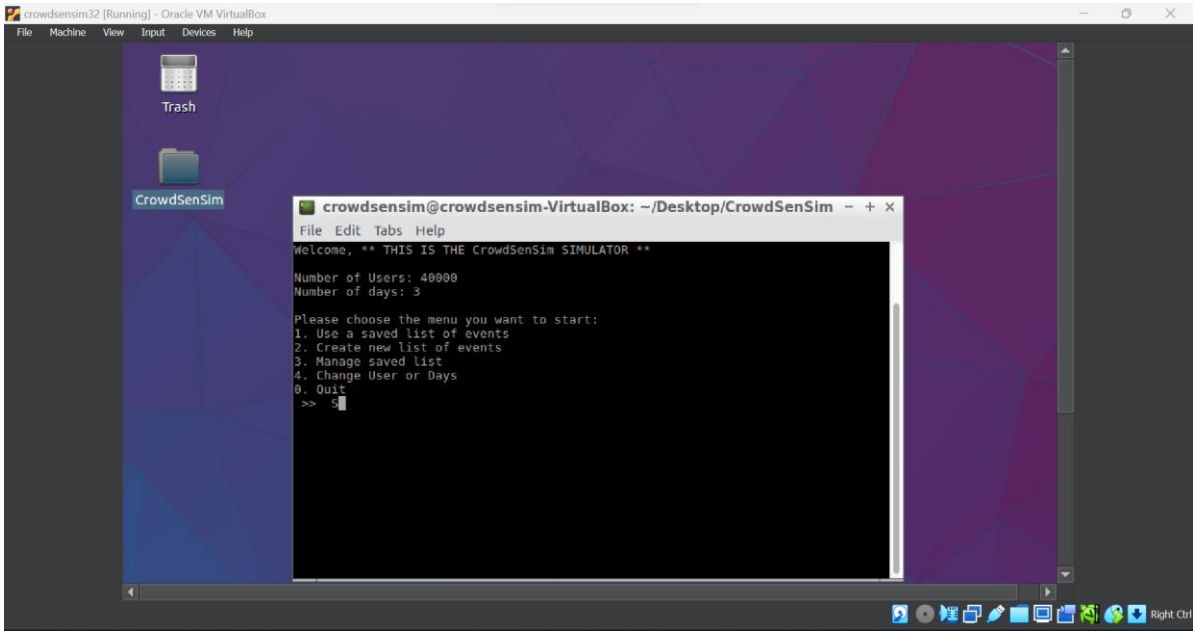


Рис. 3.2 – Текстовий інтерфейс CrowdSenSim

Додаток пропонує гнучкі налаштування симуляції – можна вказати кількість днів симуляції, кількість активних користувачів та вказати чи потрібно використовувати існуючий список подій, або створити власний. Додаток має згенеровані події для міст Барселони та Люксембургу. За бажанням користувач може вказати майже будь-яке велике місто для генерації за допомогою бібліотеки GeoPy.

Скористаємось існуючим списком подій програми та оберемо бажаним містом Люксембург. Результати роботи програми можна побачити на рисунку 3.3 нижче.

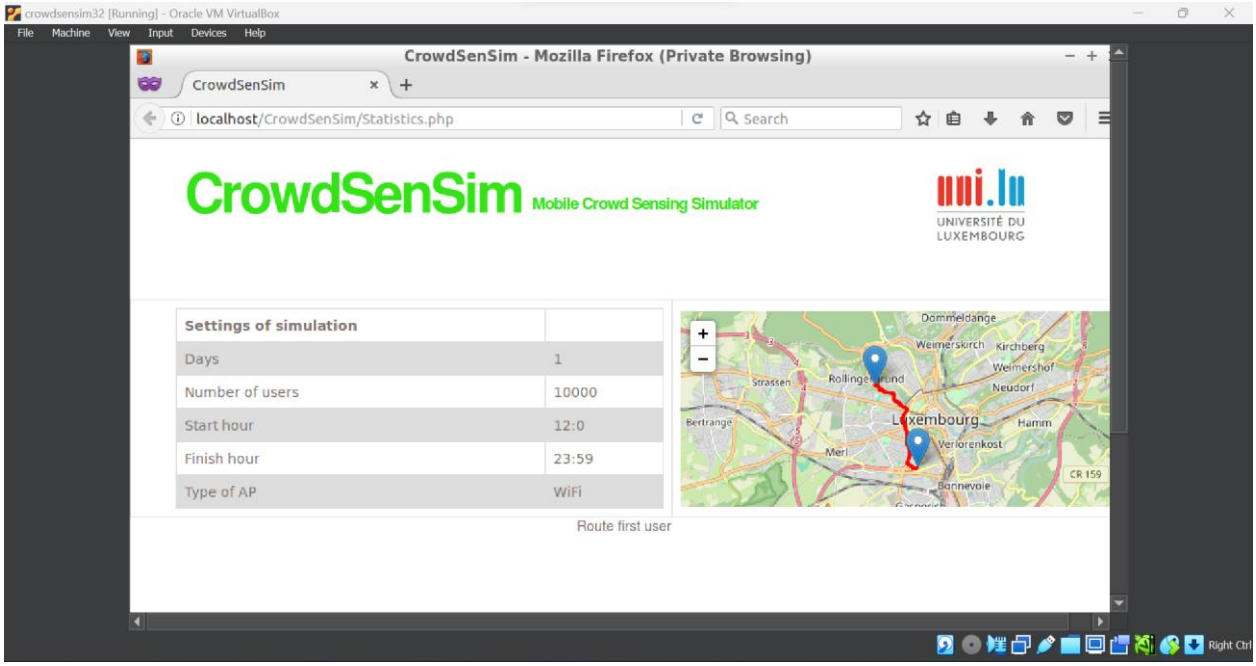
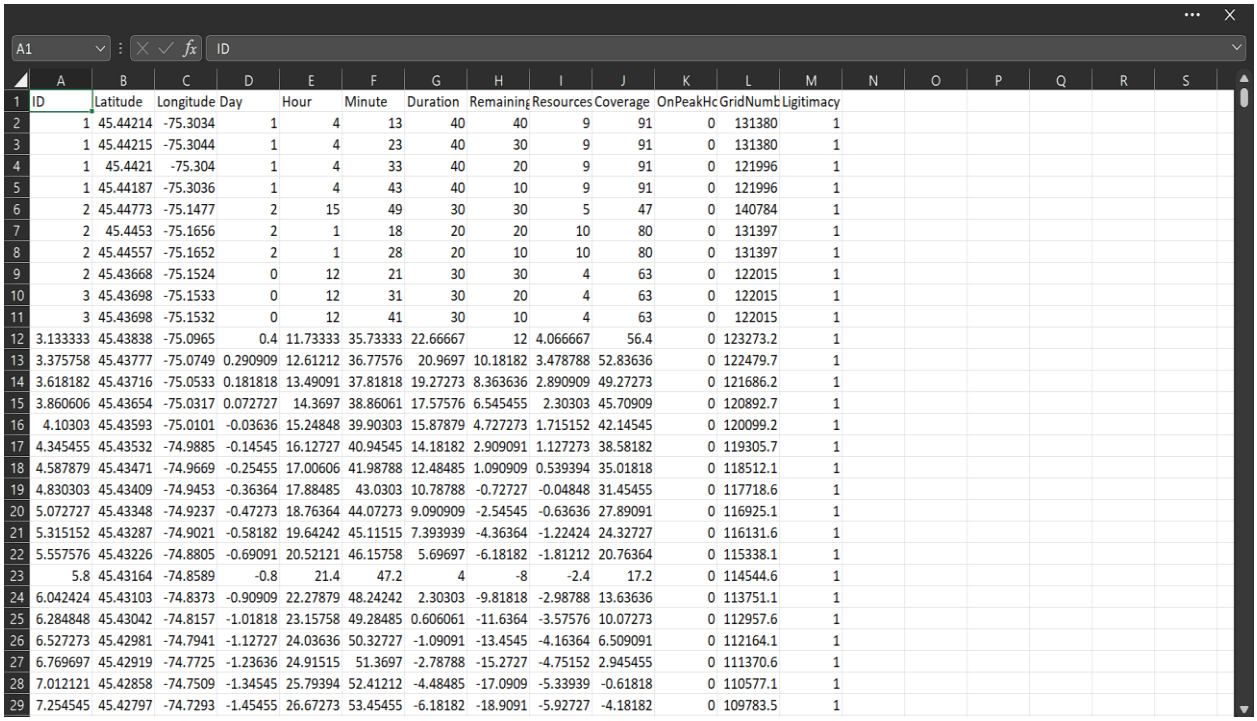


