НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.942

До захисту допущено
В. о. завідувача кафедри ММСА
_____ О.Л.Тимощук
«___» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Методи та засоби оцінки ризиків порушення захищеності в розподілених комп'ютерних системах»

Виконав:
студент ІІ курсу, групи КА-91 мп
Матюх Антон Ігорович _____

Керівник: професор кафедри ММСА
д.т.н., проф. Мухін В. Є. _____

Рецензент: доцент кафедри КТК
д.т.н., доц. Корнага Я. І. _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань
Студент _____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ММСА

_____ О. Л. Тимощук
«___» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту Матюху Антону Ігоровичу

1. Тема дисертації: «Методи та засоби оцінки ризиків порушення захищеності в розподілених комп'ютерних системах» науковий керівник дисертації Мухін Вадим Євгенович, проф, док. тех. наук., затверджені наказом по університету від «02» листопада № 31-82-с

2. Термін подання студентом дисертації: 14 грудня 2020 р.

3. Об'єкт дослідження: Історичні дані про комп'ютерні напади та інформація про те чи є програма загрозою, лог HTTP запитів.

4. Предмет дослідження: Методи прогнозування та виявлення ризиків захищеності комп'ютерних систем.

5. Перелік завдань, які потрібно розробити:

1) дослідити різні методи машинного навчання, які можна застосувати для цього типу задач, дослідити як зараз застосовується машинне навчання для аналітики логів;

2) створити програмний продукт для прогнозування ризиків захищеності комп'ютерних систем;

3) створити програмний продукт для виявлення аномалій в HTTP логу;

4) пошук даних, щоб застосувати їх програмі;

5) нормалізація та інтерпритація даних;

6) розробити стартап-проект виведення на ринок результатів дослідження;

7) розробити концептуальні висновки за результатами наукового дослідження

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

2). Робота створеного програмного продукту (рис.4.3, рис.4.4);
3). Таблиці у розділі стартап-проекту

7. Дата видачі завдання: 05 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	10.09.2020—20.09.2020
2.	Перший розділ. Огляд літературно-інформаційних джерел. Понятійно-категоріальний апарат. Характеристика об'єкта	21.09.2020—28.09.2020
3.	Другий розділ. Огляд алгоритмів класифікації та кластеризації.	28.09.2020—07.10.2020
4.	Третій розділ. Створення програмного продукту. Тестування програми	08.10.2020—19.10.2020
6.	Четвертий розділ. Стартап-проект	20.10.2020—24.10.2020
7.	Концептуальні висновки. Перспективи розвитку отриманих рішень	25.10.2020—27.10.2020

Студент _____ А. І. Матюх
(підпис)

Науковий керівник дисертації _____ В. Є. Мухін
(підпис)

РЕФЕРАТ

Магістерська дисертація: 72 с., 4 ч., 20 табл., 15 рис., 1 дод., 18 джерел.

ЗАХИЩЕНІСТЬ КОМП'ЮТЕРНИХ СИСТЕМ, ДЕРЕВА РІШЕНЬ, КЛАСТЕРИЗАЦІЯ, КЛАСИФІКАЦІЯ, XGBOOST, ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – Історичні дані про комп'ютерні напади та інформація про те чи є програма загрозою, лог HTTP запитів.

Предмет дослідження – методи прогнозування загрози ризику комп'ютерних систем та виявлення ризиків як аномалій.

Мета дослідження – проаналізувати об'єкт дослідження, побудувати і протестувати модель прогнозування та модель для виявлення аномалій.

Методи дослідження – метод прогнозування: XGBoost, метод виявлення аномалій: K-Means.

Актуальність – виявлення ризиків порушення захищеності комп'ютерних систем є дуже важливою темою, актуальність якої розвивається разом із розвитком технологій. Як для великих підприємств так і для локального використання дуже важливо вміти прогнозувати ризики порушення захищеності комп'ютерних систем.

Результати дослідження – були побудовані модель XGBoost для прогнозування ризиків захищенності та модель K-Means для виявлення аномалій у логах.

ABSTRACT

The master thesis: 72 p., 4 s., 20 tabl., 15 fig., 1 appendix, 18 references.

COMPUTER SYSTEMS SECURITY, SOLUTION TREES, CLUSTERIZATION, CLASSIFICATION, XGBOOST, HARMFUL SOFTWARE

Object of research - Historical data on computer attacks and information on whether the program is a threat, the log of HTTP requests.

The subject of research is methods for predicting the risk of computer systems and identifying risks as anomalies.

The purpose of the study is to analyze the object of study, build and test a forecasting model and a model for detecting anomalies.

Research methods - forecasting method: XGBoost, anomaly detection method: K-Means.

Relevance - Identifying the risks of security breaches of computer systems is a very important topic, the relevance of which develops with the development of technology. It is very important for both large enterprises and local use to be able to predict the risks of security breaches of computer systems.

The results of the study were the XGBoost model for predicting security risks and the K-Means model for detecting anomalies in logs.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

MSAT – інструмент безпеки Microsoft (Microsoft assessment security tool)

ІБ - інформаційна безпека

VaR – співвідношення ризику (value at risk)

DDoS – розподілена атака відмови в обслуговуванні (distributed denial-of-dervice attack)

IP – інтернет протокол (internet protocol)

IT – інформаційні технології (information technology)

ISO – міжнародна організація стандартизації (international organization for standardization)

HTTP – протокол передачі гіпертексту (hypertext transfer protocol)

ВСТУП

Ми живемо в часи, коли важко уявити собі діяльність людини, а тим паче діяльність різноманітних організацій без використання комп'ютерних систем. Наразі комп'ютерні системи допомагають людям майже у всіх робочих аспектах та використовуються у дуже різноманітних напрямках. Комп'ютерні системи кожний день зберігають та оброблюють велетенські об'єми інформації, яка може в собі містити велику кількість конфіденційної інформації, яка не повинна потрапити до стороннього кола людей, а тим паче до зловмисників, які зможуть використати її для своїх цілей. Так як наш світ постійно стрімко діджиталізується, зловмисники теж не стоять на місці, а тому з кожним роком зростає кількість комп'ютерних атак, частина з яких закінчується успіхом.

Будь-яка вдала комп'ютерна атака може стати дуже фатальною для організації. Найчастіше у сучасних реаліях метою таких атак є порушення цілісності даних, порушення цілісності комп'ютерних систем та вилучення даних компанії. Вони може нанести значний удар по репутації компанії, або по фінансам компанії, оскільки є багато випадків, коли у результаті вдалих атак у відкритий доступ потрапляло дуже багато конфіденційної інформації такої як: повна особиста інформація працівників компанії, інформація клієнтів компанії, фінансові звіти компанії та багато іншого. Аби мінімізувати ризики можливості таких випадків приймається дуже велика кількість запобіжних заходів різних типів, наймаються спеціалісти з комп'ютерної безпеки, будуються сценарії поведінки та реагування.

Метою цієї роботи є аналіз існуючих методів та засобів оцінки ризиків порушення захищеності в комп'ютерних системах та створення власної моделі на основі математичних методів та алгоритмів машинного навчання для автоматизації та спрощення виявлення ризиків порушення комп'ютерної захищеності, а також на їх основі формування та порівняння оцінки ризику.

Наукова новизна роботи полягає у тому, що запропонований метод передбачення загроз захищеності комп’ютерних систем відрізняється від відомих тим, що базується на інтеграції набору підходів до аналізу ризиків, а що дозволяє підвищити ступінь достовірності передбачення загроз та скоротити кількість ресурсів, необхідних для виконання таких задач.

Результатом роботи є створення ефективної моделі для виявлення ризиків порушення захищеності комп’ютерних систем та порівняння результату роботи цієї моделі з іншими методами аналізу та вже існуючими рішеннями.

Дана робота складається з 4 розділів. В першому розділі розглядаються поняття ризика інформаційної безпеки, існуючі проблеми галузі, методики та методи, які використовуються для аналізу ризиків. В другому розділі було описано теоретичну інформацію стосовно методів, які я використовував для аналізу. В третьому розділі було показано хід виконання мною роботи, представлення пояснення даних, які використовувались та висновки на кожному етапі виконання цієї частини. В останньому розділі було описано виконання стартап частини.

РОЗДІЛ 1 МЕТОДИ ТА ЗАСОБИ ОЦІНКИ РИЗИКІВ ПОРУШЕННЯ ЗАХИЩЕНОСТІ

1.1 Поняття ризику інформаційної безпеки, види ризиків

У широкому розумінні ризик інформаційної безпеки - ймовірність виникнення негативної події, яка принесе збитки певній організації, або фізичній особі. Математично ризик можна описати як ймовірність виникнення інциденту інформаційної безпеки помножена на розмір збитків у наслідку враження комп'ютерної системи.

Існують чотири основні положення відносно інформації та інформаційної безпеки:

а) Цілісність інформаційних даних - здатність інформації зберігати початковий вигляд і структуру як в процесі зберігання, так і після багаторазової передачі. Вносити зміни, видаляти або доповнювати інформацію можуть лише власник або користувач з легальним доступом до даних.

б) Конфіденційність - необхідність обмеження доступу до інформаційних ресурсів для певного кола осіб. У процесі дій і операцій інформація стає доступною тільки користувачам, які включені в інформаційні системи та успішно пройшли ідентифікацію.

в) Доступність інформаційних ресурсів означає, що інформація, яка знаходиться у вільному доступі, повинна надаватися повноправним користувачам ресурсів своєчасно і безперешкодно.

г) Достовірність вказує на приналежність інформації довіреній особі або власнику, який одночасно виступає в ролі джерела інформації.

Кожний інцидент пов'язаний з порушенням комп'ютерної безпеки виникає через певну комбінацію подій, далі більш детально буде розглянуто

причини види, а також наслідки порушення захищеності комп’ютерних систем. Причини інциденту інформаційної безпеки можуть бути наступними:

а) зовнішні атаки на інформаційні системи;
а) людський фактор, а саме нерегламентовані дії внутрішніх співробітників компанії;
б) наявність доступу до потенційно небезпечних об’єктів у зовнішній мережі;
в) програмне забезпечення створене для блокування, крадіжки, або видалення інформації;
г) використання неліцензійного програмного забезпечення;
д) несформована політика інформаційної безпеки компанії, або її недотримання співробітниками;
е) погана захищеність серверів, або наявність важливої інформації на публічних серверах;
ж) неточність протоколів обміну інформацією та інтерфейсів;
з) важкі умови експлуатації та розташування інформації.

Існує умовний розподіл вразливостей за 3 класами:

а) об'єктивні вразливості;
б) суб’єктивні вразливості;
в) випадкові вразливості.

Об'єктивні вразливості – вразливості, що напряму залежать від технічного побудови обладнання на об'єкті, який вимагає захисту, та його характеристик [1]. Повноцінне позбавлення від цих чинників неможливо, але їх часткове усунення досягається за допомогою інженерно-технічних рішень наступним чином:

а) Зміни, які пов'язані з технічними засобами випромінювання (електромагнітні методики, звукові варіанти, електричні (прослизання сигналів в ланцюжки електричної мережі, за наведенням на лінії і провідники, по нерівномірного розподілу струму)).

б) Інформаційні керуючі (шкідливі ПО, нелегальні програми, технологічні виходи з програм, що об'єднується терміном «програмні закладки»).

в) Створені особливостями об'єкта, який знаходиться під захистом (розташування об'єкта, організація каналів обміну інформацією (застосування радіоканалів, оренда частот або використання загальних мереж)).

г) Такі, що залежать від особливостей елементів-носіїв (деталі, що володіють електроакустичними модифікаціями, речі, які підпадають під вплив електромагнітного поля).

Випадкові вразливості – такі вразливості, які залежать від непередбачуваних обставин та особливостей оточення інформаційного середовища [2]. Їх майже неможливо передбачити в інформаційному просторі, але потрібно бути готовим до швидкого реагування на них, та їх швидкого усунення. До таких збоїв відносяться два наступні види:

а) Збої і відмови роботи систем - внаслідок несправності технічних засобів на різних рівнях обробки та зберігання інформації, несправності і старіння окремих елементів, збої різного програмного забезпечення, яке підтримує всі ланки в ланцюзі зберігання і обробки інформації, перебої в роботі допоміжного обладнання інформаційних систем.

б) Послаблюючі інформаційну безпеку фактори - пошкодження комунікацій на зразок водопостачання або електропостачання, а також вентиляції, каналізації, несправності в роботі захисних пристроїв.

Суб’єктивні вразливості – вразливості, які виникають через неправильні дії співробітників на рівні розробки систем зберігання та захисту інформації. Існує декілька підвидів таких вразливостей:

а) Неточності і грубі помилки, що порушують інформаційну безпеку - на етапі завантаження готового програмного забезпечення або

попередньої розробки алгоритмів, а також в момент його використання, на етапі управління програмами і інформаційними системами, під час користування технічної апаратурою.

