МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова
праця на правах рукопису

МІЩЕНКО МАКСИМ ВАЛЕРІЙОВИЧ

УДК [004.056:004.89](043.5)

ДИСЕРТАЦІЯ

ПРОГНОЗУВАННЯ ТА ВИЯВЛЕННЯ ЗАГРОЗ ДЛЯ КОРПОРАТИВНИХ КОМП'ЮТЕРНИХ МЕРЕЖ ЗАСОБАМИ ЕКСПЕРТНИХ СИСТЕМ

122 – Комп'ютерні науки

12 – Інформаційні технології

галузь знань

Подається на здобуття наукового ступеня доктора філософії.

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Міщенко Максим Валерійович

підпис

Науковий керівник Дорош Марія Сергіївна
доктор технічних наук, професор

Чернігів – 2025

АНОТАЦІЯ

Міщенко Максим. Прогнозування та виявлення загроз для корпоративних комп’ютерних мереж засобами експертних систем. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 – «Комп’ютерні науки» (12 – «Інформаційні технології»). – Національний Університет «Чернігівська політехніка», МОН України, Чернігів, 2025.

В роботі вирішено актуальне наукове завдання з розробки моделей та методів виявлення та прогнозування загроз для корпоративних комп’ютерних мереж засобами експертних систем для підвищення точності виявлень та оперативності прийняття рішень щодо реагування на наявні та можливі загрози.

Об’єктом дослідження є інформаційні процеси захисту корпоративних комп’ютерних мереж від кіберзагроз.

Предметом дослідження є методи, моделі та елементи інформаційної технології виявлення та прогнозування кіберзагроз для корпоративних комп’ютерних мереж засобами експертних систем.

Метою дисертаційного дослідження є підвищення оперативності виявлення загроз та забезпечення підтримки прийняття рішень щодо реагування на наявні та можливі вразливості корпоративних комп’ютерних мереж. Завдання дослідження полягає у створенні методів виявлення та прогнозування загроз для корпоративних комп'ютерних мереж, а також створення моделі інформаційної технології, що враховує комплексне використання запропонованих методів та засобів експертних систем для підтримки прийняття рішень.

В основу методології дослідження покладено методи моделювання функціональних діаграм IDEF0 для проектування інформаційної системи виявлення та прогнозування загроз для корпоративних комп’ютерних мереж, діаграми UML для моделювання варіантів використання системи експертами з

кібербезпеки та інженерами з кіберзахисту. Модель «сутність-зв’язок» використана при моделюванні загрози для корпоративних комп’ютерних мереж та при проектуванні бази даних для інформаційної системи виявлення та прогнозування загроз. Для оцінки значення кількісної метрики CVSS для загроз корпоративним комп'ютерним мережам, використано метод експертних оцінок. Також застосовано методи статистичного аналізу для виявлення аномалій мережевого трафіку, методи машинного навчання та методи NLP для ідентифікації шкідливих файлів.

У вступі обґрунтовано актуальність теми дослідження, сформульовані мета, задачі та методи дослідження, наведено зв'язок дослідження з науковими програмами кафедри, описано наукову новизну і практичне значення результатів дисертаційної роботи. Визначено, що зловмисники вдаються до модифікації існуючих атак шляхом додавання нової логіки, зміни порядку команд, пошуку нових експлойтів, що надає актуальності виявлення загроз з використанням статистичних та методів машинного навчання. Також прийняття рішень при розподілі ресурсів щодо захисту потребує визначення рівня критичності загрози, що робить актуальним використання засобів експертних систем в інформаційних технологіях управління безпекою комп’ютерних мереж.

У першому розділі проведено аналіз основних типів загроз інформаційній безпеці корпоративних комп’ютерних мереж, шляхів їх потрапляння до корпоративної мережі та механізмів їх дії. Було визначено модель загрози для корпоративної комп’ютерної мережі та змодельовано процес виявлення загроз. Представлено порівняльний аналіз існуючих систем та підходів виявлення загроз для корпоративних мереж.

У другому розділі запропоновано модель комплексної інформаційної технології виявлення загроз з використанням експертних оцінок для подальшого прогнозування ймовірності реалізації визначених векторів вразливостей та формування рекомендацій з протидії загрозам на основі Теорії Ігор. Також

розроблено методи виявлення загроз, результати яких використовуються для наповнення бази знань експертної системи. Серед методів виявлення загроз запропоновано метод визначення та класифікації секції Linux ELF файлу для ідентифікації шкідливого ПЗ та обґрунтовано актуальність виявлення загроз для UNIX-подібних систем. В порівнянні з існуючим дослідженим методом ідентифікації шкідливого ПЗ для UNIX-подібних систем, запропонований метод показав статистично значуще покращення точності та F1-міри. Досліджено застосування моделей NLP до виявлення загроз, в результаті чого запропоновано метод ідентифікації шкідливих Windows PE файлі з використанням моделі word2vec. Для виявлення мережевих аномалій запропоновано метод, що здатний виявляти DDoS атаки з урахуванням часового контексту рядів спостережень мережевих параметрів. У якості методу прогнозування загроз запропоновано використання мереж Баєса для формування ймовірності настання загрози на основі аналізу мережевого трафіку. Також було проаналізовано систему кількісної оцінки загроз CVSS, яка стала основою для побудови рушія висновків інформаційної технології підтримки прийняття рішень щодо захисту корпоративних комп’ютерних мереж.

У третьому розділі представлено загальну функціональну модель інформаційної системи виявлення та прогнозування кіберзагроз для корпоративних комп’ютерних мереж, що визначає основні вхідні та вихідні параметри, обмеження та механізми виконання з трьома рівнями деталізації. Проведено моделювання інформаційної системи з використанням UML діаграм, що дозволило виділити основні варіанти використання інформаційної технології.

Четвертий розділ містить розроблені модулі інформаційної системи виявлення та прогнозування загроз для корпоративних комп’ютерних мереж. Наведені результати експериментів, які підтверджують ефективність запропонованих методів та моделей.

Основні результати дослідження та наукова новизна роботи полягають в розробці та удосконаленні методів та моделей виявлення та прогнозування загроз для корпоративних комп'ютерних мереж з використанням статистичних моделей, методів машинного навчання та засобів експертних систем. Визначено основні сутності, що в сукупності являють собою модель кіберзагрози та аналізуються в процесі її виявлення, що дозволило сформувати набір методів для забезпечення захисту корпоративних комп'ютерних мереж. Представлені методи підвищують ефективність процесу виявлення загроз за рахунок зменшення часу на їх ідентифікацію та підвищення точності. В роботі запропоновано поєднання засобів експертних систем з запропонованими методами виявлення та прогнозування загроз, що забезпечує процес підтримки прийняття рішень щодо реагування на існуючі та прогнозовані вразливості.

В роботі набула подальшого розвитку модель інформаційної системи виявлення та прогнозування загроз для корпоративних комп'ютерних мереж, яка враховує комплексне використання модулей ідентифікації шкідливого ПЗ та рушій висновків, що дозволяють виконати прогнозування ймовірності реалізації визначених векторів вразливостей для підтримки прийняття рішень щодо реагування на загрози.