Рис. 3.3 – Результати роботи програми

Додаток надає дані про проведену симуляцію та навіть розміщує дані про сигнали на мапі. Також після симуляції всі дані про події зберігаються в csv файл.

Дані з файлу зображено на рисунку 3.4 нижче.



ID	Latitude	Longitude	Day	Hour	Minute	Duration	Remaining	Resources	Coverage	OnPeakHc	GridNumb	Ligitimacy
1	45.44214	-75.3034	1	4	13	40	40	9	91	0	131380	1
1	45.44215	-75.3044	1	4	23	40	30	9	91	0	131380	1
1	45.4421	-75.304	1	4	33	40	20	9	91	0	121996	1
1	45.44187	-75.3036	1	4	43	40	10	9	91	0	121996	1
2	45.44773	-75.1477	2	15	49	30	30	5	47	0	140784	1
2	45.4453	-75.1656	2	1	18	20	20	10	80	0	131397	1
2	45.44557	-75.1652	2	1	28	20	10	10	80	0	131397	1
2	45.43668	-75.1524	0	12	21	30	30	4	63	0	122015	1
3	45.43698	-75.1533	0	12	31	30	20	4	63	0	122015	1
3	45.43698	-75.1532	0	12	41	30	10	4	63	0	122015	1
3.133333	45.43838	-75.0965	0.4	11.73333	35.73333	22.66667	12	4.066667	56.4	0	123273.2	1
3.375758	45.43777	-75.0749	0.290909	12.61212	36.77576	20.9697	10.18182	3.478788	52.83636	0	122479.7	1
3.618182	45.43716	-75.0533	0.181818	13.49091	37.81818	19.27273	8.363636	2.890909	49.27273	0	121686.2	1
3.860606	45.43654	-75.0317	0.072727	14.3697	38.86061	17.57576	6.545455	2.30303	45.70909	0	120892.7	1
4.10303	45.43593	-75.0101	-0.03636	15.24848	39.90303	15.87879	4.727273	1.715152	42.14545	0	120099.2	1
4.345455	45.43532	-74.9885	-0.14545	16.12727	40.94545	14.18182	2.909091	1.127273	38.58182	0	119305.7	1
4.587879	45.43471	-74.9669	-0.25455	17.00606	41.98788	12.48485	1.090909	0.539394	35.01818	0	118512.1	1
4.830303	45.43409	-74.9453	-0.36364	17.88485	43.0303	10.78788	-0.72727	-0.04848	31.45455	0	117718.6	1
5.072727	45.43348	-74.9237	-0.47273	18.76364	44.07273	9.090909	-2.54545	-0.63636	27.89091	0	116925.1	1
5.315152	45.43287	-74.9021	-0.58182	19.64242	45.11515	7.393939	-4.36364	-1.22424	24.32727	0	116131.6	1
5.557576	45.43226	-74.8805	-0.69091	20.52121	46.15758	5.69697	-6.18182	-1.81212	20.76364	0	115338.1	1
5.8	45.43164	-74.8589	-0.8	21.4	47.2	4	-8	-2.4	17.2	0	114544.6	1
6.042424	45.43103	-74.8373	-0.90909	22.27879	48.24242	2.30303	-9.81818	-2.98788	13.63636	0	113751.1	1
6.284848	45.43042	-74.8157	-1.01818	23.15758	49.28485	0.606061	-11.6364	-3.57576	10.07273	0	112957.6	1
6.527273	45.42981	-74.7941	-1.12727	24.03636	50.32727	-1.09091	-13.4545	-4.16364	6.509091	0	112164.1	1
6.769697	45.42919	-74.7725	-1.23636	24.91515	51.3697	-2.78788	-15.2727	-4.75152	2.945455	0	111370.6	1
7.012121	45.42858	-74.7509	-1.34545	25.79394	52.41212	-4.48485	-17.0909	-5.33939	-0.61818	0	110577.1	1
7.254545	45.42797	-74.7293	-1.45455	26.67273	53.45455	-6.18182	-18.9091	-5.92727	-4.18182	0	109783.5	1

Рис. 3.4 - Датасет

Набір даних генерується інструментом моделювання CrowdSenSim. Набір даних містить законні завдання та підроблені завдання. Атрибути завдань такі: 'ID', 'latitude', 'longitude', 'day', 'hour', 'minute', 'duration', 'remaining time', 'battery requirements %', 'Coverage', 'GridNumber', 'OnpeakHour'.

Розташування завдань вказується разом «широтою» та «довготою». Крім того, «день», «година» та «хвилина» описують час публікації завдання. «Тривалість» позначає тривалість активного завдання в хвилинах. «Час, що залишився» означає час, що залишився від завдання зондування до його завершення. «Потреба в заряді батареї» — це відсоток заряду батареї, який потрібен для виконання завдання. «Покриття» означає відстань визначення завдання. «GridNumber» отримується шляхом поділу зондової карти міста на невеликі сітки з числами, що починаються з 1. «OnpeakHour» — це двійковий прапорець, який вказує, чи час початку завдання відбувається з 7:00 до 11:00. Ми визначаємо з 7:00 до 11:00 годину пік, а інші години не є піковими для спрощення моделювання. Також задля впровадження можливості навчання було додано колонку “legitimacy”, яка позначає легітимність запиту. Значення 1

означає, що запит був від справжнього користувача, а 0 в свою чергу – що запит був від хакера, який маскується під користувача. Набір даних включає 14 484 запити, з яких 14 075 легітимних та 409 фальшивих відповідно.

3.3 Імплементація проекту

Цей проект зосереджений на каскадній структурі на основі GAN. Пропонується каскадна структура, яка реалізує дворівневу каскадну архітектуру класифікатора для прогнозування згенерованих зразків атак і оригінальних (емпірично розроблених) зразків атак і фільтрації їх перед розповсюдженням їх учасникам MCS.