б) Порушення роботи систем в інформаційному просторі - режиму захисту особистих даних, режиму збереження і захищеності, під час роботи з технічними пристроями, під час роботи з даними.

Також, хочеться описати ранжування вразливостей. Кожна вразливість повинна бути оцінена та врахована фахівцями. Саме тому важливо визначити критерії оцінки небезпеки виникнення загрози та ймовірності поломки або обходу захисту інформації. Показники рахуються за допомогою застосування ранжирування. Серед всіх критеріїв виділяють три основних:

а) Доступність - критерій, що враховує, наскільки зручно джерелу загроз використовувати певний вид вразливості, щоб порушити інформаційну безпеку. У показник входять технічні дані носія інформації (наприклад габарити апаратури, її складності і вартості, а також можливості використання для злому інформаційних систем неспеціалізованих систем і пристроїв).

б) Фатальність - характеристика, яка оцінює глибину впливу вразливості на можливість розробників та адміністраторів впоратися з наслідками створеної загрози для інформаційних систем. Якщо оцінювати тільки об'єктивні вразливості, то визначається їх інформативність - здатність передати в інше місце корисний сигнал з конфіденційними даними без його деформації.

в) Кількість - характеристика підрахунку деталей системи зберігання та реалізації інформації, яким притаманний будь-який вид вразливості в системі.

Досить важливо описати і наслідки порушення захищеності комп’ютерних систем, щоб розуміти наскільки критично вміти правильно оцінювати ризики інформаційної безпеки [3]. Прояви та масштаби збитків

можуть бути абсолютно різними, далі описано найбільш основні втрати для компанії:

а) моральний і матеріальний збиток, який був нанесений фізичним особам, чия інформація була викрадена;
б) фінансовий збиток, нанесений шахраєм в зв'язку з витратами на відновлення систем інформації;
в) матеріальні витрати, пов'язані з неможливістю виконання роботи через зміни в системі захисту інформації;
г) моральна шкода, пов'язана з діловою репутацією компанії або спричиненням порушення взаємин на світовому рівні.

1.2 Методики оцінки ризику захищеності інформаційних систем

Типова методика аналізу захищеності інформаційної системи підприємства включає:

а) вивчення вихідних даних інформаційної системи;
б) оцінку ризиків, пов'язаних із здійсненням погроз безпеки відносно ресурсів підприємства;
в) аналіз механізмів безпеки організаційного рівня, політик безпеки організації і організаційно-розпорядчої документації відносно забезпечення режиму інформаційної безпеки та оцінку їх відповідність вимогам існуючих нормативних документів, а також їх адекватності стосовно існуючих ризиків;
г) механічний аналіз конфігураційних файлів маршрутизаторів і проксі-серверів, поштових і DNS-серверів;
д) сканування зовнішніх мережевих адрес локальної мережі;
е) сканування ресурсів локальної мережі зсередини;

ж) аналіз конфігурації серверів і робочих станцій за допомогою спеціалізованих програмних агентів.

Перераховані технічні методи передбачають застосування як активного, так і пасивного тестування системи захисту. Активне тестування полягає в емуляції дій потенційного зловмисника, а пасивне передбачає аналіз конфігурацій операційної системи та додатків по шаблонах з використанням списків перевірки. Тестування може проводитися вручну або з використанням спеціалізованих програмних засобів [4].

Після того як було описано типову методику аналізу захищеності інформаційних систем важливо детально описати дані, які необхідні для аналізу захищеності та попереднього огляду:

а) точне і повне найменування об’єкту інформатизації та його призначення. Характер оброблюваної інформації і рівень її секретності, який визначається відповідно до певного переліку (державний, галузевий, відомчий, підприємницький);

б) організаційна структура об’єкту інформатизації;

в) перелік приміщень, склад комплексу технічних засобів, що входять в об’єкт інформатизації, в яких обробляється зазначена інформація. Схема розташування об’єкт інформатизації із зазначенням меж контрольованої зони;

г) загальна функціональна схема об’єкт інформатизації, включаючи схему джерел живлення і режими обробки інформації, що захищається;

д) наявність і характер взаємодії з іншими об’єктами інформатизації;

е) склад і структура системи захисту інформації на атестованому об’єкту інформатизації;

ж) перелік засобів захисту і контролю, які використовуються на атестованому об’єкту інформатизації та мають відповідний сертифікат, припис на експлуатацію;

з) відомості про розробників системи ЗІ, наявність у сторонніх розробників ліцензій на проведення подібних робіт;

и) наявність на об’єкті інформатизації служби безпеки;

к) наявність і основні характеристики фізичного захисту об'єкта;

л) наявність проектної та експлуатаційної документації на об’єкт інформатизації та інші вихідні дані по об'єкту, що впливають на інформаційну безпеку.

Для оцінки поточного стану справ із забезпеченням безпеки найбільш значимо надання перерахованих нижче відомостей про об’єкт інформатизації:

а) нормативно-розпорядча документація з проведення регламентних робіт і забезпечення політики безпеки, посадові інструкції, процедури і плани запобігання та реагування на спроби несанкціонованого доступу до інформаційних ресурсів, топологія корпоративної мережі, структура інформаційних ресурсів із зазначенням ступеня критичності або конфіденційності кожного з них, розміщення інформаційних ресурсів в інформаційних системах, організаційна структура користувачів та обслуговуючих підрозділів, розміщення ліній передачі даних, схеми і характеристики систем електроживлення і заземлення об'єктів, що використовуються системи мережевого управління та моніторингу;

б) проектна документація - функціональні схеми, опис автоматизованих функцій, опис основних технічних рішень;

в) експлуатаційна документація - керівництва користувачів і адміністраторів, які використовують програмні і технічні засоби захисту інформації.

Існують різні підходи до оцінки ризиків. Вибір підходу залежить від рівня вимог, що встановлюються в організації відповідно до режиму інформаційної безпеки, характеру загроз, що беруться до уваги і ефективності потенційних контрзаходів.

Мінімальним вимогам до режиму інформаційної безпеки відповідає базовий рівень інформаційної безпеки. Звичайною галуззю використання цього рівня є базові проектні рішення. Існує ряд стандартів і специфікацій, в яких розглядається мінімальний (типової) набір найбільш ймовірних загроз, таких як віруси, НСД і т.д. Для нейтралізації цих загроз обов'язково повинні бути прийняті контрзаходи незалежно від ймовірності їх здійснення і уразливості ресурсів [5].

Підвищені вимоги. У випадках, коли порушення режиму інформаційної безпеки несе за собою тяжкі наслідки, базовий рівень вимог до режиму інформаційної безпеки є недостатнім. Для того щоб сформувати додаткові вимоги, необхідно:

а) визначити цінність ресурсів;
б) до стандартного набору додати список загроз, актуальних для досліджуваної інформаційної системи;
в) оцінити ймовірність загроз;
г) визначити вразливість ресурсів.

Вихідними даними є результати опитування співробітників, бази даних зі статистикою по класами ризиків. В результаті виконання цього етапу повинен бути написаний документ "Аналіз ризиків".

Для базового рівня інформаційної безпеки документ буде містити розділ: "Класи ризиків, що беруться до уваги при побудові підсистеми інформаційної безпеки " [6]. Для підвищеного рівня ІБ документ буде містити розділи:

а) оцінка значущості інформаційних ресурсів;
б) потенційні шляхи порушення режиму інформаційної безпеки, приведена модель загроз;
в) модель порушника за обраними класами загроз;
г) оцінка параметрів загроз і вразливих місць інформаційної системи.

Існує чотири підходи щодо управління ризиками:

а) Зменшення ризику. Велика кількість ризиків може бути значно зменшена як наслідок шляхом використання досить простих і дешевих контрзаходів (як приклад управління парольними параметрами).

б) Ухилення від ризику. Від певних класів ризиків можна ухилитися. Як приклад можна розглянути організацію Web-сервера за межами локальної мережі, що дозволяє уникнути ризику НСД в локальну мережу з боку Web-клієнтів.

в) Зміна характеру ризику. При неможливості ухилиення від ризику або його значного зменшення, можна використати певні заходи страховки.

г) Прийняття ризику. Багато ризики не можуть бути зменшені до незначної кількості. На практиці, після прийняття стандартного набору контрзаходів, ряд ризиків зменшується, але залишається все ще значущим.

Дуже важливо знати залишкову величину ризику. Вихідними даними є результати опитування співробітників, експертні оцінки, можливості застосування стандартних підходів до управління ризиками [7]. В результаті виконання етапу для прийнятих до уваги ризиків повинна бути запропонована стратегія управління, що викладається в документі "Управління ризиками":

а) виділення ризиків, рівень яких неприпустимо великий;

б) стратегія управління ризиками;

в) вибір варіанту контрзаходів.

1.3 Методи оцінки ризику захищеності інформаційних систем

Існує дуже багато методів та підходів до аналізу ризиків захищеності інформаційних систем. Далі будуть розглянуті основні методи аналізу ризиків захищеності інформаційних систем.

а) Табличний метод

Найбільшого поширення серед методів оцінки ризиків отримав метод «матриці ризиків». Це доволі легкий метод аналізу ризиків. В процесі оцінки, експертами визначається ймовірність виникнення кожного ризику і розмір пов'язаних з ним втрат (вартість ризику). Причому оцінювання проводиться за шкалою з трьома градаціями: «висока», «середня», «низька». На базі оцінок для окремих ризиків виставляється оцінка системі в цілому (у вигляді клітинки матриці), а самі ризики ранжуються. Дана методика дозволяє швидко і коректно провести оцінку. Але, на жаль, дати інтерпретацію отриманих результатів не завжди можливо.

б) Метод оцінки на основі нечіткої логіки

В даній області розроблено механізм отримання оцінок ризиків на основі нечіткої логіки, який дозволяє замінити наближені табличні методи грубої оцінки ризиків сучасним математичним методом, адекватним рішенням задачі.

Механізм оцінювання ризиків на основі нечіткої логіки є експертною системою, в якій базу знань становлять правила, що відображають логіку взаємозв'язку вхідних величин і ризику. У найпростішому випадку це «таблична» логіка, в загальному випадку більш складна логіка, яка відображає реальні взаємозв'язки, що можуть бути формалізовані за допомогою правил виду «Якщо ..., то» [8].

Крім того, механізм нечіткої логіки вимагає формування оцінок ключових параметрів та подання їх у вигляді нечітких змінних. При цьому необхідно враховувати безліч джерел інформації і якість самої інформації. У загальному

випадку це досить складне завдання. Однак в кожному конкретному випадку можуть бути знайдені і формально обґрунтоване її рішення.

в) Метод вагових коефіцієнтів

Вхідними даними для проведення оцінки та аналізу служать результати анкетування суб'єктів відносин, призначені для з'ясування спрямованості їх діяльності, передбачуваних пріоритетів цілей безпеки, завдань, що вирішуються автоматизованою системою і умов розташування та експлуатації об'єкта. Завдяки такому підходу можливо:

1) встановити пріоритети цілей безпеки для суб'єкта відносин;
2) визначити перелік актуальних джерел загроз;
3) визначити перелік актуальних вразливостей;
4) оцінити взаємозв'язок загроз, джерел загроз та вразливостей;
5) визначити перелік можливих атак на об'єкт;
6) описати можливі наслідки реалізації загроз.

Результати проведення оцінки та аналізу можуть бути використані при обранні адекватних оптимальних методів парирування загрозам, а також при аудиті реального стану інформаційної безпеки об'єкта з метою його страхування. При визначенні актуальних загроз, експертно-аналітичним методом визначаються об'єкти захисту, схильні до дії тієї або іншої загрози, характерні джерела цих загроз і вразливості, що сприяють реалізації загроз.

На підставі аналізу складається матриця взаємозв'язку джерел загроз і вразливостей, з якої визначаються можливі наслідки реалізації загроз (атаки) і обчислюється коефіцієнт небезпеки цих атак як добуток коефіцієнтів небезпеки відповідних загроз і джерел загроз, визначених раніше.

1.4 Засоби оцінки ризику захищеності інформаційних систем

Цінність інструментального засобу аналізу ризиків визначається в першу чергу тією методикою, яка покладена в його основу. В наш час популярність здобули такі програмні продукти, як Risk Watch, CRAMM, COBRA, «Авангард», Тріфо, КОНДОР і ряд інших. Ці програмні продукти базуються на різних підходах до аналізу ризиків і вирішення різних аудиторських завдань [9]. Можна виділити наступні підходи розробників програмних засобів аналізу ризиків до вирішення поставленого завдання:

а) отримання оцінок ризиків тільки на якісному рівні;

б) висновок кількісних оцінок ризиків на базі якісних, отриманих від експертів;

в) отримання точних кількісних оцінок для кожного з ризиків;

г) отримання оцінок механізмом нечіткої логіки.