У якості модулю ідентифікації шкідливого ПЗ інформаційної системи запропоновано метод визначення секції Linux ELF файлу UNIX-подібних операційних систем. Метод містить процеси семантичного аналізу та вибору моделі класифікації шкідливого програмного забезпечення та дає високі показники точності та F1-міри класифікації. Удосконалений метод ідентифікації Windows PE файлів базується на використанні секції «Таблиця імпортів» в поєднанні з техніками word2vec та ансамблю дерев рішень, що забезпечує високу точність та оперативну ідентифікацію шкідливих файлів.

Для наповнення бази даних інформаційної системи запропоновано удосконалений метод виявлення DDoS атак, який містить поєднання моделей

Isolation Forest та EWMA-статистики, що дозволяє враховувати часовий контекст рядів спостережень мережевих параметрів для підтримки прийняття рішення про існування загрози.

Створена функціональна модель інформаційної системи виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж, що містить інформацію про основні вхідні та вихідні параметри, обмеження, ресурси, деталізацію процесів виявлення та прогнозування загроз та може бути основою для проектування систем захисту корпоративних комп'ютерних мереж.

Також визначено практичне значення отриманих результатів, яке полягає в тому, що вони в сукупності утворюють інформаційну технологію виявлення та прогнозування загроз для корпоративних комп'ютерних мереж з використанням засобів експертних систем. Представлено розроблений інтерфейс інформаційної системи для формування звітів про виявлені загрози, які містять рекомендації щодо визначення оптимальних стратегій протидії загрозам на основі Теорії Ігор та експертних оцінок. Запропонована інформаційна технологія містить модулі виявлення, ідентифікації та прогнозування ймовірності реалізації визначених векторів вразливостей, а також модуль введення експертних оцінок CVSS загроз та стратегій протидії. Наведені бізнес-процеси та архітектура можуть бути адаптовані для подальшої розробки та розвитку інформаційних корпоративних комп'ютерних систем виявлення вторгнень. Також запропонована інформаційна технологія може бути використана для спрощення роботи та підвищення продуктивності спеціалістів з кіберзахисту або мережевих адміністраторів корпоративних комп'ютерних мереж.

Ключові слова: інформаційна технологія, інформаційна безпека, експертна система, корпоративна комп'ютерна мережа, система виявлення вторгнень, Теорія Ігор, NLP, машинне навчання.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові публікації, в яких опубліковані основні результати дисертації:

1. А.Г. Гребенник, О.В. Трунова, В.В. Казимир, М.В. Міщенко «Виявлення та прогнозування загроз для корпоративної комп'ютерної мережі.» *Технічні науки та технології, 2020. - № 2 – с.175–184.* (0,5 ум. друк. арк.) (Особистий внесок здобувача: загальна архітектура системи прогнозування та виявлення загроз ККМ, аналіз методу виявлення кіберзагроз з використанням EWMA-статистики) (0,4 ум. друк. арк.)
2. Mishchenko M.V., Dorosh M.S.. "Semantic analysis and classification of malware for UNIX-like operating systems with the use of machine learning methods". Applied Aspects of Information Technology. 2022; Vol. 5, No. 4: 371-386, DOI: https://doi.org/10.15276/aait.05.2022.25. (0,6 ум. друк. арк.) (Особистий внесок здобувача: метод семантичного аналізу та класифікації шкідливого програмного забезпечення для UNIX-подібних систем) (0,5 ум. друк. арк.)
3. Mishchenko, M. V., Dorosh, M. S.. (2024). «An expert system of recommendations for combating cyber threats using CVSS metrics and game theory.» Вісник сучасних інформаційних технологій, 7(3), 284–295. https://doi.org/10.15276/hait.07.2024.20 (0,6 ум. друк. арк.) (Особистий внесок здобувача: експертна система рекомендації протидії загрозам з використанням метрик CVSS та теорії ігор) (0,5 ум. друк. арк.)
4. Міщенко, М. «Функціональна модель системи виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж з використанням експертних оцінок.» Технічні науки та технології – 2024. № 3 (37), - с. 143–152. https://doi.org/10.25140/2411-5363-2024-3(37)-143-152 (0,6 ум. друк. арк.)
5. Mishchenko, M., & Dorosh, M. (2024). Detection of Windows Portable Executable Malware using NLP Techniques and Proxy-server. International

Journal of Computing, 23(4), 663-672. https://doi.org/10.47839/ijc.23.4.3765 (0,6 ум. друк. арк.) (Особистий внесок здобувача: метод ідентифікації шкідливих Windows PE файлів) (0,5 ум. друк. арк.)

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Міщенко М.В., Гребенник А.Г., Трунова О.В.. “Прогнозування рівня загроз з використанням мереж Байєса” XV міжнародна науково-практична конференція математичне та імітаційне моделювання систем МОДС 2020. С. 120-123. Тези доповідей. (0,2 ум. друк. арк.) (Особистий внесок здобувача: дослідження існуючих методів прогнозування загроз з використанням мережі Баєса) (0,1 ум. друк. арк.)
2. Міщенко М.В. «Створення сервісу для виявлення шкідливих elf файлів за допомогою машинного навчання з використанням хмарних технологій aws» Всеукраїнська науково-практична конференція студентів, аспірантів та молодих учених НОВІТНІ ТЕХНОЛОГІЇ У НАУКОВІЙ ДІЯЛЬНОСТІ І НАВЧАЛЬНОМУ ПРОЦЕСІ. – 2023 – с.107-108. Тези доповідей. (0,1 ум. друк. арк.)
3. Міщенко М.В. «Створення експертної системи генерації рекомендацій з протидії кібератакам з використанням теорії ігор.» XIV Міжнародна науково-практична конференція студентів, аспірантів і молодих вчених ЮНІСТЬ НАУКИ – 2024 – с. 1175-1176. Тези доповідей. (0,1 ум. друк. арк.)

ABSTRACT

Maksym Mishchenko. Forecasting and detection of threats to corporate computer networks using expert systems. – Qualification scientific work in the form of a manuscript.

PhD in Engineering Science under Specialty 122 – "Computer Science – National University "Chernihiv Polytechnic", Ministry of Education and Science of Ukraine, Chernihiv, 2025.

The thesis solves a *current scientific task* of developing models and methods for detecting and predicting threats to corporate computer networks using expert systems to increase the accuracy of detection and the efficiency of decision-making regarding response to existing and potential threats.

The object of the research is the information processes for protecting corporate computer networks from cyber threats.

The subject of the research is methods, models, and elements of information technology for detecting and predicting cyber threats for corporate computer networks using expert systems.

The purpose of the research is to increase the efficiency of threat detection and provide decision support for responding to existing and potential vulnerabilities in corporate computer networks. The research objective is to create methods for detecting and predicting threats to corporate computer networks, as well as to create an information technology model that takes into account the comprehensive use of the proposed methods and tools of expert systems to support decision-making.

