НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

«На правах рукопису»
УДК 004.891.2

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інтегровані інформаційні системи»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Система виявлення вторгнень у комп'ютерну мережу»

Виконав (-ла):
студент (-ка) VI курсу, групи ІА-92мп
Сокирко Дмитро Борисович _____

Керівник:
Професор кафедри АУТС, д.т.н., проф.
Корнієнко Богдан Ярославович _____

Консультант: _____

Рецензент:
Доцент кафедри технічних та програмних
засобів автоматизації ІХФ, к.т.н., доцент
Ладієва Леся Ростиславівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Сокирку Дмитру Борисовичу

1. Тема дисертації «Система виявлення вторгнень у комп'ютерну мережу», науковий керівник дисертації Корнієнко Богдан Ярославович, д.т.н., професор, затверджені наказом по університету від «26» 10 2020 р. №3132-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження вторгнення у комп'ютерну мережу

4. Вихідні дані виявлення вторгнень у комп'ютерну мережу

5. Перелік завдань, які потрібно розробити: аналіз існуючих рішень, формування вимог до системи, розробка системи, стартап-проект

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: структурна схема розміщення системи в мережі, сценарії використання системи, структурна схема фізичних елементів системи, блок-схема роботи системи, структурна схема логічних елементів, блок-схема алгоритму роботи системи, блок схема роботи елементу «препроцесор», діаграма класів системи.

7. Орієнтовний перелік публікацій

8. Консультанти розділів дисертації*

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

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз існуючих рішень	2.09.2020 р.	
2	Формування вимог до системи	8.09.2020 р.	
3	Вибір та обґрунтування технологій	28.09.2020 р.	
4	Розробка моделі системи	10.10.2020 р.	
5	Розробка структурної схеми	7.11.2020 р.	
6	Розробка системи	11.11.2020 р.	
7	Розробка стартап-проекту	15.11.2020 р.	
8	Оформлення текстової документації	30.11.2020 р.	

Студент Дмитро СОКИРКО

Науковий керівник Богдан, КОРНІЄНКО

* Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Сокирко Д. Б. Система виявлення вторгнень у комп’ютерну мережу. КПІ ім. Ігоря Сікорського, Київ, 2020.

Робота містить 102 с. тексту, 37 рисунків, 32 таблиці та 57 джерел.

З постійним зростом введення комп’ютерних технологій у різних сферах діяльності, з’являється проблема їх безпеки, створення системи виявлення вторгнень у комп’ютерну мережу є актуальною задачею.

Об’єкт дослідження – вторгнення в комп’ютерну мережу

Метою магістерської дисертації є підвищення ефективності виявлення вторгнень за рахунок алгоритмів машинного навчання.

Предметом дослідження є виявлення вторгнень у комп’ютерній мережі.

Ключові слова: система виявлення вторгнень, датасет, машинне навчання.

ABSTRACT

Sokyrko D. “Computer network intrusion detection system”. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2020.

The work contains 102 pages of text, 37 figures, 32 tables and 57 sources.

With the constant increase in the introduction of computer technologies in various fields of activity, there is a problem of their security, the creation of a system for detecting intrusions into the computer network is an urgent task.

The object of research is an intrusion into a computer network

The aim of the master's thesis is to improve the efficiency of intrusion detection through machine learning algorithms.

The subject of research is intrusion detection in a computer network.

Keywords: intrusion detection system, dataset, machine learning.

Зміст

Перелік умовних позначень

Датафрейм - це двовимірна маркована структура

КМ – комп’ютерна мережа

IDS – система виявлення вторгнень

KNN – метод машинного навчання k-найближчих сусідів

ML – машинне навчання

RF – метод машинного навчання випадковий ліс

SVM – метод машинного навчання, метод опорних векторів

ВСТУП

В даний час жодна компанія не може уявити роботу без використання комп'ютерних технологій. Обов’язковим атрибутом офісу будь-якої сучасної компанії є комп'ютери. А отже необхідно створення локальних або по іншому корпоративних мереж, в яких зазвичай задіяні практично всі комп'ютери компанії.

З впровадженням комп’ютерної мережі та підключенням її до глобальної мережі інтернет, компанія отримує майже необмежені інформаційні можливості, оперативний обмін інформацією(отримання новин, листів, замовлень, тощо), скорочення паперового документообігу всередині підприємства, підвищення продуктивності праці, скорочення часу на обробку інформації.

Стрімке зростання популярності інтернет-технологій супроводжується зростанням серйозних загроз розголошення персональних даних, критично важливих корпоративних ресурсів, державних таємниць і т.д. У міру розвитку і ускладнення засобів, методів і форм автоматизації процесів обробки інформації з'явилася проблема і їх безпеки використовуваних їм інформаційних технологій. Тому забезпечення інформаційної безпеки комп'ютерних систем і мереж є одним з провідних напрямків розвитку інформаційних технологій.

Багато керівників фірм навіть замислюються про те, як може вплинути на роботу організації несанкціоноване проникнення в корпоративну мережу, але часто цим нехтують, покладаючи роботу на плечі системного адміністратора. Кожен керівник повинен розуміти важливість захисту своєї мережі від несанкціонованих посягань, атак з глобальної мережі і т.д. Кожна поважаюча себе фірма зв'язується з іншими локальними мережами через глобальну мережу Інтернет, так як налагоджувати окремі канали зв'язку між окремими філіями досить дорого, а таке можуть дозволити собі тільки найбільші корпорації. Отже очікувати загрозу безпеки мережі можна очікувати не тільки з боку співробітників своєї корпорації, а й з боку хакерів через інтернет і з боку конкуруючих фірм, які можуть найняти тих же хакерів і спрямувати їх діяльність на псування конкретного майна.

Метою дослідження є створення системи виявлення вторгнень в комп’ютерній мережі.

Об’єкт дослідження – процес виявлення аномальної активності в роботі комп’ютерної мережі.

Предмет дослідження – методи побудови систем виявлення проникнень в комп’ютерній мережі.

В процесі виконання магістерської дисертації планується розв’язання наступних задач:

- обґрунтування доцільності розробки системи виявлення вторгнень в КМ;
- аналіз трафіку для виявлення вторгнень в КМ;
- опис принципів аналізу КМ на вторгнення;
- розробка системи виявлення вторгнень у комп'ютерній мережі.

Модель системи виявлення вторгнень було реалізовано з використанням мови програмування Python.

Магістерська дисертація складається з наступних розділів: вступ, огляд існуючих рішень, формування вимог до системи, вибір та обґрунтування технологій, розробка системи, стартап-проект, висновки, список використаних джерел.

Графічна система включає 8 креслеників формату А3.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Основні поняття та доцільність використання систем виявлення вторгнень

Система виявлення вторгнень (IDS) - це пристрій або програмний додаток, який відстежує мережу або системи на предмет зловмисних дій чи порушень політики. Про будь-які вторгнення або порушеннях зазвичай повідомляється адміністратору або збирається централізовано за допомогою системи управління інформацією і подіями безпеки (SIEM). Система SIEM об'єднує вихідні дані з декількох джерел і використовує методи фільтрації сигналів тривоги, щоб відрізнити шкідливу активність від хибних сигналів.

Для вирішення цього завдання IDS повинні виконувати наступні основні функції:

- моніторинг подій з метою виявлення інцидентів інформаційної безпеки;);
- запис інформації про інциденти як локально, так і з відправкою в будь-яку централізовану систему збору журналів або SIEM-систему;
- повідомлення адміністраторів про інциденти ІБ (електронна пошта, трапів SNMP і СМС, ідентифікатори системи консолі управління);
- створювати звіти, які їх уточнюють, навпаки, узагальнюють інформацію по одному або декільком подіям.

Типи систем виявлення вторгнень варіюються від окремих комп'ютерів до великих мереж. Найбільш поширеними класифікаціями є системи виявлення мережевих вторгнень (NIDS) і системи виявлення вторгнень на основі хостів (HIDS). Система, яка відстежує важливі файли операційної системи, є прикладом HIDS, а система, яка аналізує вхідний мережевий трафік, є прикладом NIDS. Також можливо класифікувати IDS за методом виявлення. Найбільш відомі варіанти - це виявлення на основі сигнатур (розпізнавання поганих шаблонів, наприклад шкідливих програм) і виявлення на основі аномалій (виявлення відхилень від моделі «доброго» трафіку, яка часто заснована на машинному навчанні). Приклад схеми розміщення бази сигнатур системи виявлення на основі сигнатур зображено на рисунку 1.1.

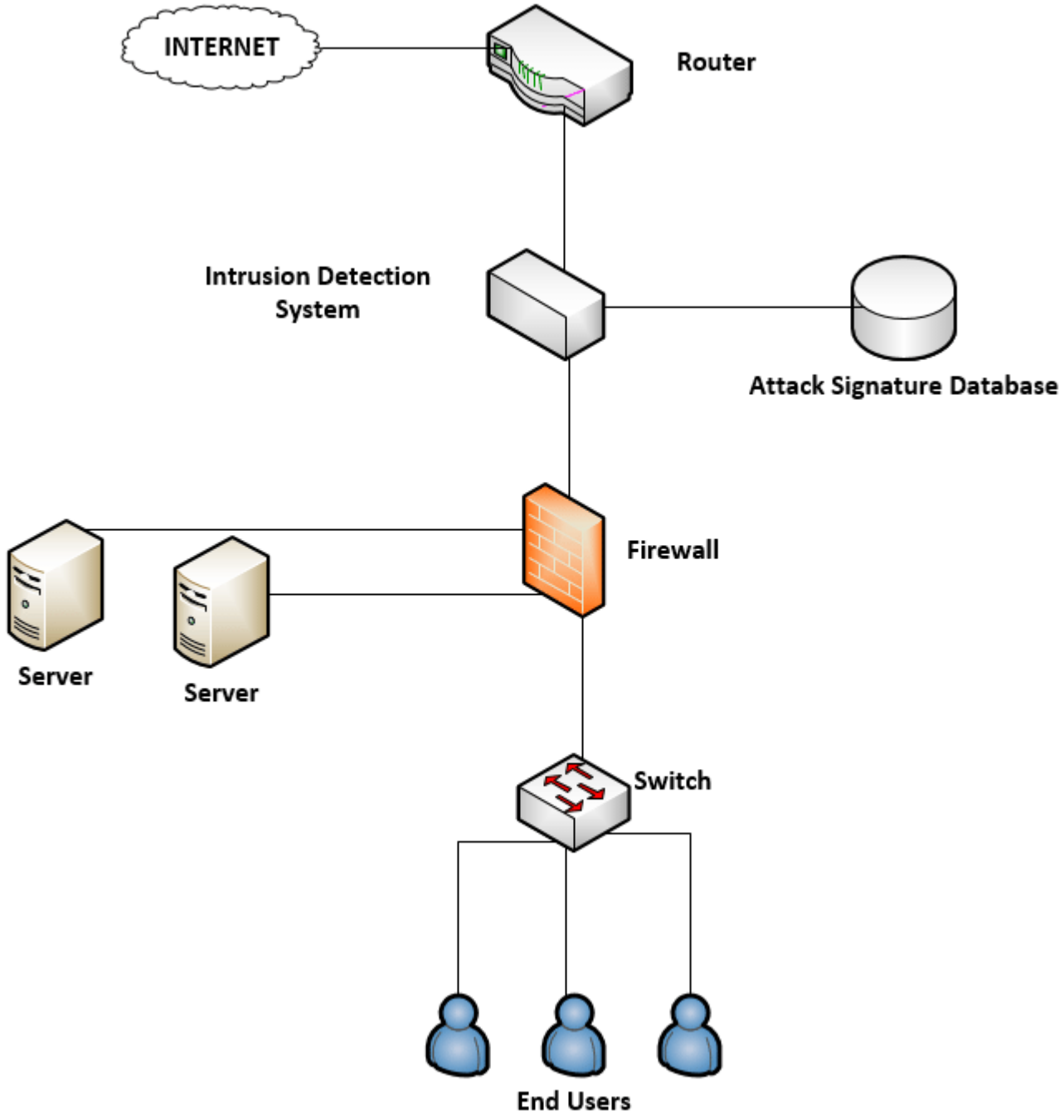

Рисунок 1.1 – Приклад розміщення системи виявлення на основі сигнатур

Інший поширений варіант - це виявлення на основі репутації (розпізнавання потенційної загрози за оцінками репутації). Деякі продукти IDS можуть реагувати на виявлені вторгнення. Системи з можливістю реагування зазвичай називають системами запобігання вторгнень. Системи виявлення вторгнень також можуть служити певним цілям, доповнюючи їх налаштованим інструментами, такими як

використання приманки для залучення і визначення характеристик шкідливого трафіку.

1.2 Порівняння з міжмережевими екранами

Хоча обидва вони відносяться до елементів мережевої безпеки, IDS відрізняється від мережевого екрану тим, що традиційний мережевий між мережевий екран (на відміну від між мережевих екранів нового покоління) використовує статичний набір правил для дозволу або заборони здійснювати підключення до мережі. Розміщення між мережевого екрану зображено на рисунку 1.2.

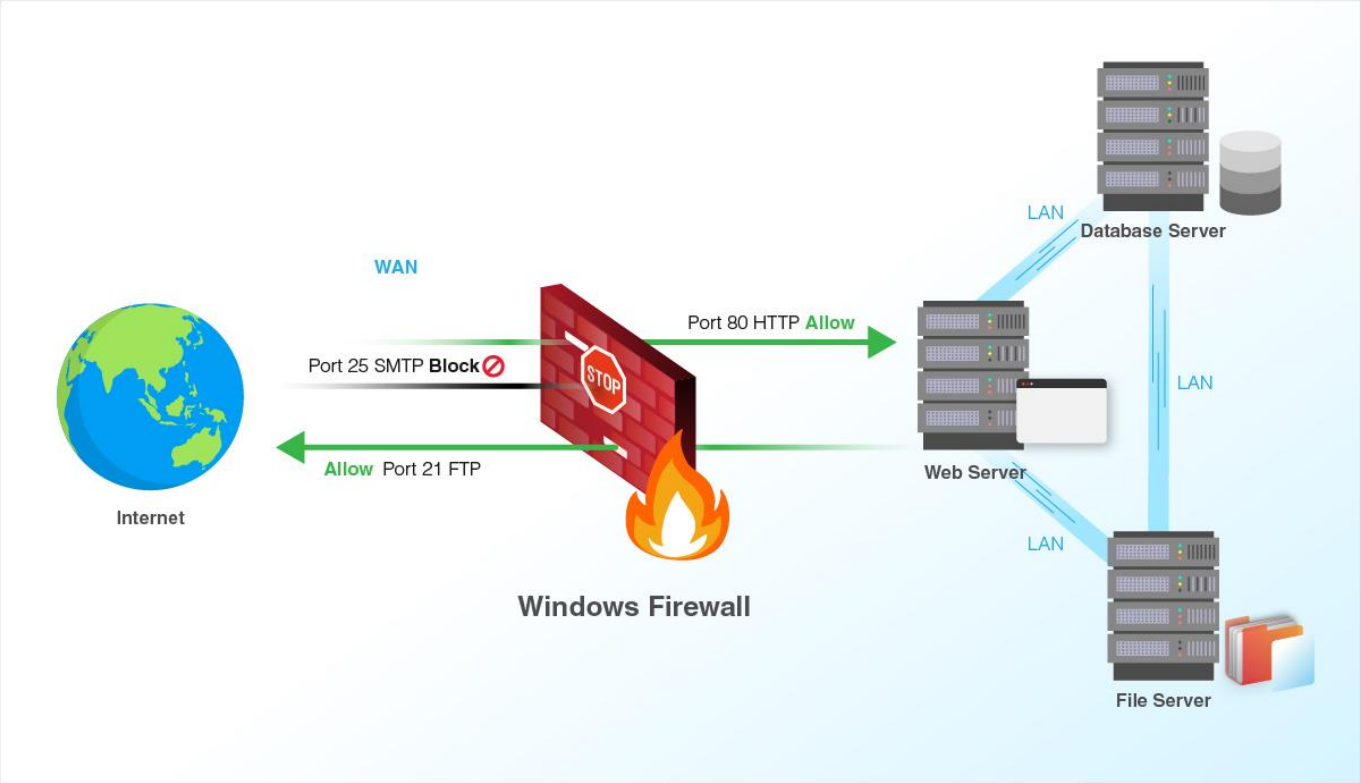

Рисунок 1.2 – Розміщення міжмережевого екрану

Він неявно запобігає вторгнення за умови, що визначений відповідний набір правил. По суті, між мережевий екрани обмежують доступ між мережами, щоб запобігти вторгненню, і не сигналізують про атаку зсередини мережі. IDS описує передбачуване вторгнення після того, як воно сталося, і сигналізує про тривогу. IDS

також відстежує атаки, які виходять із середини системи. Це зазвичай досягається шляхом вивчення мережевих комунікацій, виявлення евристик і шаблонів (часто званих сигнатурами) поширених комп'ютерних атак та вжиття заходів щодо попередження операторів. Система, яка завершує з'єднання, називається системою запобігання вторгнень і виконує контроль доступу, як між мережевий екран прикладного рівня.

IDS можна класифікувати по тому, де відбувається виявлення (мережа або хост), або по використовуваному методу виявлення (на основі сигнатури або аномалії).

1.3 Типи IDS

Існує класичне поділення IDS на:

- системи рівня мережі, на які відводиться трафік з маршрутизатора;
- системи рівня хосту, які виявляють зміни на окремо взятій машині, наприклад аналізуючи журнали або мережеву активність;
- системи, засновані на оцінці вразливостей.

Метою будь-якої такої системи є вирішення задач, а вирішення виноситься на основі отриманих даних. Задачі системи складаються з:

- отримання даних;
- інтерпретація отриманих даних;
- представлення результату.

Всі системи можна позиціонувати по значенню наступних ознак:

- тип даних, що збирається;
- метод отримання даних;
- метод інтерпретації даних;
- метод представлення результату.

Позасистемною характеристикою можна вважати тип реакції на результат:

- інформативний;
- активний.

У першому випадку відбувається інформування зацікавлених персон.

У другому – активні дії, наприклад блокування діапазону адрес джерела атак. По цій характеристиці дані системи зазвичай розділяють на IDS та IPS. Характеристика позасистемна, бо припускається розділення системи на «розвідувальну» та «силову» частини, та будь-яка IDS може бути включена до складу IPS.

1.3.1 Системи виявлення мережевих вторгнень

Системи виявлення вторгнень розміщуються в стратегічній точці або точках в мережі для відстеження трафіку (рис. 1.3), що надходить і виходить від всіх пристроїв в мережі. Він виконує аналіз проходить трафіку по всій підмережі і зіставляє трафік, який проходить по підмережах, з бібліотекою відомих атак. Після виявлення атаки або виявлення аномальної поведінки попередження може бути відправлено адміністратору. Прикладом IDS може бути його установка в підмережі, де розташовані брандмауери, щоб побачити, чи не намагається хтось зламати брандмауер. В ідеалі потрібно сканувати весь вхідний і вихідний трафік, проте це може створити вузьке місце, яке знизить загальну швидкість мережі. OPNET і NetSim – широко використовуються інструменти для моделювання систем виявлення мережевих вторгнень. Системи NID також здатні порівнювати підписи для подібних пакетів, щоб зв'язати і відкинути виявлені шкідливі IDS, підпис яких збігається з записами в NIDS. Коли ми класифікуємо дизайн IDS відповідно до властивістю інтерактивності системи, ми виділяємо два типи: on-line і off-line IDS, часто звані вбудованим режимом і режимом торкання відповідно. Он-лайн IDS працює з мережею в реальному часі. Він аналізує пакети Ethernet і застосовує деякі правила, щоб вирішити, атака це чи ні. Off-line IDS має справу з зберігаються даними і передає їх через деякі процеси, щоб вирішити, атака це чи ні.

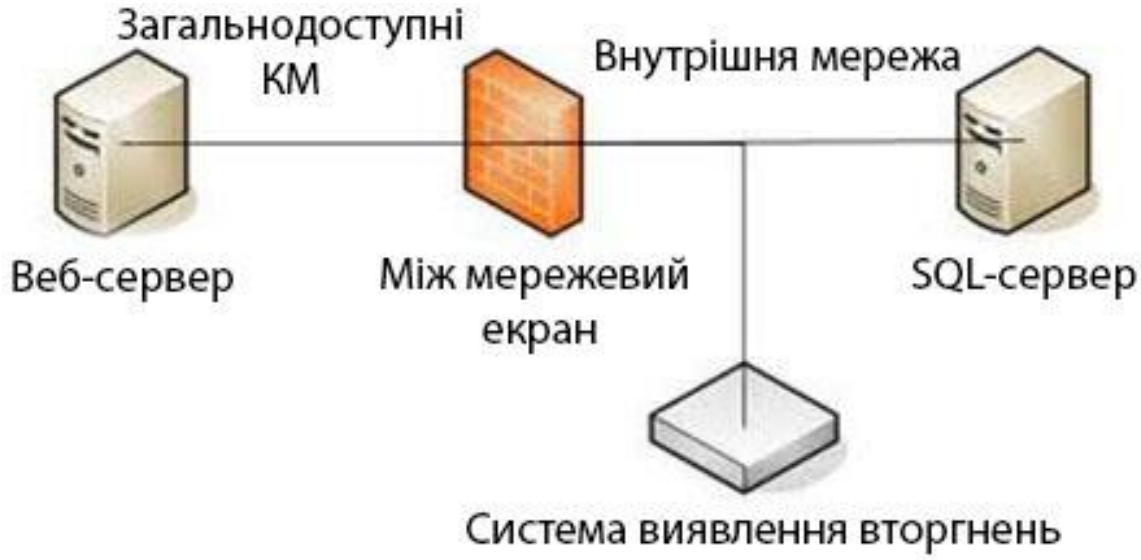

Рисунок 1.3 – Розміщення системи виявлення вторгнень.

IDS також можна комбінувати з іншими технологіями для підвищення швидкості виявлення і прогнозування. IDS на основі штучної нейронної мережі здатні аналізувати величезні обсяги даних розумним способом завдяки самоорганізованій структурі, яка дозволяє INS IDS більш ефективно розпізнавати вторгнення. Для навчання такої системи необхідно використовувати велику базу сигнатур, не правильно навчена модель може призвести до хибних спрацювань системи. Також необхідно використовувати свіжі бази сигнатур. Приклад системи зображено на рисунку 1.4.

Нейронні мережі допомагають IDS передбачати атаки, навчаючись на помилках; INN IDS допомагає розробити систему раннього попередження, засновану на двох рівнях. Перший рівень приймає поодинокі значення, а другий рівень приймає вихідні дані першого рівня в якості вхідних; цикл повторюється і дозволяє системі автоматично розпізнавати нові непередбачені моделі в мережі. Ця система може в середньому виявляти і класифікувати 99,9%, грунтуючись на результатах дослідження 24 мережевих атак, розділених на чотири категорії: DOS, Probe, Remote-to-Local і user-to-root.