Перший рівень формується дискримінатором GAN, тоді як другий рівень є двійковим класифікатором. Навчальний набір даних використовується для навчання бінарних класифікаторів (наприклад, RF, Adaboost) і GAN. Навчальний набір містить оригінальні фейкові завдання. GAN беруть вхідні зразки шуму та виводять змагальні (тобто синтетичні) зразки. GAN складається з нейронних мереж генератора і дискримінатора. Роль генератора полягає у створенні синтетичних зразків, подібних до реальних, тоді як дискримінатор розрізняє справжні зразки від синтетичних. Мотив змагання полягає в тому, щоб зменшити розрив між згенерованими зразками для генератору і підвищити точність виявлення для дискримінатору.

Приступаємо до написання програмного коду.

Код програми:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay,
accuracy_score
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
import seaborn as sns
sns.reset_defaults()
from keras import backend as K
from keras.layers import Input, Dense, Reshape, Flatten, Concatenate
from keras.layers import BatchNormalization, Activation, Embedding, multiply
from keras.layers.advanced_activations import LeakyReLU
from keras.layers.convolutional import Conv2D,Conv1D, Conv2DTranspose
from tensorflow.keras.optimizers import Adam
from keras.models import Model, Sequential
from tensorflow.keras.utils import to_categorical
from keras.utils.vis_utils import plot_model
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
import tensorflow as tf
from sklearn.preprocessing import StandardScaler
from numpy.random import seed
seed(42)
tf.random.set_seed(42)
df=pd.read_csv('../input/mcsdatasetnextconlab/MCSDatasetNEXTCONLab.csv')
df
df=df.drop('ID',axis=1)
df['Ligitimacy'].value_counts()
X = df.iloc[:,:-1]
```

```
y = df.iloc[:,-1]
scaler = StandardScaler()
X=scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42, stratify=y)
print(X_train.shape)
print(X_test.shape)
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
adaboost = AdaBoostClassifier(n_estimators=100, random_state=0)
adaboost.fit(X_train, y_train)
def test_and_compare(models, x_test, y_test, labels1=['Fake', 'Legitimate'],
title='campare between models'):
    models_name = []
    Accuracies = []
    for model in models:
        print("Evaluated {} model".format(model))
        models_name.append(str(model))
        y_pred = model.predict(x_test)
        print(classification_report(y_test, y_pred))
        cm = confusion_matrix(y_test, y_pred)
        disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels1)
        disp.plot()
        plt.show()
        Accuracies.append(accuracy_score(y_test, y_pred) * 100)
    plt.figure(figsize=(20, 10))
    ax = sns.barplot(x=models_name, y=Accuracies)
    ax.set_title(title, size=26)
    counter = 0
```

```
    for value in Accuracies:
        v = str(np.round(value, 2)) + ' %'
        ax.text(counter, value, v, color='black', ha="center")
        counter += 1
    mi = min(Accuracies)
    ma = max(Accuracies)
    range = ma - mi
    plt.ylim(mi - (range), ma + range)
    plt.show()
test_and_compare([rf,adaboost],X_test,y_test)
n_classes=2
batch_size = 64
num_channels = 1
row_size = 11
latent_dim = 128
dis_input=num_channels+n_classes
generator_in_channels=latent_dim+n_classes
#Створення дискримінатора
discriminator = keras.Sequential(
    [
        keras.layers.InputLayer((11,dis_input)),
        keras.layers.Flatten(),
        Dense(64),
        layers.LeakyReLU(alpha=0.2),
        Dense(128),
        layers.LeakyReLU(alpha=0.2),
        Dense(256),
        layers.LeakyReLU(alpha=0.2),
```

```
        layers.Dropout(0.2),
        Dense(1, activation='sigmoid')
    ],
    name="discriminator",
)
# Створення генератора
generator = keras.Sequential(
    [
        keras.layers.InputLayer((generator_in_channels)),
        layers.Dense(11 * generator_in_channels),
        layers.LeakyReLU(alpha=0.2),
        layers.Reshape((11, generator_in_channels)),
        layers.Dense(128),
        layers.LeakyReLU(alpha=0.2),
        layers.BatchNormalization(),
        layers.Dense(128),
        layers.LeakyReLU(alpha=0.2),
        layers.BatchNormalization(),
        layers.Dense(256),
        layers.LeakyReLU(alpha=0.2),
        layers.BatchNormalization(),
        layers.Dropout(0.3),
        layers.Dense(1, activation="sigmoid"),
        layers.Reshape((11, 1)),
    ],
    name="generator",
)
plot_model(discriminator, to_file='model_plot.png', show_shapes=True,
show_layer_names=True)
```

```
plot_model(generator, to_file='model_plot.png', show_shapes=True,
show_layer_names=True)
class ConditionalGAN(keras.Model):
    def __init__(self, discriminator, generator, latent_dim):
        super(ConditionalGAN, self).__init__()
        self.discriminator = discriminator
        self.generator = generator
        self.latent_dim = latent_dim
        self.gen_loss_tracker = keras.metrics.Mean(name="generator_loss")
        self.disc_loss_tracker = keras.metrics.Mean(name="discriminator_loss")
    @property
    def metrics(self):
        return [self.gen_loss_tracker, self.disc_loss_tracker]
    def compile(self, d_optimizer, g_optimizer, loss_fn):
        super(ConditionalGAN, self).compile()
        self.d_optimizer = d_optimizer
        self.g_optimizer = g_optimizer
        self.loss_fn = loss_fn
    def train_step(self, data):
        real_data, one_hot_labels = data
        row_one_hot_labels = one_hot_labels[:, None, None]
        row_one_hot_labels = tf.repeat(
            row_one_hot_labels, repeats=[row_size]
        )
        row_one_hot_labels = tf.reshape(
            row_one_hot_labels, (-1, row_size, n_classes)
        )
        #    nx11x2
        batch_size = tf.shape(real_data)[0]
```

```
#    n
random_latent_vectors = tf.random.normal(shape=(batch_size, self.latent_dim))
#    nx128
random_vector_labels = tf.concat(
    [random_latent_vectors, one_hot_labels], axis=1
)
#  nx128 + nx2= nx130
generated_row = self.generator(random_vector_labels)
#   nx11X1
fake_and_labels = tf.concat([generated_row, row_one_hot_labels], -1)
# nx11x1 +nx11x2 =nx11x3
real_and_labels = tf.concat([real_data, row_one_hot_labels], -1)
# nx11x1 +nx11x2 =nx11x3
combined_data = tf.concat(
    [fake_and_labels, real_and_labels], axis=0
)
# nx11x3
labels = tf.concat(
    [tf.zeros((batch_size, 1)), tf.ones((batch_size, 1))], axis=0
)
# 2nx2
# 0 for generator and 1 for real
with tf.GradientTape() as tape:
    predictions = self.discriminator(combined_data)
    d_loss = self.loss_fn(labels, predictions)
# Навчання дискримінатора
grads = tape.gradient(d_loss, self.discriminator.trainable_weights)
self.d_optimizer.apply_gradients(
```