The research methodology is based on the methods of modeling IDEF0 functional diagrams for designing an information system for detecting and predicting threats to corporate computer networks, UML diagrams for modeling the use cases of the system by cybersecurity experts and cyber defense engineers. The entity-relationship model was used in modeling threats to corporate computer networks and in designing a database for the information system for detecting and predicting threats. The expert

assessment method was used to assess the value of the quantitative CVSS metric for threats to corporate computer networks. Statistical analysis methods were also used to detect network traffic anomalies, machine learning methods, and NLP methods to identify malicious files.

The introduction substantiates the relevance of the research topic, formulates the goal, objectives and methods of the research and provides a connection between the research and the scientific programs of the department, describes the scientific novelty and practical significance of the results of the dissertation work. It is determined that attackers resort to modifying existing attacks by adding new logic, changing the order of commands, searching for new exploits, which makes it relevant to detect threats using adaptive methods: statistical and machine learning methods. Also, making decisions when allocating resources for protection requires determining the level of threat criticality, which makes the use of expert systems in information technologies for managing the security of computer networks relevant.

The first chapter analyzes the main types of threats to information security for corporate computer networks, the ways they enter the corporate network and the mechanisms of their action. The threat model for the corporate computer network was defined and the threat detection process was simulated. A comparative analysis of approaches and existing threat detection systems for corporate networks is presented.

The second chapter proposes a model of comprehensive information technology for threat detection using expert assessments for further prediction of the probability of implementation of certain vulnerability vectors and generation of recommendations for countering threats using Game Theory. Threat detection methods are also developed, the results of which are used to fill the expert system's knowledge base. Among the threat detection methods, a method for determining and classifying the Linux ELF file section for identifying malicious software is proposed and the relevance of threat detection for UNIX-like systems is substantiated. Compared with the existing researched method for identifying malicious software for UNIX-like systems, the

proposed method showed a statistically significant improvement in accuracy and F1-measure. The application of NLP models to threat detection is investigated, as a result of which a method for identifying malicious Windows PE files using the word2vec model is proposed. To detect network anomalies, a method is proposed that is capable of detecting DDoS attacks taking into account the time context of series of observations of network parameters. As a threat prediction method, the use of Bayesian networks is proposed to form the probability of a threat based on network traffic analysis. The CVSS threat quantification system was also analyzed, which became the basis for building an information technology inference engine to support decision-making in protecting corporate computer networks.

The third chapter presents a general functional model of an information system for detecting and predicting cyber threats for corporate computer networks, which defines the main input and output parameters, constraints, and execution mechanisms with three levels of detail. Modeling of the information system was carried out using UML diagrams, which allowed us to identify the main options for using information technology.

The fourth chapter contains the developed modules of the information system for detecting and predicting threats to corporate computer networks. The results of experiments are presented, which confirm the effectiveness of the proposed methods and models.

The main results of the research and the scientific novelty of the work consist in improving the methods and models of detecting and predicting threats to corporate computer networks using statistical models, machine learning methods, and expert systems. The main entities that collectively constitute a cyberthreat model and are analyzed in the process of its detection have been identified, which has allowed the formation of a set of methods for ensuring the protection of corporate computer networks. The presented methods increase the efficiency of the threat detection process by reducing the time for their identification and increasing accuracy. The work proposes

a combination of expert system tools with the proposed methods of detecting and predicting threats, which, due to expert assessments and the driver of conclusions, provides a decision-making support process for responding to existing and predicted vulnerabilities.

The work further developed a model of an information system for detecting and predicting threats for corporate computer networks, which takes into account the integrated use of malware identification modules and inference engine that allows predicting the probability of implementation of certain vulnerability vectors to support decision-making regarding response to threats.

As a module for identifying malicious software of an information system, a method for determining the Linux ELF file section of UNIX-like operating systems is proposed. The method includes processes of semantic analysis and selection of a malware classification model and provides high accuracy and F1-measures of classification. The improved method for identifying Windows PE files is based on the use of the Import Table section in combination with word2vec techniques and an ensemble of decision trees, which provides high accuracy and operational identification of malicious files.

To fill the database of the information system, an improved method for detecting DDoS attacks has been proposed, which contains a combination of Isolation Forest models and EWMA statistics, which allows taking into account the time context of observation series of network parameters to support decision-making about the existence of a threat.

A functional model of an information system for detecting and predicting cyber threats for corporate computer networks has been created, which contains information about the main input and output parameters, limitations, resources, and details of threat detection and prediction processes and can be the basis for designing corporate computer network protection systems.

The practical significance of the results obtained is also determined, which is that they collectively form an information technology for detecting and predicting threats for corporate computer networks using expert systems. The developed interface of the information system for generating reports on detected threats is presented, which contains recommendations for determining optimal strategies for countering threats based on Game Theory and expert assessments. The proposed information technology also contains modules for detecting, identifying, and predicting the probability of implementation of certain vulnerability vectors, as well as a module for entering expert assessments of CVSS threats and countermeasure strategies. The presented business processes and architecture can be adapted for further development and development of information corporate computer intrusion detection systems. The proposed information technology can also be used to simplify the work and increase the productivity of cyber security specialists or network administrators of corporate computer networks.

Keywords: information technology, information security, expert system, corporate computer network, intrusion detection system, Game Theory, NLP, machine learning.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CBR – Case base reasoning (Міркування на основі прецендентів)

CIA – Confidentiality, Integrity and Availability (Конфіденційність, цілісність, доступність)

CISA – Cybersecurity and Infrastructure Security Agency (Агенство з кібербезпеки та захисту інфраструктури)

CVSS – Common vulnerability scoring system (Загальна система оцінки вразливостей)

DDOS – Distributed Denial of Service (Розподілена відмова в обслуговуванні)

DLL – Dynamic-link library (Динамічно приєднувана бібліотека)

DMZ – Demilitarized zone (Демілітаризована зона)

DOS – Denial of service (Відмова в обслуговуванні)

EDR – Endpoit detection and response (Виявлення кінцевої точки та відповідь)

ELF – Executable and Linkable format (Виконуваний та приєднуваний формат)

EWMA – Exponentially weighted moving average (Експоненційно зважене ковзке середнє)

IDS – Intrusion detection system (Система виявлення вторгнень)

IPS – Intrusion prevention system (Система запобігання вторгненням)

MVP – Minimum viable product (Мінімально життєздатний продукт)

NIST – National Institute of Standrads and Technology (Національний інститут Стандартів та Технологій)

KNN – K-Nearest neighbourhood (Метод K найближчих сусідів)

NLP – Natural language processing (Обробка натуральної мови)

PE – Portable executable (Переносний виконуваний)

RBR – Rule based reasoning (Міркування на основі правил)

SGD – Stochastic gradient descent (Стохастичний градієнтний спуск)

SIEM – Security information and event management (Управління інформацією безпеки та подіями)

STIX – Structured threat information expression (Структуроване вираження інформації про загрозу)

SVM – Support Vector Machine (Метод опорних вектоірв)

ПЗ – програмне забезпечення

ВСТУП

Актуальність теми дослідження. В сучасному світі кібербезпека відіграє ключову роль у захисті цифрових даних організацій. Як відомо, переважна більшість сучасних компаній чи організацій – як приватних так і державних, використовують корпоративні комп’ютерні мережі для своєї діяльності. Такі мережі часто стають ціллю атак, які завдають великих збитків та шкоди організаціям-жертвам. Ключовим фактором у захисті корпоративних мереж є оперативне виявлення та прогнозування ймовірності виникнення загроз. Тому удосконалення та розробка методів виявлення та прогнозування загроз для корпоративних мереж є актуальною проблемою сьогодення.