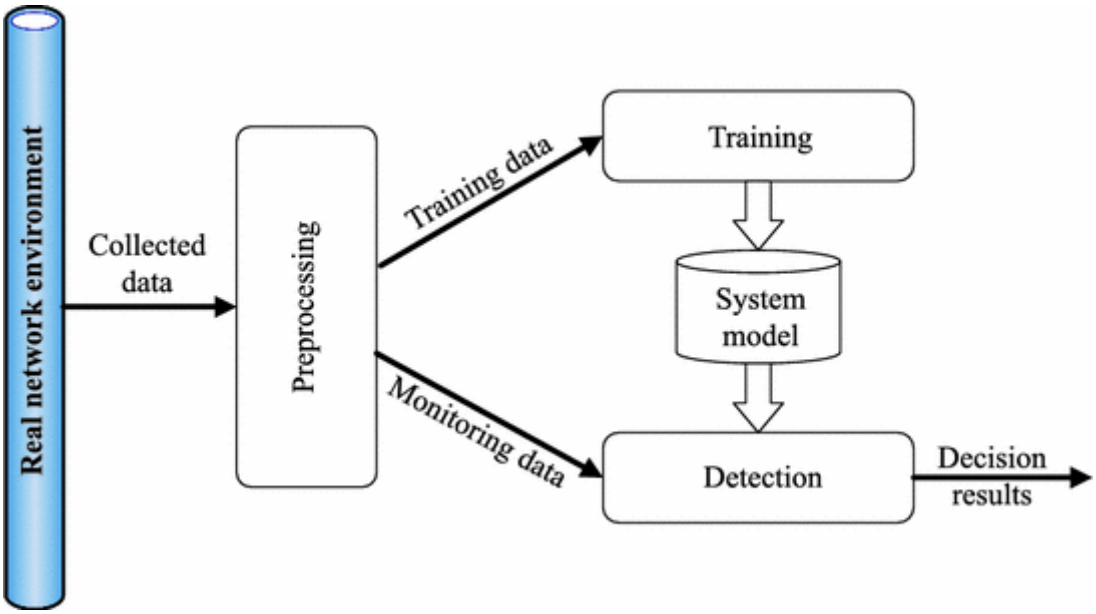

Рисунок 1.4 – Приклад системи, що базується на машинному навчанні[1]

1.3.2 Хост-системи виявлення вторгнень

Системи виявлення вторгнень хосту працюють на окремих хостах або пристроях в мережі. Система відстежує вхідні і вихідні пакети тільки від пристрою і попереджає користувача або адміністратора при виявленні підозрілої активності. Він робить знімок існуючих системних файлів і зіставляє його з попереднім знімком. Якщо критичні системні файли були змінені або видалені, адміністратору відправляється попередження для розслідування. Приклад використання можна побачити на критично важливих машинах, від яких не очікується зміна своєї конфігурації.

1.3.3 Класи типів даних, що збираються

Дотримуючись класичного поділу, можна ввести три класи:

- дані, що збираються про вузол мережі;
- дані, що збираються про всю мережу;
- гібридна система.

1.3.4 Дані, що збираються про вузол мережі

Типи даних, що збираються про вузол мережі – це дані, які стосуються тільки одного вузла та частково тих, які з ним взаємодіють. Аналіз таких даних дозволяє визначити чи відбувається атака на даний хост. Як правило, ці дані зручно збирати безпосередньо на вузлі, але це не обов'язково. Деякі нові сканери можуть отримати список відкритих портів на конкретному вузлі, не маючи можливості запустити код на ньому.

Цей клас включає в себе дані наступних типів, до кожного з яких відносяться конкретні показники, що збираються:

- мережева активність вузла;
- мережеві налаштування вузла;
- данні про файли(списки та контрольні суми, метадані, операції з файлами та інші);
- данні про процеси.

При цьому, вузли можуть бути, як робочими станціями, які не передбачають їх використання в якості серверів, що надають служби, так і серверами. Також можна виділити окремий випадок, коли хост може бути спеціально зроблений вразливим, з ціллю вивчення методів атак та виявлення атакуючих вузлів. Можна припустити, що кожна взаємодія з даним вузлом буде спробою атаки.

1.3.5 Дані, що збираються про всю мережу

Дані, що збираються про мережу – це загальна картина мережевої взаємодії. Як правило, повні мережеві дані не збираються, через те, що це ресурсо-затратно та вважається, що порушник не може знаходитися всередині мережі, або йому обов'язково необхідний зв'язок з зовнішнім середовищем(крім техніки «Подолання повітряного зазору»).

В цьому випадку, IDS аналізує трафік, що йде через маршрутизатор, для чого в маршрутизаторі є SPAN порт, який використовується для перенаправлення трафіку в

IDS. Також можна зібрати дані з вузла на якому працює IDS, що в свою чергу буде являти собою додатковий контроль.

1.3.6 На основі сигнатур

IDS на основі сигнатур відноситься до виявлення атак шляхом пошуку певних шаблонів, таких як послідовності байтів в мережевому трафіку або відомі послідовності шкідливих інструкцій, які використовуються шкідливим ПЗ. Ця термінологія походить від антивірусного програмного забезпечення, яке називає ці виявлені шаблони сигнатурами. Хоча IDS на основі сигнатур може легко виявляти відомі атаки, важко виявити нові атаки, для яких немає шаблонів.

В IDS на основі підпису підписи випускаються постачальником для всіх його продуктів. Своєчасне оновлення IDS за допомогою підпису є ключовим аспектом.

1.3.7 На основі аномалій

Системи виявлення вторгнень на основі аномалій були в першу чергу введені для виявлення невідомих атак, частково через швидкий розвиток шкідливих програм. Основний підхід - використовувати машинне навчання для створення моделі яка є «нормальною», а потім порівнювати нову поведінку з цією моделлю. Оскільки ці моделі можна навчати відповідно з додатками і конфігураціями обладнання, метод на основі машинного навчання має краще узагальнене властивість у порівнянні з традиційними IDS на основі сигнатур. Хоча цей підхід дозволяє виявляти раніше невідомі атаки, він може страждати від помилкових спрацьовувань: раніше невідома законна діяльність також може бути класифікована як шкідлива. Більшість існуючих IDS страждають від того, що процес виявлення займає багато часу, що знижує продуктивність IDS. Ефективний алгоритм вибору ознак робить процес класифікації, що використовується при виявленні, більш надійним.

Gartner розглядає нові типи того, що можна назвати системами виявлення вторгнень на основі аномалій, як аналіз поведінки користувачів і об'єктів (UEBA)

(еволюція категорії аналітики поведінки користувачів) і аналіз мережевого трафіку (NTA). Зокрема, NTA має справу з зловмисними інсайдерами, а також з цільовими зовнішніми атаками, які скомпрометували комп'ютер або обліковий запис користувача. Gartner відзначає, що деякі організації зволіли NTA більш традиційної IDS.

1.4 Розміщення IDS

Розміщення систем виявлення вторгнень має вирішальне значення і залежить від мережі. Найбільш поширене розміщення – за між мережевим екраном на краю мережі. Ця практика забезпечує IDS високу видимість трафіку, що входить у вашу мережу, і не буде отримувати трафік між користувачами в мережі. Край мережі - це точка, в якій мережа підключається до екстрамережі. Ще одна практика, яку можна виконати, якщо доступно більше ресурсів, - це стратегія, при якій технічний фахівець поміщає свою першу IDS в точку максимальної видимості, а в залежності від доступності ресурсів розміщує іншу в наступній найвищій точці, продовжуючи цей процес до тих пір, поки мережі накриті.

Якщо IDS розміщується за мережевим брандмауером, його основною метою буде захист від шуму з Інтернету, але, що більш важливо, захист від поширених атак, таких як сканування портів і відображення мережі. IDS в цій позиції буде контролювати рівні з 4 по 7 моделі OSI і буде грунтуватися на сигнатурі. Це дуже корисна практика, тому що замість того, щоб показувати фактичні порушення в мережі, які пройшли через брандмауер, будуть показані спроби порушення, що знижує кількість помилкових спрацьовувань. IDS в цьому положенні також допомагає скоротити час, необхідний для виявлення успішних атак на мережу.

Іноді IDS з більш просунутими функціями інтегрується з міжмережевим екраном, щоб мати можливість перехоплювати складні атаки, що проникають в мережу. Приклади додаткових функцій можуть включати кілька контекстів безпеки на рівні маршрутизації і в режимі моста. Все це, в свою чергу, потенційно знижує вартість і складність експлуатації.

Інший варіант розміщення IDS - в реальній мережі. Це дозволить виявити атаки або підозрілу активність в мережі. Ігнорування безпеки в мережі може викликати безліч проблем: це або дозволить користувачам створювати ризики для безпеки, або дозволити зловмиснику, який вже проник в мережу, вільно переміщатися по ній. Інтенсивна безпека інтрамережі заважає навіть хакерам в мережі маневрувати і підвищувати свої привілеї.

1.5 Огляд існуючих рішень

1.5.1 Snort

Вільна мережева система запобігання вторгнень (IPS) і виявлення вторгнень (IDS) з відкритим вихідним кодом, здатна виконувати реєстрацію пакетів і в реальному часі здійснювати аналіз трафіку в IP-мережах. Логотип системи зображено на рисунку 1.2.

Багато людей знають 1998 рік як рік виходу Windows 98, але це був також час, коли Мартін Рош вперше випустив у світ Snort. Хоча тоді Snort не був справжнім IDS, зараз все змінилося. З недавніх пір він став де-факто стандартом для IDS, завдяки величезному внеску в IT-співтовариство. Важливо відзначити, що Snort не має графічного інтерфейсу або простої у використанні адміністративної консолі, хоча до цього були вже створені багато інших інструментів з відкритим вихідним кодом, такі як BASE і Sguil. Ці утиліти мають веб-інтерфейс для запитів і аналізу попереджень, що надходять від Snort IDS.

Виконує протоколювання, аналіз, пошук по вмісту, а також широко використовується для активного блокування або пасивного виявлення цілого ряду нападів і зондувань, таких як спроби атак на переповнення буфера, приховане сканування портів, атаки на веб-додатки, SMB-зондування і спроби визначення операційної системи. Програмне забезпечення в основному використовується для запобігання проникнення, блокування атак, якщо вони мають місце. Недоліки: необхідність додаткового ПЗ для більш глибокого сканування та аналізу данних, складне налаштування та поглиблене знання функцій Snort.

1.5.2 Zeek

Zeek використовує два окремих кроки для виявлення вторгнень, включаючи перевірку трафіку і окремий аналіз.. Аналіз відбувається за допомогою перетворення потоків пакетів в події і дивиться, що відбувається, а потім використовує сценарії, щоб визначити, як реагувати. Користувач також може налаштувати ці сценарії. Логотип Zeek зображено на рисунку 1.3.

Аналіз відбувається за допомогою перетворення потоків пакетів в події і дивиться, що відбувається, а потім використовує сценарії, щоб визначити, як реагувати. Користувач також може налаштувати ці сценарії. Це дозволяє вирішити, які саме повідомлення адміністратор хоче отримувати про проблеми, і включає можливість реєструвати дані для подальшого використання і виконання програми за запитом. За допомогою Zeek можна отримати швидкий і широкий огляд мережевої активності, включаючи мережеві пристрої, типи файлів у системі і встановлене або програмне забезпечення, що використовується. Потім цю інформацію можна експортувати в засоби візуалізації, щоб допомогти адміністратору розібратися в даних. Недоліки: необхідно самостійно налаштовувати скрипти для аналізу[2].

1.5.3 Suricata

Suricata - це безкоштовний і відкритий вихідний код, зрілий, швидкий і надійний механізм виявлення мережевих загроз. Логотип зображено на рисунку 1.4.

Ядро Suricata здатний виявляти вторгнення в реальному часі (IDS), вбудоване запобігання вторгнень (IPS), моніторинг мережевої безпеки (NSM) і автономну обробку pcap.

Suricata перевіряє мережевий трафік, використовуючи потужний і великий мову правил і сигнатур, а також має потужну підтримку сценаріїв Lua для виявлення складних загроз.

Завдяки стандартним форматам введення та виведення, таким як YAML та JSON, інтеграція з такими інструментами, як існуючі SIEMs, Splunk, Logstash/Elasticsearch, Kibana та іншими базами даних, стає легкою.

Проект і код Suricata належать і підтримуються Фондом відкритої інформаційної безпеки(OISF), некомерційним фондом, що прагне забезпечити розвиток Suricata і стійкий успіх в якості проекту з відкритим вихідним кодом[3].

1.6 Висновки до розділу

У розділі розглянуто основні поняття та доцільність використання систем виявлення вторгнень та порівняння їх з між мережевими екранами. Представлено тими даних, що збираються про мережу, а саме: дані , що збираються про вузол мережі; дані, що збираються про всю мережу. Типи IDS ,що базуються на двох методах виявлення вторгнень: на основі сигнатур; на основі аномалій. Представлено огляд існуючих рішень систем виявлення, які використовуються для виявлення вторгнень, а саме, Snort, Zeek та Suricata.

2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

У сучасному світі системи виявлення та запобігання вторгнень – необхідний елемент захисту від мережевих атак. Основне завдання даних систем – виявлення фактів несанкціонованого доступу в корпоративну мережу або несанкціонованого управління нею, з виконанням відповідних заходів протидії (інформування адміністраторів про факт вторгнення, обрив з'єднання або пере-налаштування брандмауера для блокування подальших дій зловмисника і т.д.). Для вирішення поставленого завдання IDS зазвичай виконують такі основні функції:

- моніторинг подій з метою виявлення інцидентів інформаційної безпеки (ІБ);
- запис інформації про дані інциденти як локально, так і з відправкою в будь-яку централізовану систему збору логів або SIEM-систему;
- повідомлення адміністраторів ІБ про інциденти;
- створення звітів, що уточнюють або, навпаки, узагальнюючих інформацію по одному або декільком подіям.

Вимоги до системи, що розробляється можна поділити на функціональні та не функціональні. Функціональні вимоги визначають що система повинна робити, а не функціональні вимоги визначають якою система повинна бути.

2.1 Функціональні вимоги до системи

Функціональні вимоги - це вимоги які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, опрацювання даних, і інші специфічні функції які повинна виконувати система[4].

Вимоги яким повинна відповідати система:

- зчитування даних з журналу захопленого трафіку;
- конвертація даних для змоги їх класифікації;
- класифікація даних за їх характеристиками;
- видача результату класифікації;
- передача інформації для усунення проблем іншим системам захисту;

– оповіщення Адміністратора про відхилення в мережі.

2.2 Не функціональні вимоги

Не функціональні вимоги – це вимоги, що визначають критерії, за якими можна робити результати про роботу системи, а не про її конкретну поведінку. Нефункціональні вимоги можна розділити на дві основні категорії:

– виконавчі якості, такі як безпека та зручність використання, які можна спостерігати під час роботи;
– еволюційні якості, такі як тестованість, ремонтопридатність, розширюваність, та масштабованість, які знаходять своє втілення у статичній структурі програмної системи[5].

Вимоги яким повинна відповідати система:

– продуктивність;
– швидка обробка даних;
– високі показники точності;
– високі показники тестування.

2.3 Сценарії використання системи

Між мережеві екрани дозволяють отримати захист від багатьох неприємностей, фільтруючи трафік на різних рівнях, але вони призначені для пасивного захисту. Однак у багатьох ситуаціях для більш ефективного захисту необхідні ще елементи оповіщення, які повідомлять про вторгнення в мережу (або на важливий сервер мережі) чи про спробу проникнення.

Система, що була розроблена може використовуватися в декількох напрямках:

– статистичний аналіз потоків даних, заснований на математичних моделях відомих атак. для них не важлива послідовність подій, що ускладнює обхід такої системи;

- моніторинг та аналіз активності користувачів та систем. звичайно здійснюється через відповідність потоків даних визначеному набору правил. використовувані в ids правила представляють собою опис найпопулярніших векторів атак. хоча навіть невелика зміна в ході проведення атаки дозволяє зловмиснику обійти даний фільтр;
- перевірка цілості критичних даних;
- визначення підозрілих дій.

Схема сценаріїв використання, наведена в Додатку А, показує вигляд випадків використання системи виявлення вторгнень. На рисунку 2.1 зображено схему сценаріїв використання системи.

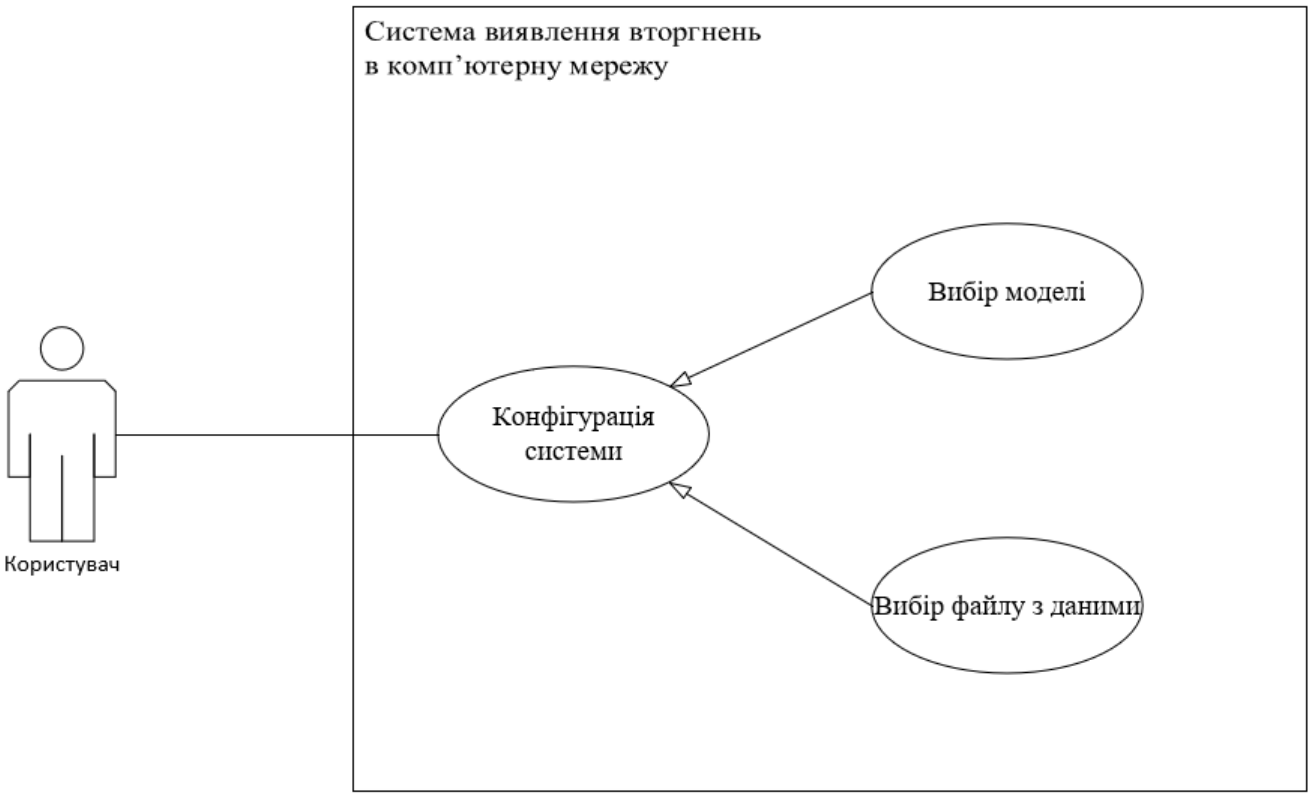

Рисунок 2.1 – Схема сценаріїв використання системи

З рисунку можна виділити основні процеси, що виконує система:

- оповіщення адміністратора про виявлені аномалії в мережі;
- передача інформації на систему виявлення вторгнень.

Система запобіганню вторгнень отримує сигнал про виявлення шкідливого трафіку в мережі. Оповіщення аномального трафіку включає в себе виявлення аномалії та її класифікації, при наявності схожих сигнатур у навченій моделі.

Послідовність кроків за якими система працює в звичайному режимі:

- крок 1: система зчитує логи з журналу аналізатора zeek;
- крок 2: система порівнює отриманий трафік з базою сигнатур доброякісного трафіку. при виявленні відхилення від базового трафіку система переходить до кроку 3:
- крок 3: запис та виведення інформації про виявлене відхилення, а саме передача його характеристик: звідки було здійснено атаку - ip-адреса джерела, порт джерела, id пакетів; та куди було здійснено атаку - ip-адреса призначення, порт призначення. та запис характеристик в журнал аномалій для подальшого аналізу, наприклад виявлення причини відхилення від звичного шаблону мережі чи виявлення типу атаки.

2.4 Висновки до розділу

В даному розділі було встановлено функціональні та нефункціональні вимоги до системи, що розробляється. Було обрано головні ідеї та функції систем, для ефективної роботи. Розглянуто сценарій використання системи, при якому вона виконує виявлення аномалії. Виявлення аномалії в трафіку дозволяє передати інформацію необхідну системі запобіганню вторгнень та передачі інформації про неї адміністраторам мережі, для вживання заходів для запобіганню викраденню даних чи управління мережею.

3 РОЗРОБКА СИСТЕМИ

3.1 Структурна схема системи

Базуючись на інформації отриманій в попередніх розділах було вирішено розробляти систему виявлення вторгнень на базі детектування аномалій, через те що система повинна працювати в умовах великої кількості невідомих атак. Структурна схема зображено на рисунку 3.1.