Зупинимося дещо докладніше на аналізі цих продуктів.

Програмне забезпечення RiskWatch є потужним засобом аналізу та управління ризиками, більш орієнтованим на точну кількісну оцінку співвідношення втрат від загроз безпеки і витрат на створення системи захисту. Треба також зазначити, що в цьому продукті ризики в сфері інформаційної та фізичної безпеки комп'ютерної мережі підприємства розглядаються спільно. У сімейство RiskWatch входять наступні програмні продукти: для фізичних методів захисту, для інформаційних ризиків, для оцінки вимог до стандарту ISO 17799 [10].

В основу продукту RiskWatch покладено методику аналізу ризиків, яка складається з чотирьох етапів:

а) перший - визначення предмета дослідження. Тут описуються такі параметри, як тип організації, склад досліджуваної системи, базові вимоги в області безпеки;

б) другий - введення даних, що характеризують основні параметри системи. На цьому етапі докладно описуються ресурси, втрати і класи інцидентів. Останні виводяться шляхом зіставлення категорії втрат і категорії ресурсів. Крім того, задаються частота виникнення кожної з виділених загроз, ступінь уразливості і цінність ресурсів. Все це використовується в подальшому для розрахунку ефекту від впровадження засобів захисту;

в) третій - кількісна оцінка. На цьому етапі розраховується профіль ризиків, і вибираються заходи забезпечення безпеки. Фактично ризик оцінюється за допомогою математичного очікування втрат за рік. Ефект від впровадження засобів захисту кількісно описується за допомогою показника ROI (Return on Investment віддача від інвестицій), який показує віддачу від зроблених інвестицій за певний період часу;

г) четвертий - генерація звітів.

Програмне забезпечення RiskWatch має велику кількість переваг, а до недоліків продукту можна віднести його відносно високу вартість.

CRAMM - інструментальне засіб, що реалізує однойменну методику, яка була розроблена компанією BIS Applied Systems Limited за замовленням британського уряду. Метод CRAMM дозволяє проводити аналіз ризиків і вирішувати ряд інших аудиторських завдань таких як: обстеження інформаційної системи, проведення аудиту відповідно до вимог стандарту BS 7799, розробка політики безпеки [11].

Ця методика спирається на оцінки якісного характеру, які одержуються від експертів, але на їх базі будує вже кількісну оцінку. Метод є універсальним і підходить для великих та дрібних організацій як урядового, так і комерційного сектору. Кваліфіковане використання методу CRAMM дозволяє отримати дуже гарні результати, найбільш важливим з яких, мабуть, є можливість економічного забезпечення організації для забезпечення інформаційної безпеки неперервності бізнесу. Економічно обґрунтована

стратегія управління ризиками дозволяє в кінцевому підсумку уникати невиправданих витрат.

CRAMM передбачає поділ всієї процедури на три послідовні етапи:

а) Завданням першого етапу є визначення достатності для захисту системи застосування засобів базового рівня, що реалізують традиційні функції безпеки, або необхідність проведення більш детального аналізу.

б) На другому етапі проводиться ідентифікація ризиків і оцінюється їх величина.

в) На третьому етапі вирішується питання про вибір адекватних контрзаходів. Для кожного етапу визначаються набір вихідних даних, послідовність заходів, анкети для проведення інтерв'ю, списки перевірки і набір звітних документів.

Переваги методу CRAMM: добре структурований і широко випробуваний метод аналізу ризиків; може використовуватися на всіх стадіях проведення аудиту безпеки інформаційних систем; в основі програмного продукту лежить об'ємна база знань по контрзаходам в області інформаційної безпеки, гнучкість і універсальність даного методу дозволяють його використовувати для аудиту інформаційної системи будь-якого рівня складності і призначення; даний метод дозволяє розробляти план безперервності бізнесу. До недоліків методу CRAMM можна віднести наступне: для його використання потрібно висококваліфікований аудитор; аудит за цим методом процес досить складний і може займати місяці безперервної роботи; генерує велику кількість паперової документації, яка не завжди виявляється корисною на практиці; неможливо доповнити базу знань CRAMM, що викликає певні труднощі при адаптації цього методу до потреб конкретної організації [12].

Система COBRA є засобом аналізу ризиків та оцінки відповідності інформаційної системи стандарту ISO 17799. Дана система реалізує методи кількісної оцінки ризиків, а також інструменти для консалтингу та проведення

оглядів безпеки. У систему COBRA закладені принцип побудови експертних систем, велика база знань з питань загроз і вразливостей, велика кількість анкет, має великий успіх у застосовуються на практиці.

Програмний продукт КОНДОР + дозволяє фахівцям (IT-менеджерам, офіцерам безпеки) перевірити політику інформаційної безпеки компанії на відповідність вимогам ISO 17799. КОНДОР + включає в себе більше 200 питань, відповівши на які фахівець отримує детальний звіт про стан існуючої політики безпеки, а також модуль оцінки рівня ризиків відповідності вимогам ISO 17799. У звіті відображаються всі положення політики безпеки, які відповідають і не відповідають стандарту, а також існуючий рівень ризику невиконання вимог політики безпеки відповідно до стандарту. Елементам, які не виконуються, надаються коментарі та рекомендації експертів. За бажанням фахівця, який працює з програмою, можуть бути обрані генерація звіту, наприклад, по якомусь одному або декількох розділів стандарту ISO 17799, загальний детальний звіт з коментарями, загальний звіт про стан політики безпеки без коментарів для подання керівництву [13]. Всі варіанти звітів супроводжуються діаграмами. КОНДОР + дає можливість фахівцеві відстежувати вносяться на основі виданих рекомендацій зміни в політику безпеки, поступово приводити її в повну відповідність до вимог стандарту. Дана система реалізує метод якісної оцінки ризиків за рівневою шкалою ризиків: високий, середній, низький.

ГРИФ - це програмний комплекс аналізу та контролю ризиків інформаційних систем компаній. У ньому розроблено гнучке і, незважаючи на прихований від користувача складний алгоритм, що враховує більш 100 параметрів, максимально просте у використанні програмне рішення, основне завдання якого дати можливість IT-менеджеру самостійно (без залучення сторонніх експертів) оцінити рівень ризиків в інформаційній системі і ефективність існуючої практики забезпечення безпеки компанії. Даний комплекс робить оцінку ризиків за різними інформаційними ресурсами, підраховує сумарний ризик за ресурсами компанії, а також веде підрахунок

співвідношення шкоди і ризику, а також видає недоліки існуючої політики безпеки. В основі продукту ГРИФ закладена методика аналізу ризиків, яка складається з п'яти етапів:

а) на першому етапі визначається повний список інформаційних ресурсів, які мають цінність у досліджуваної автоматизованої паспортної системи, які об'єднуються в мережеві групи;

б) на другому здійснюється введення в систему всіх видів інформації, що надає цінність для інформаційної системи. Введені групи цінної інформації повинні бути розміщені користувачем на раніше зазначених на попередньому етапі об'єктах зберігання інформації (серверах, робочих станціях і т. Д.); вказується збиток по кожній групі цінної інформації, розташованої на відповідних ресурсах, за всіма видами загроз (конфіденційності, цілісності, відмови обслуговування);

в) на третьому етапі спочатку відбувається визначення всіх видів користувацьких груп, потім визначається, до яких груп інформації на ресурсах має доступ кожна з груп користувачів. На закінчення визначаються види (локальний і / або віддалений) і права (читання, запис, видалення) доступу користувачів до всіх ресурсів, що містять цінну інформацію;

г) на четвертому вказується, якими засобами захисту інформації захищені цінна інформація на ресурсах і робочі місця груп користувачів. Вводиться інформація про витрати на придбання всіх засобів захисту інформації та щорічних витратах на їх технічну підтримку, а також на супровід системи інформаційної безпеки;

д) на фінальному етапі необхідно відповісти на список запитань з політики безпеки, реалізованої в системі, що дозволяє оцінити реальний рівень захищеності системи і деталізувати оцінки

ризиків. Цей етап необхідний для отримання достовірних оцінок існуючих в системі ризиків.

Звітна система програмного комплексу ГРИФ складається з трьох частин: перша «Інформаційні ризики ресурсів», друга «Співвідношення збитків і ризику», третя «Загальний висновок про існуючі ризики інформаційної системи» [14].

1.5 Висновки до розділу 1

У даному розділі було досліджено основні поняття про ризики захищеності комп'ютерних систем. Також було розглянуто різні методології, методи та механізми для захисту комп'ютерних систем від різних типів загроз інформаційної безпеки. Було порівняно різні методології для оцінки ризиків, а також розроблено детальне порівняння існуючих програм різних типів для аналізу ризиків захищеності комп'ютерних систем. Кожна з цих програм має певні переваги та недоліки, які було описано у розділі. Також для кожного з програмних забезпечень було детально описано про те, які методології використовуються при її використанні та для яких ситуацій вона підходить, а також було описано характер результатів роботи програм.

РОЗДІЛ 2 МОДЕЛЬ ВИЯВЛЕННЯ ТА ПЕРЕДБАЧЕННЯ РИЗИКІВ ЗАХИЩЕНОСТІ

2.1 Основні поняття

Для виявлення та передбачення ризиків захищеності я буду використовувати декілька підходів та моделей. Так як дані різні, потрібно використовувати і різні алгоритми та підходи до вирішення цієї проблематики. Першою частиною моєї роботи буде задача прогнозування комп'ютерних атак, які можуть спричиняти пошкодження системі. У цій частині я буду виявляти загрози комп'ютерної безпеки на основі великого об'єму даних, що в мене є та таким чином прогнозувати за рядом факторів чи є певний файл загрозою для системи. У другій частині своєї роботи я буду знаходити загрози у логу HTTP. Загрозою вважаються аномальні HTTP запити, які я буду знаходити за допомогою методу кластеризації.

Особливість обраного підходу в тому, що комплексно вирішується проблема захищеності комп'ютерних систем та поєднується 2 методи, що можуть з різних сторін відловлювати небезпечні для системи події. Методи є різними з точки зору логіки роботи та різними з точки зору використання даних тому покривають собою більше простору загроз та модель є більш універсальною з точки зору практичного використання.

Спочатку розглянемо метод, що допоможе нам прогнозувати та виявляти ризики комп'ютерних атак. Щоб це зробити треба класифікувати новий об'єкт за його характеристиками, тобто віднести його до певної групи. Для цього я використовую алгоритм XGBoost (градієнтне підсилювання для дерев рішень). В основі XGBoost лежить алгоритм градієнтного бустингу дерев рішень. Метод найшвидшого бустингу - така техніка машинного навчання для задач класифікації і регресії, що будує модель передбачення в формі ансамблю слабких передбачаючих моделей, найчастіше дерев рішень. Навчання ансамблю проводиться послідовно відмінно, наприклад від беггінга.

На кожній ітерації обчислюються відхилення прогнозів вже навченого ансамблю на навчальній вибірці. Наступна модель, що буде додана в ансамбль буде передбачати дані відхилення. Таким чином, якщо додати передбачення нового дерева до передбачень навченого ансамблю ми можемо зменшити середнє відхилення моделі, що є метою оптимізаційної задачі. Нові дерева додаються в ансамбль до тих пір, доки помилка зменшується, або поки не виконується одне з правил "ранньої зупинки".

Тепер розглянемо метод, що допоможе нам знайти аномалії у наших даних. Це дозволить знайти підозрілу активність виходячи в логу HTTP запитів. Для цього я буду використовувати алгоритм k-means. k- means кластеризація - це метод векторного квантування, спочатку отриманий з обробки сигналів, що спрямований на розбиття n спостережень на k кластерів, в яких кожне спостереження належить до кластера з найближчим середнім значенням (центри кластера або центроїд кластера), що слугує прототипом кластеру. Це призводить до розподілу простору даних на комірки Вороного. Він популярний для кластерного аналізу при видобутку даних. Кластеризація k- means мінімізує дисперсії всередині кластеру (квадратні відстані Евкліда), але не регулярні відстані Евкліда, що було б більш складною задачею Вебера: середнє значення оптимізує квадратичні помилки, тоді як лише геометрична медіана мінімізує евклідові відстані.

Алгоритм методу передбачення та ідентифікації загроз захищеності комп’ютерних систем:

а) Завантажуємо лог файл з історією HTTP запитів. На основі цих даних ми будемо знаходити аномалії та виділяти їх як ризики захищеності комп’ютерних систем.

б) Після цього формуємо метрики, що пов’язані з часовими рамками та на їх основі за допомогою методу кластеризації K-Means отримаємо аномалії, які будемо вважати загрозами захищеності комп’ютерних систем.

в) Завантажуємо та обробляємо файл з історичними даними в якому знаходяться змінні та значення, що їм відповідають, а також розмічені дані відносно того є файл ризиком для комп’ютерної системи чи ні.