Сучасними викликами до забезпечення кібербезпеки є швидке реагування на відомі та нові кіберзагрози. У випадку, якщо загрози відомі і досліджені, їхні хеш-суми містяться в багатьох базах даних сигнатур антивірусного ПЗ та IDS систем, що спрощує їх виявлення та знешкодження. Однак зловмисники вдаються до модифікації існуючих загроз шляхом додавання нової логіки, зміни порядку команд, пошуку нових експлойтів, що надає актуальності визначення загроз з використанням методів, що здатні виявляти мережеві аномалії, самонавчатись та ідентифікувати знайомі шаблони в нових загрозах. До таких методів зокрема відносяться методи засновані на використанні статистичних моделей та моделей машинного навчання. Над проблемою виявлення та класифікації загроз з використанням статистичних моделей та моделей машинного навчання працювали Ian Shiel, Stephen O'Shaughnessy, Boojoong Kang, Suleiman Y. Yerima, Kieran Mclaughlin, Sakir Sezer, Zhong Fangtian, Скітер І.С., Bin Qin, Shin-Ming Chen, Ferhat Ozgur Catak, Javed Ahmed, Kevser Sahinbas, Zahid Hussain Khand, Sayed M.A., Anwar A.H., Kiekintveld C., Bosansky B., Kamhoua C., A. Ravi, V. Chaturvedi.

Прийняття рішень при розподілі ресурсів щодо захисту потребує визначення рівня критичності загрози, що робить актуальним використання засобів

експертних систем, зокрема методів експертних оцінок в інформаційних системах виявлення загроз. Застосування експертних систем до вирішення проблем кіберзахисту було досліджене Churu Matidaa, Blaauw Dewalda, Watson Brucea, Mahdavifar Samaneh, Ghorbani Ali. Дослідження з застосуванням експертних систем до проблем з кіберзахисту в основному розглядають застосування апарату нечіткої логіки або нейронних мереж для формування рушію висновків. Як альтернативу існуючим дослідженням, доцільно розглянути варіант застосування моделей теорії ігор у якості рушію висновків експертних систем, адже він потенційно менш ресурсоємний для імплементації. Також використання моделей теорії ігор дозволяє оцінювати та прогнозувати ймовірності експлуатації певних загроз зловмисниками та генерувати рекомендації з кіберзахисту, пропонуючи оптимальні стратегії спеціалістам з кіберзахисту та пріоритет їх застосування. Дане питання досліджувалось шляхом знаходження рівноваги Неша або використання мереж Маркова дослідниками Zhu Quanyan, Başar Tamer, Wang Kun, Du Miao, Yang Dejun, Zhu Chunsheng, Shen Jian, Zhang Yan, Sayed M.A., Anwar A.H., Kiekintveld C., Bosansky B., Kamhoua C., K. S. Gill, S. Saxena, A. Sharma. Удосконалення підходів до створення та вирішення антагоністичних ігор за участі кіберзлочинця та спеціаліста з кіберзахисту є актуальним питанням, що потенційно дозволить покращити методи оцінки кіберзагроз.

Отже, **актуальним науковим завданням** є розробка моделей та методів виявлення та прогнозування загроз для корпоративної комп'ютерної мережі засобами експертних систем для підвищення оперативності прийняття рішень щодо реагування на наявні та можливі загрози.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана в рамках визначених задач міжнародного науково-дослідного проекту «Cyber Rapid Analysis for Defense Awareness of Real-time Situation – CyRADARS» відповідно грантової програми NATO SPS, (grant

agreement number: G5286)» та відповідно до плану науково – дослідної роботи Національного університету «Чернігівська політехніка» - «Розробка моделей та методів захисту системи від зовнішніх атак з використанням технологій штучного інтелекту» (№0120U101931).

Мета і завдання дослідження.

Метою дисертаційного дослідження є підвищення оперативності виявлення загроз та забезпечення підтримки прийняття рішень щодо реагування на наявні та можливі вразливості корпоративних комп'ютерних мереж.

Для досягнення мети дослідження в дисертаційній роботі поставлені наступні завдання:

1. Аналіз існуючих кіберзагроз для корпоративних комп'ютерних мереж і методів їх виявлення та прогнозування.

2. Розробка методів ідентифікації шкідливого програмного забезпечення для UNIX-подібних систем та систем сімейства Windows на основі семантичного аналізу та моделей машинного навчання.

3. Розробка методу виявлення мережевих аномалій з можливістю детекції DDoS атак з використанням статистичних методів.

4. Розробка методу прогнозування виникнення загроз з використанням мереж Баєса для корпоративної комп'ютерної мережі.

5. Розробка моделі комплексної інформаційної технології для виявлення кіберзагроз та формування рекомендацій щодо вибору оптимальних стратегій кіберзахисту з використанням експертних систем.

6. Підтвердження ефективності запропонованих методів та комплексної технології в розробленій системі виявлення та прогнозування кіберзагроз для корпоративної комп'ютерної мережі.

Об'єкт дослідження – інформаційні процеси захисту корпоративних комп'ютерних мереж від кіберзагроз.

Предметом дослідження є методи, моделі та елементи інформаційної технології виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж засобами експертних систем.

Методи дослідження

Для проведення дослідження було застосовано методи моделювання функціональних діаграм IDEF0 з використанням декомпозиції основної функціональної моделі виявлення та прогнозування кіберзагроз для корпоративної комп'ютерної мережі. Використано діаграми UML для моделювання варіантів використання системи експертами з кібербезпеки та інженерами з кіберзахисту. Використано метод експертних оцінок для оцінки кількісної міри CVSS критичності виявлених та прогнозованих загроз для корпоративної мережі. Також застосовано методи статистичного аналізу, машинного навчання та методи NLP для виявлення загроз.

Наукова новизна

Вперше:

– метод визначення секції Linux ELF файлу UNIX-подібних операційних систем, який, на відміну від існуючих, містить процеси семантичного аналізу та вибору моделі класифікації і ідентифікації шкідливого програмного забезпечення для підвищення точності та оперативності виявлення загроз.

Удосконалено:

– метод ідентифікації шкідливих Windows PE файлів операційних систем сімейства Windows, який, на відміну від існуючих, використовує секцію таблиці імпортів в поєднанні з моделями word2vec та ансамблю дерева рішень, що забезпечує більшу точність виявлення загроз та F-міру класифікації.

– метод виявлення DDoS атак, який, на відміну від існуючих, містить поєднання моделей Isolation Forest та EWMA-статистики, що дозволяє враховувати часовий контекст рядів спостережень мережевих параметрів для підтримки прийняття рішення про існування загрози;

Набуло подальшого розвитку:

– модель інформаційної технології виявлення та прогнозування загроз для корпоративної комп’ютерної мережі, яка, на відміну від існуючих, враховує комплексне використання модулей ідентифікації шкідливого ПЗ та рушій висновків, що дозволяють виконати прогнозування ймовірності реалізації визначених векторів вразливостей для підтримки прийняття рішень щодо реагування на загрози.