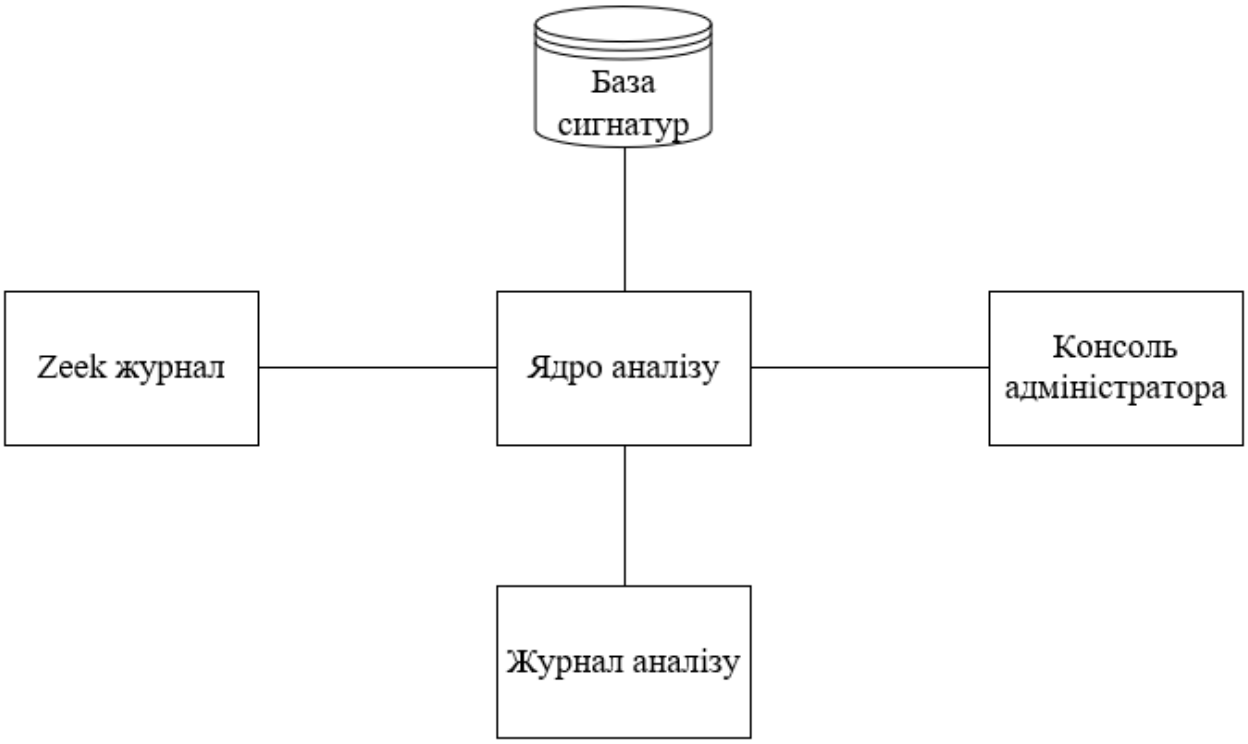

Рисунок 3.1 – Структурна схема елементів системи

Zeek монітор – це система моніторингу, яка збирає інформацію про події що відбуваються в локальному середовищі, датчики моніторингу можна розмістити в любому місці, якому необхідне пильне спостереження – наприклад це можуть бути сервери з важливими даними або спеціальні сервера HoneyPOT (рисунок 3.2), які спеціально використовуються як ціль для атак, що в свою чергу після аналізу дозволяє виявляти найбільш поширені атаки.

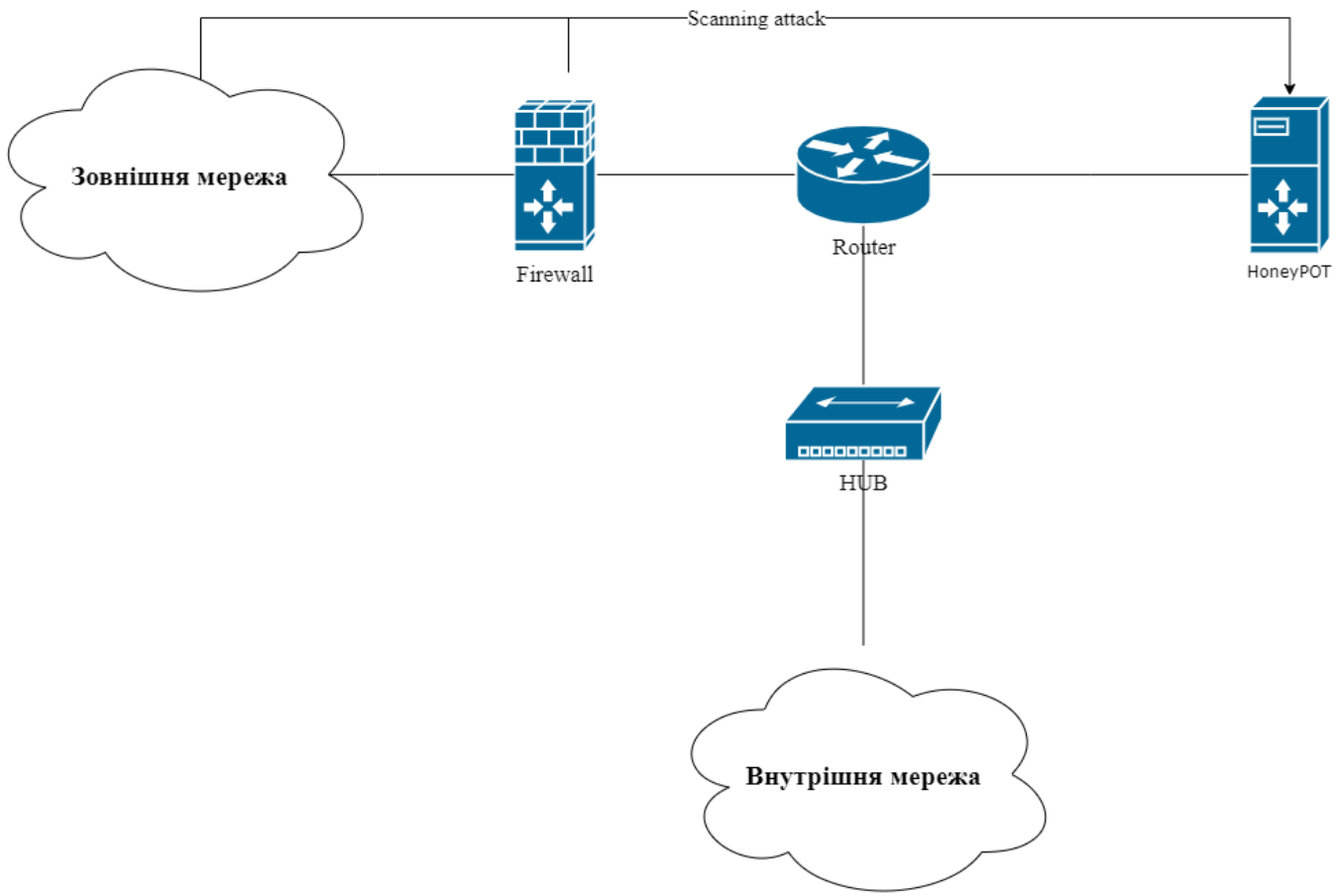

Рисунок 3.2 – Розміщення сервера HoneyPOT

IDS містить в собі всі необхідні елементи за допомогою яких виконує аналіз пакетів. В свою чергу вона тісно пов’язана з іншими елементами, з базою сигнатур в якій зберігаються дані про нормальні сценарії протікання мережевого трафіку, консоллю адміністратора в якій буде відображатись інформація про аномальні події, модуль відповідного реагування на який відправляється інформація про аномальні події.

3.2 Архітектура системи

Механізм нечіткого логічного виводу дає змогу використовувати досвід експертів, сформульований у виді нечітких предикатних правил. Що в свою чергу дозволить автоматично створювати нові правила при виявленні нових атак. В якості вхідних даних використовуються атаки описані експертами, формалізовані у виді нечітких правил.

Для IDS, що розробляється, в якості вхідних даних будуть вибрані наступні параметри:

- номер порту (хосту)
- номер порту (гостя);
- tcp-прапорці;
- статус підключення;
- втрачені байти;
- вихідна кількість пакетів;
- байти вихідного корисного навантаження;
- кінцева кількість пакетів;
- кінцева кількість байтів.

Сама система будується по структурній схемі наведеній на рисунку 3.1. Свою роботу вона починає зі зчитування журналу Zeek. Ядро аналізу виконує логічні операції та будується по структурній схемі зображеній на рисунку 3.3.

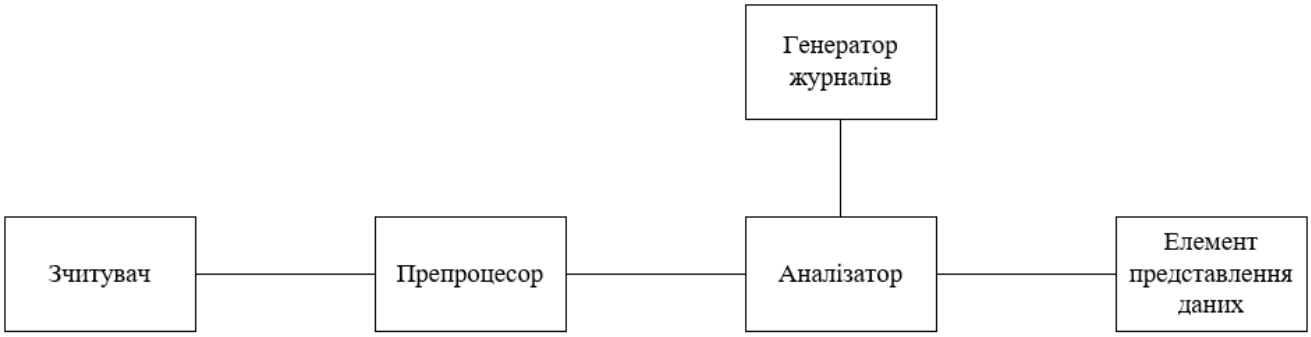

Рисунок 3.3 – Структурна схема логічних елементів

Опис логічних елементів:

- зчитувач виконує функцію зчитування даних з журналу zeek.
- препроцесор виконує функцію представлення даних у виді набору чисел, який використовує система для аналізу.
- детектор аномалії виконує порівняння з «нормальним» трафіком та виявляє до якого класу належить трафік.

– генератор оповіщень виконує оповіщення адміністратора про відхилення в мережі.

3.2.1 Zeek

Для отримання повної інформації, необхідно сканувати мережу, встановити модулі для сканування - Zeek. Система Zeek розробляється в Каліфорнійському університеті в Берклі і в даний час використовується в проектах багатьох серйозних американських компаній.

Zeek (раніше Bro) - це пасивний аналізатор мережевого трафіку. Це в першу чергу монітор безпеки, який перевіряє весь трафік за посиланням на наявність ознак підозрілої діяльності, але може також використовуватися для висвітлення багатьох різних видів поведінки мережі. Це надійний інструмент, який використовують сучасні експерти з питань мережі та безпеки. Інтерфейс Zeek зображено на рисунку 3.4.

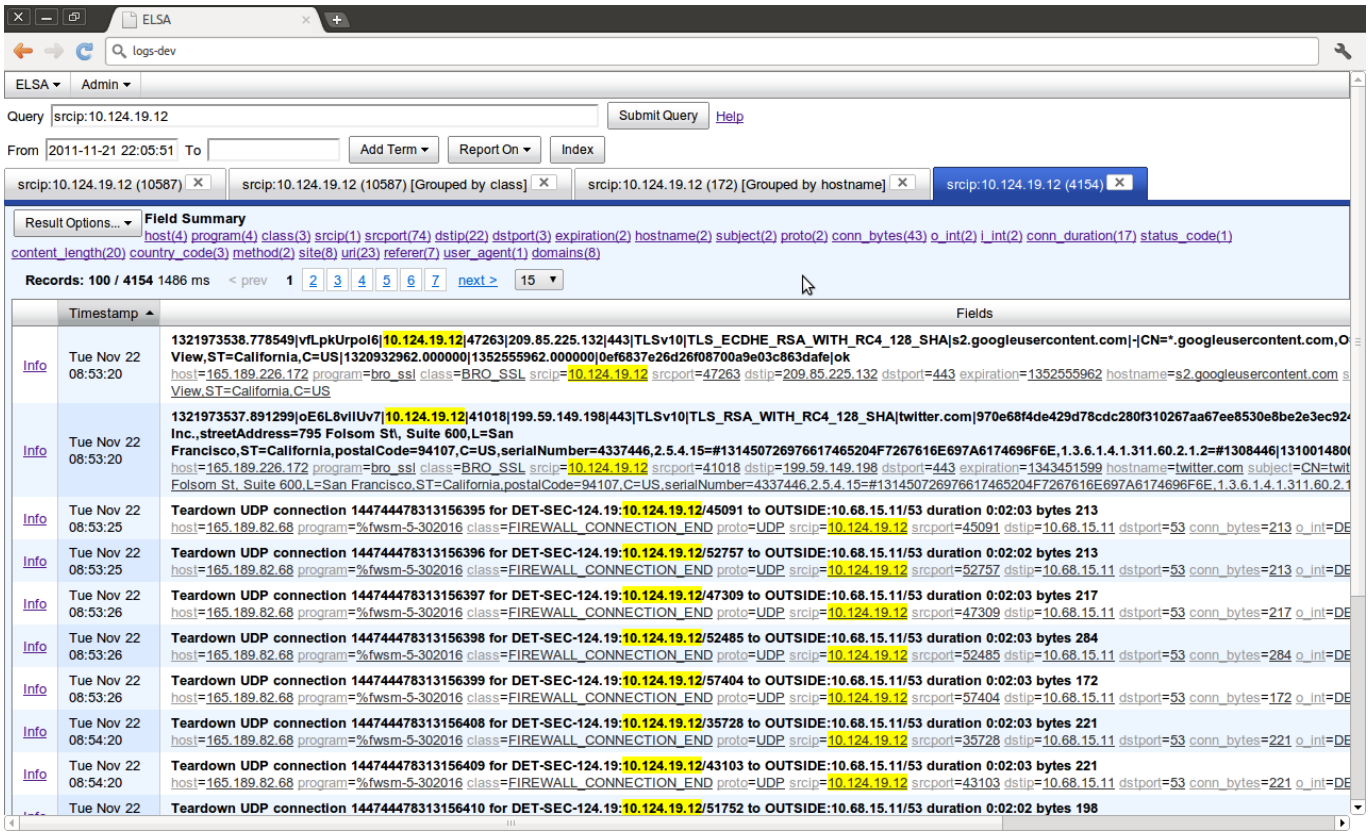

Рисунок 3.4 – Інтерфейс Zeek[6]

Як проект з відкритим кодом, Zeek підтримується керівництвом команди zeek.org і має активну спільноту, яка розширює свої можливості за допомогою мови сценаріїв, оптимізованої для аналізу мережі.

Інші системи виявлення вторгнень (IDS) або запобігання вторгненню (IPS) створюють попередження на основі конкретних правил, які називаються підписами. На відміну від цього, Zeek спокійно сидить на датчику і ненав'язливо спостерігає та перетворює необроблений мережевий трафік у всеосяжні «журнали транзакцій».

Zeek являє собою систему для створення мережевої IDS / IPS і має багаторівневу модульну структуру:

- механізм захоплення пакетів, який використовує для даних цілей libpcap (бібліотеку з відкритим вихідним кодом), що дозволяє Zeek не залежати від платформи і від нижчого мережного рівня. У цьому функціонал Zeek може замінювати відомі сніфери, наприклад, Wireshark, виділяючи і аналізуючи тільки необхідний трафік;

- механізм подій (EventEngine) перетворює отримані послідовності пакетів в первинні події. Події ці відображають базові відомості про мережеву активність. Наприклад, кожен HTTP-запит породжує відповідна подія, яка описує:

  - адресу;
  - порт;
  - запитуваний URL;
  - версію протоколу HTTP.

Цей механізм, однак, не приймає ніяких рішень щодо оцінки події - тобто на даному рівні невідомо, шкідлива вона чи ні;

- верхній рівень, інтерпретатор скриптів (PolicyScriptInterpreter): кожна реакція на будь-яку подію реєструється його оброблювачем, відповідним певному скрипту. Події ставляться в чергу FIFO. Скрипти ж визначають дії, що використовуються для виявлення шкідливого трафіку, а також політику, яка застосовується при його виявленні, і пишуться на власній скриптовій мові Zeek(Bro).

Особливістю даної системи є адаптивність до постійних змін в середовищі. Тому в розвиток IDS ставиться задача створення алгоритму самонавчання алгоритму.

При аналізі відомих IDS були виділені головні направлення розвитку:

– створення і розвиток методів та засобів виявлення атак;

– покращення способів реагування на виявлені атаки;

Найбільш підходящим інтелектуальним засобом, для вирішення задач класифікації є нейронні мережі, що в свою чергу припускає використання апарату нечіткої логіки. Перевагою такого підходу є адаптивність, інформаційна захищеність, здатність виділити нові інформаційні знання.

В своєму початковому виді дані знаходяться в звичайному текстовому файлові, який не підходить для аналізу та потребує перенесення в датафрейм. Фрагмент початкового виду мережевого трафіку зображено на рисунку 3.3.

_bytes	resp_bytes	conn_state	local_orig	local_resp	missed_bytes	history	orig_pkts
count	string	count	count	count	count	set[string]	
0	0	S0	-	-	0	S	3	180	0	0	-
-	S0	-	-	0	S	1	60	0	0	-	
-	S0	-	-	0	S	1	60	0	0	-	
149	128252	SF	-	-	2896	ShADadttcfF	94	5525	96	139044	-
0	0	S0	-	-	0	S	3	180	0	0	-
151	128348	SF	-	-	5792	ShADadttcfF	96	5699	92	133140	-
148	91961	SF	-	-	2896	ShADadtcfF	67	4148	65	92453	-
148	99303	SF	-	-	5792	ShADadtcfF	75	4412	69	97107	-
0	0	S0	-	-	0	S	3	180	0	0	-
-	S0	-	-	0	S	1	60	0	0	-	
-	S0	-	-	0	S	1	60	0	0	-	
48	48	SF	-	-	0	Dd	1	76	1	76	-
-	S0	-	-	0	S	1	60	0	0	-	
-	S0	-	-	0	S	1	60	0	0	-	
48	48	SF	-	-	0	Dd	1	76	1	76	-
0	0	S0	-	-	0	S	3	180	0	0	-
-	S0	-	-	0	S	1	60	0	0	-	
-	S0	-	-	0	S	1	60	0	0	-	
-	S0	-	-	0	D	1	76	0	0	-	

Рисунок 3.5 – фрагмент початкового вигляду мережевого трафіку в журналі Zeek

3.5 Мережевий трафік

Мережевий трафік складається з пакетів, що надійшли від джерела до точки призначення. Мережева архітектура розділена на шари з різними протоколами, які працюють на кожному шарі для виконання певного спектру функцій. OSI (Open Systems Service) – модель описує сім рівнів, від фізичного, що представляє собою

кабель або бездротовий носій, до прикладного рівня з користувацьким інтерфейсом. Найбільш актуальним для даної роботи є четвертий рівень – транспортний. Найбільш часто використовуваними протоколами на рівні 4 є TCP (Transport Control Protocol) та UDP (User Datagram Protocol)[7].

Більшість пакетів, які передаються через транспортний рівень до рівня програми, інкапсулюються або в сегмент TCP , або дейтаграму UDP. Це означає, що TCP і UDP є посередниками більшості протоколів на названих вище рівнях. Тому, для надавання характеристики трафіку, достатньо поглянути тільки на протоколи TCP і UDP. Набори даних, які використовуються, містять шкідливий трафік, що складається з TCP, UDP, а також деяких пакетів ICMP. ICMP (Internet Control Message Protocol) працює на рівні 2, який також можна прийняти до уваги для характеристики процесу.

3.6 Характеристика трафіку на основі потоку

Потік тут визначається як потік пакетів, який має ідентифікатор потоку, що характеризується ознаками: Source IP, Destination IP Source Port, Destination Port і протоколу.

Потоки є двосторонніми, що означає, що перший пакет потоку визначає, який напрямок є прямим (від джерела до пункту призначення). Відповідні пакети відправляються в зворотному напрямку (від пункту призначення до джерела). Потоки відповідають TCP-з'єднанню і UDP-потокам на транспортному рівні OSI та мають максимальну тривалість (тайм-аут). Тайм-аут означає, що після певного порогу поточний стан потоку реєструється і аналізується. Для довгих сполук або потоків це означає, що вони поділяються на декілька більш дрібних потоків.

3.6.1 Характеристики потоку

З потоку можна виділити основні характеристики:

- час прибуття пакету – час між двома пакетами;
- зворотній час прибуття – час між двома пакетами у зворотньому напрямку;

- час прибуття потоку – час між двома пакетами в будь-якому напрямку;
- активність – тривалість відправки пакетів перед переходом в режим очікування;
- холостий хід – тривалість бездіяльності перед повторною відправкою пакетів;
- потік байтів в секунду – кількість байтів, що відправляються в секунду в будь-якому напрямку;
- потік пакетів в секунду – кількість пакетів, що відправляються в секунду в будь-якому напрямку;
- тривалість – час між першим і останнім пакетом потоку.

3.6.2 Переваги потокових характеристик

Перевірка пакетів полягає у перевірці даних корисного навантаження кожного мережевого пакету для характеристики трафіку. Це процес який використовує велику кількість ресурсів, та не завжди можливий, так як корисне навантаження може бути зашифроване. Потокова характеристика набагато більш ефективна, але з іншого боку менш точна, ніж перевірка пакетів[8]. Для навчання моделі на потокових характеристиках необхідно було обрано бібліотеку Scikit-Learn.

3.7 Scikit-learn

Scikit-learn (sklearn або scikits-learn) – це безкоштовна програмна бібліотека машинного навчання для мови програмування Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг, і працює у зв'язці з бібліотеками NumPy та SciPy[9]. Scikit-learn є однією з найбільш популярних бібліотек машинного навчання. Логотип Scikit-Learn зображено на рисунку 3.6.

3.8 Загальна інформація

Для початку система машинного навчання або середовище приймає вхідні та вихідні дані. Елементи, що формують машинне навчання зображено на рисунку 3.7.

Вхідні дані а системі машинного навчання часто називають «ознаками». В машинному навчанні та розпізнаванні образів ознака — це окрема властивість або характеристика спостережуваного явища, яку можливо виміряти. Обрання інформативних, розрізнювальних і незалежних ознак є ключовим кроком алгоритмів розпізнавання образів, класифікації та регресії. Ознаки є зазвичай числовими, але в синтаксичному розпізнаванні образів використовують і структуровані ознаки, такі як стрічки та графи. Поняття «ознака» є пов'язаним із поняттям описової змінної, що застосовують у таких статистичних методиках як лінійна регресія.

Ознаки такі ж, як і змінні в науковому експерименті, що являють собою характеристики спостережуваного явища, які можна кількісно визначити або виміряти якимсь чином. Коли ці ознаки потрапляють у структуру машинного навчання, мережа намагається розпізнати відповідні закономірності між ознаками. Потім ці шаблони використовуються для генерування результатів середовища.

Виходи фреймворку часто називають "мітками", оскільки функції виводу мають певний ярлик, наданий їм мережею, певне припущення про те, до якої категорії потрапляє вихід.

У контексті машинного навчання класифікація – це тип навчання під контролем. Навчання під контролем означає, що дані, що надходять у мережу, вже позначені, а важливі особливості вже окремо розділені на окремі категорії. Після навчання мережа вже знає, які частини вхідних даних необхідні для неї, а також існує цільова або основна істина, проти якої мережа може провірити себе. Прикладом класифікації є сортування колекції різних рослин за різними категоріями.

3.9 Характеристика трафіку на основі потоку

Потік тут визначається як потік пакетів, який має ідентифікатор потоку, що характеризується ознаками: Source IP, Destination IP Source Port, Destination Port і протоколу.