г) Навчаємо модель на історичних даних та тренуємо класифікатор для того, щоб він передбачував виникнення ризиків захищеності комп’ютерних систем на основі параметрів файлів.

д) На основі двох методів будуємо модель для комплексного та оперативного реагування на комп’ютерні загрози базуючись на методах машинного навчання. Таким чином ця модель вміщує в себе як аналіз та прийняття рішень на розмічених даних так і на даних без розмітки, що робить її більш універсальною та прикладною.

2.2 Детальний опис алгоритму XGBoost

Так як алгоритм базується на деревах рішень розглянемо що це таке, а також алгоритм побудови дерев рішень вже після цього перейдемо до того, що таке бустинг, а після цього розглянемо саме той алгоритм бустингу, який я використовував у своїй роботі.

Дерево рішень - рішення задачі навчання з вчителем, засноване на тому, як вирішує завдання прогнозування людина. У загальному випадку - це k-ічне дерево з вирішальними правилами в нелистових вершинах (вузлах) і деякому заключенні про цільову функцію в листових вершинах (прогнозом). Вирішальне правило - деяка функція від об'єкта, що дозволяє визначити, в яку з дочірніх вершин потрібно помістити даний об'єкт. У листових вершинах можуть перебувати різні об'єкти: клас, який потрібно присвоїти об'єкту, що

туди потрапив (в завданні класифікації), ймовірності класів (в завданні класифікації), безпосередньо значення цільової функції (в завданні регресії).

Розглянемо алгоритм побудови:

а) Перевірити критерій зупинки алгоритму. Якщо він виконується, вибрати для вузла прогноз, що видається, це можна зробити декількома способами.

б) Інакше потрібно розбити множину на декілька множин, що не перетинаються. У загальному випадку в вершині t задається вирішальне правило Q_t (x), що приймає деякий діапазон значень. Цей діапазон розбивається на Rt непересічних множин об'єктів, S_1, S_2, ..., S_{Rt}, де R_t - кількість нащадків у вершини, а кожне S_i - це безліч об'єктів, які потрапили в і-го нащадка.

в) Множину у вузлі розбивається відповідно до обраного правила, для кожного вузла алгоритм запускається рекурсивно.

Далі поговоримо про те, що таке вирішуючі правила та як обрати оптимальне вирішуюче правило. Найчастіше в якості Q_t (x) беруть просто одну з ознак, тобто $x^{i(t)}$.

Традиційні розбиття на діапазони:

а) $S_t(j) = \{x \in \mathbb{X}: h_j \leq x^{i(t)} \leq h_{j+1}\}$ для обраних $h_1,\ldots,h_{j+1}$

б) $S_t(1) = \{x \in \mathbb{X}: \langle x, v\rangle \leq 0\}; S_t(2) = \{x \in \mathbb{X}: \langle x, v\rangle > 0\}$ перевірка кута нахилу.

в) $S_t(1) = \{x \in \mathbb{X}: \rho(x, x_0) \leq h\}; S_t(2) = \{x \in \mathbb{X}: \rho(x, x_0) > h\}$, де відстань ρ визначено в деякому метричному просторі (наприклад, ρ (x, y) = | x-y |).

г) $S_t(1) = \{x \in \mathbb{X}: x^{i(t)} \leq h\}; S_t(2) = \{x \in \mathbb{X}: x^{i(t)} > h\}$ - предикати, ⟨x, v⟩ - скалярний добуток векторів.

В цілому, взяти можна будь-які вирішальні правила, але краще - інтерпретовані, так як їх легше налаштовувати. Особливого сенсу брати щось складніше предикатів немає, так як вже з їх допомогою можна отримати

дерево з 100% -й точністю на навчальній вибірці (але при цьому і швидше за все перенавчитися) [15].

Зазвичай для побудови дерева вибирається ціла родина вирішальних правил. Щоб знайти серед них оптимальне для кожного конкретного вузла, потрібно ввести деякий критерій оптимальності. Для цього вводять деяку міру I (t) вимірювання того, наскільки розкидані об'єкти (регресія) або перемішані класи (класифікація) в деякому вузлі t. Ця міра називається критерієм інформативності. Потім для кожного варіанта вирішального правила підраховується міра того, наскільки будуть розкидані об'єкти (регресія) або перемішані класи (класифікація) при такій розбивці:

$$\Delta I(X_t, t) = I(X_t, t) - \sum_{i=1}^{R} I\left(X_{t_i}, t_i\right) \frac{N(t_i)}{N(t)},$$

де R - на скільки вузлів розбивається вузол, t - поточний вузол, t_1, ..., t_R - вузли-нащадки, що виходять при обраному розбитті, N (t_i) - кількість об'єктів навчальної вибірки, що потрапляють в нащадок i, N (t) - потрапили в поточний вузол, X_{ti} - об'єкти, що потрапили в t_i-ую вершину.

Тепер розглянемо можливі критерії зупинки:

а) обмеження максимальної глибини дерева;

б) обмеження мінімального числа об'єктів в листі;

в) обмеження максимальної кількості листя в дереві;

г) зупинка в разі, якщо всі об'єкти в вершині відносяться до одного класу;

д) вимога, що Information gain при дробленні поліпшувався як мінімум на s відсотків.

Після того як ми розглянули дерева та поговорили про алгоритми їх побудови проговоримо що таке бустинг та детально про алгоритм XGBoost. Бустинг є жадібним алгоритмом побудови композиції алгоритмів. Основна ідея полягає в тому, щоб, маючи безліч відносно слабких алгоритмів навчання, побудувати їх хорошу лінійну комбінацію. Він схожий на беггінг тим, що

базовий алгоритм навчання фіксований. Відмінність полягає в тому, що навчання базових алгоритмів для композиції відбувається ітеративно, і кожен наступний алгоритм прагне компенсувати недоліки композиції всіх попередніх алгоритмів.

На прикладі бустинга стало ясно, що гарною якістю можуть володіти як завгодно складні композиції класифікаторів, за умови, що вони правильно налаштовуються. Це розвіяло існуюче довгий час уявлення про те, що для підвищення узагальнюючої здатності необхідно обмежувати складність алгоритмів [16].

Згодом цей феномен бустинга отримав теоретичне обґрунтування. Виявилось, що зважене голосування не збільшує ефективну складність алгоритму, а лише згладжує відповіді базових алгоритмів. Ефективність бустинга пояснюється тим, що в міру додавання базових алгоритмів збільшуються відступи навчальних об'єктів. Причому бустинг продовжує розсовувати класи навіть після досягнення безпомилкової класифікації навчальної вибірки.

Загальна схема бустинга:

а) Шуканий ансамбль алгоритмів має вигляд:

$$a(x) = (\sum_{t=1}^{T} \alpha_t b_t(x)),$$

де b_t – базові алгоритми.

б) Ансамбль будується ітеративно, оптимізуючи на кожному кроці функціонал Q_t, що дорівнює кількості помилок поточної композиції на навчальній вибірці.

в) При додаванні додатку $\alpha_t b_t(x)$ в суму, функционал Q_t оптимізується лише за базовим алгоритмом $b_t(x)$ і коефіцієнту α_t при ньому, усі попередні доданки вважаються фіксованими.

г) Функціонал Q_t має вигляд суми по об'єктах навчальної вибірки порогових функцій виду:

$$\left[y_i \sum_{j=1}^{t} \alpha_j b_j(x_i) < 0\right],$$

що має настурний сенс "поточна композиція помиляється на об'єкті з номером і". Кожний такий доданок має вигляд "сходинки" і є розривної функцією. Для спрощення рішення задачі оптимізації така порогова функція замінюється на безперервно диференційовану оцінку зверху. У підсумку виходить новий функціонал $\widehat{Q_t} \geq Q_t$, мінімізація якого призводить до мінімізації вихідного функціоналу Q_t.

Проблема багатьох алгоритмів побудови дерев у тому, що в них не приділяється належної уваги регуляризації. У класичному градієнтному бустингу застосовуються такі заходи:

а) обмеження на структуру дерева: максимальна глибина (max_depth), мінімальне число об'єктів в листі (min_samples_leaf);
б) контролювання темпу навчання (learning_rate);
в) збільшення "несхожості" дерев за рахунок рандомізації, як у випадковому лісі;
г) Xgboost використовує ще більше параметрів для регуляризації базових дерев.

Цільова функція для оптимізації в Xgboost складається з двох складових: специфічною функції втрат і регуляризатора для кожного з K дерев, де f_k - прогноз k -го дерева:

$$obj(\theta) = \sum_{i}^{\ell} l(y_i - \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k).$$

Функція втрат залежить від розв'язуваної задачі (Xgboost адаптований під завдання класифікації, регресії і ранжирування, а регуляризатора виглядає наступним чином:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2,$$

де перший додаток штрафує модель за велику кількість листів T, а другий контролює суму ваг моделі в листях.

2.3 Детальний опис алгоритму k-means

Метод k-середніх - це метод кластерного аналізу, мета якого є поділ m спостережень (з простору) на k кластерів, при цьому кожне спостереження відноситься до того кластеру, до центру (центроїду) якого воно найближче.

В якості міри близькості використовується Евклідова відстань:

$$\rho(x, y) = \| x - y \| = \sqrt{\sum_{p=1}^{n} \left(x_p - y_p\right)^2},$$

де x,y Є R^n.

Отже, розглянемо ряд спостережень ($x^{(1)}$, $x^{(2)}$, …, $x^{(m)}$), x^j Є R^n. Метод k-середніх розділяє m спостережень на k груп (або кластерів) (k ≤ m), щоб мінімізувати сумарне квадратичне відхилення точок кластерів від Центроїд цих кластерів:

$$min \left[\sum_{i=1}^{k} \Sigma_{x^{(j)} \epsilon S_i} \|x^{(j)} - \mu_i\|^2\right],$$

де x^j Є R^n, μ_i Є R^n, μ_i – центроїд для кластеру S_i.

Отже, якщо міра близькості до центроїда визначена, то розбиття об'єктів на кластери зводиться до визначення центроїд цих кластерів. Число кластерів k задається дослідником заздалегідь. Розглянемо початковий набір k середніх

(Центроїд) в кластерах. На першому етапі центроїди кластерів вибираються випадково або за певним правилом (наприклад, вибрати центроїди, максимізує початкові відстані між кластерами). Відносимо спостереження до тих кластерів, чиє середнє (центр ваги) до них найближче [17]. Кожне спостереження належить тільки до одного кластеру, навіть якщо його можна віднести до двох і більше кластерам. Потім центр ваги кожного i-го кластера переобчислюють за таким правилом:

$$\mu_i = \frac{1}{s_i}\sum_{x^{(j)}\epsilon s_i} x^{(j)}.$$

Таким чином, алгоритм k-середніх полягає в перерахунку на кожному кроці центроїда для кожного кластера, отриманого на попередньому кроці. Алгоритм зупиняється, коли значення не змінюються [18].

Важливо: неправильний вибір початкового числа кластерів k може привести до некоректних результатів. Саме тому при використанні методу k-середніх важливо спочатку провести перевірку відповідного числа кластерів для даного набору даних.

Отже, ще раз підкреслимо деякі особливості методу k-середніх:

а) як метрика використовується Евклідова відстань;

б) число кластерів заздалегідь вибирається дослідником заздалегідь;

в) якість кластеризації залежить від початкового розбиття.

2.4 Висновки до розділу 2

У даному розділі було розглянуто алгоритми машинного навчання,

які будуть використовуватися у моїй практичній частині виконання дипломної роботи. Для кожного з алгоритмів було розписано математичні моделі, що використовуються в них. У першій частині було розглянуто алгоритм класифікації. Перед тим як розписувати сам алгоритм було приведені пояснення загальних термінів, що в ньому використовуються, а саме пояснення того, що так дерева, пояснення того, що таке бустинг та як він працює і вже потім детально розписано алгоритм роботи. У другій частині було розглянуто алгоритм кластеризації, його математичне обґрунтування та деякі зауваження щодо використання.

РОЗДІЛ 3 АНАЛІЗ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Класифікація загроз комп’ютерної небезпеки за допомогою алгоритму XGboost

Як вхідні дані була обрана вибірка, яка в собі містить дані про різні види комп’ютерних загроз за великий проміжок часу та розмітку до якого вигляду загрози відноситься, або не відноситься певний файл. Потрібно розуміти, що передбачення буде базуватися опираючись на змінні, якими у даному випадку є певні характеристики файлу. Тому дуже важливо отримати якомога більше інформації про кожну змінну.

Опишемо кожну зі змінних, що буде використовуватись при аналізі вхідних даних для передбачення загрози захищеності комп’ютерної системи. Покажемо як виглядає частина наших даних та напишемо про значення кожної змінної (рис. 3.1).