Практичне значення отриманих результатів. Наукові результати дисертаційного дослідження у своїй сукупності утворюють інформаційну технологію виявлення та прогнозування загроз для корпоративної комп’ютерної мережі засобами експертних систем. Розроблено програмний модуль інформаційної системи для формування звітів про виявлені загрози, які містять рекомендації щодо визначення оптимальних стратегій протидії загрозам на основі Теорії Ігор та засобів експертних систем. Запропонована інформаційна система також містить модулі виявлення, ідентифікації та прогнозування ймовірності реалізації визначених векторів вразливостей, а також модуль введення експертних оцінок CVSS загроз та стратегій протидії. Запропонована інформаційна система може бути використана системними адміністраторами або спеціалістами з кіберзахисту для покращення обізнаності щодо існуючих загроз для корпоративної комп’ютерної мережі та підтримки прийняття рішень щодо реагування на наявні загрози. Розроблені бізнес-процеси та архітектура може бути адаптована для подальшої розробки та розвитку інформаційних систем захисту корпоративних комп’ютерних мереж.

Результати дисертаційного дослідження впроваджені в навчальному процесі Національного університету «Чернігівська політехніка» при проведенні лекцій та лабораторних робіт з дисципліни «Системи штучного інтелекту» – в процесі навчання бакалаврів спеціальності 121 – «Інженерія програмного

забезпечення» на кафедрі інформаційних технологій та програмної інженерії (акт про впровадження №202/08-121/ВС від 13.01.2024, Додаток А).

Також результати були використані в ТОВ «Датчикове підприємство «Завод Рапід»» при виявленні шкідливого ПЗ на комп'ютерах в корпоративній мережі та при організації заходів щодо покращення кібербезпеки корпоративної комп'ютерної мережі (акт про впровадження №1154 від 27.12.2024, Додаток А).

Особистий внесок здобувача

Наукові результати, викладені в дисертаційній роботі, отримані автором особисто. В наукових роботах, опублікованих в співавторстві, в дисертації викладені лише ті ідеї та положення, що є результатом особистої роботи.

Апробація результатів дисертації

Основні положення дисертаційного дослідження доповідалися та обговорювалися на XV міжнародній науково-практична конференції «Математичне та імітаційне моделювання систем модс 2020», Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих учених «Новітні технології у науковій діяльності і навчальному процесі» (2023), XIV Міжнародній науково-практичній конференції студентів, аспірантів і молодих вчених «Юність науки 2024».

Публікації

За темою дисертаційного дослідження з викладенням основних результатів опубліковано 8 наукових праць, серед них 5 статей у фахових журналах, 1 з них включена до наукометричної бази Scopus; 3 праці апробаційного характеру.

Структура і обсяг роботи

Дисертаційна робота складається з 4 розділів, переліку умовних скорочень, переліку посилань із 143 джерел та 4 додатків. Загальний обсяг роботи становить 188 сторінок, з яких зміст на 3 сторінках, вступ на 6 сторінках, основний текст на 141 сторінках, список використаних джерел із 143 найменувань на 18 сторінках. Робота містить 44 рисунки та 18 таблиць.

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ ЗАГРОЗ ДЛЯ КОРПОРАТИВНИХ КОМП'ЮТЕРНИХ МЕРЕЖ І МЕТОДІВ ЇХ ВИЯВЛЕННЯ ТА ПРОГНОЗУВАННЯ

1.1 Визначення поняття кіберзагрози та інформаційної безпеки для корпоративної комп'ютерної мережі

Для проведення дослідження у сфері кібербезпеки корпоративних комп'ютерних мереж, необхідно надати визначення поняттю кіберзагрози, визначити та описати модель кіберзагрози, надати характеристику кіберзагроз для корпоративних комп'ютерних мереж та поняттю інформаційної безпеки організацій.

1.1.1 Визначення та модель кіберзагрози для корпоративної комп'ютерної мережі

Відповідно до Закону України «Про основні засади забезпечення кібербезпеки України» "кіберзагроза – це наявні та потенційно можливі явища і чинники, що створюють небезпеку, <…> кібератака - це спрямовані (навмисні) дії в кіберпросторі, які здійснюються за допомогою засобів електронних комунікацій ." [1].

Відповідно до визначення, наданого організацією CISCO, в той час як кіберзагроза є лише індикатором, що зловмисник (хакер) здатний або шукає шляхи неавторизованого доступу до системи, мережі або програми для запуску кібератаки, то кібератака є конкретною дією особи чи організації з метою нанесення шкоди або викрадення конфіденційних даних з інформаційної системи іншої особи чи організації [2]. Якщо порівняти ці два визначення, постає питання про визначення грані, між кіберзагрозою та кібератакою, тому що обидва визначення мають на увазі певну дію. У випадку кіберзагрози дія спрямована на пошук шляхів неавторизованого доступу до системи, а у випадку кібератаки

мається на увазі дія, що має на меті зламати інформаційну систему, здійснити крадіжку конфіденційних даних тобто фактично нанести шкоди жертві.

Задля більш чіткого розуміння про спільні риси та відмінності між кіберзагрозами та кібератаками доцільно дослідити найбільш поширені види кіберзагроз та кібератак. Наприклад, у своєму оглядовому дослідженні кібервразливостей, загроз, атак та рішень з кібербезпеки, Ömer Aslan та ін. [3] визначають поняття кіберзагрози, як шкідливого актора, що отримує неавторизований доступ до комп'ютерних мереж або мережі іншої людини чи організації. Автори виділяють п'ять основних кіберзагроз – комп'ютерні віруси, комп'ютерні хробаки, «троянські» програми, rootkit програми, а також хакери й кібернападники. Об'єднуючою ознакою виділених загроз є те, що вони створюють підґрунтя для подальших атак, таких як крадіжка персональних даних, сповільнення або відмова роботи мережі та ін. Також дослідники виділяють 9 основних типів кібератак: атаки на застосунки, криптографічні атаки, hijacking атаки, атаки на комп'ютерну мережу, фішинг атаки, malware атаки, botnet атаки, атаки з викрадення паролів та man-in-the-middle атаки. Всі з перелічених атак являють собою дію, що має прямі негативні наслідки для однієї людини або організації.

Компанія IBM у своїй статті «Types of cyberthreats» [4] наводить інший список типів кіберзагроз, ніж автори у дослідженні [3], розширюючи його кіберзагрозами типу "Injection attacks" , "Social engineering", "Insider attacks" та "Zero-Day Exploit", що являє собою невідомий раніше незахищений недолік в програмному забезпеченні, апаратному забезпеченні або прошивці. Прикладом zero-day загрози є вразливість Log4Shell, що була виявлена у Листопаді 2021 року та існувала в 10% глобальних цифрових активів. Деякі типи, що автори в дослідженні [3] віднесли до атак, компанія IBM представила у вигляді загроз. Додатково можемо звернутись до документації RFC 4778 [5], що є відкритим інформаційним документом, який містить технічні специфікації та стандарти, що

використовуються у всесвітній мережі Інтернет. Там вказують, що кіберзагроза може бути викликана одним або декількома чинниками, до яких відносять Man-In-The-Middle, використання вразливостей протоколів, вставка/зміна/видалення мережевих повідомлень.