Потоки є двосторонніми, що означає, що перший пакет потоку визначає, який напрямок є прямим (від джерела до пункту призначення). Відповідні пакети відправляються в зворотному напрямку (від пункту призначення до джерела). Потоки відповідають TCP-з'єднанню і UDP-потокам на транспортному рівні OSI та мають максимальну тривалість (тайм-аут). Тайм-аут означає, що після певного порогу поточний стан потоку реєструється і аналізується. Для довгих сполук або потоків це означає, що вони поділяються на декілька більш дрібних потоків.

3.9.1 Характеристики потоку

З потоку можна виділити основні ознаки:

- час прибуття пакету – час між двома пакетами;
- зворотній час прибуття – час між двома пакетами у зворотньому напрямку;
- час прибуття потоку – час між двома пакетами в будь-якому напрямку;
- активність – тривалість відправки пакетів перед переходом в режим очікування;
- холостий хід – тривалість бездіяльності перед повторною відправкою пакетів;
- потік байтів в секунду – кількість байтів, що відправляються в секунду в будь-якому напрямку;
- потік пакетів в секунду – кількість пакетів, що відправляються в секунду в будь-якому напрямку;
- тривалість – час між першим і останнім пакетом потоку.

3.9.2 Переваги потокових характеристик

Перевірка пакетів полягає у перевірці даних корисного навантаження кожного мережевого пакету для характеристики трафіку. Це процес який використовує велику кількість ресурсів, та не завжди можливий, так як корисне навантаження може бути зашифроване. Потокова характеристика набагато більш ефективна, але з іншого боку менш точна, ніж перевірка пакетів.

3.10 Набори даних для оцінки виявлення вторгнень

Для роботи було обрано набір мережевого трафіку IoT-23 з пристроїв Internet of Things (IoT).

Він має 20 захоплень зловмисного програмного забезпечення, виконаних на пристроях IoT, і 3 захоплення для доброякісного трафіку пристроїв IoT. Вперше він був опублікований у січні 2020 р., Аналіз тривав в межах з 2018 по 2019 рр. Цей мережевий трафік IoT фіксувався в лабораторії Стратосфери, група AIC, FEL, Університет CTU, Чеська Республіка. Його мета - запропонувати великий набір даних про реальні та марковані зараженнями зловмисними програмами IoT та доброякісний трафік IoT для дослідників для розробки алгоритмів машинного навчання. Цей набір даних та його дослідження фінансуються компанією Avast Software, Прага.

Набір даних IoT-23 складається з двадцяти трьох знімків (так звані сценарії) різного мережевого трафіку IoT. Ці сценарії поділяються на двадцять мережевих захоплень (файлів pcap) із заражених пристроїв IoT (на яких буде вказано ім'я зразка шкідливого програмного забезпечення, що виконується в кожному сценарії), та трьох мережевих захоплень мережевого трафіку реальних пристроїв IoT (які мають назву пристроїв, де трафік був захоплений). У кожному зловмисному сценарії запускався конкретний зразок шкідливого програмного забезпечення в Raspberry Pi, який використовував кілька протоколів та виконував різні дії. Таблиця 3.1 показує

характеристики сценаріїв бот-мереж IoT, а таблиця 3.2 - протоколи, які були знайдені в кожному захопленні мережевого трафіку.

У таблиці 3.1 наведено номер сценарію, ім'я набору даних, тривалість у годинах, кількість пакетів, кількість потоків ідентифікаторів Zeek у файлі conn.log (отримано шляхом запуску середовища аналізу мережі Zeek на вихідному pcap файл), розмір оригінального файлу pcap та можливу назву зразка шкідливого програмного забезпечення, що використовується для зараження пристрою.

Захоплення шкідливого програмного забезпечення виконується протягом тривалого періоду часу. Через великий обсяг трафіку, що генерується кожною інфекцією, файли pcap чередуються кожні 24 години. Однак у деяких випадках pcap-файл зростав занадто швидко, і було прийнято рішення зупинити збір до завершення двадцяти чотирьох годин. З цієї причини деякі захоплення відрізняються кількістю годин.

Таблиця 3.1 - Характеристики сценаріїв бот-мереж

№	Номер датасету	Тривалість (годин)	Кількість пакетів	Zeek потоки	Розмір файлу pcap	Ім’я
1	34-1	24	233,000	23,146		Mirai
2	43-1	1	82,000,000	67,321,810	121 Мб	Mirai
3	44-1	2	1,309,000	238	6 Гб	Mirai
4	49-1	8	18,000,000	5,410,562	1.7 Гб	Mirai
5	52-1	24	64,000,000	19,781,379	1.3 Гб	Mirai
6	20-1	24	50,000	3,210	4.6 Гб	Torii
7	21-1	24	50,000	3,287	3.9 Мб	Torii
8	42-1	8	24,000	4,427	3.9 Мб	Trojan
10	17-1	24	109,000,000	54,659,864	7.8 Гб	Kenjiro
11	36-1	24	13,000,000	13,645,107	992 Мб	Okiru
12	33-1	24	54,000,000	54,454,592	3.9 Гб	Kenjiro

Продовження таблиці 3.1

№	Номер датасету	Тривалість (годин)	Кількість пакетів	Zeek потоки	Розмір файлу pcap	Ім’я
13	8-1	24	23,000	10,404	2.1 Мб	Hakai
14	35-1	24	46,000,000	10,447,796	3.6 Гб	Mirai
15	48-1	24	13,000,000	3,394,347	1.2 Гб	Mirai
16	39-1	7	73,000,000	73,568,982	5.3 Гб	IRCBot
7	7-1	24	11,000,000	11,454,723	897 Мб	Linux.Mirai
18	9-1	24	6,437,000	6,378,294	472 Мб	Linux.Hajime
19	3-1	36	496,000	156,104	56 Мб	Muhstik
20	1-1	112	1,686,000	1,008,749	140 Мб	Hide and Seek

Щоб отримати додаткові дані, що до мережевого трафіку, генерованого кожним зараженим пристроєм, було використано передбачення протоколу рівня додатків від Zeek для фільтрації та узагальнення цієї інформації. У таблиці 3.2 ця інформація узагальнена, тут включено для кожного сценарію назву набору даних, кількість потоків для наступних протоколів: HTTP, DNS, DHCP, SSL та IRC. деякі протоколи не були розпізнані Zeek, є стовпець, де всі ці потоки кількісно виражені.

Таблиця 3.2 - протоколи рівня додатків, виявлених Zeek у зловмисних сценаріях.

№	Номер датасету	HTTP	DNS	DHCP	SSL	SSH	IRC	Не виявлено Zeek	Ім’я
1	34-1	12	192	2	-	-	1,641	21,298	Mirai
2	43-1	16	204	-	-	-	-	67,321,589	Mirai
3	44-1	11	-	-	-	-	-	226	Mirai

Продовження таблиці 3.2

№	Номер датасету	HTTP	DNS	DHCP	SSL	SSH	IRC	Не виявлено Zeek	Ім’я
4	49-1	19	6	1	-	-	-	5,410,535	Mirai
5	52-1	14	4	1	-	-	-	19,781,359	Mirai
6	20-1	-	592	-	-	-	-	2,617	Torii
7	21-1	-	1,924	-	-	-	-	1,362	Torii
8	42-1	33	1,680	1	2	-	-	2,710	Trojan
9	17-1	-	-	2	-	-	-	3,581,026	Gagfyt
10	36-1	4	11,902	-	-	-	-	54,647,949	Kenjiro
11	33-1	-	751	2	-	-	-	13,644,345	Okiru
12	8-1	228	80	2	-	-	-	54,454,281	Kenjiro
13	35-1	-	-	-	-	-	-	10,403	Hakai
14	48-1	36	1,479	2	9	-	-	10,446,261	Mirai
15	39-1	11	2	-	-	-	-	3,394,325	Mirai
16	7-1	14	2,308	-	6	538	914	73,565,201	IRCBot
17	9-1	-	7	1	-	-	-	11,454,706	Linux.Mirai
18	3-1	55	1,162	-	-	-	-	6,377,076	Linux.Hajime
19	1-1	-	1	3	-	5,898	6	150,195	Muhstik
20		3,238	1	1	-	-	-	1,005,507	Hide and Seek

Мережевий трафік, зафіксований для доброякісних сценаріїв, був отриманий шляхом захоплення мережевого трафіку трьох різних пристроїв IoT: розумна

світлодіодна лампа Philips HUE (Рисунок 3.5), домашній інтелектуальний особистий помічник Amazon Echo (Рисунок 3.6) та розумний дверний замок Somfy (Рисунок 3.7).

Рисунок 3.8 - Розумна світлодіодна лампа Philips HUE

Рисунок 3.9 - Домашній інтелектуальний особистий помічник Amazon Echo

Рисунок 3.10 - Розумний дверний замок Somfy

Важливо зазначити, що ці три пристрої IoT є справжнім обладнанням, а не імітуються. Це дозволяє фіксувати та аналізувати реальну поведінку мережі. Як зловмисні, так і доброякісні сценарії працюють у контрольованому мережевому середовищі з нестримним підключенням до Інтернету, як будь-який інший реальний пристрій IoT. У таблиці 3 наведено мережеві дані доброякісних сценаріїв IoT, а в таблиці 4 - протоколи, знайдені в кожному захопленні мережі.

3.10.1 Набори даних доброякісних потоків

Ці сценарії були створені шляхом збору даних про мережевий трафік незаражених реальних пристроїв IoT. Доброякісні сценарії отримуються шляхом захоплення мережевого трафіку реальних пристроїв IoT. Важливо побачити і зрозуміти, як справжні пристрої IoT поводяться в мережі, коли вони не заражені. Це дозволить виявити зміну поведінки, коли вони заражені шкідливим програмним забезпеченням або піддаються атаці. У таблиці 3.3 наведено мережеві дані для кожного зі сценаріїв

роботи, включаючи інформацію щодо тривалості, кількості пакетів, кількості потоків Zeek, файлу pcap та імені пристрою. У таблиці 3.4 наведено протоколи виявленого рівня додатків для кожного з доброякісних сценаріїв.

Таблиця 3.3 – Доброякісні сценарії

№	Назва датасету	Тривалість (годин)	Кількість пакетів	Zeek потоки	Розмір файлу pcap
1	Honeypot-Capture-7-1	1.4	8,276	139	2.094 Кб
2	Honeypot-Capture-4-1	24	21,000	461	4,594 Кб
3	Honeypot-Capture-5-1	5.4	398,000	1,383	381 Мб

Таблиця 3.4 – Розбивка протоколів рівня додатків, виявлених у доброякісних сценаріях.

№	Назва датасету	HTTP	DNS	DHCP	SSL	SSH	IRC	Не виявлено Zeek
1	Honeypot-Capture-7-1	-	54	15	1	-	-	60
2	Honeypot-Capture-4-1	54	191	2	-	-	-	205
3	Honeypot-Capture-5-1	157	521	156	86	-	-	454

3.10.2 Шкідливий мережевий трафік

Шкідливий мережевий трафік тут визначається як будь-трафік, отриманий в результаті атаки з наміром заподіяти шкоду або вторгнення в комп'ютерну систему або мережу. Набори даних, вибрані для цього проекту містять наступні класи шкідливих програм рух:

- attack: ця мітка вказує на те, що з зараженого пристрою на інший хост відбувся якийсь тип атаки. таким чином позначається атака на будь-який потік, який, аналізуючи свою корисну навантаження та поведінку, намагається скористатися якоюсь вразливою службою;
- доброзичливий (benign): ця мітка вказує на відсутність підозрілих зловмисних дій у з'єднаннях;
- c&c: ця мітка вказує на те, що заражений пристрій було підключено до сервера cc. ця активність була виявлена під час аналізу мережевого захоплення зловмисного програмного забезпечення, оскільки зв'язки з підозрілим сервером періодичні, або наш заражений пристрій завантажує з нього деякі двійкові файли, або з нього надходять і розшифровуються деякі подібні або декодовані замовлення irc;
- ddos: ця мітка вказує на те, що заражений пристрій виконує розподілену атаку відмови в обслуговуванні. ці потоки трафіку виявляються як частина ddos-атаки через кількість потоків, спрямованих на ту саму ip-адресу;
- filedownload: ця мітка вказує на те, що файл завантажується на заражений пристрій. це виявляється за допомогою фільтрації з'єднань із байтами відповіді більше 3 кб або 5 кб, зазвичай це поєднується з деяким відомим підозрілим портом призначення або ip-адресою призначення, про які відомо, що це сервер c&c;
- heartbeat: ця мітка вказує на те, що пакети, надіслані за цим підключенням, використовуються для збереження відстеження на зараженому хості сервером c&c. це було виявлено за допомогою фільтрації з'єднань з байтами

відповіді нижче 1b та періодичних подібних з'єднань, як правило, це поєднується з деяким відомим підозрілим портом призначення або ip-адресою призначення, про які відомо, що це сервер c&c;

- mirai: ця мітка вказує на те, що з'єднання мають характеристики бот-мережі mirai. ця мітка додається, коли потоки мають подібні шаблони, як найпоширеніші атаки mirai;
- okiru: ця мітка вказує на те, що з'єднання мають характеристики бот-мережі okiru. це рішення щодо маркування було прийнято з тими ж параметрами, що і у mirai, але з тією різницею, що це сімейство ботнетів зустрічається рідше;
- partofahorizontalportscan: ця мітка вказує на те, що з'єднання використовуються для горизонтального сканування портів для збору інформації для подальших атак. щоб поставити ці мітки, ми покладаємось на шаблон, за яким з'єднання використовували один і той же порт, однакову кількість переданих байтів і кілька різних ip-адрес призначення;
- torii: ця мітка вказує на те, що з'єднання мають характеристики бот-мережі torii. це рішення щодо маркування було прийнято з тими ж параметрами, що і у mirai, але з тією різницею, що це сімейство ботнетів зустрічається рідше[10].

3.11 Машинне навчання

Машинне навчання (ML)-це підкатегорія штучного інтелекту , заснована на статистичних методах навчання, які використовуються для розуміння і прогнозування даних. Машинне навчання можна розділити на дві категорії: контрольоване навчання і неконтрольоване навчання. Категорії навчання зображено на рисунку 3.1.

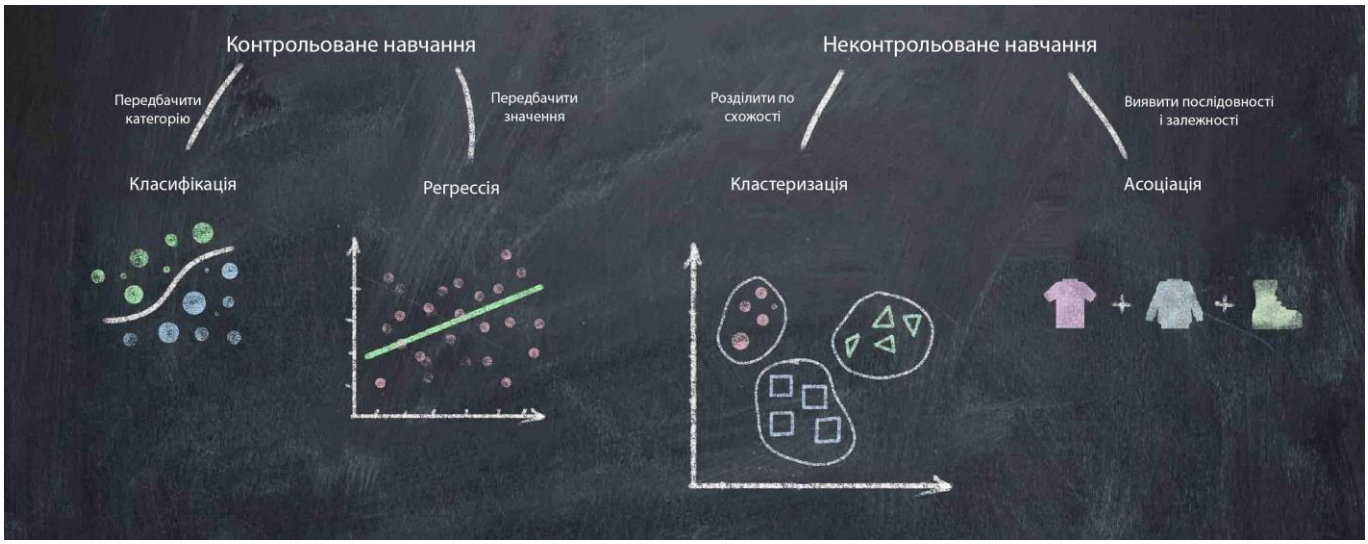

Рисунок 3.11 – Категорії машинного навчання

При контрольованому навчанні вхідні дані і позначені вихідні дані використовуються для навчання статистичної моделі з метою прогнозування виведення нових вхідних даних. Найпростіший математичний приклад – це, мабуть, метод найменших квадратів, який справляє пряму лінію (модель) з набору координат x і Y. Тоді для будь-якого нового значення x (вхід) лінія буде пророкувати значення y (вихід).

Неконтрольоване навчання може бути використано, коли в наборі немає позначених вихідних даних. Неконтрольоване навчання використовується для аналізу взаємозв'язків між точками вхідних даних. Одним з таких підходів є кластеризація, яка групує разом точки вхідних даних на основі подібності їх ознак [11].

Оскільки у наборі використовуються позначені набори даних і мета полягає в тому, щоб зробити прогнози про шкідливий трафік, виконувана тут ML потрапляє в категорію контрольованого навчання. Залежно від наборів даних, різні алгоритми ML мають різну точність і продуктивність. Складність використання ML полягає в пошуку найбільш відповідних функцій і пошуку оптимального алгоритму і параметрів.

Побудова ML-моделі з набору даних називається навчанням, рисунок 3.12.

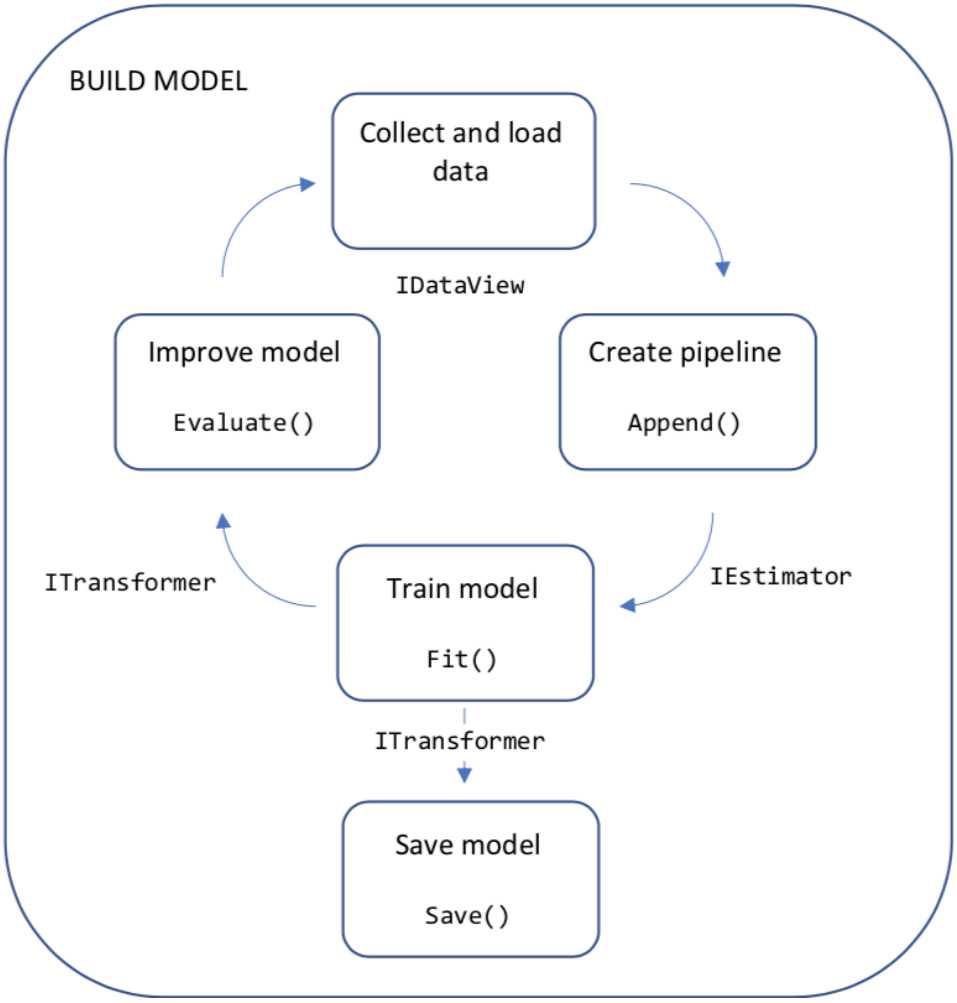

Рисунок 3.12 – Побудова ML моделі[12]

Навчання виконується на підмножині набору даних, званому навчальними даними. Інша частина набору даних використовується для перевірки або тестування моделі. Поділ зазвичай виконується з більшістю даних, використовуваних для навчання, наприклад 80/20 (%) для навчання/тестування.

При навчанні моделі важливим завданням є знаходження оптимального компромісу між зміщенням і дисперсією. Дисперсія – це міра того, наскільки сильно змінюється модель при використанні нових навчальних даних, тобто гнучкість або складність моделі. А упередженість – це помилка, яка виникає при застосуванні простої моделі до реальної проблеми. Оптимальний компроміс між зміщенням і дисперсією має найменше можливе зміщення при збереженні найменшій можливій дисперсії. Ці дві величини корелюють, оскільки проста модель з високим зміщенням (наприклад, метод найменших квадратів) має низьку дисперсію, а гнучка модель з

високою дисперсією-низьку дисперсію. Дисперсія зі зміщенням зображена на рисунку 3.8.

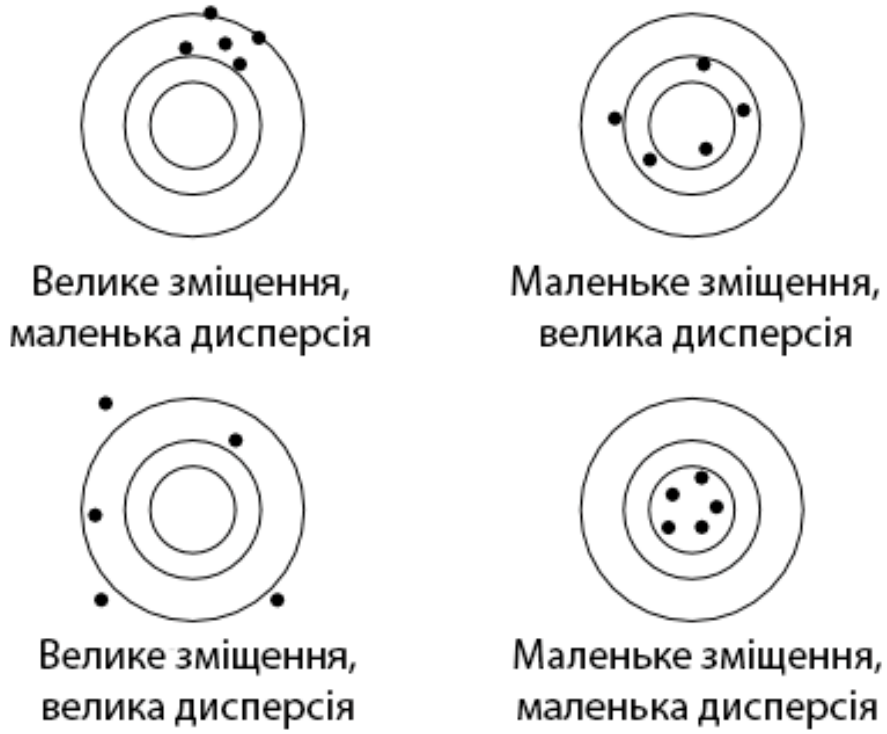

Рисунок 3.13 – Дисперсія зі зміщенням

Висока дисперсія може призвести до важливої проблеми, званої переоснащенням. Це означає, що модель дуже уважно відстежує помилки в навчальних даних, знаходячи патерни, які не повинні існувати, що призводить до надмірно складної моделі і поганої точності.