```
            zip(grads, self.discriminator.trainable_weights)
        )
        random_latent_vectors = tf.random.normal(shape=(batch_size, self.latent_dim))
        random_vector_labels = tf.concat(
            [random_latent_vectors, one_hot_labels], axis=1
        )
        # nx130
        misleading_labels = tf.ones((batch_size, 1))
        # nx2 all ones
        with tf.GradientTape() as tape:
            fake_data = self.generator(random_vector_labels)
            fake_and_labels = tf.concat([fake_data, row_one_hot_labels], -1)
            predictions = self.discriminator(fake_and_labels)
            g_loss = self.loss_fn(misleading_labels, predictions)
        grads = tape.gradient(g_loss, self.generator.trainable_weights)
        self.g_optimizer.apply_gradients(zip(grads, self.generator.trainable_weights))
        self.gen_loss_tracker.update_state(g_loss)
        self.disc_loss_tracker.update_state(d_loss)
        return {
            "g_loss": self.gen_loss_tracker.result(),
            "d_loss": self.disc_loss_tracker.result(),
        }
# Зміна значень пікслелів в [0, 1] діапазон
all_digits = np.reshape(X_train, (-1, 11)).astype('float32')
all_digits=np.array(all_digits).reshape(-1,11,1)
all_labels = keras.utils.to_categorical(y_train, 2))
# Створення датасету tf.data.Dataset.
dataset = tf.data.Dataset.from_tensor_slices((all_digits, all_labels))
dataset = dataset.shuffle(buffer_size=1024).batch(batch_size)
```

```
print(f"Shape of training images: {all_digits.shape}")
print(f"Shape of training labels: {all_labels.shape}")
cond_gan = ConditionalGAN(
    discriminator=discriminator, generator=generator, latent_dim=latent_dim
)
cond_gan.compile(
    d_optimizer=keras.optimizers.Adam(learning_rate=0.01),
    g_optimizer=keras.optimizers.Adam(learning_rate=0.01),
    loss_fn=keras.losses.BinaryCrossentropy(from_logits=True),
)
cond_gan.fit(dataset, epochs=10)
trained_gen = cond_gan.generator
trained_dic=cond_gan.discriminator
num_interpolation =   X_test.shape[0]
# Генерація шуму для інтерполяції
interpolation_noise = tf.random.normal(shape=(1, latent_dim))
interpolation_noise = tf.repeat(interpolation_noise, repeats=num_interpolation)
interpolation_noise = tf.reshape(interpolation_noise, (num_interpolation, latent_dim))
interpolation_labels = keras.utils.to_categorical([1]*num_interpolation, 2)
interpolation_labels = tf.reshape(interpolation_labels, (num_interpolation, 2))
noise_and_labels = tf.concat([interpolation_noise, interpolation_labels], axis=1)
fake = trained_gen.predict(noise_and_labels)
mixed_data=np.concatenate((np.array(X_test), fake.reshape(-1,11)), axis=0)
mixed_y=np.concatenate((np.array(y_test), [0]*y_test.shape[0]), axis=0)
mixed_data.shape
mixed_y.shape
test_and_compare([rf,adaboost],mixed_data,mixed_y)
```

```
y_true_pred=np.concatenate([[1]*y.shape[0],[0]*fake.shape[0]])
y_true_pred.shape
rows = np.reshape(X, (-1, 11)).astype('float32')
rows = np.array(rows).reshape(-1,11,1)
all_labels = keras.utils.to_categorical(y, 2)
one_hot_labels = all_labels[:, None, None]
one_hot_labels = tf.repeat(
    one_hot_labels, repeats=[row_size]
)
one_hot_labels = tf.reshape(
    one_hot_labels, (-1, row_size, n_classes)
)
# N X 11 X n
data_and_labels = tf.concat([rows, one_hot_labels], -1)
all_labels = keras.utils.to_categorical([0]*fake.shape[0], 2)
one_hot_labels = all_labels[:, None, None]
one_hot_labels = tf.repeat(
    one_hot_labels, repeats=[row_size]
)
one_hot_labels = tf.reshape(
    one_hot_labels, (-1, row_size, n_classes)
)
fake_and_labels = tf.concat([fake,one_hot_labels], -1)
# nx11x1 +nx11x2 =nx11x3
combined_data = tf.concat(
    [data_and_labels,fake_and_labels], axis=0
)
```

```
fake_test=trained_dic.predict(combined_data)
y_pred=np.array([ 1 if x>0.5 else 0 for x in fake_test])
print(classification_report(y_true_pred, y_pred))
cm = confusion_matrix(y_true_pred, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=['Generator','Real'])
disp.plot()
plt.show()
def cascade_detection_framework(models,trained_dic,X,Y):
  rows = np.reshape(X, (-1, 11)).astype('float32')
  rows = np.array(rows).reshape(-1,11,1)
  all_labels = keras.utils.to_categorical(Y, 2)
  one_hot_labels = all_labels[:, None, None]
  one_hot_labels = tf.repeat(
     one_hot_labels, repeats=[row_size]
  )
  one_hot_labels = tf.reshape(
     one_hot_labels, (-1, row_size, n_classes)
  )
  # N X 11 X 2
  data_and_labels = tf.concat([rows, one_hot_labels], -1)
  fake_test=trained_dic.predict(data_and_labels)
  y_pred=np.argmax(fake_test,axis=1)
  y_pred=np.array([ 1 if x>0.5 else 0 for x in fake_test])
  x_real=X[y_pred==1]
  y_real=Y[y_pred==1]
  test_and_compare(models,x_real,y_real)
```