Підсумовуючи проаналізовані дослідження, можемо дійти висновку, що коли мова йде про типи кіберзагроз та кібератак, багато з них відносяться як до кіберзагроз так і до кібератак, тобто один тип може вживатись для вираження як сутності загрози, так і атаки, в залежності від контексту вживання. Таким чином, можемо сформулювати основні типи кіберзагроз та кібератак:

1. Шкідливе програмне забезпечення. Шкідливе ПЗ може бути спрямоване на отримання зловмисником несанкціонованого доступу до даних, вузлів, систем, комунікацій; знищити або зашифрувати дані, викрасти конфіденційні дані або привести до збоїв операційної системи чи обладнання.
2. Denial-of-Service/Distributed Denial-of-Service. Атака полягає у перевантаженні запитами різних вузлів мережі або серверів з використанням одного (DoS) або багатьох (DDoS) скомпрометованих девайсів для запуску атаки.
3. Соціальна інженерія та фішинг. Дана загроза розрахована на людський фактор і незважаючи на відносну простоту реалізації може мати серйозні наслідки, як для особистої кібербезпеки так і для організації в цілому.
4. Man-in-the-Middle. Дана загроза або атака полягає в «підслуховуванні» обміну даних в мережевому з’єднанні, яке не є захищеним. Загроза полягає у витоку або підміні конфіденційних даних.
5. Zero-day exploits. Атаки, що спрямовані на використання вразливостей нульового дня.

6. Password attack. Атаки, що спрямовані на крадіжку паролю та логіну облікового запису користувача, можуть бути як атакою зламу типу brute force так і підвидом соціальної інженерії та фішингу.
7. Internet of Things. Атака, в якій зловмисники використовують вразливості в IoT пристроях, з метою отримати до них доступ та доєднати до мережі botnet.
8. SQL Injection/ Cross-site scripting. Атака спрямована на отримання конфіденційної інформації, зазвичай інформації про логіни, паролі, куки-файли або веб-сесії користувачів веб-застосунків, за допомогою вбудовування в форми введення або в параметри http-запиту відповідного sql коду, або скрипту, зазвичай створеного на мові програмування javascript.

У підсумку, кіберзагроза є наявними явищами, чинниками або діями, які несуть потенційну небезпеку системі, мережі або організації, а кібератака є дією що безпосередньо спрямована на нанесення шкоди інформаційній системі, що належить особі чи організації.

Для дослідження кіберзагроз у рамках інформаційних систем, в тому числі корпоративних комп'ютерних мереж, виконують моделювання загроз, що полягає у виділенні основних сутностей які є частиною експлуатації, оцінки та знешкодження загрози. На основі фреймворку Structured Threat Information eXpression (STIX) [6], та дослідження Abel Yeboah-Ofori [7], виділимо базові сутності моделі кіберзагрози:

1. Зловмисник. Являє собою актора, дії якого направлені на пошук та експлуатацію загрози, а ціллю є нанесення шкоди системі, в якій виявлена загроза.
2. Техніки, тактики та процедури. Набір технік, тактик та процедур, які зловмисник використовує для пошуку та експлуатації кіберзагрози. До таких технік можуть бути віднесені методології MITTRE ATT&CK,

Cyber Kill Chain. Також зловмисник може винаходити та застосовувати новітні унікальні техніки для ускладнення виявлення його дій.

3. Вектор вразливості. Визначає тип вразливості та механізм її поширення.
4. Вразливість. Сутність, що є частиною вектору вразливості та використовується для ідентифікації загрози, та оцінки рівня критичності загрози з використанням різних технік, зокрема CVSS.
5. Цілі. Сутність, що являє собою цілі зловмисника по експлуатації загрози, та визначається типом організації та активами всередині організації, на які спрямовані дії зловмисника.
6. Вплив. Є результатом досягнення цілі зловмисником, та класифікується за типом та виміром критичності. Серед типів впливу виділяють витік, пошкодження або знищення даних, фінансові втрати, пошкодження або знищення сервісу. Критичність впливу вимірюється кількістю та якістю завданих збитків.
7. Протидія. Протидія спрямована на виявлення та усунення вразливості та вектора загрози в цілому, що досягається використанням різних спеціалізованих систем: IDS, IPS, EDR, SIEM.

Відповідно до виділених сутностей та побудованих зв’язків між ними, була створена модель «сутність-зв’язок» кіберзагрози для корпоративної комп’ютерної мережі, що представлена на рисунку 1.1. З побудованої моделі бачимо, що для формування методів виявлення та прогнозування кіберзагроз для корпоративних комп’ютерних мереж, доцільно зосередитись на виявленні та характеристиці вектора вразливості. Ціллю зловмисника, що розглядається в даному дослідженні, є корпоративна комп’ютерна мережа, що може містити в собі такі активи як кінцеві пристрої, сайти та програмне забезпечення. Для оцінки впливу та протидії для вектора вразливості може бути використаний метод експертних оцінок.