	FileName	DebugSize	IatRVA	ExportSize	ImageVersion	ResourceSize	VirtualSize2	NumberOfSections	Class
4120	Unknown	0	102400	0	600000	45568	2884	8	Rbot
4121	Unknown	0	29932	0	600000	20280	4894	5	Rbot
4122	Unknown	0	25820	0	0	8912	2654	4	Rbot
4123	Unknown	0	90112	0	100000	0	12	8	Rbot
4124	Unknown	0	229376	0	100000	0	264	8	Rbot

Рисунок 3.1 – Приклади даних та змінних

Пояснення кожної змінної:

а) Debug Size: Файли містять необов’язковий каталог налагодження, який вказує, яка форма інформації присутня і де вона знаходиться. Поле, яке враховується, - це розмір.

б) latRVA: Відносна віртуальна адреса у файлі зображення, що адресує елемент після його завантаження в пам’ять., Із відніманням з нього базової адреси зображення. RVA елемента

майже завжди відрізняється від його положення у файлі на диску.

в) Export Size: Експортний розмір інформації про таблицю каталогів.

г) Image Version: Версія зображення, яка використовується для обробки операційною системою.

д) TResource Size: Таблиця каталогів ресурсів має формат зміщення, розмір і поле. Використовуване поле - це розмір ресурсу.

е) Virtual Size: Віртуальний розмір, зайнятий конкретною вибіркою.

ж) Number Of Sections: Основна одиниця коду або даних у портативному виконуваному файлі. Віртуальний розмір, зайнятий конкретною вибіркою.

Після того, як ми ознайомились з кожною змінною, подивимось на статистичні показники кожної змінної (рис. 3.2).

	DebugSize	IatRVA	ExportSize	ImageVersion	ResourceSize	VirtualSize2	NumberOfSections
count	4125.000000	4.125000e+03	4.125000e+03	4.125000e+03	4.125000e+03	4.125000e+03	4125.000000
mean	12.713697	3.550893e+05	4.562058e+05	4.539298e+06	1.359324e+05	3.422452e+04	3.796606
std	16.822077	1.637739e+07	2.068512e+07	9.467515e+07	4.573090e+06	7.188397e+04	1.006568
min	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.000000e+00	2.000000
25%	0.000000	2.078400e+04	0.000000e+00	1.000000e+05	0.000000e+00	1.784000e+03	3.000000
50%	0.000000	6.212800e+04	0.000000e+00	5.010000e+05	3.360000e+02	5.004000e+03	4.000000
75%	28.000000	1.064960e+05	8.700000e+01	5.010000e+05	6.824000e+03	7.782400e+04	4.000000
max	56.000000	1.051890e+09	9.395241e+08	2.152012e+09	2.073968e+08	1.737006e+06	11.000000

Рисунок 3.2 – Статистичні характеристики змінних

Було отримано певну додаткову інформацію про кожну змінну та розуміння про її природу, також ми отримали певне розуміння про всю вибірку в цілому.

Далі було побудовано матрицю, яка відображує кореляцію між змінними для того, щоб зрозуміти чи є сильно корелюючі змінні в моделі. Це робиться для того, щоб не допустити мультиколінеарність, яка може збільшити дисперсію оцінок. Для кожної пари було обраховано коефіцієнти Пірсона (рис. 3.3).

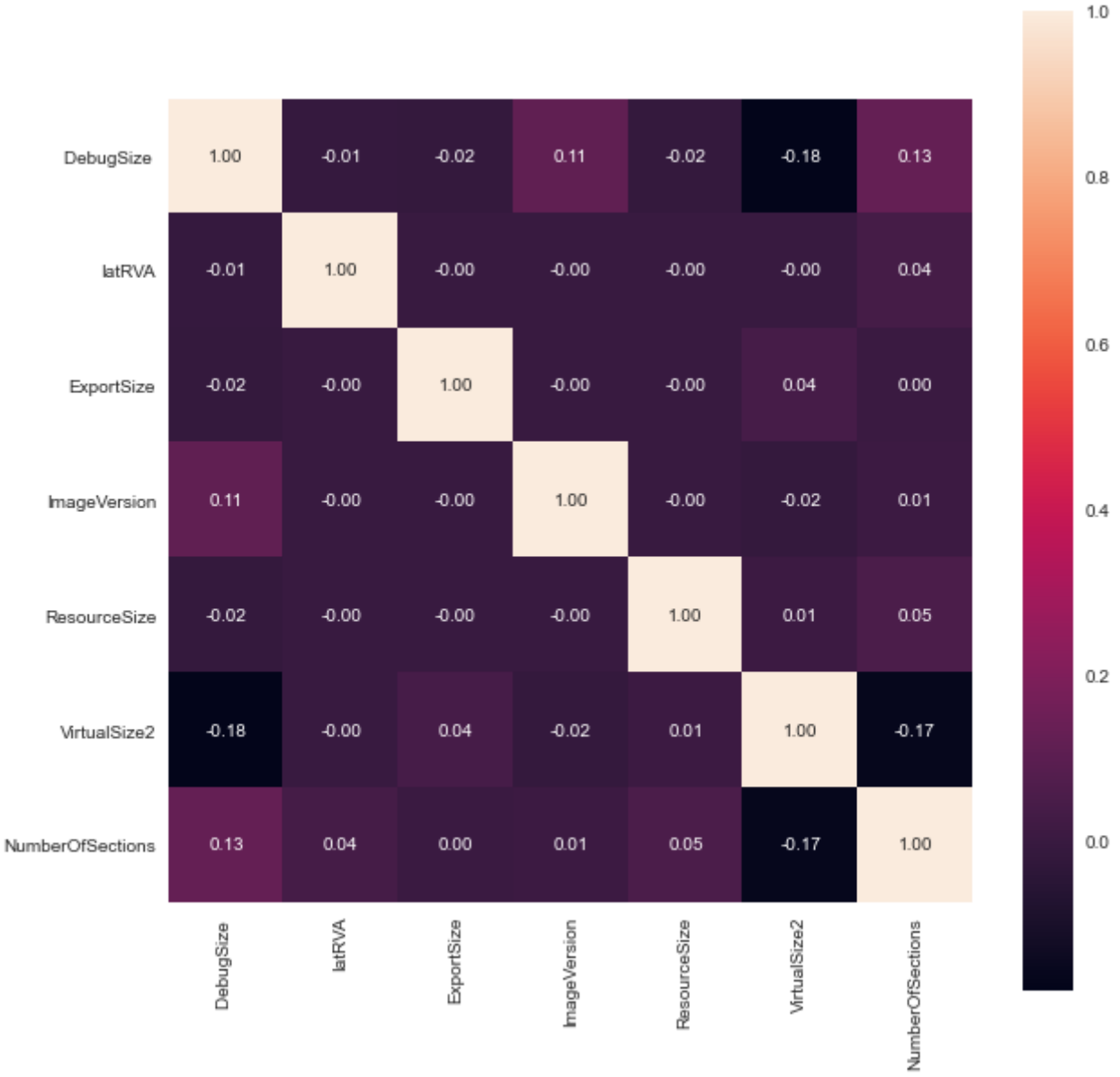

Рисунок 3.3 – Кореляційна матриця для змінних

З отриманої кореляційної матриці можна побачити, що максимальний за модулем коефіцієнт кореляції дорівнює 0.18 і є низьким. Теоретично

вважається, що коефіцієнти кореляції менше ніж 0.7 є допустимими та не сприяють виникненню мультиколінеарності.

Можна побачити, що змінні, які є параметрами вибірки мають різний порядок, тому було прийнято рішення нормалізувати змінні. Ми проведемо розбиття даних у відношенні 70% та 30%, де на 70% даних ми будемо навчати нашу модель, а потім перевіряти якість роботи моделі на 30% даних. Після того як дані були підготовлені та оброблені для подальшої роботи перейдемо до самої моделі.

Метод для класифікації XGboost має декілька основних параметрів, кожний з яких я опишу нижче:

а) Максимальна глибина дерева

б) Швидкість навчання моделі

в) Кількість прогнозів

г) Кількість раундів для зупинки

Наступною ітерацією був підбір найкращих параметрів для моделі. Для Цього модель тестувалася на різних даних після чого було отримано оптимальні параметри, що дають найкращі результати (рис. 3.4).

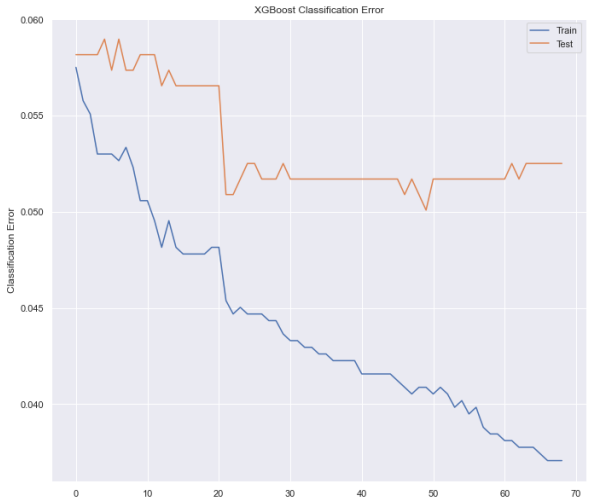

Рисунок 3.4 – Графіки помилки класифікації

Після того як було підібрано оптимальні параметри для запуску моделі модель була запущена та були отримані наступні результати (рис. 3.5 – 3.6).

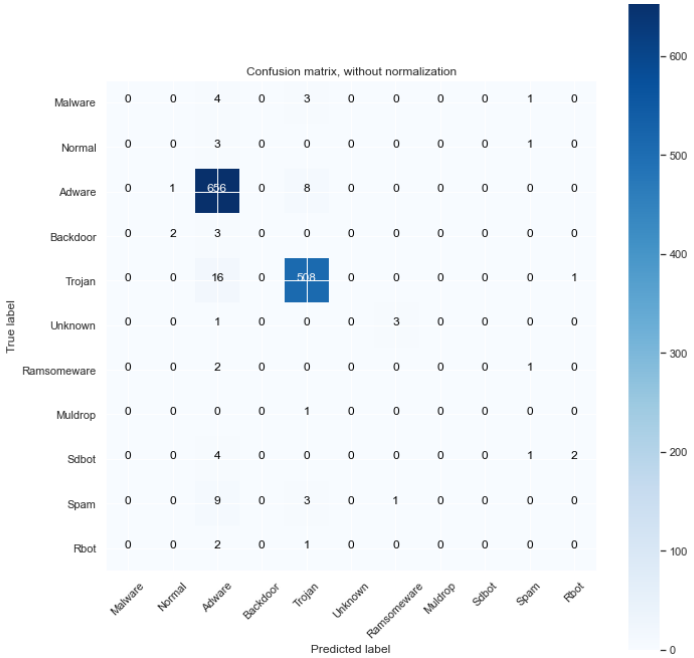

Рисунок 3.5 – Матриця помилок

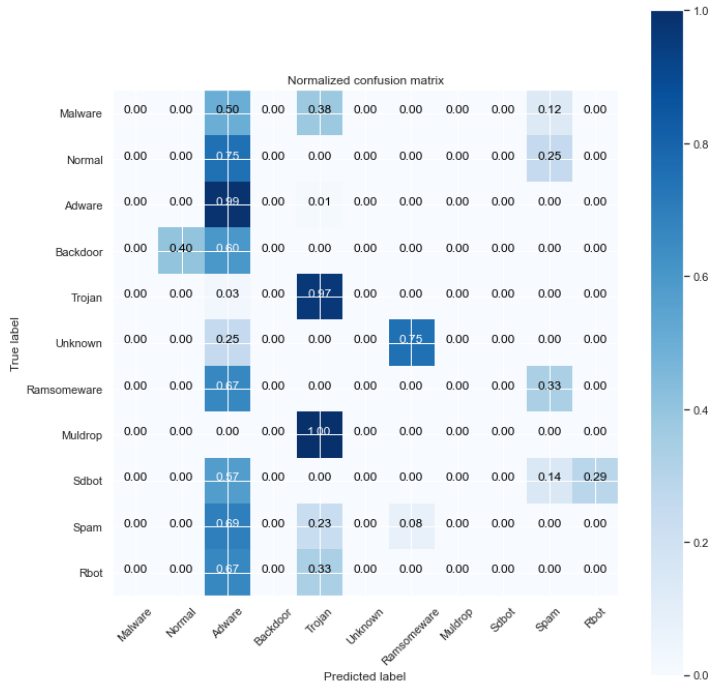

Рисунок 3.6 – Матриця помилок для нормалізованих даних

Як основну метрику оцінки якості роботи моделі будемо використовувати accuracy. Ця метрика відображає як відсоток правильно розпізнаних об’єктів відносно всіх об’єктів. Розроблена модель дала результат

0.9483, а значить можна сказати, що з такою ймовірністю загроза для комп’ютерної безпеки буде класифікована вірно.