```
trained_dic=cond_gan.discriminator
cascade_detection_framework([rf,adaboost],trained_dic,mixed_data,mixed_y)
```

Симуляція буде проходити в 3 етапи: на першому обидві команди будуть використовувати класичні методи атаки та захисту, на другому червона команда буде використовувати GAN алгоритм для нападу, синя команда ж буде відбиватися звичайними алгоритмами RF та AdaBoost, на третьому етапі вже синя команда буде використовувати GAN алгоритм для покращення захисту. Після кожного етапу буде оцінено ефективність моделей та результати їхніх передбачень буде проілюстровано за допомогою бібліотек matplotlib та seaborn.

Розглянемо схему першого етапу.

Блок-схему першого етапу зображено на рисунку 3.5 нижче.

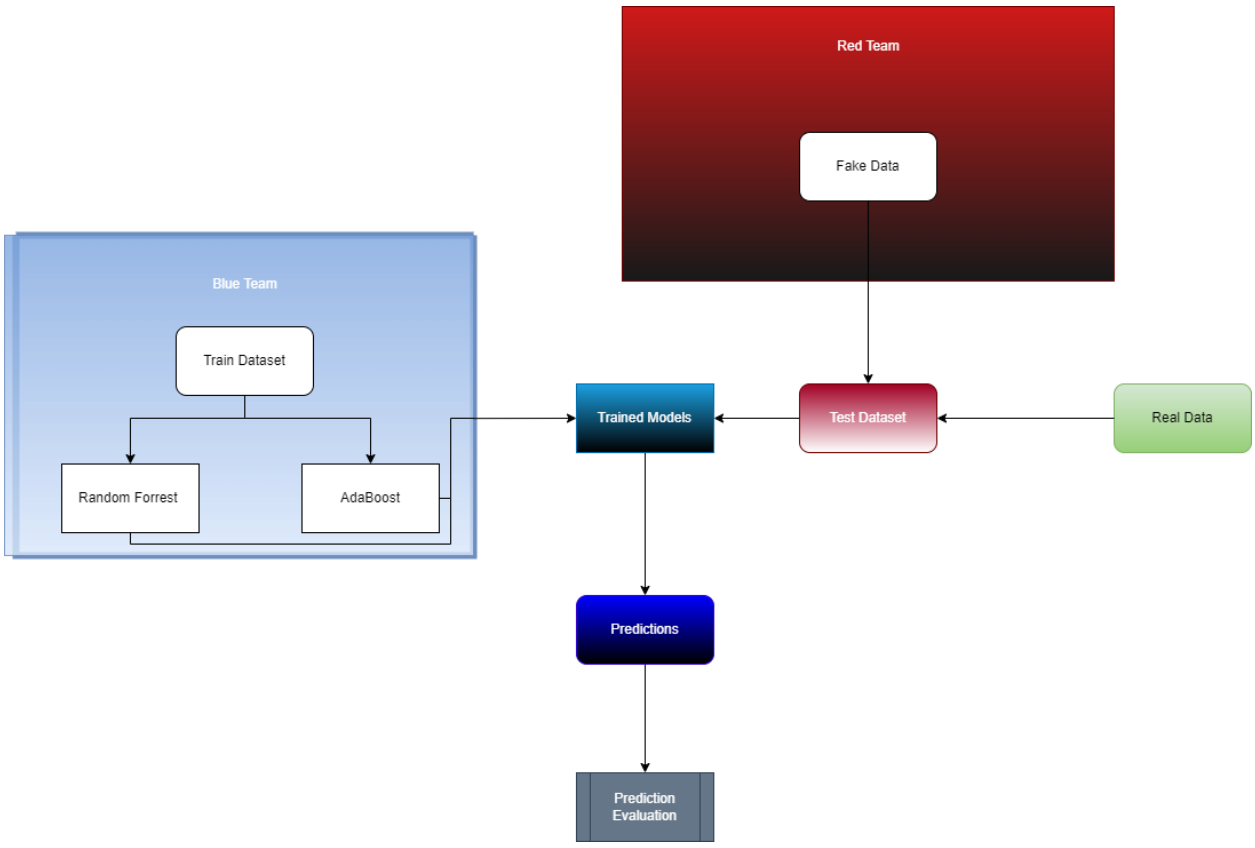


Рис. 3.5 – Схема першого етапу

Червона команда за попередніми умовами має дані від облікових записів користувачів, отримані за допомогою методів соціальної інженерії. Задача червоної команди – залогінитися в систему та під виглядом звичайного користувача уникнути викриття. На даному етапі червона команда користується тільки класичними методами втручання.

Синя команда, аналізуючи логи користувачів має виявити нелегітимні запити до системи. Для цього синя команда використовує алгоритми RF та AdaBoost. Спочатку команда навчає ці алгоритми на тестовому блоці даних. Результати навчання можна побачити на рисунку 3.6 нижче.

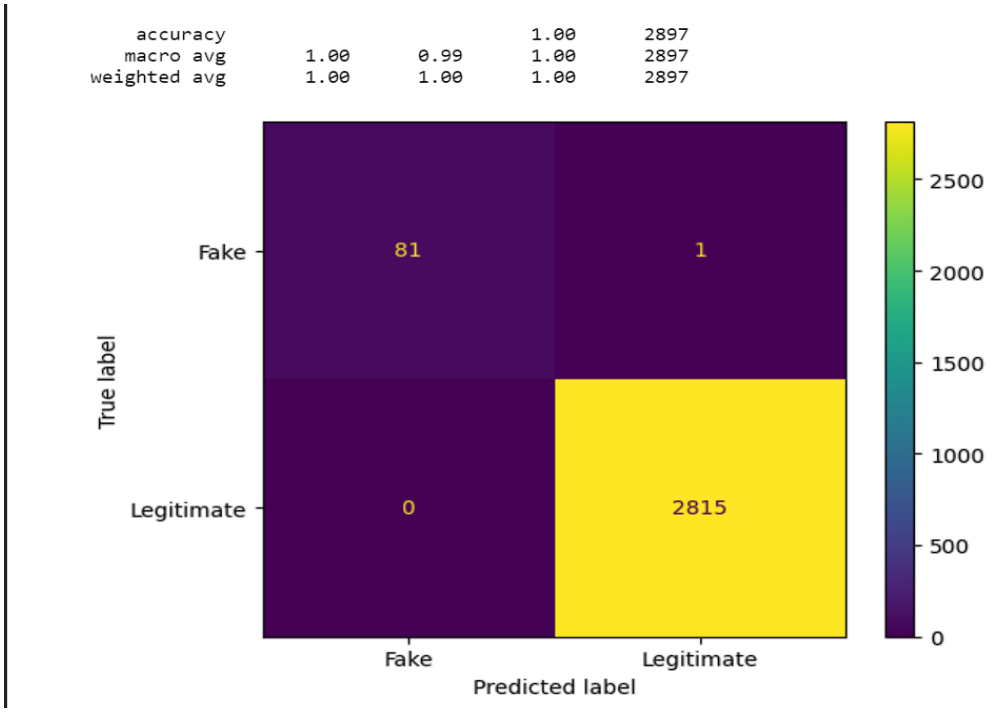


Рис. 3.6 – Результати навчання

Як можна побачити, моделі показують майже 100% точність, що в свою чергу може бути не дуже добрим знаком та свідчити про перенавчання. Щоб перевірити цю гіпотезу модель перевіряють на валідаційних даних зі справжніх запитів, легітимних та нелегітимних.

Результати застосування моделі на реальних даних зображені на рисунку 3.7 нижче.

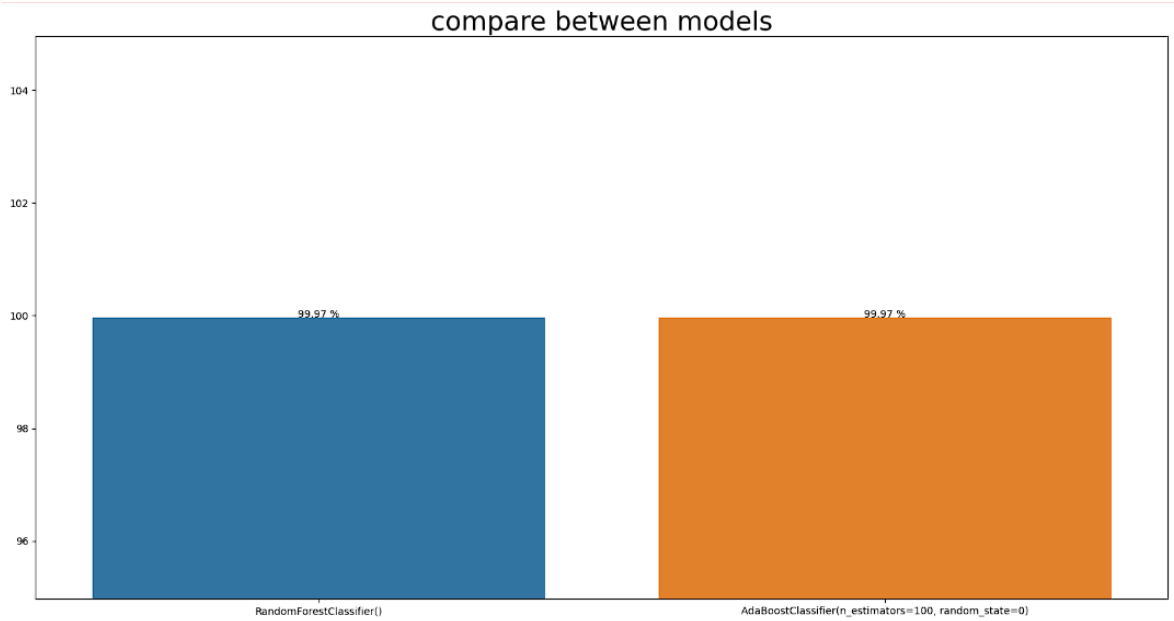


Рис. 3.7 – Ефективність моделей на першому етапі

Як можна побачити, гіпотезу спростовано – навіть на реальних даних моделі показують високу ефективність. У червоної команди не було жодних шансів проти моделей. Але в другому раунді все може змінитися.

Схему другого раунду зображено на рисунку 3.8 нижче.

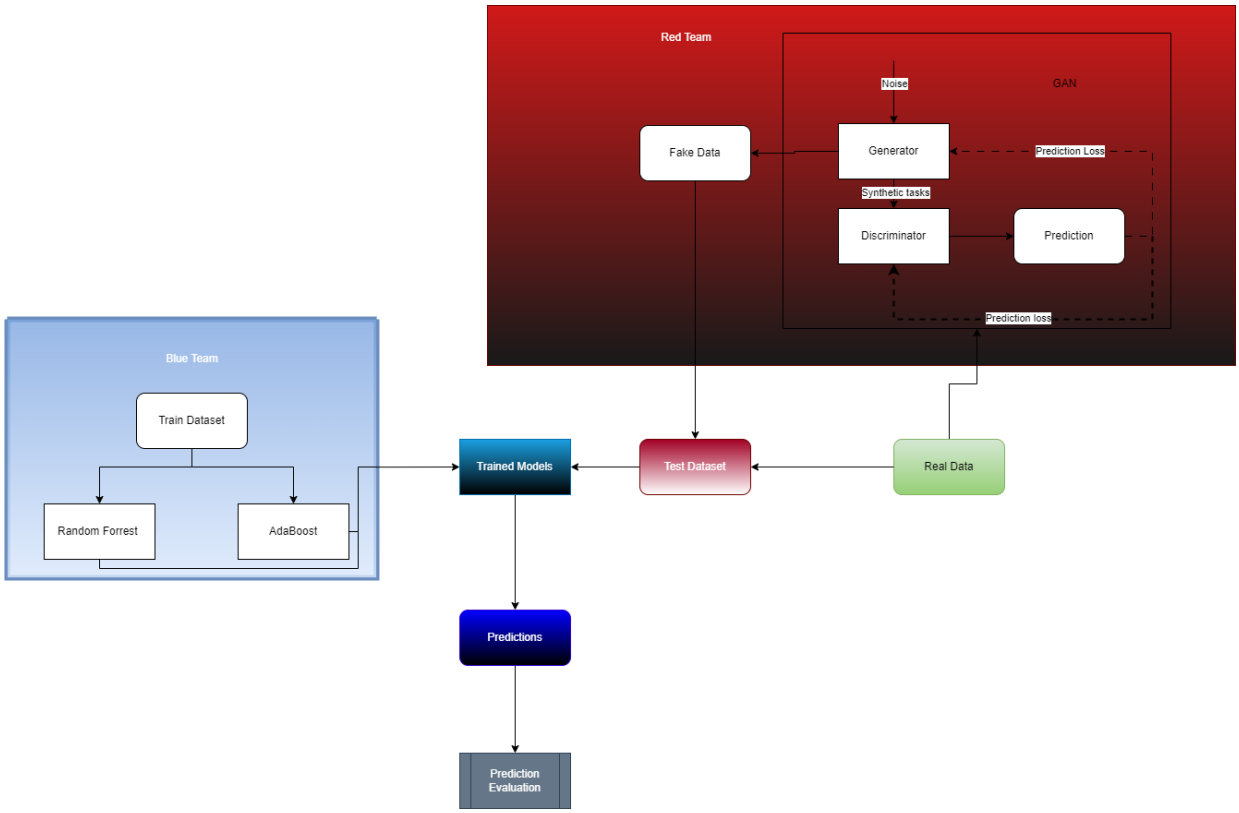