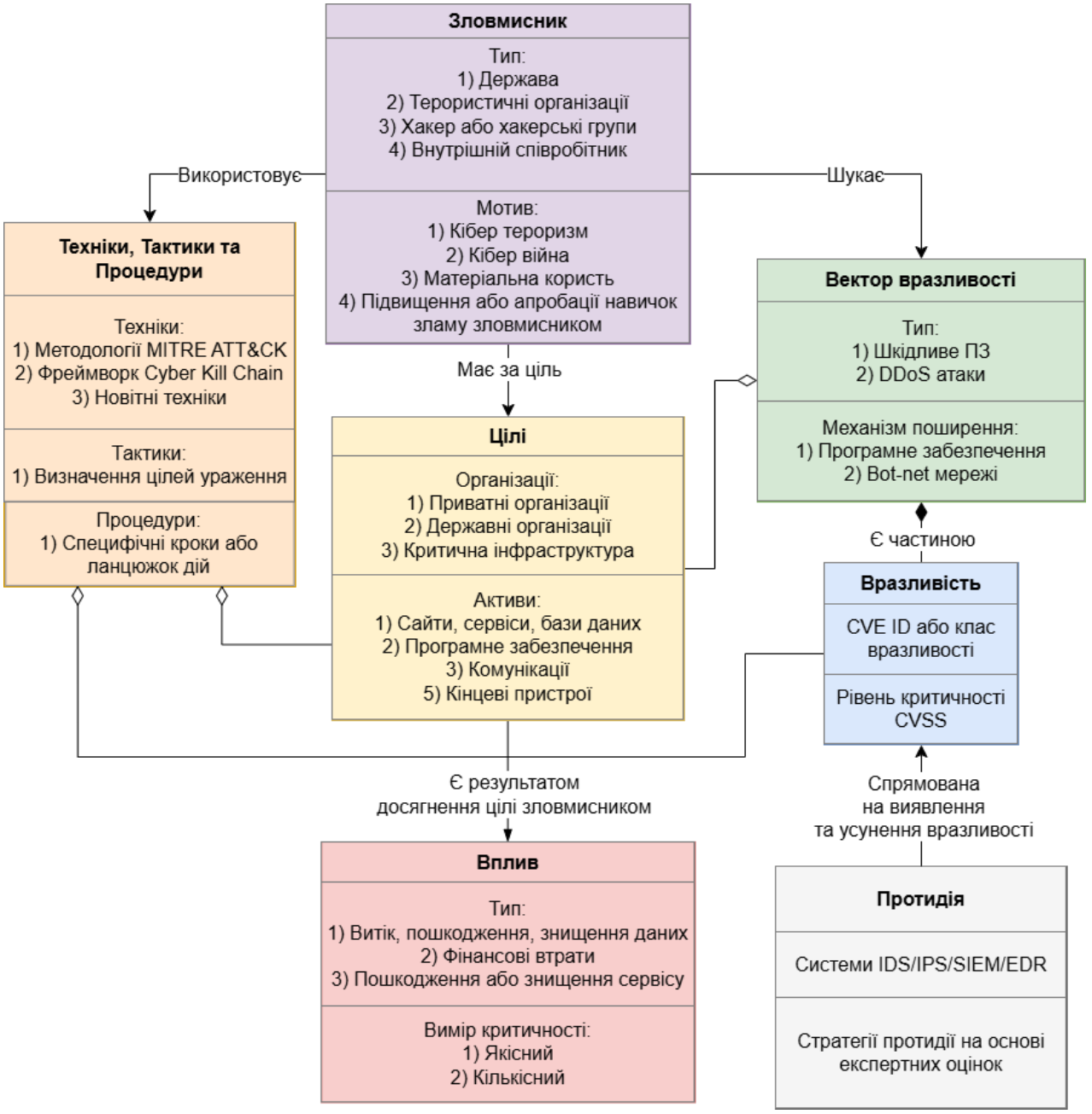


Рисунок 1.1 – Модель кіберзагрози для корпоративної комп’ютерної мережі

Підсумовуючи викладену інформацію, було представлено визначення кіберзагрози, виділено основні сутності кіберзагроз для корпоративних комп’ютерних мереж, на основі яких побудовано модель, що відображає дані сутності та зв’язки між ними. Засновуючись на побудованій моделі кіберзагрози для корпоративних комп‘ютерних мереж, також було визначено основні

сутності, які необхідно аналізувати у процесі виявлення кіберзагроз для корпоративної комп'ютерної мережі.

1.1.2 Характеристика кіберзагроз для корпоративних комп'ютерних мереж

Корпоративна комп'ютерна мережа – це мережа, основним призначенням якої є підтримка роботи організації чи підприємства, яке нею володіє. Таким чином, кіберзагрорзи для корпоративної комп'ютерної мережі, в першу чергу є загрозами для організації, що володіє даною мережею. У своєму дослідженні Bahuguna, A. та ін. [10] стверджують, що основними проблемами в реалізації кібербезпеки для організації є брак робочої сили, забезпечення відповідності вимогам безпеки зберігання даних та неправильний вибір фреймворків для реалізації інформаційних систем та застосунків. Дані проблеми слугують підгрунтям для відповідних кібератак на організацію.

Дослідники Karakaya M. та Sevin A. у своїй роботі [11] виділяють такі кібератаки та кіберзагрози, характерні для організацій: атаки на веб-застосунки, серед яких broken-access control, що надає доступ до системи неавторизованому користувачу, dos та ddos атаки, sql-ін'єкції. Аналізуючи запропонований перелік атак, можемо дійти висновку, що він включає основні види кіберзагроз, досліджені нами у розділі 1.1.1, які направлені на конкретну компанію або організацію.

Ключовою ознакою кібератаки, що здійснюється на корпоративну комп'ютерну мережу, яка належить державній чи приватній організації, є цільові активи даної організації, серед яких можуть бути сайти, сервіси, бази даних, кінцеві пристрої, комунікації, тощо. В той же час мотивація здійснення атаки може не залежати від кінцевої мети або фактичних наслідків атаки та включає в себе кібертероризм, кібервійну, отримання матеріальної користі або підвищення чи апробацію нових навичок зламу зловмисником. Malik, Annas та ін. у своєму

дослідженні [12], запропонували розділити кіберзагрози та кібератаки за мотивацією умовно на 2 основні категорії: атаки на організації, компанії, що мають на меті ідеологічне підґрунтя – кібертероризм, та ведення агресивних дій або війни проти організації або цілої країни з використанням кібератак – кібервійна.

У якості прикладу атак, що були спрямовані як на окремі корпоративні мережі, так і на організації, та мають ознаки кібертероризму та ведення кібервійни є серія кібератак проти приватних та державних корпорацій України. Однією з найбільш деструктивних кібератак, починаючи з 2014 року, став вірус-вимагач “Petya”, розповсюдження якого розпочалось у квітні 2017 року. Бекдор був зашитий у файли DLL ZvitPublishedObjects.dll, що потрапляв на комп’ютери в мережі компаній-жертв разом з оновленням для бухгалтерської програми M.E.Doc. Вірус не тільки шифрував дані та вимагав викуп, але й збирав корпоративну інформацію, таку як код ЄРДПОУ компанії, логіни й паролі до поштових скриньок та проксі серверів. Оскільки код ЄРДПОУ є унікальним для кожної окремої компанії, це дозволило кіберзлочинцям адаптувати методи атак для кожної окремої компанії щоб максимізувати нанесені збитки. Жертвами атаки стали щонайменше 2000 компаній до червня 2017 року [13], [14].

Прикладом кібератак, спрямованих проти енергетичних компаній України, можна привести вірус “Industroyer”, що був спрямований на виведення з ладу енергетичної інфраструктури. Атака з використанням першої версії вірусу “Industroyer” була успішно виконана 17 грудня 2016 року, що стало результатом масових відключень світла в Україні. Модифікований різновид вірусу “Industoyer 2” був запущений у квітні 2022 року, та мав на меті фізичне виведення з ладу обладнання на енергетичних станціях та знищення даних в корпоративних мережах енергетичних компаній. Вірус мав 2 модулі, один з яких, вбудований у PE файли, відповідав за інфікування операційних систем “Windows”, інший, вбудований у Bash скрипти, – за інфікування операційних систем “Linux” та

знищення слідів діяльності вірусу. Завдяки сукупності чинників, таких як недоліки в програмній частині вірусу “Industoyer 2”, вчасній реакції української урядової організації з кібербезпеки UA-CERT та з допомогою компанії ESET, руйнівних наслідків вдалось уникнути [15].