Мітки в наборі даних можуть бути як кількісними, так і якісними. Якісні мітки-це дискретні значення, що належать класу, в цьому проекті мітки є якісними, так як мітки шкідливого трафіку є іменами атак.

Кількісні мітки з іншого боку, є неперервними значеннями. Дані з якісними мітками вимагають алгоритмів класифікації, в той час як дані з кількісними мітками вимагають алгоритмів регресії. Характеристики даних можуть бути як якісними, так і кількісними, але це менш важливо при виборі правильного алгоритму.

Об'єкти в наборі даних не завжди корелюють з вихідними даними в мітці. Такі особливості слід виключити, оскільки вони можуть призвести до надто складної моделі. Вибір об'єкта може бути виконаний для видалення об'єктів з низькою кореляцією до виходу.

3.11.1 Методи машинного навчання

Для ефективного навчання та роботи системи виявлення вторгнень, необхідно обрати метод який буде задовольняти певним критеріям, а саме:

- точність навчання;
- час навчання.

Оптимальними методами з контрольованим навчанням для задоволення даних критеріїв було обрано:

- метод опорних векторів (SVM);
- метод k-найближчих сусідів (KNN);
- метод випадковий ліс (RF).

Для неконтрольованого навчання було обрано методи

- аналіз головних компонентів(PCA)
- локальний фактор відхилення(LOF)

3.11.1.1 Метод опорних векторів

В машинному навчанні метод опорних векторів — це метод аналізу даних для класифікації та регресійного аналізу за допомогою моделей з керованим навчанням з пов'язаними алгоритмами навчання, які називаються опорно-векторними машинами або опорно-векторними машинами (SVM)[13]. Для заданого набору тренувальних зразків, кожен із яких помічено як належний до однієї чи іншої з двох категорій, алгоритм тренування ОВМ будує модель, яка відносить нові зразки до однієї чи іншої категорії, роблячи це ефективним бінарним лінійним класифікатором. Модель опорних векторів є представленням зразків як точок у просторі, відображених таким

чином, що зразки з окремих категорій розділено чистою широкою прогалиною, зображено на рисунку 3.14. Нові зразки тоді відображуються до цього ж простору, й робиться передбачення про їхню належність до категорії на основі того, на який бік прогалини вони потрапляють.

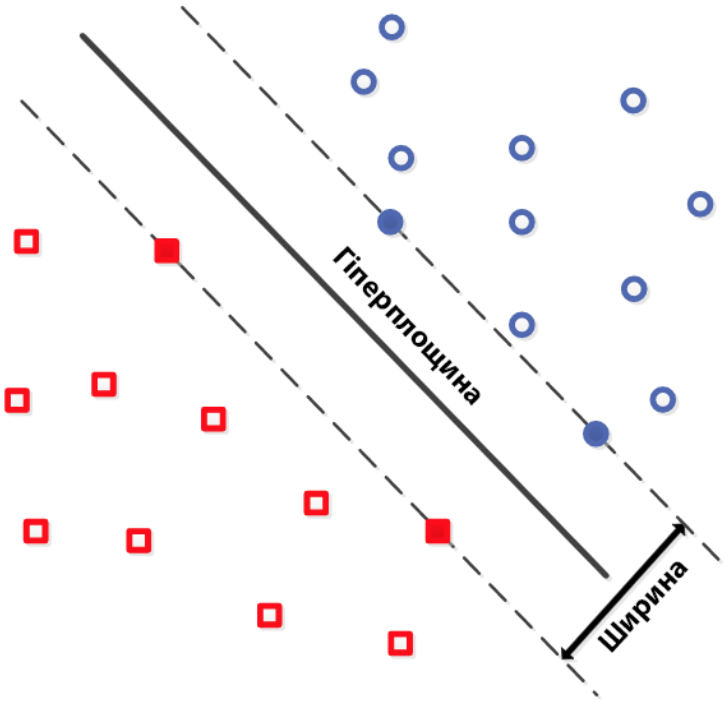

Рисунок 3.14 - SVM

На додачу до виконання лінійної класифікації, SVM можуть ефективно виконувати нелінійну класифікацію при застосуванні так званого ядрового трюку, неявно відображуючи свої входи до просторів ознак високої вимірності.

Коли дані не є міченими, кероване навчання є неможливим, і виникає необхідність у спонтанному навчанні, яке намагається знайти природне кластерування даних на групи, а потім відображувати нові дані на ці сформовані групи. Алгоритм кластерування, який забезпечує вдосконалення опорно-векторним машинам, називається опорно-векторним кластеруванням[14] і часто використовується в промислових застосуваннях, коли дані або не є міченими, або коли лише деякі дані є міченими як попередня обробка перед проходом класифікації.

SVM можуть застосовуватися для розв'язання різноманітних практичних задач:

- SVM є корисними для класифікації текстів та гіпертекстів, оскільки їхнє застосування може значно знижувати потребу в мічених тренувальних зразках як у стандартній індуктивній, так і в трансдуктивній постановках.
- із застосуванням SVM може виконуватися й класифікація зображень. експериментальні результати показують, що SVM можуть досягати значно вищої точності пошуку, ніж традиційні схеми уточнення запиту, всього лише після трьох-чотирьох раундів зворотного зв'язку про відповідність. це є вірним і для систем сегментування зображень, включно з тими, які використовують видозмінену версію SVM, яка застосовує привілейований підхід, запропонований вапником[15;16].
- за допомогою SVM може здійснюватися розпізнавання рукописних символів.
- алгоритм SVM широко застосовується в біологічних та інших науках. їх було використано для класифікації білків з правильною класифікацією до 90 % складу. як механізм інтерпретування моделей SVM було запропоновано перестановну перевірку на основі вагових коефіцієнтів SVM[17;18]. вагові коефіцієнти опорно-векторних машин використовувалися для інтерпретування моделей SVM і в минулому[19]. ретроспективне інтерпретування моделей опорно-векторних машин з метою розпізнавання ознак, які використовує модель для здійснення передбачень, є відносно новою областю досліджень з особливим значенням у біологічних науках.

Потенційні недоліки SVM включають наступні аспекти:

- вимагають повністю мічених вхідних даних;
- некалібровані ймовірності приналежності до класів;
- SVM застосовні напряму лише до двокласових задач – мають застосовуватися алгоритми, які зводять багатокласову задачу до кількох бінарних задач;
- інтерпретувати параметри розв'язаної моделі важко.

Переваги методу опорних векторів:

- висока швидкодія;
- єдино вірне рішення;
- знаходження максимальної ширини смуги поділу, внаслідок чого виробляється впевнена класифікація[20].

3.11.1.2 Метод k-найближчих сусідів

Метод k-найближчих сусідів — метричний алгоритм для автоматичної класифікації об'єктів. Основним принципом методу найближчих сусідів є те, що об'єкт присвоюється тому класу, який є найбільш поширеним серед сусідів даного елемента. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і, виходячи з ключового для даного методу значення k, вираховується, який клас є найчисленнішим серед них, вибір сусідів зображено на рисунку 3.15. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей)[21].

Передбачається, що існує певний набір об'єктів з уже наявною класифікацією.

- у класифікації k-NN результатом є приналежність до класу. об'єкт класифікується за допомогою множинного голосування сусідів, при цьому об'єкт присвоюється класу, найпоширенішому серед його k найближчих сусідів (k - ціле додатне число , як правило, невелике). якщо k = 1, тоді об’єкт просто присвоюється класу цього найближчого сусіда;
- у регресії k-NN вихідним результатом є значення властивості для об'єкта. це значення є середнім значенням k найближчих сусідів[22].

k-найближчих сусідів – це тип навчання на основі екземпляру , або лінивого навчання, де функція апроксимується лише локально, а всі обчислення відкладаються до оцінки функції. Оскільки цей алгоритм покладається на відстань для класифікації, нормалізація навчальних даних може значно покращити його точність[23].

На етапі класифікації k - це визначена користувачем константа, а немічений вектор (запит чи тестова точка) класифікується шляхом присвоєння мітки, яка найчастіше зустрічається серед k- навчальних зразків, найближчих до цієї точки запиту.

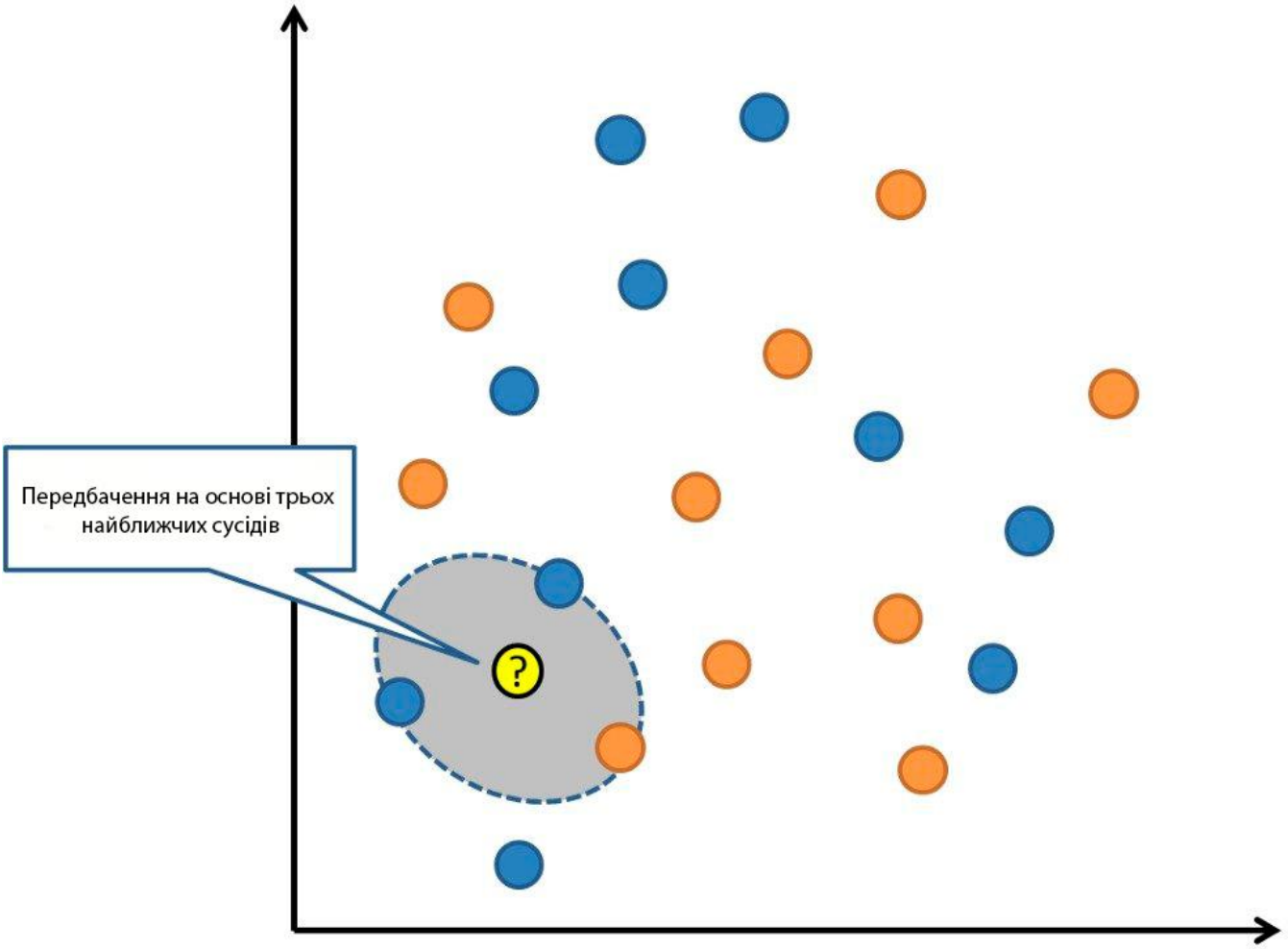

Рисунок 3.15 - Вибір сусідів

Часто використовуваною метрикою відстані для неперервних змінних є евклідова відстань. Для дискретних змінних, наприклад для класифікації тексту, може використовуватися інша метрика, така як метрика перекриття (або відстань Хеммінга). Наприклад, у контексті даних мікрочипів експресії генів, k-найближчих сусідів використовували з коефіцієнтами кореляції, такими як Пірсон та Спірмен, як метрику[24]. Часто точність класифікації k-найближчих сусідів можна значно покращити, якщо метрику відстані вивчити за допомогою спеціалізованих алгоритмів, таких як аналіз компонентів з найближчим сусідством із великим полем чи сусідством .

Недолік базової класифікації "більшості голосів" виникає, коли розподіл класів є нерівним. Тобто, приклади більш частого класу, як правило, домінують у прогнозуванні нового прикладу, оскільки вони, як правило, поширені серед k

найближчих сусідів через їх велику кількість[25]. Один з способів вирішення цієї проблеми є вага класифікації, беручи до уваги відстань від контрольної точки до кожного з його до найближчих сусідів. Клас (або значення в задачах регресії) кожної з k найближчих точок множиться на вагу, пропорційну оберненій відстані від цієї точки до контрольної точки. Іншим способом подолання перекосів є абстракція у поданні даних.

k-найближчих сусідів - це приватний випадок оцінки пропускної здатності змінної пропускної здатності "повітряної кулі" щільності ядра з рівномірним ядром[26;27].

Наївну версію алгоритму легко реалізувати, обчислюючи відстані від тестового прикладу до всіх збережених прикладів, але вона обчислювально інтенсивна для великих навчальних наборів. Використання наближеного алгоритму пошуку найближчого сусіда робить k-найближчих сусідів обчислюваним для обчислення навіть для великих наборів даних. За ці роки було запропоновано багато алгоритмів пошуку найближчих сусідів; вони, як правило, спрямовані на зменшення кількості фактично проведених оцінок відстані.

3.11.1.3 Метод випадковий ліс

Випадковий ліс — ансамблевий метод (використовує кілька навчальних алгоритмів з метою отримання кращої ефективності прогнозування) машинного навчання для класифікації, регресії та інших завдань, які оперують за допомогою побудови численних дерев прийняття рішень під час тренування моделі і продукують моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев. Недоліком є схильність до перенавчання. Приклад роботи методу зображено на рисунку 3.16.

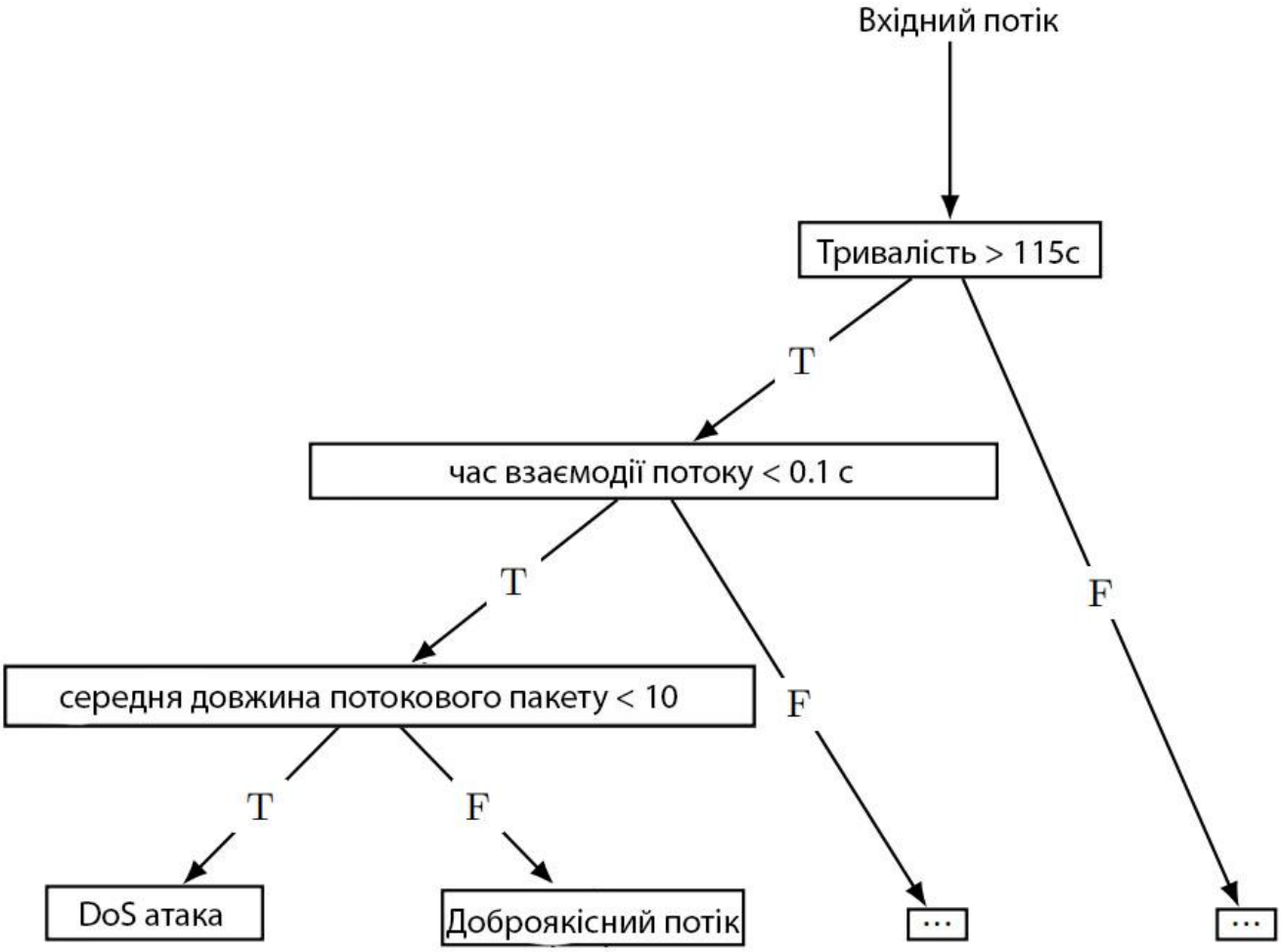

Рисунок 3.16 – Приклад роботи методу випадковий ліс

Випадкові ліси, можуть бути природним чином використані для оцінки важливості змінних в задачах регресії та класифікації. Наступний спосіб такої оцінки був описаний Брейманом.

Перший крок в оцінці важливості змінної в тренувальному наборі — тренування випадкового лісу на цьому наборі. Під час процесу побудови моделі для кожного елемента тренувального набору вважається так звана «out-of-bag» — помилка. Потім для кожної сутності така помилка опосередковується по всьому випадковому лісі.

Для того, щоб оцінити важливість j -ого параметра після тренування, значення j-ого параметра перемішуються для всіх записів тренувального набору та «out-of-bag» — помилка рахується знову. Важливість параметра оцінюється шляхом усереднення по всіх деревах різниці показників «out-of-bag» — помилок до і після перемішування

значень. При цьому значення таких помилок нормалізуються на стандартне відхилення.

Параметри вибірки, які дають більші значення, вважаються більш важливими для тренувального набору. Метод має наступний потенційний недолік — для категоріальних змінних з великою кількістю значень метод схильний вважати такі змінні більш важливими. Часткове переваження значень в цьому випадку може знижувати вплив цього ефекту[28;29]. Якщо дані містять групи корельованих ознак, що мають подібне значення для результату, то більш дрібні групи мають переваги над більшими групами[30].

Переваги методу випадковий ліс:

- здатність ефективно обробляти дані з великим числом ознак і класів;
- нечутливість до масштабування (і взагалі до будь-яких монотонних перетворень) значень ознак;
- однаково добре обробляються як безперервні, так і дискретні ознаки. існують методи побудови дерев за даними з пропущеними значеннями ознак;
- існують методи оцінювання значущості окремих ознак в моделі;
- внутрішня оцінка здатності моделі до узагальнення (тест out-of-bag);
- здатність працювати паралельно в багато потоків;
- масштабованість.

Недоліки:

- Алгоритм схильний до перенавчання на деяких завданнях, особливо з великою кількістю шумів[31];
- Великий розмір отримуваних моделей. Потрібно $O(NK)$ пам'яті для зберігання моделі, де K — число дерев.

3.12 Метод головних компонентів

Аналіз головних компонентів (PCA) може бути використаний для виявлення відхилень.

PCA - це зменшення лінійної розмірності за допомогою розкладання одиничного значення даних для проектування їх на нижчий розмірний простір, рисунок 3.17. У цій процедурі матриця коваріації даних може бути розкладена до ортогональних векторів, званих власними векторами, асоційованими з власними значеннями. У власних векторах з високими власними значеннями захоплення більшої дисперсії в даних.

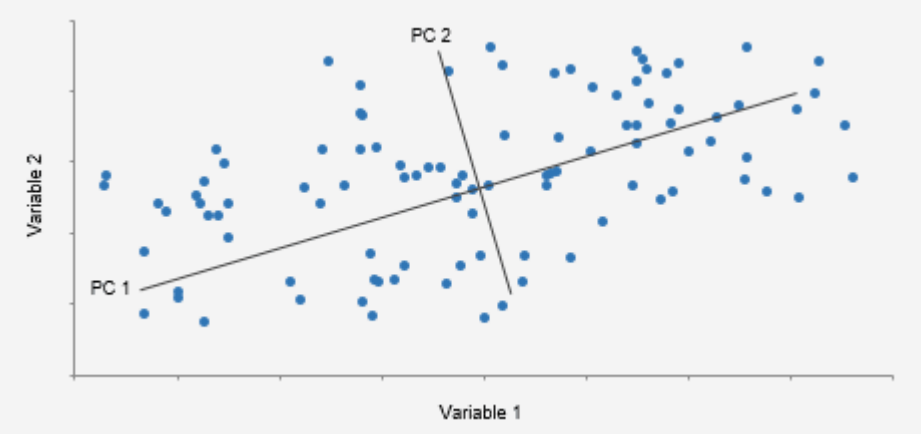

Рисунок 3.17 – аналіз головних компонентів[32]

Отже, низьковимірна гіперплощина, побудована за допомогою власних векторів, може захоплювати більшу частину дисперсії даних. Однак викиди відрізняються від звичайних точок даних, що більш очевидно на гіперплощині, побудованій власними векторами з малими власними значеннями[33].