3.2 Знаходження аномалій у логу HTTP запитів

У першій частині моєї дипломної роботи було розглянуто задачу передбачення загроз комп’ютерної безпеки на основі розмічених даних за допомогою методу класифікації. Однак у реальному житті далеко не завжди є підготовлені дані для аналізу. Іноді передбачати загрози потрібно на основі логу. У цій частині було зроблено пошук аномалій у логу за допомогою одного з методів машинного навчання (рис. 3.7). Знайшовши аномалії можна знаходити ризики комп’ютерної безпеки для дуже великих масивів даних не перебираючи їх руками.

0	July 8th 2019, 14:43:03.000	XswJ0msBoTGddM7vxMDB	10.1.1.285
1	July 8th 2019, 14:43:01.000	dKQJ0msB7mP0GwVzvJjz	10.1.2.389
2	July 8th 2019, 14:42:59.000	CcwJ0msBoTGddM7vtb8y	10.1.1.415
3	July 8th 2019, 14:42:57.000	bKQJ0msB7mP0GwVzrZdT	10.1.1.79
4	July 8th 2019, 14:42:55.000	L6QJ0msB7mP0GwVzpZeI	10.1.1.60
5	July 8th 2019, 14:42:53.000	O8wJ0msBoTGddM7vnb2w	10.1.2.66
6	July 8th 2019, 14:42:51.000	Z8wJ0msBoTGddM7vlbze	10.1.2.25
7	July 8th 2019, 14:42:49.000	J6QJ0msB7mP0GwVzjpUY	10.1.2.247
8	July 8th 2019, 14:42:46.000	#NAME?	10.1.1.199
9	July 8th 2019, 14:42:46.000	#NAME?	10.1.1.387
10	July 8th 2019, 14:42:30.000	36QJ0msB7mP0GwVzQ4_c	10.1.2.128
11	July 8th 2019, 14:12:45.000	W8ru0WsBoTGddM7vB6Mx	10.1.1.243
12	July 8th 2019, 14:12:40.000	2qLt0WsB7mP0GwVz84Cr	10.1.2.106
13	July 8th 2019, 14:12:36.000	Ycrt0WsBoTGddM7v5KAI	10.1.2.500
14	July 8th 2019, 14:12:32.000	BKLt0WsB7mP0GwVz1H9p	10.1.2.100
15	July 8th 2019, 14:12:28.000	t8rt0WsBoTGddM7vxJ3J	10.1.2.314
16	July 8th 2019, 14:12:23.000	rcrt0WsBoTGddM7vsZw_	10.1.1.383

Рисунок 3.7 – Приклад даних

На основі дат, часу та днів тижню сформуємо змінні, які потім будемо використовувати як характеристики.

Було сформовано початкові характеристики, що пов’язані з часом та датою запитів (рис. 3.8). Такими характеристиками стали характеристики, що позначають чи є день вихідним, день тижня, час запиту, різниця між запитами в хвилинах та інші.

	@timestamp	_id	ip_address	shift_time	time_diff	date	dow	hour	is_weekend	hour_bucket
721473	2019-06-09 00:06:09	DBuOOWsB7mP0GwVzhZ9U	10.1.1.1	NaT	NaN	2019-06-09	6	0	1	0
720483	2019-06-09 01:28:39	bB7aOWsB7mP0GwVzDY5G	10.1.1.1	2019-06-09 00:06:09	82.0	2019-06-09	6	1	1	0
719233	2019-06-09 03:12:49	R0w5OmsBoTGddM7vayZT	10.1.1.1	2019-06-09 01:28:39	104.0	2019-06-09	6	3	1	0
719222	2019-06-09 03:13:45	U0w6OmsBoTGddM7vRi8R	10.1.1.1	2019-06-09 03:12:49	0.0	2019-06-09	6	3	1	0
718875	2019-06-09 03:42:39	z01UOmsBoTGddM7vuzyC	10.1.1.1	2019-06-09 03:13:45	28.0	2019-06-09	6	3	1	0
718730	2019-06-09 03:54:45	x01fOmsBoTGddM7vz6d3	10.1.1.1	2019-06-09 03:42:39	12.0	2019-06-09	6	3	1	0
718240	2019-06-09 04:35:35	nk-FOmsBoTGddM7vLRjb	10.1.1.1	2019-06-09 03:54:45	40.0	2019-06-09	6	4	1	1
717685	2019-06-09 05:21:49	N1CvOmsBoTGddM7vhbSz	10.1.1.1	2019-06-09 04:35:35	46.0	2019-06-09	6	5	1	1

Рисунок 3.8 – Приклад нових характеристик

Після цього було обраховано медіанне значення кількості переходів з кожної ip-адреси. Приклад даних приведено на рис. 3.9.

	ip_address	daily_counts
0	10.1.1.1	40.0
1	10.1.1.100	78.0
2	10.1.1.101	40.0
3	10.1.1.106	35.5
4	10.1.1.109	42.5
5	10.1.1.110	41.0
6	10.1.1.114	37.0
7	10.1.1.118	42.0

Рисунок 3.9 – Приклад медіанних значень за день

Після цього було обраховано характеристики, що відповідають середньому та максимальному часу логіну для кожної ip-адреси. Таким чином було сформовано та згруповано кінцеві характеристики за допомогою яких

буде проводитися знаходження аномальних значень. Приклад фінальних даних приведено на рис. 3.10.

	ip_address	total_count	daily_counts	is_weekend_ratio	td_mean	td_max
0	10.1.1.1	1446	40.0	2.070064	28.999308	362.0
1	10.1.1.100	2860	78.0	2.177778	14.427072	185.0
2	10.1.1.101	1465	40.0	2.191721	28.520492	211.0
3	10.1.1.106	1408	35.5	2.229358	29.771144	319.0
4	10.1.1.109	1459	42.5	2.206593	28.711934	278.0
5	10.1.1.110	1482	41.0	2.242888	28.249831	240.0
6	10.1.1.114	1407	37.0	2.140625	29.827169	300.0
7	10.1.1.118	1446	42.0	2.308924	28.976471	267.0

Рисунок 3.10 – Фінальний вигляд даних для пошуку аномалій

Після того як було отримано всі характеристики настав час переходити до самого методу кластеризації. У цій роботі використовувався метод кластеризації K-means. Цей метод потребує як вхідний параметр кількість кластерів. Зазвичай кількість кластерів вказують в залежності від конкретної задачі, але нажаль ця задача не є таким випадком. Коли кількість кластерів невідома, її підбирають використовуючи цикл і перебирання декількох різних варіантів. Після того як перебирається достатня кількість варіантів для кількості кластерів оптимальна кількість визначається за допомогою евристичного методу ліктю. У нашому випадку оптимальною кількістю кластерів є 5 кластерів. На рис. 3.11 приведено графік виходячи з якого було прийняте таке рішення.

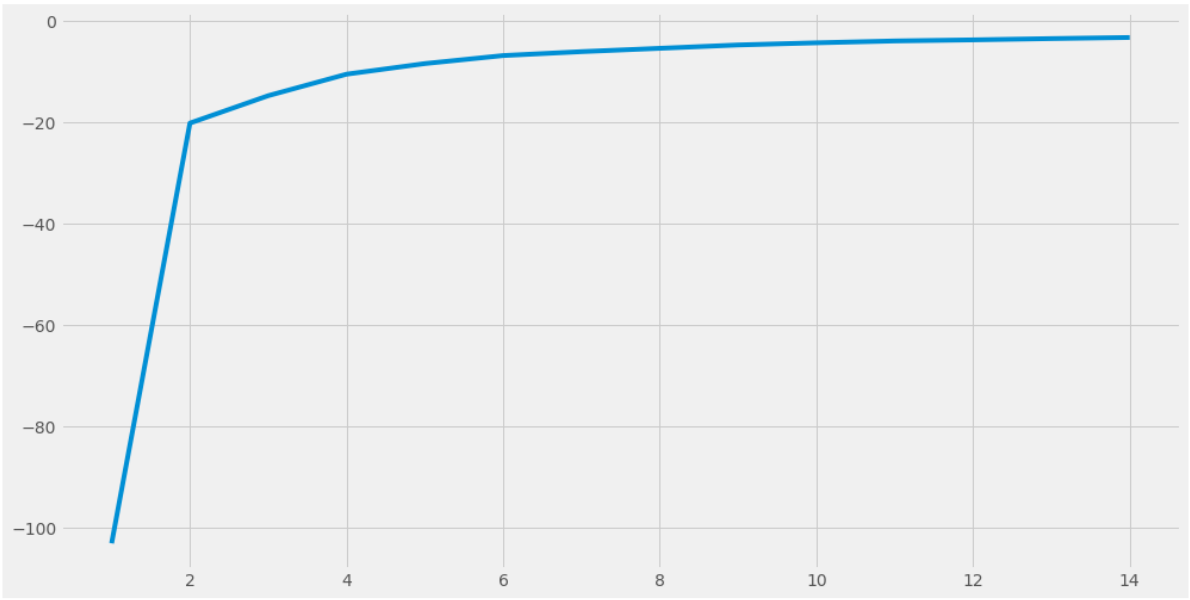

Рисунок 3.11 – Графік для вибору кількості кластерів

Після того як ми отримали кількість кластерів застосуємо алгоритм кластеризації до наших даних. Після цього дані було розбито на 5 кластерів та зображено це на рис. 3.12.

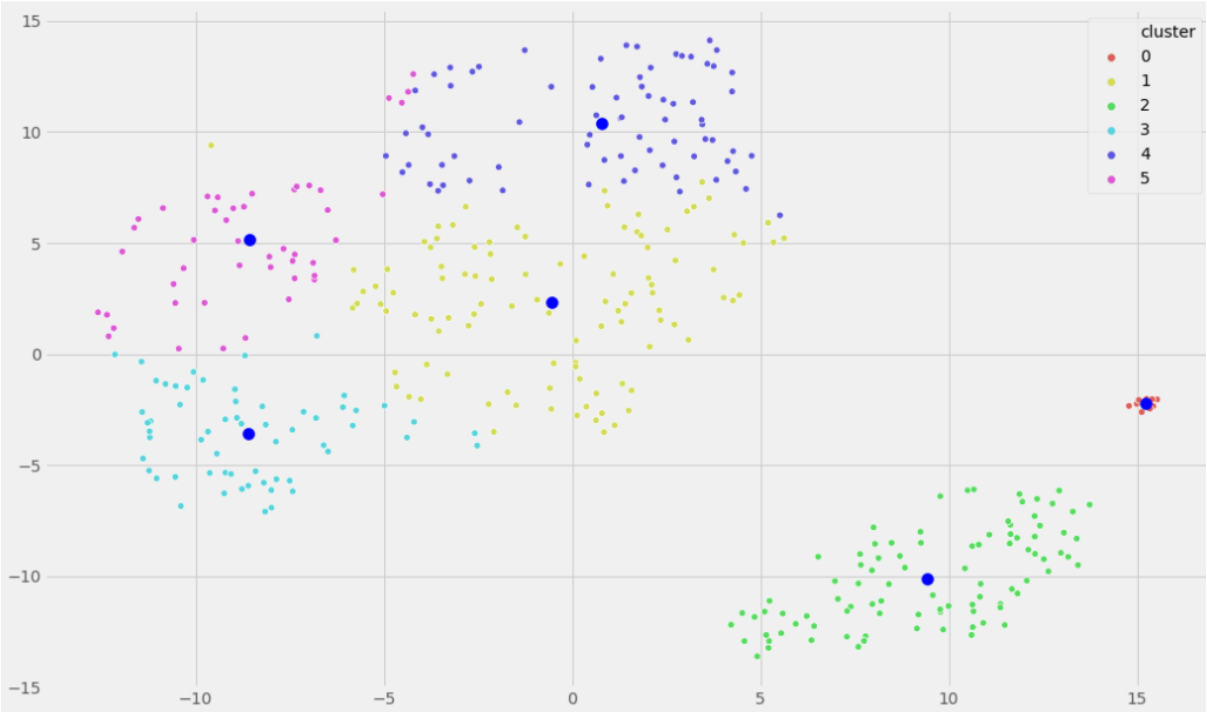

Рисунок 3.12 – Візуалізація розбиття на кластери

На наступному кроці було побудовано гістограму, яка позначає відстані (рис. 3.13).