Рис. 3.9 – Схема 2 етапу

Цього разу червона команда буде навчати модель GAN, для більш успішної підробки запитів. Як було зазначено вище – модель складається з двох частин – генератора та дискримінатора. Червона команда використає зразки реальних даних, щоб навчити генератор маскуватися під запити реальних користувачів. Потім команда почне також тренувати дискримінатор, щоб він вчився розпізнавати підроблені запити. Як наслідок, протягом кожної ітерації дискримінатор буде вчитися краще розпізнавати нелегітимні запити, а генератор буде все краще їх підробляти. Після навчання червона команда використає генератор для створення нелегітимних запитів.

Синя команда так само, як і минулого разу, має розпізнати нелегітимні запити за допомогою RF та AdaBoost. Синя команда знов навчає моделі на тренувальних даних та застосовує їх проти запитів, згенерованих GAN.

Результати роботи моделей можна побачити на рисунках 3.10 та 3.11 нижче.

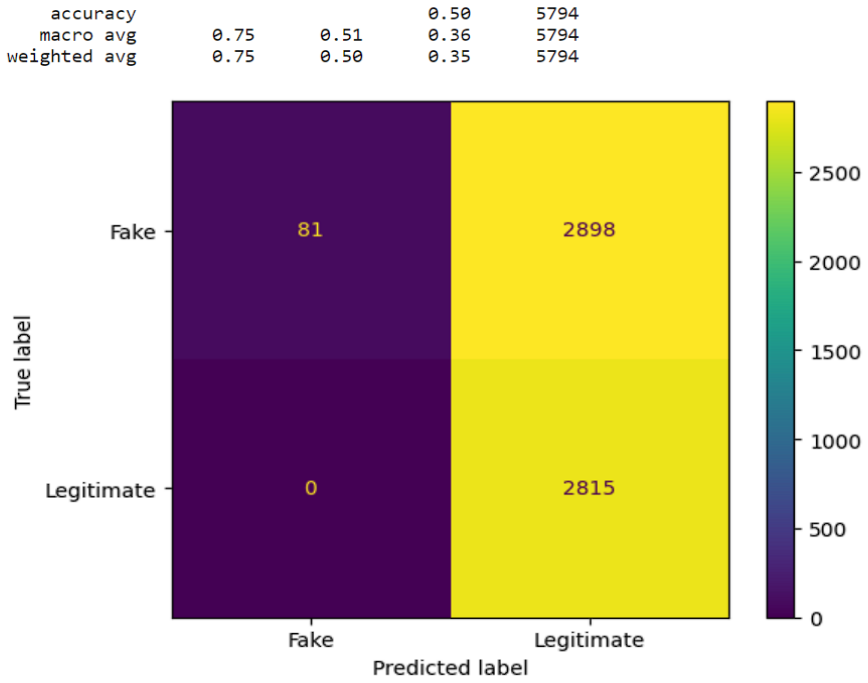


Рис. 3.10 – Результати застосування GAN

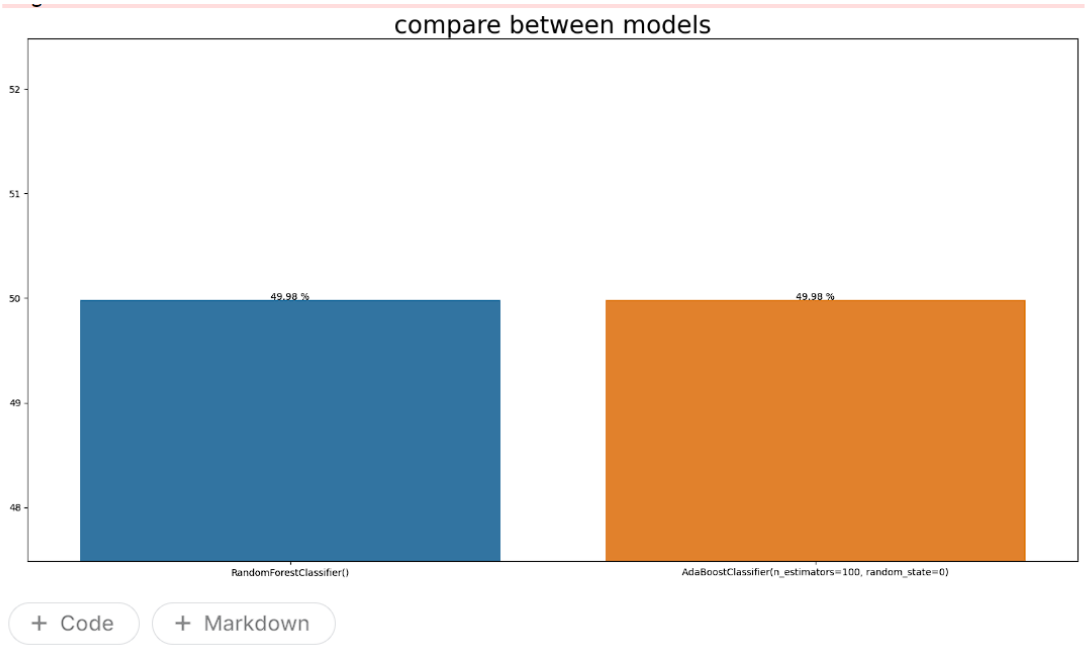


Рис. 3.11 – Ефективність моделей на 2 етапі

Як можна побачити, ефективність моделей впала дуже значно. Моделі тепер показують ефективність нижче 50%, тобто передбачення шляхом підкидання монетки буде більш ефективним ніж використання моделей.

Використання GAN алгоритмів для атаки показало високу ефективність, однак в 3 раунді синя команда зможе взяти реванш шляхом використання GAN вже для захисту системи.

Схему третього раунду зображено на рисунку 3.12 нижче.

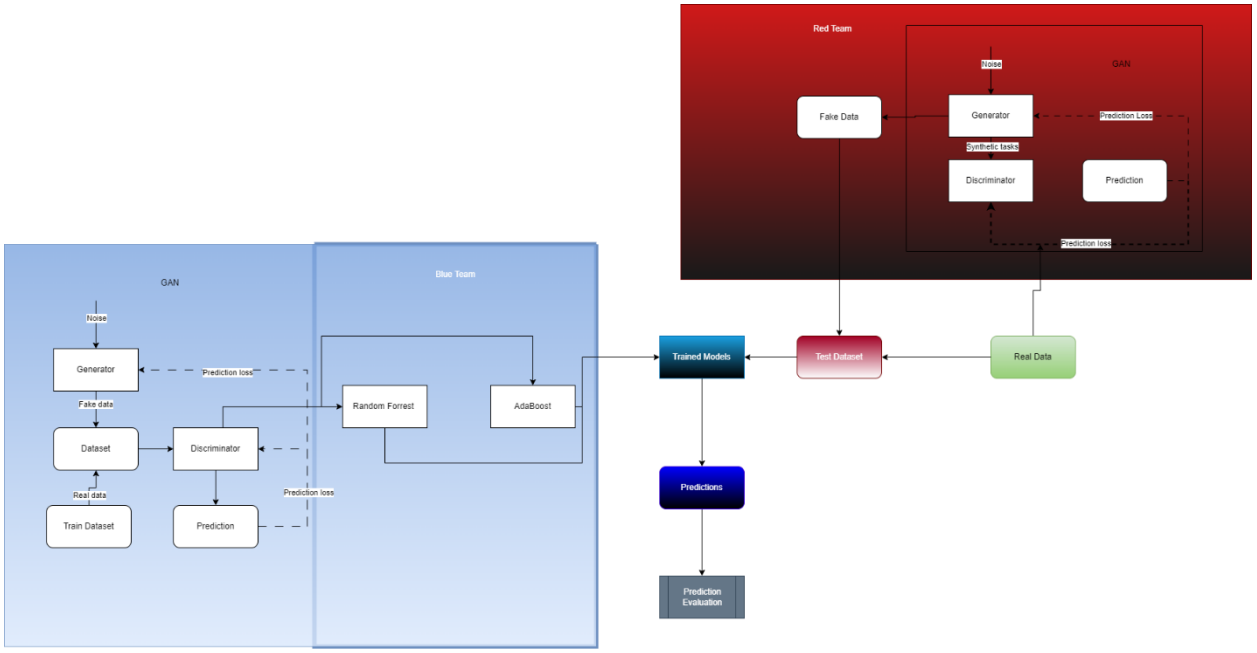


Рис. 3.12 – Схема 3 етапу

В цьому раунді вже синя команда також навчає GAN мережу, але задля своїх потреб команда використовує дискримінатор. Після навчання дискримінатор інтегрується в існуючі моделі як перший шар обробки інформації. Вихідні дані дискримінатора подаються на моделі RF та AdaBoost як на другий шар задля коригування передбачень та підвищення надійності.

Результати застосування моделі зображено на рисунках 3.13 та 3.14 нижче.

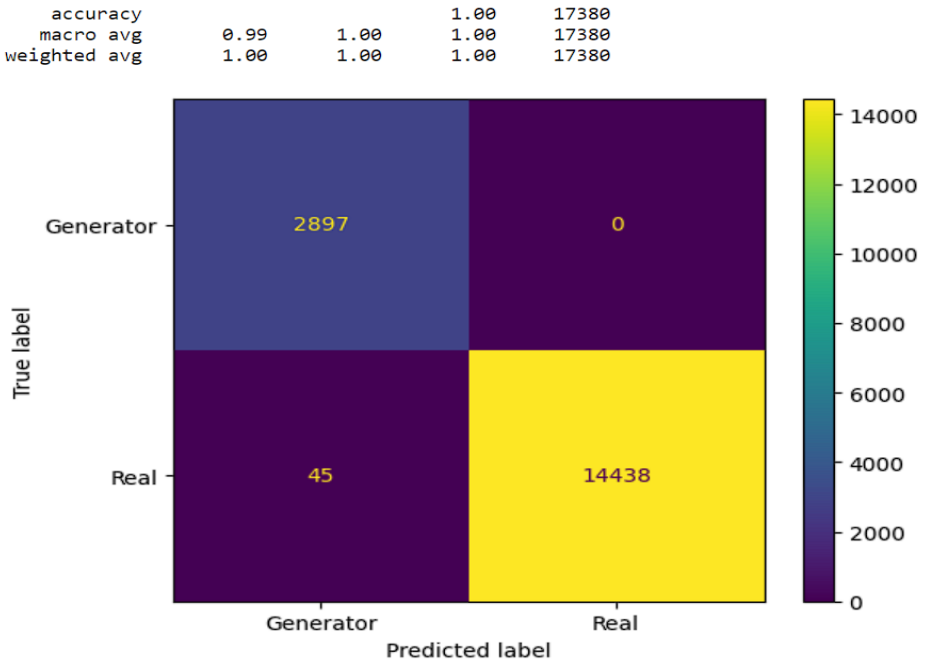


Рис. 3.13 – Результати застосування дискримінатора

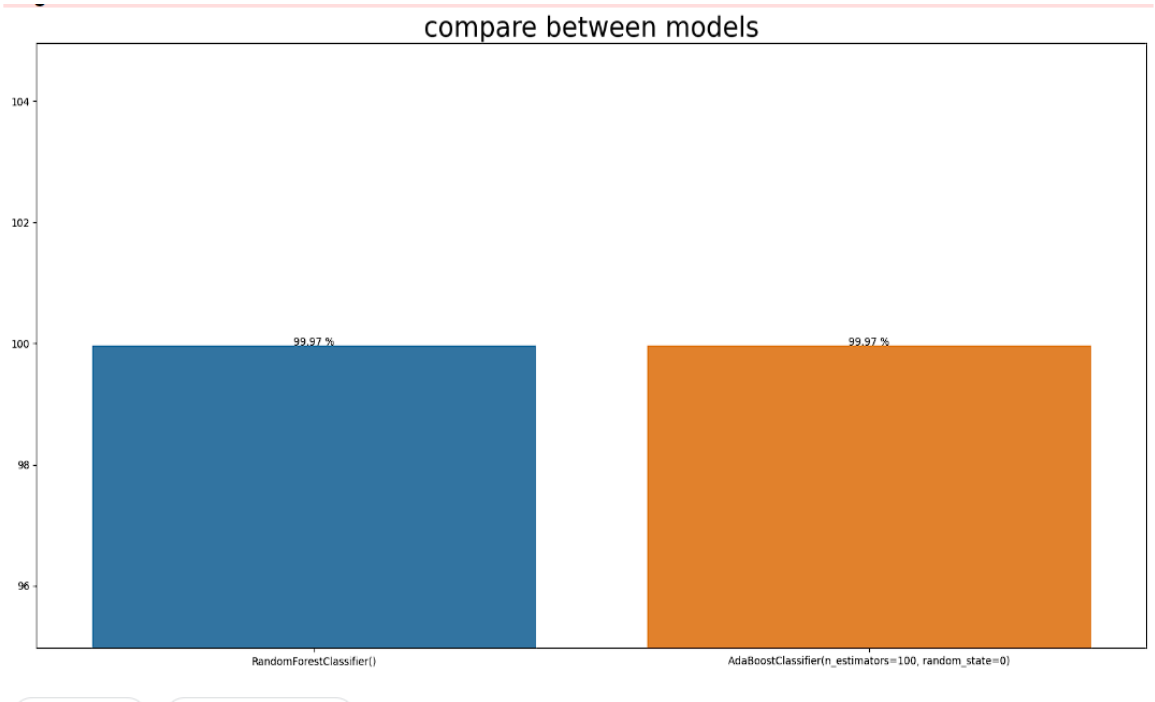