Метод головних компонент — один із найпоширеніших методів факторного аналізу[34].

Серед інших подібних методів, що дозволяють узагальнювати значення елементарних ознак, МГК виділяється простою логічною конструкцією, і, разом з тим, на його прикладі стають зрозумілими загальна ідея й цілі чисельних методів факторного аналізу.

Метод головних компонент дає можливість за m — числом початкових ознак виділити r головних компонентів, або узагальнених ознак. Простір головних компонентів ортогональний. Математична модель методу головних компонентів

базується на логічному припущенні, що значення множини взаємозалежних ознак породжують деякий загальний результат.

3.13 Локальний фактор відхилення

Алгоритм локального фактору відхилення (LOF) - це неконтрольований метод виявлення аномалій, який обчислює локальне відхилення щільності даної точки даних щодо її сусідів, рисунок 3.18.

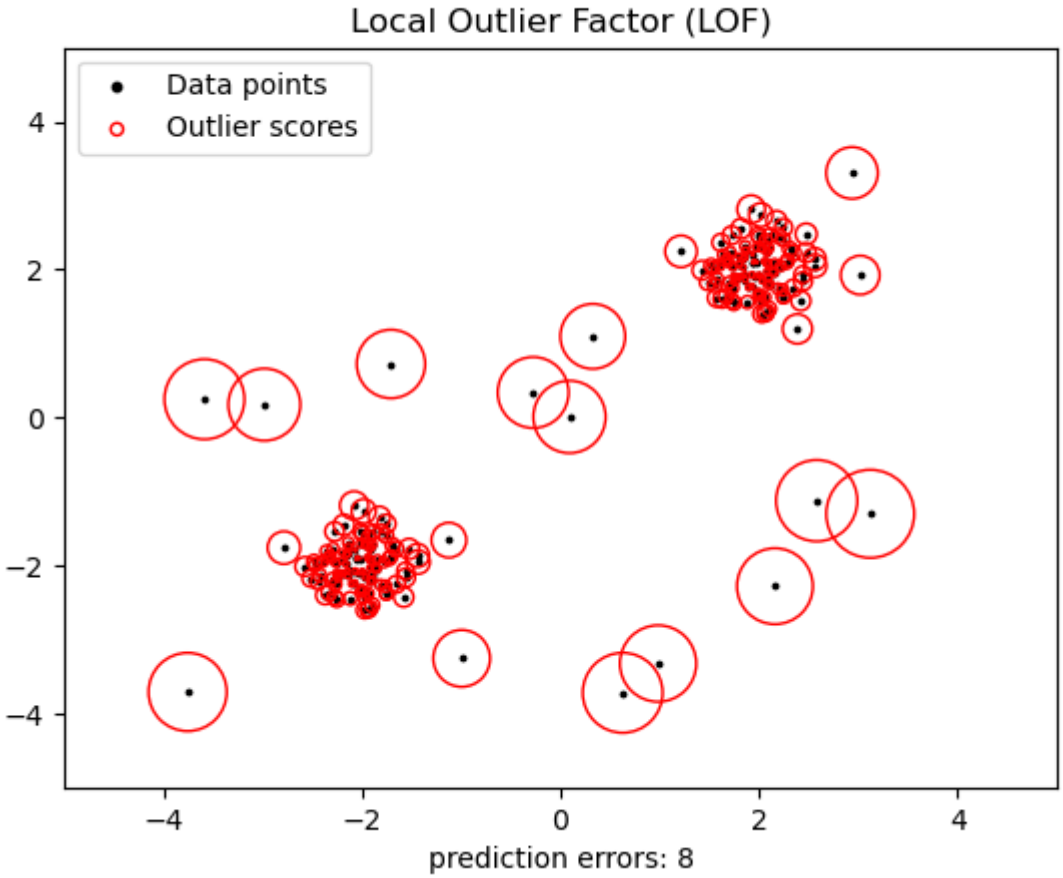

Рисунок 3.18 – локальний фактор відхилення

Він розглядає в якості викидів зразки, що мають істотно меншу щільність, ніж їх сусіди. Число розглянутих сусідів зазвичай встановлюється 1) більше мінімального числа вибірок, які повинен містити кластер, так що інші вибірки можуть бути локальними викидами щодо цього кластера, і 2) менше максимального числа

близьких вибірок, які потенційно можуть бути локальними викидами. На практиці така інформація, як правило, недоступна, і взяття n_neighbors=20, мабуть, добре працює в цілому[35].

3.14 Підготовка даних до машинного навчання

Задля ефективної роботи системи виявлення вторгнень необхідно вибрати один з методів машинного навчання, який покаже найбільшу точність у виявленні аномалій в трафіку. Для тестування методів машинного навчання було вибрано сценарій «CTU-IoT-Malware-Capture-34-1 (Mirai)», який містить 3 класифікації трафіку, в тому числі і доброякісний. Види трафіку представлено в таблиці 3.5.

Таблиця 3.5 – Види трафіку в «CTU-IoT-Malware-Capture-1-1 (Mirai)»

Назва класу трафіку	Потоки
Доброякісний	469,275
C&C	8
Port Scan	539,465

Середовищем для проведення тестування алгоритмів було обрано «Google Colab» - це безкоштовний Хмарний сервіс на основі Jupyter Notebook. Google Colab надає все необхідне для машинного навчання прямо в браузері, дає безкоштовний доступ до неймовірно швидкими GPU і TPU[36]. Для навчання моделі, дані необхідно перенести в формат, що необхідний для навчання системи, який називається «dataframe». Для цієї цілі необхідно використовувати пакет ZAT. Після перенесення даних в dataframe можна ознайомитись з елементами в ньому. Елементи продемонстровано в таблиці 3.6.

Таблиця 3.6 – Інформація в датафреймі

Змінна	Трафік	Трафік	Трафік
ts	2019-02-28 18:15:34.188184023	2019-02-28 18:15:36.206260920	2019-02-28 18:15:38.426746130
uid	C9mqzS28ln5Lv41J09	CUQLPn2ZXYFN1lNgo6	CSw2Xf4Q5Yb2mOblGe
id.orig_h	192.168.1.200	192.168.1.200	192.168.1.200
id.orig_p	52724	52726	52728
id.resp_h	167.99.182.238	167.99.182.238	167.99.182.238
id.resp_p	80	80	80
proto	tcp	tcp	tcp
service	http	http	http
duration	0 days 00:00:01.97859	0 days 00:00:02.182247	0 days 00:00:01.675550
orig_bytes	149	149	152
resp_bytes	119442	119442	83118
conn_state	SF	SF	SF
local_orig	-	-	-
local_resp	-	-	-
missed_bytes	0	0	0
history	ShADadttfF	ShADadttfF	ShADadttfF
orig_pkts	174	172	124
orig_ip_bytes	11698	11570	8120
resp_pkts	172	170	122
resp_ip_bytes	247844	247740	172596

Продовження таблиці 3.6

Змінна	Трафік	Трафік	Трафік
attack	C&C-HeartBeat-FileDownload	C&C-HeartBeat-FileDownload	C&C-HeartBeat-FileDownload

Характеристика елементів трафіку:

- ts: час;
- id.orig_h: IP-адреса джерела;
- id.orig_p порт джерела;
- id.resp_h: IP адреса призначення;
- id.resp_p: порт призначення;
- proto: транспортний протокол;
- service: сервіс;
- duration: тривалість з’єднання;
- orig_bytes: байти корисного навантаження джерела;
- resp_bytes: байти корисного навантаження призначення;;
- conn_state: статус з’єднання;
- local_orig;
- local_resp;
- missed_bytes: кількість втрачених байтів;
- history;
- orig_pkts: загальна кількість пакетів, відправлених з IP на порт;
- orig_ip_bytes: : загальна кількість байтів, відправлених з IP на порт;
- resp_pkts: загальна кількість пакетів, отриманих з IP на порт;
- resp_ip_bytes: загальна кількість байтів, отриманих з IP на порт;
- attack: тип атаки.

Після ознайомлення елементами з яких складаються дані, необхідно виділити ті, які будуть задіяні в навчанні. Для цього можна скористатися кореляцією даних.

Кореляція, або кореляційний залежність - статистична взаємозв'язок двох або більше випадкових величин[37].

Для кореляції можна скористатись бібліотекою Python, що називається «plt», та знайти сильно корельовані змінні за допомогою теплової карти та ігнорувати їх для аналізу. Теплова карта зображена на рисунку 3.19.

Логи необхідно перенести з журналу та представити у виді який буде необхідний для подальшої їх обробки системою. Для виконання даної частини необхідний зчитувач, який зможе виконувати функцію. Для цих цілей було обрано Zeek Analysis Tool (ZAT) спеціально створений для цих цілей.

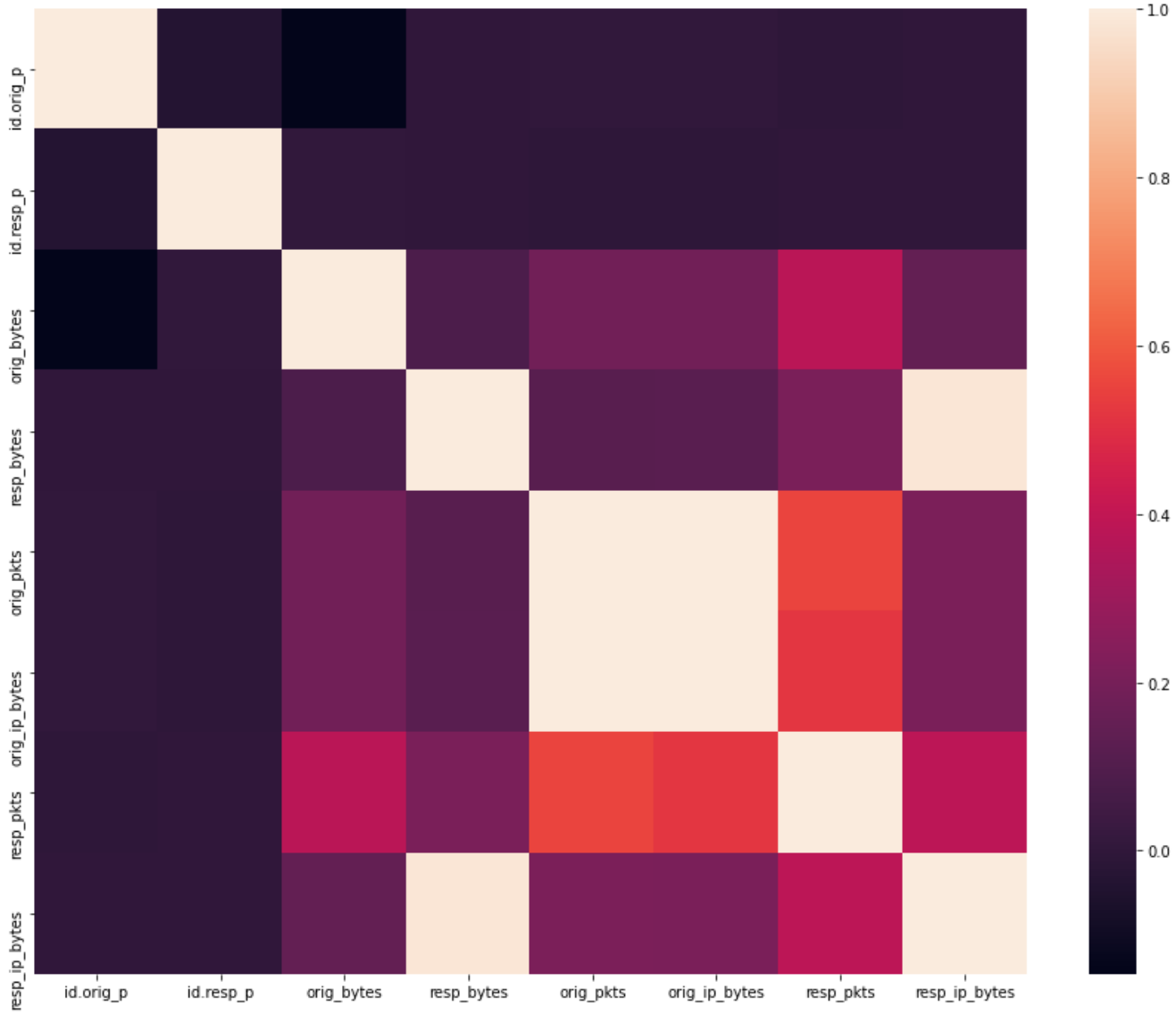

Рисунок 3.19 – Теплова карта кореляції змінних

Змінна resp_bytes сильно корелює з resp_ip_bytes, тому необхідно її ігнорувати. Загалом для навчання були обрано змінні, такі як: id.orig_p, id.resp_p, proto, conn_state, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes, attack.

3.15 Зчитування даних

Хоча Zeek вже має гнучку, потужну мову сценаріїв для широкого аналізу трафіку, але використання ZAT значно зменшує навантаженння та полегшує роботу Zeek**.**

Розвантаження: Запуск складних завдань, таких як статистика, стан машини, машинне навчання тощо. необхідно зменшити навантаження на функціональність Zeek, щоб Zeek міг зосередитися на ефективній обробці великого обсягу мережевого трафіку.

Аналіз даних: ZAT має великий набір класів підтримки, які допомагають перейти від необроблених даних Zeek до таких пакетів, як Pandas, scikit-learn та Spark. На софіційному сайті можна переглянути документацію та приклади використання ZAT[38].

Використання ZAT дозволяє швидко та в кілька рядків коду, зчитати та занести в датафрейм великі об’єми інформації. Фрагмент прикладу перенесених даних зображено на рисунку 3.20.

ig_bytes	resp_bytes	conn_state	local_orig	local_resp	missed_bytes	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	tunnel_parents
0	0	S0	NaN	NaN	0	S	3	180	0	0	NaN
<NA>	<NA>	S0	NaN	NaN	0	S	1	60	0	0	NaN
<NA>	<NA>	S0	NaN	NaN	0	S	1	60	0	0	NaN
149	128252	SF	NaN	NaN	2896	ShADadttcfF	94	5525	96	139044	NaN
0	0	S0	NaN	NaN	0	S	3	180	0	0	NaN
151	128348	SF	NaN	NaN	5792	ShADadttcfF	96	5699	92	133140	NaN
148	91961	SF	NaN	NaN	2896	ShADadtcfF	67	4148	65	92453	NaN

Рисунок 3.20 – фрагмент вигляду даних в датафреймі

3.16 Вибір необхідних даних

Занесення даних в датафрейм надає можливість легко з ними працювати, в першу чергу необхідно виділити характеристики, які необхідні для процесу класифікації трафіку, такі характеристики також необхідні для навчання моделі. Це характеристики: id.orig_p, id.resp_p, proto, conn_state, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes. Процес вибору характеристик описано в підрозділі 3.6. Після вибору характеристик датарейм перезаписується з обраними значеннями та набуває вигляду, який зображено на рисунку 3.21.

ts	id.orig_p	id.resp_p	proto	conn_state	missed_bytes	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes
2019-02-28 18:15:34.188184023	52724	80	0	0	0	174	11698	172	247844
2019-02-28 18:15:36.206260920	52726	80	0	0	0	172	11570	170	247740
2019-02-28 18:15:38.426746130	52728	80	0	0	0	124	8120	122	172596
2019-02-28 18:15:40.140019894	52730	80	0	0	0	136	8578	134	193266
2019-02-28 18:15:41.976972103	52732	80	0	0	0	170	11602	170	244256
2019-02-28 18:15:44.163718939	38448	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.163727045	47056	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.163966894	40544	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.163973093	36926	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.163976908	33046	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.164216042	40242	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.164222956	46220	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.164226055	44566	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.164465904	39052	23	0	1	0	6	360	0	0
2019-02-28 18:15:44.164473057	36996	23	0	1	0	6	360	0	0

Рисунок 3.21 – Оброблений датасет, що готовий до класифікації

3.17 Препроцесинг даних

Для коректного навчання необхідно замінити текстові змінні на числові, таким чином були замінені змінні proto – таблиця 3.7 та conn_state – таблиця 3.8.

Таблиця 3.7 – Заміна змінних proto

До		Після
tcp	→	0
udp	→	1
icmp	→	2

Таблиця 3.8 – Заміна змінних conn_state

До		Після
SF	→	0
S0	→	1
REJ	→	2
RSTR	→	3
RSTO	→	4
SH	→	5
S1	→	6
S2	→	7
RSTOS0	→	8
S3	→	9
OTH	→	10

Для заміни використовувались стандартні можливості мови Python. Заміна на числові дані відбувається для того, щоб модель могла класифікувати особливості вхідних даних.

Після виконання заміни можна поглянути на готовий до навчання набір даних, зображено на рисунку 3.22.

ts	id.orig_p	id.resp_p	proto	conn_state	missed_bytes	orig_pkts	proto	orig_ip_bytes	resp_pkts	resp_ip_bytes	attack
2019-02-28 18:15:34.188184023	52724	80	0	0	0	174	0	11698	172	247844	C&C-HeartBeat-FileDownload
2019-02-28 18:15:36.206260920	52726	80	0	0	0	172	0	11570	170	247740	C&C-HeartBeat-FileDownload
2019-02-28 18:15:38.426746130	52728	80	0	0	0	124	0	8120	122	172596	C&C-HeartBeat-FileDownload
2019-02-28 18:15:40.140019894	52730	80	0	0	0	136	0	8578	134	193266	C&C-HeartBeat-FileDownload
2019-02-28 18:15:41.976972103	52732	80	0	0	0	170	0	11602	170	244256	C&C-HeartBeat-FileDownload
...	...	...	...	...	...	...	...	...	...	...	...
2019-02-28 18:58:50.047991991	37212	23	0	1	0	6	0	360	0	0	PortScan
2019-02-28 18:58:50.047994852	36810	23	0	1	0	6	0	360	0	0	PortScan
2019-02-28 18:58:50.048230171	45848	23	0	1	0	6	0	360	0	0	PortScan
2019-02-28 18:58:50.048235893	52904	23	0	1	0	6	0	360	0	0	PortScan
2019-02-28 18:58:50.048238993	38532	23	0	1	0	6	0	360	0	0	PortScan

250000 rows × 11 columns

Рисунок 3.22 – Готовий до навчання набір даних

Якщо порівняти датасет до та після обробки, можна помітити, що у необробленому датасеті деякі характеристики мають значення NaN, це зроблено в цілях зменшення хибних спрацювань системи, тобто хибній класифікації трафіку, що може призвести до постійного хибного спрацювання системи. Для класифікації обрані дані які в повному обсязі описують подію та в своєму звичайному вигляді не мають пустих значень.

Після обробки дані направляються на класифікатор, що в свою чергу вже визначає до якого класу відноситься трафік. Для більш точної класифікації трафіку систему необхідно навчати великим датасетом, для якого необхідні великі обчислювальні потужності.

3.9 Порівняння результатів навчання моделей

Навчання на великих наборах даних потребує багато часу та ресурсів, саме тому, для навчання було обрано не весь датасет, а декілька із його частин. Після тренування кожним з методів можна порівняти час за який було витрачено на тренувався кожен з методів, що зображено на рисунку 3.23, де RF – випадковий ліс, SVM – метод опорних векторів, KNN – k-найближчих сусідів.

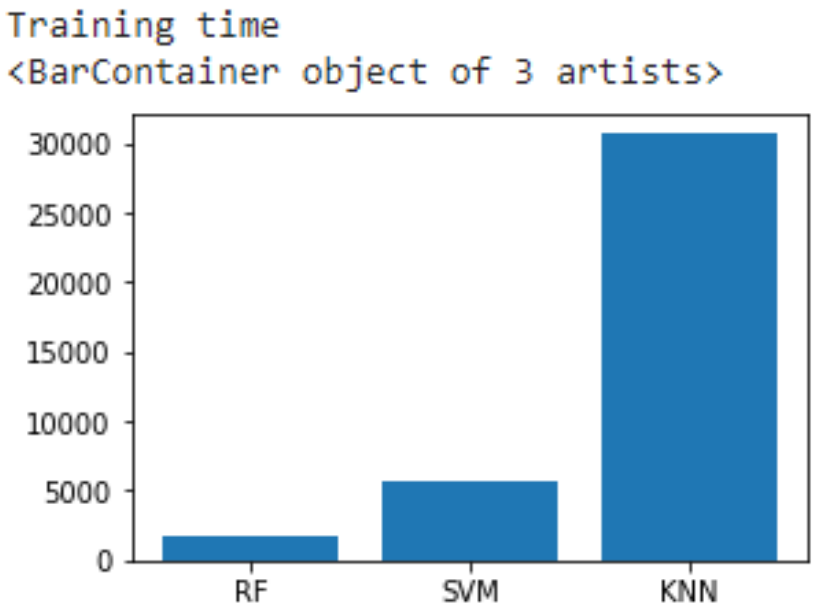

Рисунок 3.23 – Час витрачений на тренування алгоритмів

Модель алгоритму k-найближчих сусідів займає найбільше часу для навчання, це доволі довго для такої системи, якщо брати до уваги те, що датасет має невелику базу вхідних даних. Метод SVM та RF займають значно менше часу для навчання моделі.

Але довге навчання моделі не є показником її бездіяльності, необхідно також здійснити інші метрики для вибору підходящого алгоритму навчання та навченої моделі. Після навчання системи можна порівняти час за який система буде виконувати аналіз даних, в даному випадку система аналізує 20% від датасету, що важливо ця частина даних не використовувалась в навчанні, а призначена саме для виконання тестування моделі. Діаграма порівняння часу тестування моделей зображено на рисунку 3.24.

Як можна помітити час тестування для кожної з систем є різний. Модель KNN показуй не найгірший результат, тестування методу SVM показує дуже великий час тестування на фоні інших, що не підходить для використання системи, оскільки необхідна система, що зможе швидко виконувати аналіз даних, які до неї подаються. Метод SVM в даній ситуації виглядає особливо привабливим, оскільки тестування 20% датасету відбулося дуже швидко. .