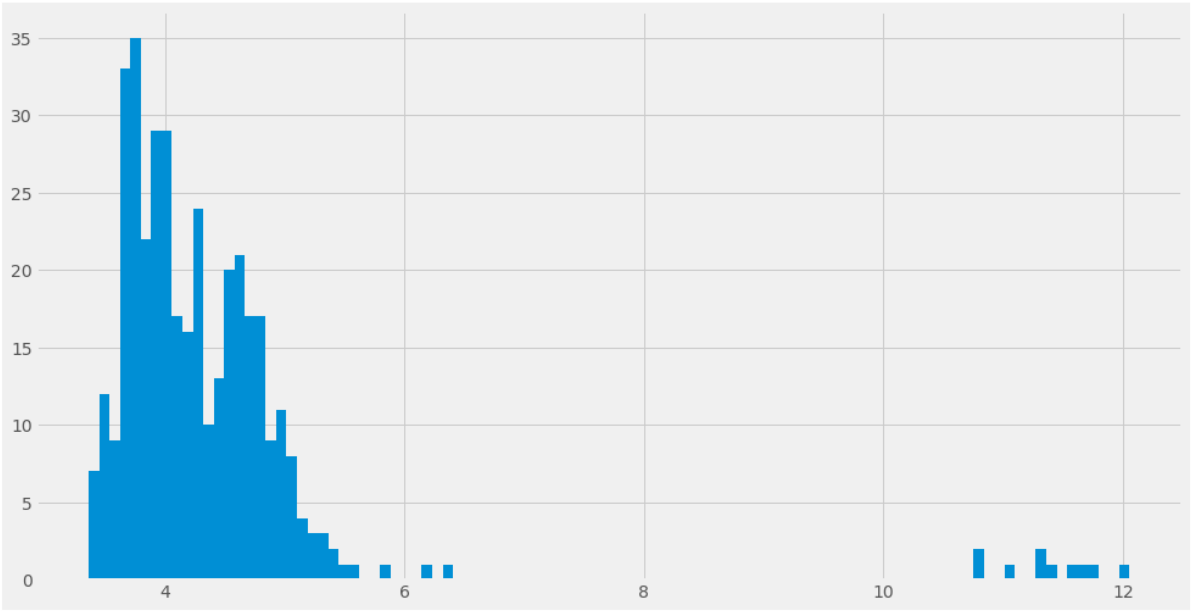

Рисунок 3.13 – Гістограма отриманих даних

Було прийнято рішення, що об’єкти, що мають відстань більше 6 вважаються аномаліями. Зобразимо ці об’єкти візуально та побачимо де вони знаходяться (рис. 3.14).

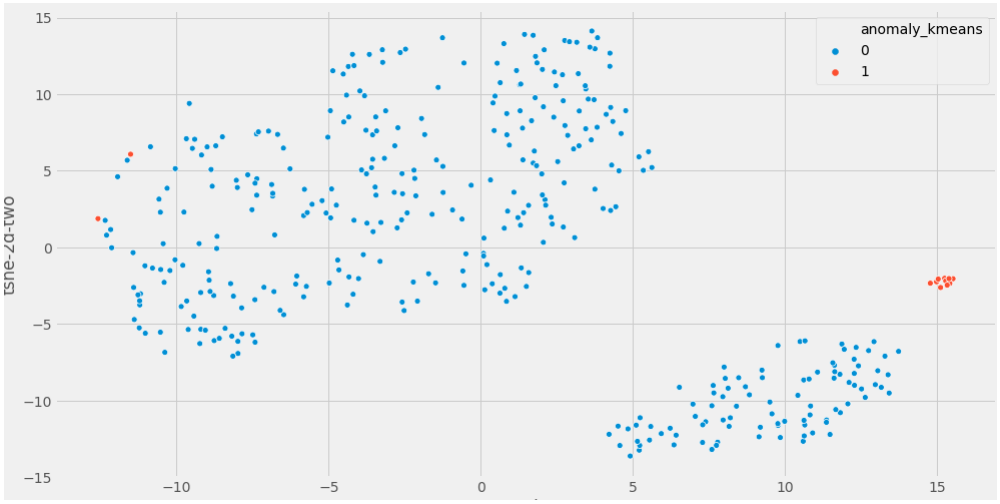

Рисунок 3.14 – Об’єкти, що є аномаліями

Після того як було сформовано правило для аномалій та знайдено такі аномалії на рисунку знайдемо їх більш аналітично серед наших даних. Далі покажемо аномальні ip-адреси серед логу запитів (рис. 3.15).

ip_address	total_count	daily_counts	is_weekend_ratio	td_mean	td_max	cluster
10.1.1.199	1365	40.5	2.123570	30.801320	455.0	5
10.1.1.249	4301	116.5	2.236268	9.459535	101.0	0
10.1.1.386	4300	118.5	2.127273	9.453361	104.0	0
10.1.1.486	4317	117.0	2.315668	9.417285	108.0	0
10.1.1.63	4339	112.0	2.148766	9.368142	101.0	0
10.1.1.86	4293	113.0	2.203731	9.456897	110.0	0
10.1.2.249	4353	112.0	2.250934	9.332721	102.0	0
10.1.2.386	4326	108.0	2.250188	9.392370	110.0	0
10.1.2.432	1437	40.5	2.413302	29.176880	466.0	5
10.1.2.486	4251	114.0	2.056075	9.571059	99.0	0
10.1.2.63	4372	121.0	2.184268	9.268588	118.0	0
10.1.2.86	4307	111.0	2.209389	9.441013	122.0	0

Рисунок 3.15 – Аномальні ip-адреси

Таким чином серед дуже великої кількості даних логу запитів за допомогою алгоритму кластеризації було знайдено аномальні ip-адреси, що дозволяє адміністратору серверу детальніше вивчити їх та запобігти ризику комп’ютерної атаки.

3.3 Висновки до розділу 3

У даній частині диплому було на практиці використано алгоритми машинного навчання для того, щоб за допомогою алгоритму класифікації передбачити загрози комп’ютерної безпеки за допомогою інформації про файл та класифікувати їх до певного типу. Для цього було використано алгоритм, що базується на деревах рішень. У другій частині було проаналізовано лог

файл та в ньому визначено аномальні значення, які можна вважати комп’ютерною загрозою. Це було зроблено за допомогою використання методу кластеризації на основі відстаней між кластерами на які було поділено лог файл.

РОЗДІЛ 4 СТАРТАП ПРОЕКТ «РИЗИКХЕЛПЕР»

Суть магістерської дисертації у тому, щоб створити модель яка б допомагала знаходити серед великої кількості даних аномальні події і таким чином допомагати аналізувати ризик захищеності комп'ютерних систем. Цю модель можна використовувати, щоб допомагати спеціалістам інформаційної безпеки автоматизовано відслідковувати ризики.

4.1. Опис ідеї проекту

У табл. 4.1 надано зміст ідеї, можливі напрямки застосування та основні вигоди, що може отримати користувач товару.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програмного продукту «РИЗИКХЕЛПЕР»	1. Допомога моніторингу нестандартних випадків	Швидкість та економія часу
	2. Зниження ризику захищеності	Мінімізація ймовірності шкоди системі
	3. Автоматизація процесів	Менші трати людської сили

Виділимо такі техніко-економічні характеристики ідеї:

1) велика точність знаходження підозрілих даних;
2) зрозумілість та простота для користувача;

3) широка галузь для застосування та універсальність.

Для порівняння цього продукту з іншими представниками на ринку, у якості конкурентів виберемо такі три програмні продукти:

1) Гриф;
2) РизикВОТЧ;
3) Кобра.

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

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Ризикхелпер	Гриф	Ризик ВОТЧ	Кобра			
1.	автоматизація	+	-	-	+			+
2.	Легкість застосування	+	-	+	+		+	
3.	аналіз	-	+	+	+	+		

4.2. Технологічний аудит ідеї проекту

Визначення технологічної здійсненності ідеї проекту передбачає аналіз складових, наведених у табл. 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Більш ретельний аналіз	Створення анкетування користувачів	Наявні	+
2	Більше методів аналізу	Залучення датасаєнтистів	Наявні	+
3	Створення інтерфейсу	Залучення розробників	Наявні	+
Обрана технологія реалізації ідеї проекту: залучення інвесторів для вкладання коштів для залучення більшої кількості кваліфікованих спеціалістів				

4.3. Аналіз ринкових можливостей запуску стартап-проекту

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн	2 млн

Продовження таблиці 4.4

3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Прозорість аналізу
5	Специфічні вимоги до стандартизації та сертифікації	-
6	Середня норма рентабельності в галузі (або по ринку), %	40

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 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	Автоматизація процесів аналізу ризиків	Компанії будь-якого профілю	-	- ефективність; - легкість; - швидкість.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6 і 4.7). Фактори в таблиці подані в порядку зменшення значущості.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Мала кількість даних	Чим менше даних для цієї галузі тим гірше буде фінальний прогноз	Збереження як можна більшої кількості даних
2	Людський фактор	Невірне використання продукту адміністратором	Повторний інструктаж користувача

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Покращення моделей	Впровадження нових методів	Покращення рівня праці
2	Нові методи та фактори прийняття рішень	Більш комплексний аналіз	Покращення умов праці співробітника

Далі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (табл. 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. Вказати тип конкуренції - монополія / олігополія / монополістична / чиста	Чиста	Хороші перспективи розвитку
2. За рівнем конкурентної боротьби - локальний / національний / …	Національний	Ведучи конкуренцію на національному рівні, компанії необхідно прикласти дуже значні зусилля, щоб охопити весь національний ринок.
3. За галузевою ознакою - міжгалузева / внутрішньогалузева	Внутрішньогалузева	Зосередження зусиль на пошук конкурентних переваг, що дозволять компанії займати стійкі конкурентні позиції в даній галузі.
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Товарно-видова	Зосередження на перевагах цього програмного продукту серед існуючих.
5. За характером конкурентних переваг - цінова / нецінова	Нецінова	Зосередити зусилля на точності прийняття рішення та ефективності продукту

Продовження таблиці 4.8

6. За інтенсивністю - марочна / не марочна	Марочна	Дослідити якість послуг та цінову категорію конкурентів

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (табл. 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки	На ринку відносно стабільна кількість конкурентів.	Прозорість розробки продукту.	Відсутня залежність від постачальників.	Клієнти у значній мірі мають вплив на загальний попит	Субститутів немає.

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію (табл. 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Чіткі алгоритми	Аналіз на основі математичних алгоритмів
2	Зрозумілість у використанні	Незначний поріг входження для використання
3	Велика точність	Доволі точні висновки

За визначеними факторами конкурентоспроможності (табл. 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін «ГРИФ»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «Рискхелпер»						
			–3	–2	–1	0	+1	+2	+3
1	Чіткі алгоритми	18		+					
2	Зрозумілість у використанні	19			+				
3	Велика точність	20				+			

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12).

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Сильні сторони: Висока ефективність, зручність у використанні	Слабкі сторони: відсутність значної кількості даних, відсутність значної кількості кваліфікованих співробітникі
Можливості: Можливість покращення працівниками компанії	Загрози: Невірна інтерпретація та помилки через недостатню кваліфікацію

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 9, аналіз потенційних конкурентів).

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Залучення інвестицій	Пошук інвесторів, щоб залучити кошти задля співпраці з програмістами та датасаєнтистами	4 місяці
2	Розвиток ринку	Вихід продукту на європейський ринок	18 місяців

4.4. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Важкість входу у сегмент
1	ІТ-компанії	середня	високий	висока	висока
2	Інші клієнти	висока	середній	висока	середня
Які цільові групи обрано: інших клієнтів.					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Спеціалізація	Зосередження на певному сегменті ринку	Висока ефективність, зручність у використанні	Стратегія спеціалізації

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» 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
1	Ні	Забиратиме існуючих у конкурентів	Напрямок у сторону виділення фіч на фоні конкурентів	Стратегія заняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до продукту (див. табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (табл. 4.16) розробляється стратегія

позиціонування (табл. 4.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати проект.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	ефективність, зручність, ситуаційність.	Стратегія спеціалізації	гарантії ефективної роботи, зручність у використанні	ефективність, зручність, ситуаційність

4.5. Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

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	Автоматизація роботи	Значна економія ресурсів	Більшість конкурентів максимізують цей параметр
2	Легкість використання	Гнучкість та зрозумілість продукту	Використання систем з конфіденційною логікою

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>I. Товар за задумом</td><td colspan="3">Програмний продукт для автоматизації роботи з можливістю аналізувати.</td></tr>
<tr><td rowspan="3">II. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Функціональність
2. Зручність
3. Зовнішній вигляд</td><td>М
М
Нм</td><td>Тх
Е
Е/Ор</td></tr>
<tr><td colspan="3">Якість: продукт має відповідати міжнародним стандартам</td></tr>
<tr><td>III. Товар із підкріпленням</td><td colspan="3">Використовуються тимчасові знижки</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: інтелектуальна власність.</td></tr>
</table>

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Немає	Вищий	Високий	від 15% до 35% від зборів

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21).

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Дані (логи) з комп’ютерної системи	Звіт та аналіз виконаної роботи	Без посередників	Напряму з клієнтом

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Конфіденційність інформації	Електронна пошта	Ефективність та легкість	Привернути увагу цільових клієнтів.	Знижки на перші пів року користування продуктом.