Рис. 3.14 – Ефективність моделей у 3 раунді

Підсумки всіх раундів симуляції зображено у таблиці 3.1 нижче.

Табл. 3.1 – Підсумки симуляції

№ Раунду	Технології червоної команди	Технології синьої команди	Ефективність моделей захисту
1	Стандартні технології фішингу та атак на MIC	Random Forrest Classifier, AdaBoost, стандартні методи захисту MIC	99,97%
2	GAN	Random Forrest Classifier, AdaBoost, стандартні методи захисту MIC	49,98%
3	GAN	Все вищезазначене + GAN	99,98%

У підсумку симуляції можна констатувати, що призначені дві команди, червона та синя використовували методології кібербезпеки та науки про дані заради підвищення ефективності виконання власної задачі. Синя команда використовувала випадковий ліс і AdaBoost для виявлення нелегітимних запитів, щоб захистити MIC від шахрайства, команді вдалося виявити майже 99% фальшивих завдань, але класифікатор виявився слабким для виявлення продвинутих атака, тож червона команда намагається атакувати систему

виявлення за допомогою мереж GAN, і їй вдалося знизити точність з 99% до 49%, тож синя команда теж починає використовувати GAN, щоб відфільтрувати будь-які фальшиві завдання створені за допомогою генератора GAN, які допомагають моделі повернути високу точність і виявити всі підробки, створені випадковим чином і створені за допомогою передових методів.

ВИСНОВОК

Інтеграція науки про дані в практику кібербезпеки дає можливість організаціям проактивно захищати свої цифрові активи від нових загроз. Надані джерела пропонують подальше розуміння застосування та значення науки про дані в цій важливій сфері.

Загалом, використання методів Data Science в кібербезпеці дозволяє підвищити рівень захисту інформації, швидко реагувати на загрози та ефективно боротися з кібератаками. Даний напрямок залишається однією з ключових складових стратегії забезпечення кібербезпеки в сучасному цифровому середовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Statistics and Data Science for Cybersecurity / [A. Hero, S. Kar, J. Moura та ін.]., 2023. – 36 с. – (Harvard Data Science Review).