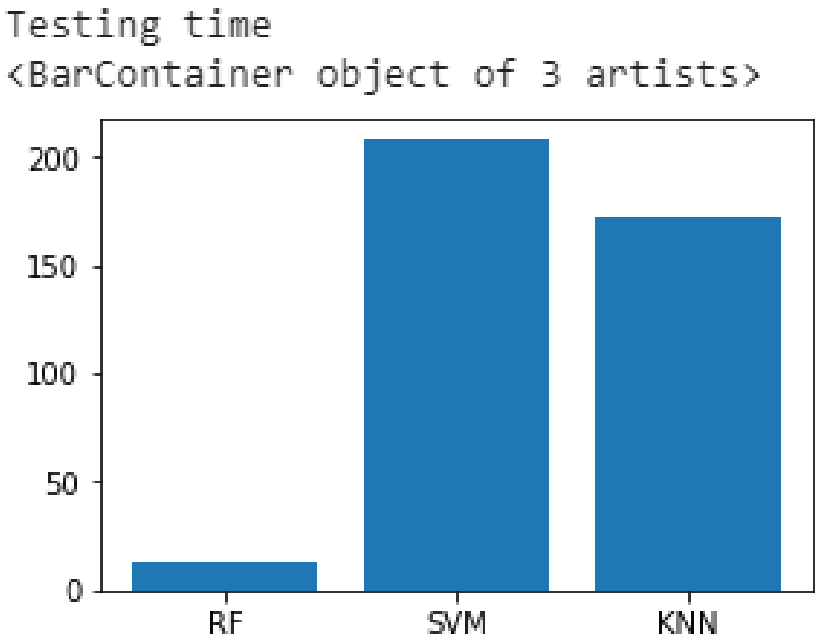

Рисунок 3.24 - Час витрачений на тестування алгоритмів

Тепер необхідно визначити, яка з моделей буде найбільш точно класифікувати атаки та доброякісний трафік. Найбільш важливою критерією необхідно вважати виявлення саме доброякісного трафіку, оскільки виявлення аномальної поведінки зможе знаходити навіть, так звані, атаки нульового дня – атаки не відомі раніше для системи.

Також така особливість системи дозволить виявляти підключення до системи нових пристроїв, що не були додані адміністратором мережі, це дозволить виключити можливість виконувати зловмисні дії з середини мережі. Для такої особливості необхідно правильно класифікувати доброякісний трафік, діаграма точності моделі зображено на рисунку 3.25.

Як можна побачити з діаграми зображеної на рисунку 3.25 точність моделей є доволі гарним показником у кожної з них. Після отримання метрик необхідних для вибору моделі, можна переглянути числові дані, та порівняти їх між моделями. Отримані дані представлено в таблиці 3.9.

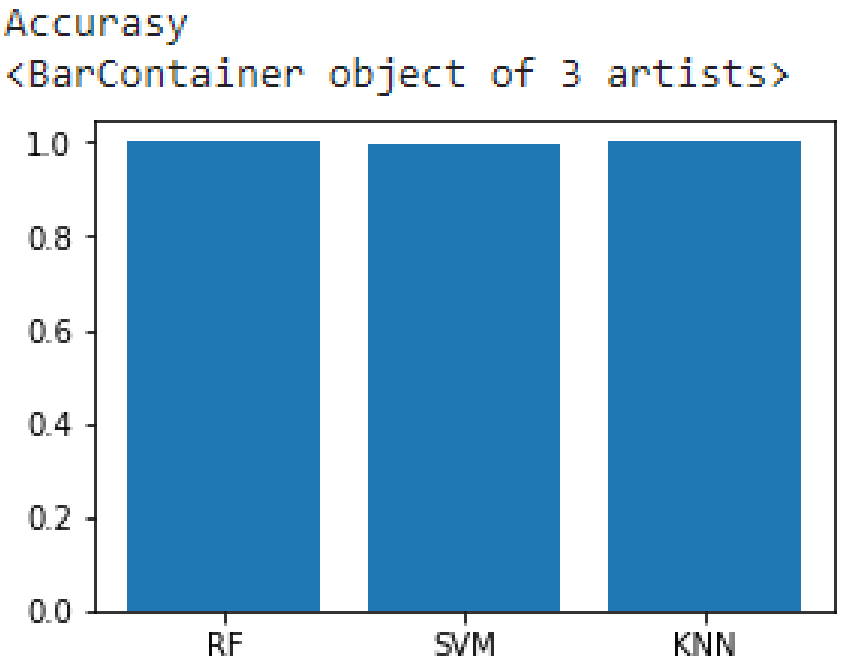

Рисунок 3.25 – Діаграма показників точності

Таблиця 3.9– Порівняння методів машинного навчання

	RF	SVM	KNN
Час тренування	1665.1454224586487	5718.65680193011	30627.3205475
Час тестування	13.658202886581421	207.48963451385498	171.8562672138214
Точність	0.9993989995109506	0.9987243470011843	0.9993430239752058

Можна провести аналіз для обрання ефективної моделі, підходящими характеристиками для моделі є:

– оптимальний час тренування;

– швидке тестування;

– висока точність.

Порівняльний аналіз підходящих характеристик зображено в таблиці 3.10.

Таблиця 3.10 - Порівняльний підходящих характеристик моделей

№ п/п	Характеристика моделі	RF	SVM	KNN
1.	Оптимальний час тренування	+		
2.	Швидке тестування	+		
3.	Висока точність	+		

З порівняльного аналізу можна побачити, що модель RF відповідає всім критеріям для використання її в системі виявлення вторгнень.

Для створення моделей за допомогою неконтрольованих методів навчання необхідні великі кількості вхідних даних. Такий підхід допоможе виявляти шкідливий трафік на основі відхилень від норми, що в свою чергу потребуватиме виставлення порогу чутливості та виникне потреба в постійному моніторингові даних. Для більш точного аналізу в даній роботі було прийнято рішення використовувати моделі навчені за допомогою контрольованого методу.

3.18 Алгоритм роботи системи

Блок-схему роботи системи зображено на рисунку 3.17.

Свою роботу система починає з отримання даних, дані подаються в своєму першочерговому вигляді, файл conn.log – це журнал в який записуються всі події, виявлені за допомогою аналізатора мережі Zeek. Оскільки conn.log є звичайним текстовим файлом в першу чергу дані необхідно помістити у вигляд зручний для роботи. Для цього застосовуються методи бібліотеки ZAT, що дозволяє в кілька рядків перенести та сегментувати дані в датафрейм, наступний крок обробка за допомогою препроцесору.

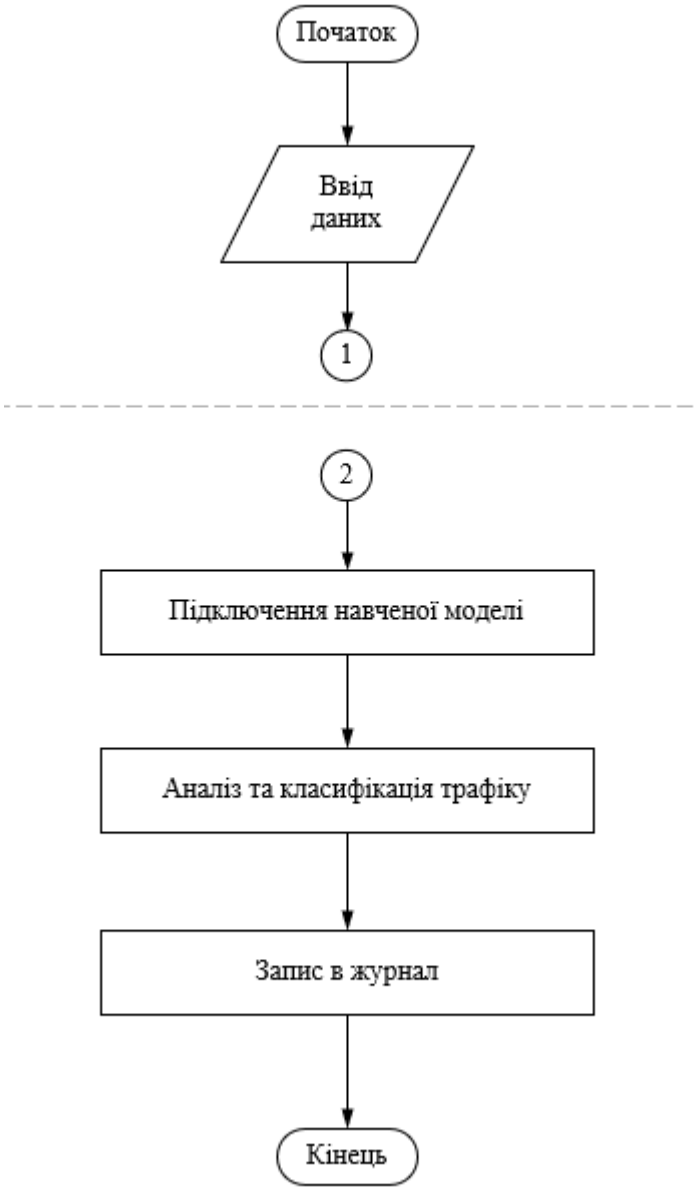

Рисунок 3.26 – Блок-схема роботи системи

Після занесення даних в датафрейм препроцесор, починає діяти за алгоритмом зображеним на рисунку 3.27. Перший крок виконання підсистеми – отримання даних. Підсистема просто отримує готовий датафрейм. Після отримання даних, відбувається вибір характеристик за допомогою яких система зможе класифікувати подію. До таких характеристик відносяться: id.orig_p, id.resp_p, proto, conn_state, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes. Після вибору даних характеристик відбувається заміна нечислових значень числовими, а саме для proto та conn_state. Заміна цих характеристик показана у таблиці 3.7 та 3.8 відповідно.

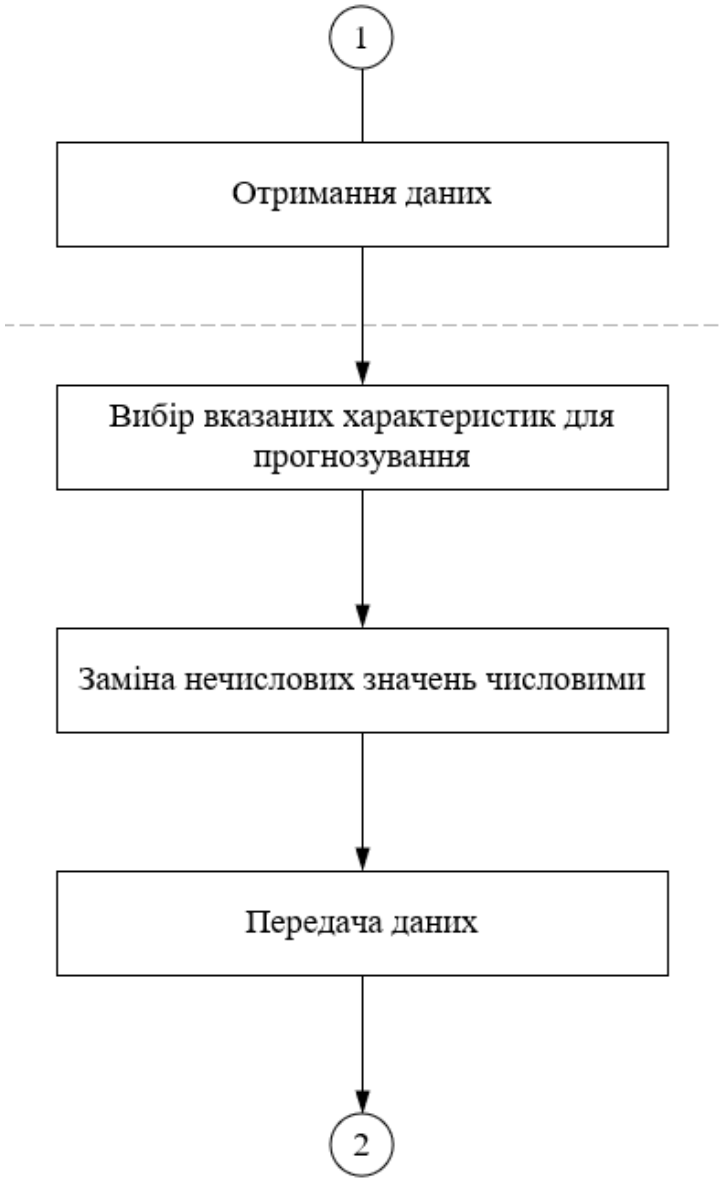

Рисунок 3.27 – Блок-схема роботи препроцесора

Також відбувається заміна характеристик у яких відсутні значення, якщо такі дані помістити в аналізатор, програма дасть збій, таким чином це дозволяє використовувати систему без збоїв.

Якщо коротко препроцесор в даній системі виконує функції:

- перенесення даних в вид зручний для їх перетворення;
- перетворення даних для класифікації.

Далі дані направляються до аналізатора який проводить класифікацію трафіку в залежності від вибраної моделі, це може бути відхилення від норми в лінійних методах та класифікація трафіку як доброякісного чи шкідливого.

3.19 Тестування системи

Систему вирішено було розробляти за допомогою мови програмування Python. За допомогою консольної команди «python main.py -h» можна здійснити огляд необхідних команд для роботи системи, рисунок 3.17. На рисунку зображені основні команди для роботи системи, а саме:

- -m model: обов’язкове вказання навченої моделі моделі для виконання аналізу трафіку;
- -f file: обов’язкове вказання файлу, в якому знаходяться дані для аналізу;
- -d dumptocsv: запис результату аналізу до файлу csv для подальшої обробки, після вказання аргументу необхідно дати назву файлу до якого буде записаний результат;
- -r results: відобразити результати аналізу, які були спрогнозовані як шкідливі події;
- -b benign: відобразити доброякісні результати аналізу.

```
(system) C:\Users\keepu\PycharmProjects\system>python main.py -h
usage: main.py [-h] [-m MODEL] -f FILE [-d] [-r] [-b]

optional arguments:
  -h, --help            show this help message and exit
  -m MODEL, --model MODEL
                        Choose a model RF
  -f FILE, --file FILE  Path to the conn.log input file to read.
  -d, --dumptocsv       Dump the conn.log DataFrame to a csv file
  -r, --results         Show all results
  -b, --benign          Show benign results
```

Рисунок 3.28 – Огляд консольних команд системи

Така модель роботи система розроблена для гнучкості у використанні та розробці системи. ЇЇ можна допрацьовувати та реалізовувати різні сценарії використання:

– змінювати навчену модель;
– змінювати файл для аналізу;
– модернізувати саму систему під необхідності використання.

Для запуску необхідно вказати необхідні параметри наведені вище, запуск програми зображено на рисунку 3.29, результат запуск із записом даних в csv файл на рисунку 3.30.

```
(system) C:\Users\keepu\PycharmProjects\system>python main.py -m finalized_model_RF.sav -f conn.log.labeled -r
C:\Users\keepu\anaconda3\envs\system\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle esti
de or invalid results. Use at your own risk.
  warnings.warn(
C:\Users\keepu\anaconda3\envs\system\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle esti
de or invalid results. Use at your own risk.
```

Рисунок 3.29– команда для запуску системи

```
(system) C:\Users\keepu\PycharmProjects\system>python main.py -f conn.log.txt -m finalized_model_RF_3.sav -d data
Simple Anomaly Detector for Zeek conn.log files. Version: 0.2
C:\Users\keepu\anaconda3\envs\system\lib\site-packages\pandas\core\missing.py:49: FutureWarning: elementwise compa
  mask = arr == x
C:\Users\keepu\anaconda3\envs\system\lib\site-packages\pandas\core\missing.py:49: FutureWarning: elementwise compa
  mask = arr == x
C:\Users\keepu\anaconda3\envs\system\lib\site-packages\sklearn\base.py:329: UserWarning: Trying to unpickle estima
or invalid results. Use at your own risk.
```

Рисунок 3.30 – команда для запуску системи із записом даних

Для виведення в консоль даних які були класифіковані як шкідливі необхідно ввести команду, яка відповідає за це. Виведення даних зображено на рисунку 3.31.

Для запуску системи порядок вказаних елементів не важливий оскільки програма автоматично зчитує інформацію та передає до системи. Для роботи файли не обов’язково переміщати в спільну директорію з системою, достатню вказати повну адресу до необхідних елементів для роботи. Результатом роботи є класифікація подій на доброякісні та шкідливі, дані виводяться в консоль адміністратора та записуються в файл csv для подальшої обробки.

ts	uid	id.orig_h	id.orig_p	...	resp_ip_bytes	tunnel_parents	pred
2018-12-21 14:50:48.981338024	CDrkrSobGYxHhYfth	192.168.1.195	41040	...	139044	NaN	Malware
2018-12-21 14:50:57.781319857	CvyynC4Sabj9BNXFRi	192.168.1.195	41042	...	133140	NaN	Malware
2018-12-21 14:50:59.183341026	CWYyiA2sgRijwk2jEd	192.168.1.195	41044	...	92453	NaN	Malware
2018-12-21 14:51:00.282392025	CYttPy2pqOIcen7UDh	192.168.1.195	41046	...	97107	NaN	Malware
2018-12-21 14:55:30.023056984	CcSgZloEzkJknQDIk	192.168.1.195	41052	...	168910	NaN	Malware
...	...	...	...	...	...	...	...
2018-12-22 14:46:17.702588081	CinFTF4QlQyRoBn2Ph	192.168.1.195	57094	...	589	NaN	Malware
2018-12-22 14:47:41.142472029	CG2jvv2N8tYuhv8fo7	192.168.1.195	57100	...	589	NaN	Malware
2018-12-22 14:44:40.763553858	CRPM7qYJN4QudEl47	192.168.1.195	57086	...	589	NaN	Malware
2018-12-22 14:48:53.961380005	C2F17zSUnGOcWzBa7	192.168.1.195	57110	...	589	NaN	Malware
2018-12-22 14:45:34.221598148	C93P4z4k5IRJD1rXJg	192.168.1.195	57092	...	632	NaN	Malware

Рисунок 3.31 – Виведення даних в консолі

Для запуску системи порядок вказаних елементів не важливий оскільки програма автоматично зчитує інформацію та передає до системи. Для роботи файли не обов’язково переміщати в спільну директорію з системою, достатню вказати повну адресу до необхідних елементів для роботи. Результатом роботи є класифікація подій на доброякісні та шкідливі, дані виводяться в консоль адміністратора та є можливість записати їх у файл csv для подальшої обробки. Система має можливість загального відображення класифікованих даних та занесення їх у файл. Приклад загального відображення зображено на рисунку 3.32.

proto	service	...	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	tunnel_parents	pred
		...							
0	NaN	...	S	3	180	0	0	NaN	Benign
0	NaN	...	S	1	60	0	0	NaN	Benign
0	NaN	...	S	1	60	0	0	NaN	Benign
0	http	...	ShADadttcfF	94	5525	96	139044	NaN	Malware
0	NaN	...	S	3	180	0	0	NaN	Benign
...	...	...	...	...	...	...	...	...	...
0	irc	...	ShAdDaf	7	434	6	589	NaN	Malware
0	irc	...	ShAdDaf	10	606	7	632	NaN	Malware
1	NaN	...	D	1	76	0	0	NaN	Benign
1	NaN	...	D	1	76	0	0	NaN	Benign
1	NaN	...	D	1	76	0	0	NaN	Benign

3.32 – класифікація даних

Результат класифікації відображається в останній колонці з назвою pred, всі інші колонки, що відображуються, система залишає, оскільки дані які збираються можуть бути корисними для інших систем з якими може взаємодіяти дана система. Далі ці

дані можуть бути направлені в місця вказані адміністратором. Однією з необхідних функцій такої системи є збереження аналізованих даних в файл csv, рисунок 3.32.

local_orig	local_resp	missed_b	history	orig_pkts	orig_ip_b	resp_pkts	resp_ip_b	tunnel_pa	pred
		0	S	3	180	0	0		Benign
		0	S	1	60	0	0		Benign
		0	S	1	60	0	0		Benign
		2896	ShADadtt	94	5525	96	139044		Malware
		0	S	3	180	0	0		Benign
		5792	ShADadtt	96	5699	92	133140		Malware
		2896	ShADadtt	67	4148	65	92453		Malware
		5792	ShADadtt	75	4412	69	97107		Malware
		0	S	3	180	0	0		Benign
		0	S	1	60	0	0		Benign
		0	S	1	60	0	0		Benign
		0	Dd	1	76	1	76		Benign
		0	S	1	60	0	0		Benign
		0	S	1	60	0	0		Benign
		0	Dd	1	76	1	76		Benign
		0	S	3	180	0	0		Benign
		0	S	1	60	0	0		Benign
		0	S	1	60	0	0		Benign
		0	D	1	76	0	0		Benign
		0	S	1	60	0	0		Benign
		0	Dd	1	76	1	76		Benign
		0	D	1	76	0	0		Benign

Рисунок 3.33 – Вигляд даних занесених у файл csv

Вигляд даних такий же як і у консолі, та поділений для зручності подальшого використання. Створення таких файлів дозволяє їх застосування для подальшого навчання моделі, або ж статистиці. Також не є виключенням повторна перевірка для більш обширного пошуку аномалій та знаходження атак, що не було виявлено. Постійне перенавчання моделі на свіжих даних дозволить системі виявляти нові види атак.

3.20 Висновки до розділу

У даному розділі було розглянуто та вибрано технології необхідні для створення системи виявлення вторгнень. Зокрема у розділі було обґрунтовано вибір аналізатора мережі, з використанням якого буде написана дана система, вибір датасету для навчання системи, модель алгоритму машинного навчання за допомогою якого буде проводитися навчання моделі. Після огляду та тестування алгоритмів було обрано алгоритм Random Forest, який відповідає функціональним вимогам наведеним в розділі 2. Також було розглянуто структурну схему системи виявлення вторгнень у комп’ютерну мережу. Проаналізовано роботу системи від зчитування інформації до виведення результату, який спрогнозувала система. Дані яка система видає в якості інформування можна перенаправити на інші системи безпеки. Система повідомляє саме ці дані для того, щоб система чи адміністратор могли блокувати з’єднання та запобігти аномальній активності в мережі.

4 СТАРТАП ПРОЕКТ

Даний розділ створений для проведення маркетингового аналізу розробленої системи як стартап проекту. Проект розглядатиметься як система виявлення вторгнень.