4.6. Висновки до розділу 4

У даному розділі було розглянуто спроможності розробленого програмного продукту аналізу ризиків захищеності комп’ютерних систем. Було проаналізовано ринок та конкурентів, описані основні стратегії та можливості. Даний продукт доцільно розвивати та реалізовувати в сучасних умовах.

ВИСНОВКИ

У даній роботі було запропоновано метод, що дозволяє з досить високою точністю прогнозувати різні типи ризиків захищеності комп'ютерних систем, а також виявляти аномалії у великих обсягах даних.

Було проведено аналіз логу HTTP запитів та аналіз даних, що відображають захищеність комп'ютерних систем. На цих даних було створено дві моделі, одна з яких з певною точністю передбачає ризик загрози комп'ютерній системі, а друга на великому обсягу даних знаходить аномалії, які є показниками ризику безпеки комп'ютерної системи.

Для кращого результату прогнозування, дані було нормалізовано, а також перевірено на мультиколінеарність. Після цього було підібрано оптимальні параметри моделі за допомогою тестування використовуючи графік помилки методу.

Після цього був пророоблений алгоритм класифікації та було отримано передбачення щодо загроз ризику захищеності комп'ютерної системи. Результати були протестовані метрики точності роботи моделі.

Другою частиною роботи було знаходження аномалій у логу HTTP запитів для виявлення ризиків інформаційної безпеки. Спочатку на основі логу були створені часові параметри, які залежали від багатьох показників. Потім на основі цих параметрів було пророоблено кластеризацію та таки чиною виявлено аномальні результати, які можуть бути загрозою для безпеки комп'ютеру.

ПЕРЕЛІК ПОСИЛАНЬ

1. Конеев И., Беляев А. Информационная безопасность предприятия. Санкт-Петербург: БХВ-Петербург, 2003. 256 с.
2. Бармен. С. Разработка правил информационной безопасности. Москва: Издательский дом "Вильямс", 2002. 208 с.
3. Бартон Т., Шенкир У. Риск-менеджмент. Практика ведущих компаний. Москва: Издательский дом "Вильямс", 2002. 208 с.
4. Астахов А. Искусство управления информационными рисками. Москва: МК Пресс, 2009. 312 с.
5. Галатенко В.А. Основы информационной безопасности. Москва: Издательский дом "Вильямс", 2004. 57 с.
6. Ярочкин В. И. Информационная безопасность. Москва: Академический проспект, 2004. 89 с.
7. Баранова Е.К., Бабаш А.В. Информационная безопасность и защита информации. Москва: ИНФРА-М_РИОР, 2014. 38 с.
8. Баранова Е.К. Методики и программное обеспечение для оценки рисков в сфере информационной безопасности. Управление риском. Москва: ИНФРА-М_РИОР, 2009. 108 с.
9. Петренко С.А., Симонов С.В. Управление информационными рисками. Экономически оправданная безопасность. Москва: Компания АйТи; ДМК Пресс, 2004. 47 с.
10. Международный стандарт ISO/IEC 27005:2008. Информационная технология. Методы защиты. Менеджмент рисков информационной безопасности. URL: https://www.praxiom.com/iso-17799-4.htm 88 (дата звернення: 10.10.2020)
11. Левченко В.Н. Этапы анализа рисков. URL: http://www.cfin.ru/finanalysis/risk/stages.shtml (дата звернення: 13.10.2020)

12. Методики и программные продукты для оценки рисков. URL: https://www.intuit.ru/studies/courses/531/387/lecture/8996?page=4 (дата звернення: 15.10.2020)
13. Юрьев В.Н., Эрман С.А. ТЕОРЕТИКО-ВЕРОЯТНОСТНАЯ МОДЕЛЬ ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРЕДПРИЯТИЯ. URL: https://cyberleninka.ru/article/v/teoretikoveroyatnostnaya-model-otsenki-riskov-informatsionnoy-bezopasnostipredpriyatiya (дата звернення: 20.10.2020)
14. Мазов Н.А., Федотов А.М. КЛАССИФИКАЦИЯ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ. URL: https://cyberleninka.ru/article/v/klassifikatsiya-riskovinformatsionnoy-bezopasnosti (дата звернення: 21.10.2020)
15. Tianqi Chen XGBOOST DOCUMENTATION. URL: https://xgboost.readthedocs.io/en/latest/ (Last accessed: 22.10.2020)
16. Jason Brownlee GRADIENT BOOSTED TREES WITH XGBOOST AND SCIKIT-LEARN. URL: https://s3.amazonaws.com/MLMastery/xgboost_with_python_sample.pdf (Last accessed: 15.11.2020)
17. By Joaquín Pérez-Ortega, Nelva Nely Almanza-Ortega, Andrea Vega-Villalobos, Rodolfo Pazos-Rangel, Crispín Zavala-Díaz and Alicia Martínez-Rebollar THE K-MEANS ALGORITHM EVOLUTION. URL: https://www.intechopen.com/books/introduction-to-data-science-and-machine-learning/the-em-k-em-means-algorithm-evolution (Last accessed: 05.11.2020)
18. Chris Piech K MEANS. URL: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html (Last accessed: 10.11.2020)

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
import pandas as pd
import numpy as np
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import svm
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier,ExtraTreesClassifier
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix
#from pandas_ml import ConfusionMatrix
from sklearn.metrics import precision_recall_fscore_support
import itertools
df=pd.read_excel("C:/Users/admin/Desktop/Диплом_Магистратура/data.xlsx")
df.head()
df['Class'].unique()
df.describe()
df1=df.iloc[:,1:8]
col = df1.columns
cm = np.corrcoef(df1.values.T)
sns.set(font_scale=1.5)
sns.set(rc={'figure.figsize':(10,10)})
hm = sns.heatmap(cm,cbar=True,annot=True,square=True,fmt='.2f',annot_kws={'size':
10},yticklabels=col,xticklabels=col)
from sklearn.model_selection import train_test_split
```

```
y = df['Class']
X = df.iloc[:,1:8]
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3)
y_train=np.array(y_train).reshape(-1,1)
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        #print("Normalized confusion matrix")
    else:
        #print('Confusion matrix, without normalization')
        pass

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
```

```
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
from sklearn.metrics import accuracy_score
model = xgb.XGBClassifier(learning_rate=0.1, max_depth=5, n_estimators=300)
eval_set = [(X_train, y_train), (X_test, y_test)]
model.fit(X_train, y_train, eval_metric=["merror", "mlogloss"], eval_set=eval_set, early_stopping_rounds=10)
score = model.score(X_test, y_test)
y_pred=model.predict(X_test)
model.evals_result()
accuracy = accuracy_score(y_test, y_pred)
accuracy
# retrieve performance metrics
from matplotlib import pyplot
results = model.evals_result()
epochs = len(results['validation_0']['merror'])
x_axis = range(0, epochs)
results['validation_0']
#plot log loss
fig, ax = pyplot.subplots()
ax.plot(x_axis, results['validation_0']['mlogloss'], label='Train')
ax.plot(x_axis, results['validation_1']['mlogloss'], label='Test')
ax.legend()
pyplot.ylabel('Log Loss')
pyplot.title('XGBoost Log Loss')
pyplot.show()
# plot classification error
fig, ax = pyplot.subplots()
ax.plot(x_axis, results['validation_0']['merror'], label='Train')
ax.plot(x_axis, results['validation_1']['merror'], label='Test')
ax.legend()
pyplot.ylabel('Classification Error')
pyplot.title('XGBoost Classification Error')
```

```
pyplot.show()
plt.style.use('ggplot')
import matplotlib.style as style
style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = (15, 8)
import seaborn as sns
from sklearn.manifold import TSNE
from mpl_toolkits.mplot3d import Axes3D
from sklearn import preprocessing
from sklearn.cluster import KMeans
data = pd.read_csv('C:/Users/admin/Desktop/Диплом_Магистратура/data1.csv', delimiter=';')
data['@timestamp'] = pd.to_datetime(data['@timestamp'])
data.sort_values(['ip_address', '@timestamp'], inplace=True)
data['shift_time'] = data.groupby(['ip_address'])['@timestamp'].shift(1)
data['time_diff'] = (data['@timestamp'] - data['shift_time']).dt.seconds//60
data['date'] = data['@timestamp'].dt.date
data['dow'] = data['@timestamp'].dt.weekday
data['hour'] = data['@timestamp'].dt.hour
data['is_weekend'] = ((data['dow']==5)|(data['dow']==6)).astype(int)
data['hour_bucket'] = data['hour']//4
ip_col = 'ip_address'
ip_counts = data.groupby(ip_col)['@timestamp'].count().reset_index()
ip_counts = ip_counts.rename(columns={'@timestamp':'total_count'})
daily_counts = data.groupby([ip_col, 'date'])['@timestamp'].count().reset_index()
daily_counts = daily_counts.rename(columns={'@timestamp':'daily_counts'})
daily_counts_agg = daily_counts.groupby(ip_col).daily_counts.median().reset_index()
weekend_counts = data.groupby([ip_col, 'is_weekend'])['@timestamp'].count().reset_index()
weekend_counts = weekend_counts.rename(columns={'@timestamp':'weekend_counts'})
weekend_counts_agg = weekend_counts.pivot_table(index=ip_col,
columns='is_weekend').reset_index([0])
weekend_counts_agg.columns = weekend_counts_agg.columns.droplevel()
weekend_counts_agg.columns = [ip_col, 'week_day', 'weekend']
weekend_counts_agg['is_weekend_ratio'] = weekend_counts_agg['week_day']/
weekend_counts_agg['weekend']
```

```
lean_weekend_counts_agg = weekend_counts_agg[[ip_col, 'is_weekend_ratio']]

avg_timedelta_data = data.groupby(ip_col).agg({'time_diff':['mean','max']}).reset_index()

avg_timedelta_data.columns = avg_timedelta_data.columns.droplevel()

avg_timedelta_data.columns = [ip_col, 'td_mean', 'td_max']

merge_1 = ip_counts.merge(daily_counts_agg, on=ip_col, how='left')

merge_2 = merge_1.merge(lean_weekend_counts_agg, on=ip_col, how='left')

final_data = merge_2.merge(avg_timedelta_data, on=ip_col, how='left')

ip_map = final_data[ip_col].to_dict()

RANDOM_STATE = 123

feature_cols = ['total_count', 'daily_counts', 'is_weekend_ratio', 'td_mean', 'td_max']

data_new = final_data[feature_cols]

min_max_scaler = preprocessing.MinMaxScaler()

np_scaled = min_max_scaler.fit_transform(data_new)

data_new = pd.DataFrame(np_scaled, columns=feature_cols)

sns.pairplot(final_data[feature_cols])

n_cluster = range(1, 15)

kmeans = [KMeans(n_clusters=i, random_state=RANDOM_STATE).fit(data_new) for i in
n_cluster]

scores = [kmeans[i].score(data_new) for i in range(len(kmeans))]

fig, ax = plt.subplots()

ax.plot(n_cluster, scores)

plt.show()

cluster_model = kmeans[5]

final_data['cluster'] = cluster_model.predict(data_new)

final_data['cluster'].value_counts()

tsne = TSNE(n_components=2, verbose=1, perplexity=40, n_iter=300,
random_state=RANDOM_STATE)

tsne_results = tsne.fit_transform(data_new)

final_data['tsne-2d-one'] = tsne_results[:,0]

final_data['tsne-2d-two'] = tsne_results[:,1]

tsne_cluster = final_data.groupby('cluster').agg({'tsne-2d-one':'mean', 'tsne-2d-
two':'mean'}).reset_index()

plt.figure(figsize=(16,10))
```

```
sns.scatterplot(
    x="tsne-2d-one", y="tsne-2d-two",
    hue="cluster",
    palette=sns.color_palette("hls", 6),
    data=final_data,
    legend="full",
    alpha=1
)
plt.scatter(x="tsne-2d-one", y="tsne-2d-two", data=tsne_cluster, s=100, c='b')
plt.show()
centers = cluster_model.cluster_centers_
points = np.asarray(data_new)
total_distance = pd.Series()
def get_sum_square_distance(data, cluster_model):
    centers = cluster_model.cluster_centers_
    points = np.asarray(data[feature_cols])
    total_distance = pd.Series()
    for i in range(len(points)):
        distance = 0
        for j in range(len(centers)):
            a = np.linalg.norm(points[i] - centers[j])
            distance += a**2
        total_distance.set_value(i, distance)

    return total_distance
final_data['ssd'] = get_sum_square_distance(data_new, cluster_model)
plt.hist(final_data['ssd'], bins=100)
cutoff = 6
final_data['anomaly_kmeans'] = (final_data['ssd'] >= cutoff).astype(int)
sns.scatterplot(
    x="tsne-2d-one", y="tsne-2d-two",
    hue="anomaly_kmeans",
#     palette=sns.color_palette("hls", 10),
```

```
    data=final_data,
    legend="full",
    alpha=1
)
final_data.loc[final_data['anomaly_kmeans']==1]
```