2. Data Science and Cybersecurity: Trends and Innovations [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://www.researchgate.net/publication/326383877_Data_Science_and_Cybersecurity_Trends_and_Innovations.

3. Schick T. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference / T. Schick, H. Schütze., 2021. – (EACL2021).

4. The Role of Big Data in Cybersecurity [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.weforum.org/agenda/2018/02/the-role-of-big-data-in-cybersecurity.

5. Bhardwaj A. Predictive Analytics-Based Cybersecurity Framework for Cloud Infrastructure / A. Bhardwaj, K. Kaushik. – Dehradun, India, 2022. – 20 с. – (International Journal of Cloud Applications and Computing).

6. Ankita. Analysis of Machine Learning and Deep Learning Intrusion Detection System in Internet of Things Network / Ankita, A. Bashir, S. Rani., 2022. – 10 с. – (International Conference on Data Analytics for Business and Industry (ICDABI)).

7. Duplicated Replay Buffer for Asynchronous Deep Deterministic Policy Gradient / [S. Motehayeri, V. Baghi, E. Miandoab та ін.]. – Teheran: IEEE, 2021. – 173 с.

8. Al-Qarni E. Cybersecurity in Healthcare: A Review of Recent Attacks and Mitigation Strategies / Elham Abdullah Al-Qarni // (IJACSA) International Journal of Advanced Computer Science and Applications / Elham Abdullah Al-Qarni. – Bisha, 2023. – С. 135–140.

9. Bell G. HEALTH CARE AND CYBER SECURITY: Increasing Threats Require Increased Capabilities / G. Bell, M. Ebert., 2015. – 8 с. – (HEALTH CARE AND CYBER SECURITY).

10. Cutler A. Random Forests / A. Cutler, J. Stevens, D. R. Cutler // Ensemble Machine Learning: Methods and Applications / A. Cutler, J. Stevens, D. R. Cutler., 2011. – С. 157–176.

11. Hertzmann A. AdaBoost / A. Hertzmann, D. Fleet, M. Brubaker // Machine Learning and Data Mining Lecture Notes / A. Hertzmann, D. Fleet, M. Brubaker. – Toronto: Department of Computer and Mathematical Sciences University of Toronto Scarborough, 2015. – С. 123–129.

12. Wang S. Generative Adversarial Networks (GAN) A Gentle Introduction / S. Wang. – Texas: Department of Statistics and Data Science University of Texas at Austin, 2017. – 10 с.

13. CrowdSenSim Simulator Version 1.0.0 User Manual – Luxembourg: University of Luxembourg, 2016. – 9 с.