4.1 Опис ідеї проекту

Для початку відбувається аналіз та подання змісту ідеї стартап-проекту, вигоди та можливі напрямки, якими може бути зацікавлений користувач продукту. Опис ідеї стартап-проекту показано в таблиці 4.1.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система виявлення вторгнень (IDS) для виявлення шкідливого трафіку в мережі	Встановлення на серверне обладнання та дозволяє адміністратору мережі покращити рівень моніторингу мережі на випадки проникнення	Для власника IDS: поява прибутку за рахунок продажу системи. Для кінцевого користувача – це можливість навчання моделі під особливості мережі.
	Можливість аналізувати дані зібрані за деякий період, на любій платформі	Зменшення витрат за рахунок необхідності розгортання системи на певній платформі

Аналіз потенційних техніко-економічних переваг ідеї в порівнянні з пропозиціями конкурентів. Результати аналізу зображено в таблиці 4.2.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідей проекту

№ п/п	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Проект	Snort	Suricata			
1.	Вартість	20000$/рік	40000$/рік	50000$/рік			+
2.	Прибуток	150000$/рік	200000$/рік	100000$/рік		+	
3.	Контроль якості	Програмісти	Аналітики, програмісти	Аналітики та програмісти	+		
4.	Динаміка галузі	Швид-ка	Швид-ка	Швид-ка		+	
5.	Витрати	5000$/рік	19000$/рік	17000$/рік			+
6.	Витрати, що варіюються	5000$ - 10000$/рік	10000$ - 15000$/рік	15000$ - 20000$/рік	+		
7.	Патенти на продукти	Відсутні	Наявні патенти на винахід	Наявні патенти на винахід	+		
8.	Гнучкість вартості	Вартість єдина	Вартість варію-ється	Вартість єдина	+		

4.2 Технологічний аудит ідеї проекту

Визначення технологічної здійсненності ідеї проекту за допомогою аналізу таких складових, як технології, за якою буде виготовлено товар згідно ідеї проекту, існування таких технологій, чи їх необхідно розробити / доробити, доступність таких технологій авторам проекту. Результати аналізу зображено в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Зчитування мережевого трафіку з журналів Zeek	Система має можливість зчитувати готові журнали мережевого трафіку Zeek	Є у наявності – Zeek Analysis Tool	Є відкритими
Аналіз мережевого трафіку	Система має можливість аналізувати дані за допомогою ML моделі	Є у наявності (Scikit-Learn, Keras, TensorFlow)	Необхідна розробка
Обробка мережевого трафіку	Можливість системи обробляти трафік перед аналізом	Необхідна розробка	Необхідна розробка

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Проведення аналізу попиту: наявність попиту, обсяг, динаміка розвитку ринку. Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту Результати аналізу зображено в таблиці 4.4.

Таблиця 4.4 – Характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	700000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Здатність працювати на серверному обладнанні
5	Специфічні вимоги до стандартизації та сертифікації	Дотримання формату IDMEF
6	Середня норма рентабельності в галузі (або по ринку), %	90

Таким чином, після попереднього аналізу попиту , ринок є придатнім для входження.

Надалі відбувається визначення потенційних групи клієнтів, їх характеристики, та формування орієнтовного переліку вимог до товару для кожної групи користувачів. Характеристика потенційних клієнтів зображена в таблиці 4.5.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Захищеність інфраструктури	ІТ компанія	ІТ компанія прагне знати про спроби проникнення в мережу.	ІТ компанія прагне захистити свою інфраструктуру.
2	Аналітика методів шкідливої діяльності	ІТ компанія	ІТ компанія потребує постійного моніторингу та аналізу мережі	ІТ компанія прагне розширити базу можливих випадків шкідливої діяльності

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складання таблиці факторів, що сприяють ринковому впровадженню проекту (Таблиця 4.6), та факторів, що йому перешкоджають (Таблиця 4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Відсутність попиту	Використання звичних IDS.	Зосередитися на клієнтах, що вже використовують продукт, збір даних про результативність системи з подальшим її використанням в рекламній компанії.
2	Високий рівень хибних результатів системи	Високий рівень хибних результатів через особливості середовища.	Збір інформації та створення моделі під конкретне середовище.

Таблиця 4.7 – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Відсутність альтернатив	Існуючі системи не задовольняють спектром своїх можливостей	Розширення спектру можливостей

Надалі проводиться аналіз пропозиції: відбувається визначення загальних рис конкуренції на ринку. Результати аналізу зображені в таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Чиста конкуренція	Продукти не мають сильних відмінностей	Презентація на виставках, вигідні умови для користувачів
2. Регіональна конкуренція	Гравці ринку – міжнародні компації	Зайняття вільних ринків
3. Внутрішньогалузева конкуренція	Гравці ринку знаходяться в одній галузі	Розвивати напрямки нерозвинуті конкурентами
4. Товарно-видова конкуренція	Продукти мають різне призначення	Розвивати напрямки відмінні від конкурентів
5. Конкурентні переваги неціновi	Продукти відрізняються	Надання послуг, які не надають конкуренти
6. Марочна конкуренція	Акцент уваги на бренд, що розробив продукт	Надання послуг, які не надають конкуренти

Далі необхідно визначити та обґрунтувати фактори конкурентоспроможності Обґрунтування факторів конкурентоспроможності зображено в таблиці 4.9.

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Здатність працювати на любому залізі.	Відсутність необхідності у покупці дорогого заліза.

Продовження таблиці 4.9

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
2	Швидкодія	Аналіз відбувається швидше ніж для повної класифікації
3	Захист від 0-day атак	Здатність розпізнавати нові, невідомі атаки
4	Гнучкість	Система може масштабуватися через особливості ML.

Далі проводиться більш детальний аналіз умов конкуренції в галузі за М. Портером. Аналіз конкуренції зображено в таблиці 4.10.

Таблиця 4.10 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Динаміка галузі, подібні продукти, недосяжність ринку	Патенти на продукти, товарні знаки	Наявність постачальників, більш вигідні пропозиції	Чутливість до зміни цін, контроль якості	Лояльність споживачів, ціна

Продовження таблиці 4.10

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Висновки:	Конкуренція не є інтенсивною, оскільки ринок ще не до кінця сформований.	Обов'язкове створення торгового знаку, та налагодження програмного продукту.	Програмному продукту не потрібно постачання.	Клієнти задають роботу ринку та динаміку цін.	Необхідні постійні інвестиції для покращення якості системи.

Після визначення факторів конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту. Результати аналізу зображено в таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з проектом*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	Здатність працювати на любому обладнанні	13				*			
2	Швидкодія	17			*				
3	Захист від 0-day атак	16		*					
4	Гнучкість	14				*			

Фінальним етапом ринкового аналізу можливостей впровадження проекту є формування SWOT – аналізу, на основі виділених ринкових загроз та можливостей, та сильних та слабких сторін. За допомогою даного аналізу розробляються

альтернативи ринкової поведінки, що сприяють виведенню стартап-проекту на ринок. Також розробляється орієнтовний оптимальний час їх ринкової реалізації з рівнянням на потенційні проекти конкурентів, які мають можливість бути виведенеми на ринок. SWOT аналіз складається з аналізу сильних, слабких сторін, загроз та можливостей. SWOT – аналіз представлено у таблиці 4.12.

Таблиця 4.12 – SWOT – аналіз стартап-проекту

Сильні сторони: Захист від 0-day атак, швидкодія	Слабкі сторони: Відсутність моніторингу в реальному часі
Можливості: відсутність альтернатив	Загрози: Обмеженість функцій

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтований оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Альтернативи ринкового впровадження зображено в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Реалізація IDS функціоналу	Висока	6 місяців
2	Створення системи аналізу трафіку	Висока	10 місяців

Серед даних альтернатив було обрано першу альтернативу, адже строки її реалізації найменші та є ймовірність високого отримання ресурсів.

4.4 Розроблення ринкової стратегії проекту

Для розроблення ринкової стратегії спочатку необхідно виділити та описати цільові групи потенційних клієнтів. Вибір цільових груп потенційних клієнтів зображено в таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних клієнтів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Малі компанії	Відсутня	Не є цільова група	-	-
2.	Середні компанії	Середня	8-10 підприємств в рік	Середня	Середня
3.	Великі компанії	Готові	1-3 заклади в рік	Висока	Складна
Було обрано цільову групу: середні компанії.					

Для роботи в обраних сегментах ринку формується базова стратегію розвитку. Визначення базової стратегії розвитку зображено в таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвитку

Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Зосередження на цільових сегментах компанії.	Запропонований продукт має меншу вартість та більшу кількість функцій	Спеціалізація.

Наступним кроком є визначення стратегії конкурентної поведінки, таблиця 4.16.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохід-цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Так.	Так	За потреби буде проводитися розробка нових функцій продукту відмінних від конкурентів	Стратегія диференціації

Далі розробляється стратегія позиціонування, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати

торгівельну марку/проект. Визначення стратегії позиціонування зображено в таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспро-можні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Здатність виявлення нових атак, швидкодія	Створення нового функціоналу, проведення досліджень	Товар інноваційний та дешевий у порівнянні з аналогами	Швидкість, виявлення нових атак, дешевизна

4.5 Проведення маркетингової програми стартап-проекту

Відбувається формування маркетингової концепції товару, який отримає споживач. Визначення ключових переваг концепції потенційного товару зображено в таблиці 4.18

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Актуальність.	Виявлення невідомих атак	Не залежить від сигнатур

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
2.	Гнучкість	Здатність працювати на любому обладнанні	Не потребує потужного заліза
3.	Швидкодія	Швидкодія системи	Швидкі алгоритми аналізу даних

Надалі розробляються трирівневу маркетингову модель товару: уточнюється ідея продукту, його фізичні складові та особливості процесу його надання. Опис трьох рівнів моделі товару зображено в таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмний продукт – система виявлення вторгнень у комп’ютерну мережу.
	Властивості / характеристики: 1. Класифікатор подій в мережі 2. Збереження класифікованих даних
	Якість: продукт пройшов тестування з реальними даними
	Пакування: інсталятор
	Марка: назва організації-розробника «ММ», назва товару «MIDS».
	Користувач отримує консультацію з використання системи

Рівні товару	Сутність та складові
III. Товар із підкріпленням	Розробник підтримує актуальну версію програмного забезпечення

Наступний крок – визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар. Визначення проводиться експертним методом, результати зображено в таблиці 4.20.

Таблиця 4.20 – Визначення меж встановлення цін

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
20000-30000 $/рік	40000-50000 $/рік	400000-500000 $/рік	Нижня межа – 15000 $/рік, верхня межа - 30000 $/рік

Надалі визначається оптимальна система збуту, в межах якої приймається рішення. Формування системи збуту зображено в таблиці 4.21.

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Товар постачається з усіма необхідними засобами	Просте встановлення та оплата послуг	Розробник-користувач.	Канал збуту одного рівня.

Останньою складовою є розробка концепції маркетингової комунікації, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Концепцію маркетингових комунікацій зображено в таблиці 4.22.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнт намагається знайти нові методи захисту КМ.	Мережа Інтернет, конференції.	Швидкодія, кросплатформність, актуальність.	Демонстрація швидкодії, можливість виявлення нових атак, актуальність.	Швидка робота та точне виявлення атак

4.6 Висновки до розділу

В даному розділі було виконано етап розроблення стартап-проекту – маркетинговий аналіз стартап-проекту.

За результатами можна сказати, що проект має можливість ринкової комерціалізації. З огляду на потенційну групу клієнтів та інноваційність технології проект має перспективи для впровадження.

ВИСНОВКИ

Магістерська дисертація присвячена задачі розроблення системи виявлення вторгнень к комп’ютерну мережу.

Для ознайомлення з принципами будови систем виявлення вторгнень було проаналізовано актуальні існуючі рішення. Огляд наведено у розділі 1. На основі аналізу інформації, наведеної в розділі було прийнято рішення щодо необхідних компонентів системи.

Огляд існуючих рішень дозволив виявити вимоги та сценарії, яким повинна відповідати система. Огляд властивостей та сценаріїв наведено в розділі 2.

На основі вимог та сценаріїв відбувався вибір технологій, за допомогою яких була розроблена система. Технології, що було використано для створення цієї системи було обрано для спрощення процесу розробки. Вибір алгоритму за допомогою якого було навчено систему відбувався експерементальним шляхом. Модель за навченим алгоритмом повинна відповідати характеристикам, таким як, високий коефіцієнт точності та швидкий аналіз. Для аналізу необхідно підібрати набір даних який би задовільнив своєю кількістю даних, для даної системи було обрано набір IoT-23. Після навчання моделі різними алгоритмами було виявлено той, який найбільше відповідав усім вимогам. Вибір технологій наведено в розділі 3.

При розробленні системи виявлення вторгнень у мережу було використано мову програмування Python, алгоритм машинного навчання Random Forest, який показав найбільшу відповідність до вимог наведених в розділі 2. В середині системи налаштовано передачу даних між сегментами як на кожному кроці проводять певну обробку даних для подальшого успішного використання в системі.

Для повного опису системи та її функцій було розроблено та описано схему сценаріїв використання системи та структурну схему. Опис схем наведено у розділах 2 та 4.

Розроблено систему виявлення вторгнень у комп’ютерну мережу мережу, система складається з методів та технологій, які допомагають значно спростить розробку подібних систем. Методи та технології розглянуті та застосовані в даній

роботі поширені в розробці систем на основі машинного навчання попередньо розглянутих та протестованих. Проведено тестування системи на наборі даних, що показує високі показники точності при виявленні аномалій.

Було проведено дослідження можливості комерціалізації запропонованої ідеї реалізації. Для цього, проведено дослідження стартап-проекту за різними показниками, оцінено стан та тип конкуренції, можливість виходу на ринок, можливі перешкоди та виділено переваги даної реалізації продукту відносно конкурентів. На основі аналізу було зроблено висновок про доцільність подальшого впровадження проекту. Розроблення стартап – проекту описано у розділі 5.

Запропонована реалізація системи виявлення вторгнень у комп’ютерну мережу є досить ефективним засобом в порівнянні з аналогічними системами та має можливість бути комерціалізованою.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Machine Learning Algorithms for Network Intrusion Detection [Електронний ресурс]: Режим доступу https://link.springer.com/chapter/10.1007/978-3-319-98842-9_6
2. Open source IDS tools [Електронний ресурс]: Режим доступу: https://cisoclub.ru/open-source-ids-tools-sravnenie-suricata-snort-bro-zeek-linux/
3. Suricata [Електронний ресурс]: Режим доступу: https://suricata-ids.org
4. Функціональні та Не Функціональні Вимоги [Електронний ресурс]: Режим доступу: http://lvivqaclub.blogspot.com/2008/10/blog-post_17.html
5. What is the difference between functional and non functional requirement? [Електронний ресурс]: Режим доступу: https://stackoverflow.com/questions/16475979/what-is-the-difference-between-functional-and-non-functional-requirement
6. Monster Logs [Електронний ресурс]: Режим доступу: https://zeek.org/2012/01/04/monster-logs/
7. B. A. Forouzan, Data Communications and Networking, 5:th ed. Mcgroy-Hill, 2013
8. H. Alaidaros i M. Mahmuddin, " Flow-Based approach on Bro Intru-Sion Detection, " Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, № 2-2, С. 139-145, 2017.
9. Aposemat IoT-23 [Електронний ресурс]: Режим доступу: https://www.stratosphereips.org/datasets-iot23
10. Что такое ML.NET и принципы работы этой системы [Електронний ресурс]: Режим доступу: https://docs.microsoft.com/ru-ru/dotnet/machine-learning/how-does-mldotnet-work
11. G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction toStatistical Learning, 1:st ed.Springer, 2017
12. Support-vector networks [Електронний ресурс]: Режим доступу: https://link.springer.com/article/10.1007/BF00994018/ - 01.11.2020 р.

13. Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; "Support vector clustering" (2001) Journal of Machine Learning Research, 2: 125–137
14. Vapnik, V.: Invited Speaker. IPMU Information Processing and Management of Uncertainty in Knowlendge-Based Systems (2014)
15. Barghout, Lauren. "Spatial-Taxon Information Granules as Used in Iterative Fuzzy-Decision-Making for Image Segmentation." Granular Computing and Decision-Making. Springer International Publishing, 2015. 285-318.
16. Bilwaj Gaonkar, Christos Davatzikos "Analytic estimation of statistical significance maps for support vector machine based multi-variate image analysis and classification", April 2013.
17. R. Cuingnet, C. Rosso, M. Chupin, S. Lehéricy, D. Dormont, H. Benali, Y. Samson and O. Colliot, "Spatial regularization of SVM for the detection of diffusion alterations associated with stroke outcome, Medical Image Analysis", 2011, 15 (5): 729–737
18. Statnikov, A., Hardin, D., & Aliferis, C., "Using SVM weight-based methods to identify causally relevant and non-causally relevant variables". 2006.
19. Баев Н.О. Использование метода опорных векторов в задачах классификации // Международный журнал информационных технологий и энергоэффективности. – 2017. – Т.2 №2(4) с. 17-21
20. Глосарій термінів з хімії / уклад. Й. Опейда, О. Швайка ; Ін-т фізико-органічної хімії та вуглехімії ім. Л. М. Литвиненка НАН України, Донецький національний університет. — Донецьк : Вебер, 2008. — 738 с.
21. Piryonesi S. Madeh; El-Diraby Tamer E. (2020-06-01). "Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems". Journal of Transportation Engineering, Part B: Pavements.
22. Hastie, Trevor. The elements of statistical learning : data mining, inference, and prediction : with 200 full-color illustrations. Tibshirani, Robert., Friedman, J. H. (Jerome H.). New York: Springer. 2001.

23. askowiak, Pablo A.; Campello, Ricardo J. G. B. "Comparing Correlation Coefficients as Dissimilarity Measures for Cancer Classification in Gene Expression Data". Brazilian Symposium on Bioinformatics (BSB 2011): 1–8.
24. Coomans, Danny; Massart, Desire L, "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules". Analytica Chimica Acta,1982. - 136: 15–27.
25. Terrell, George R.; Scott, David W. (1992). "Variable kernel density estimation". Annals of Statistics, 1992. – 1236–1265.
26. Mills, Peter. "Efficient statistical classification of satellite measurements". International Journal of Remote Sensing, 2012.
27. Deng,H.; Runger, G.; Tuv, E. Bias of importance measures for multi-valued attributes and solutions Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN), 2011.- 293–300 с.
28. Altmann A, Tolosi L, Sander O, Lengauer T. Permutation importance:a corrected feature importance measure. Bioinformatics, 2010.
29. Tolosi L, Lengauer T, Classification with correlated features: unreliability of feature ranking and solutions.. Bioinformatics. Volume 27, Issue 14, 15 July 2011, 1986–1994 с..
30. Machine Learning Benchmarks and Random Forest Regression. Center for Bioinformatics & Molecular Biostatistics [Електронний ресурс]: Режим доступу: https://escholarship.org/uc/item/35x3v9t4
31. Principal Component Analysis [Електронний ресурс]: Режим доступу: https://www.statistixl.com/features/principal-components/
32. Source code for pyod.models.pca [Електронний ресурс]: Режим доступу: https://pyod.readthedocs.io/en/latest/_modules/pyod/models/pca.html
33. Abdi H., Williams L.J. (2010). Principal component analysis.. Wiley Interdisciplinary Reviews: Computational Statistics, 2: 433–459.
34. Outlier detection with Local Outlier Factor (LOF) [Електронний ресурс]: Режим доступу:https://scikit-learn.org/stable/auto_examples/neighbors/plot_lof_outlier_detection.html#:~:text=T

he%20Local%20Outlier%20Factor%20(LOF,lower%20density%20than%20their%20neighbors.

35. Молчание вентиляторов. Google Colab, Javascript и TensorflowJS [Електронний ресурс]: Режим доступу: https://habr.com/ru/company/avito/blog/488936/
36. Основы теории статистики : [учеб. пособие] / В. В. Полякова, Н. В. Шаброва ; М-во образования и науки Рос. Федерации, Урал. федер. ун-т. – 2-е изд., испр. и доп. – Екатеринбург : Изд-во Урал. ун-та, 2015. – 148 с.
37. ZAT Documentation [Електронний ресурс]: Режим доступу: https://supercowpowers.github.io/zat/
38. Корнієнко Б.Я. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK. Моделювання та інформаційні технології. 2018. Вип. 83. С. 34-41.
39. Корнієнко Б.Я. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів. Наукоємні технології. 2017. № 4 (36). С. 316 - 322.
40. Korniyenko B., Yudin A., Galata L. Risk estimation of information system.Wschodnioeuropejskie Czasopismo Naukowe. 2016. № 5. P. 35 - 40.
41. Корнієнко Б.Я., Юдін О.К., Снігур О.С.Безпека аутентифікації у web-ресурсах. Захист інформації. 2012. № 1 (54). С. 20 -25. DOI: 10.18372/2410-7840.14.2056 (ukr).
42. Корнієнко Б.Я., Максімов Ю.О., Марутовська Н.М. Прикладні програми управління інформаційними ризиками. Захист інформації. 2012. № 4 (57). С. 60 – 64. DOI: 10.18372/2410-7840.14.3493 (ukr).
43. Galata, L., Korniyenko, B., Yudin, A.: Research of the simulation polygon for the protection of critical information resources. In: CEUR Workshop Proceedings, Information Technologies and Security, Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017), 30 Nov 2017, Kyiv, Ukraine. vol. 2067. pp. 23–31., urn:nbn:de:0074-2067-8.

44. Корнієнко Б.Я. Безпека інформаційно-комунікаційних систем та мереж. Навчальний посібник для студентів спеціальності 125 «Кібербезпека». – К.:НАУ, 2018. – 226 с.
45. Корнієнко Б.Я., Галата Л.П. Оптимізація системи захисту інформації корпоративної мережі. Математичне та комп'ютерне моделювання. Серія: Технічні науки, Випуск 19, 2019. - С. 56-62.
46. Korniyenko B., Galata L. Implementation of the information resources protection based on the CentOS operating system. Conference Proceedings of 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON -2019) July 2 – 6, 2019, Lviv, Ukraine. - pp. 1007-1011.
47. Галата Л.П., Корнієнко Б.Я., Заболотний В.В. Математична модель протидії загрозам у системі захисту критичних інформаційних ресурсів. Наукоємні технології, Том 43, № 3, 2019. – С. 300 – 306.
48. Корнієнко Б.Я. Modeling of information security system in computer network. Безпека інформаційних систем і технологій, Том №1 (1), 2019. – С.36-41.
49. Korniyenko B., Galata L., Ladieva L. Research of Information Protection System of Corporate Network Based on GNS3. Conference Proceedings of 2019 IEEE Inernational Conference on Advanced Trends in Information Theory (IEEE ATIT - 2019) Dezember 18 – 20, 2019, Kyiv, Ukraine. - pp. 244-248.
50. Korniyenko B., Galata L., Ladieva L. Mathematical model of threats resistance in the critical information resources protection system. CEUR Workshop Proceedings, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019) Kyiv, Ukraine, November 28, 2019. Vol-2577. P.281-291.
51. Корниенко Б.Я. Кибернетическая безопасность – операционные системы и протоколы. ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2017. – 122 с.
52. Korniyenko B.Y., Galata L.P. Design and research of mathematical model for information security system in computer network. Науковий журнал «Наукоємні технології». – 2017, № 2 (34), С. 114 - 118.

53. Корниенко Б.Я. Информационная безопасность и технологии компьютерных сетей : монография. ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland. − 2016. – 102 с.
54. Korniyenko B., Galata L., Kozuberda O. Modeling of security and risk assessment in information and communication system. Sciences of Europe. – 2016. – V. 2. – No 2 (2). – P. 61 -63.
55. Korniyenko B. The classification of information technologies and control systems. International scientific journal. – 2016. –№ 2. – P. 78 - 81.
56. Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив :монографія. – К.: Вид-во Аграр Медіа Груп, 2014. – 288 с.
57. Korniyenko B., Galata L., Ladieva L. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources / B. Korniyenko, //CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018) Kyiv, Ukraine, November 27, 2018, Vol-2318, - P. 176-187, urn:nbn:de:0074-2318-4