Приведені атаки були здійснені шляхом вживляння в корпоративну комп’ютерну мережу шкідливого програмного забезпечення – malware, що є однією з найбільш розповсюджених загроз. Важливим аспектом у дослідженні кіберзагроз для корпоративної комп’ютерної мережі є розуміння джерела та рушію поширення загрози. Наприклад, автори сайту VirusTotal – одного з найбільш популярних сайтів з відкритою публічної інформацією про зловмисне ПЗ (malware), опублікували звіт про тренди застосування malware станом на 2023 рік [16]. Відповідно до опублікованого звіту, до найпопулярніших джерел атак з використанням malware можна віднести додатки до email, файли типу excel, word, pdf, iso тощо. Стабільно потягом 2020 – 2023 років, поширеними є віруси-файли типу Windows Portable Executable, що можуть бути EXE та DLL типів. Серед механізмів поширення malware автори звіту виділяють скачування неперевірених прикріплених файлів до email-повідомлень. Також, відповідно до наведеного прикладу з вірусом, що розповсюджувався разом з оновленням бухгалтерської програми M.E.Doc, можемо визначити, що можливим механізмом потрапляння шкідливого ПЗ в корпоративну мережу є бекдори в оновленні нешкідливих програм.

Виділяють 8 основних типів шкідливого ПЗ [17,18] :

1. Хробак (Worm) – шкідливе ПЗ, що самостійно поширюється комп’ютерною мережею, сповільнюючи її дію або несучи віруси іншого типу, такі як Trojan або інші.
2. Trojan – шкідливе ПЗ, що приховує справжнє своє призначення.

3. Вірус-вимагач (Ransomware) – шкідливе ПЗ, направлене на віддалене блокування операційної системи та шифрування файлів, метою якого є шантажування з метою отримання грошової винагороди.
4. Шпигунське ПЗ (Spyware) – шкідливе ПЗ, що встановлюється на кінцевий пристрій без згоди користувача, та збирає конфіденційну інформацію, облікові та особисті дані користувача кінцевого пристрою.
5. Рекламне ПЗ (Adware) – шкідливе ПЗ, що ускладнює виконання повсякденних задач користувачем кінцевого девайсу, виводячи рекламні банери, вносячи несанкціоновані зміни налаштування пошукових систем та встановлюючи додаткові програми.
6. Botnet – шкідливе ПЗ, що інфікує кінцевий пристрій, який буде використано віддалено для проведення DDoS-атаки.
7. Rootkit – шкідливе ПЗ, що отримує доступ до системних засобів керування процесами, файлами, каталогами, драйверами адміністратора системи та використовується в сукупності з іншими типами вірусів для приховання слідів їх діяльності.
8. Fileless – вид кіберзагрози, що фізично не зберігається на жорсткому диску, а напряму завантажується в оперативну пам’ять шляхом запуску легітимних системних програм, таких як PowerShell.

Існують 2 основні способи виявлення шкідливого ПЗ: з використанням статичного аналізу та використанням динамічного аналізу [19]. Виявлення шкідливого ПЗ з використанням статистичного аналізу передбачає сканування файлу без його запуску. Таким чином, статичний аналіз дозволяє уникнути шкідливого впливу ПЗ, не вимагаючи при цьому його ізоляції для визначення наявності загроз у файлу.

Базовим підходом у здійсненні статичного аналізу шкідливого ПЗ є аналіз з використанням сигнатури файлу. Він полягає у формуванні унікального ідентифікатора файлу шкідливого ПЗ та порівняння його значення з відомими

існуючим шкідливим ПЗ. Зазвичай у якості ідентифікатора використовують хеш-суму (алгоритм MD5 або SHA256) цілого файлу. Перевагою даного методу є його простота реалізації та швидкість виявлення, але недоліком є нестійкість до будь-яких змін існуючих видів шкідливого ПЗ, обфускації тощо.

Для покращення сигнатурного методу виявлення шкідливого ПЗ використовують метод нечіткого хешування – Context Triggered Piecewise Hashing, що був розроблений Jesse Kornblum у роботі [20] та полягає у розбитті файлу на різні фрагменти та обчисленні хеш-суми окремо для кожного з фрагментів файлу. Це дозволяє зберігати схожість хеш-суми при зміні окремих фрагментів файлу, на відміну від алгоритмів чіткого хешування. Даний підхід дозволяє виявити нові варіації існуючого шкідливого ПЗ, шляхом порівняння нечітких хеш-сум. Наприклад, в 2021 році для антивіруса Microsoft 365 Defender був розроблений новий алгоритм виявлення шкідливого ПЗ с застосуванням нечіткого хешування, NLP та машинного навчання [21]. Його суть полягає в застосуванні техніки вкладання слів до нечітких хеш-сум вірусів для подальшої класифікації з використанням багатошарового перцептрона. Завдяки цьому алгоритму, антивірусу вдалось ідентифікувати нову варіацію вірусу GoldMax, яка пізніше була підтверджена та опублікована у найбільш відомих базах даних вірусів, зокрема VirusTotal.

Іншим популярним підходом до статичного аналізу шкідливого ПЗ є поєднання технік добування даних та машинного навчання. В якості вхідних даних для моделей машинного навчання можуть використовуватись байт-коди файлу, список імпортованих бібліотек, список текстових значень змінних strings. Вхідні дані можуть бути векторизовані та класифіковані з використанням NLP алгоритмів, згорткових нейромереж або нейромережами глибинного навчання. Також до векторизованих даних додатково можуть бути застосовані інші алгоритми машинного навчання, для класифікації або регресійного аналізу.

Динамічний підхід виявлення шкідливого ПЗ передбачає запуск ПЗ та спостереження за його діями, побудову профілю активності та аналізу слідів роботи ПЗ. Зазвичай динамічний аналіз проводять в ізольованому середовищі для уникнення інфікування основної системи. Серед даних, які можуть бути використані для формування висновків про безпечність файлу можуть бути логи про виконання, послідовність API викликів, інформація про використання системних реєстрів.

Іншим типом загроз, з якими часто стикаються організації – це DoS та DDoS атаки. Атаки такого типу поєднують простоту реалізації для великих хакерських угрупувань та водночас несуть велику шкоду жертвам. Проведення даних атак не вимагає детального знання про структуру та топологію корпоративних мереж та не вимагає проникнення в середину мережі. Для проведення атаки достатньо визначити хост-жертву, яка може виступати http-сервером, API сервісом, cloud-based сервісом та запустити атаку. Основними типами DDoS атак є Volumetric attack, Protocol attack або Application Layer attack. Volumetric attack не вимагає встановлення зв’язку з жертвою і діє по UDP протоколу, застосовуючи UDP-flood техніку, що полягає у завантаженні кінцевого пристроя або розподіленого сервісу великим обсягом даних, що вимірюються в бітах на секунду або гігабітах на секунду. Такі атаки можуть бути виявлені визначенням аномальної зміни обсягу трафіку до кінцевого пристрою. Протидією таким атакам є імплементація або використання готового scrubbing центру, призначенням якого є масштабування для обробки, виявлення аномалій та блокування великих обсягів трафіку. Protocol attack вимагає встановлення зв’язку з кінцевим пристроєм або сервісом та надсилання великої кількості мережевих пакетів, на які пристрій зобов’язаний відповісти. Дані типи атак включають SYN flood, ICMP flood, DNS amplification, SYN-ACK flood та можуть бути виявлені за допомогою аналізу аномалій мережевого трафіку. Application layer атаки полягають у завантаженні шкідливих скриптів на сервери шляхом виконання SQL та XSS ін’єкцій. Дані атаки

вимірюються в запитах за секунду, та характеризуються невисоким обсягом мережевого трафіку. Захистом від таких атак може бути використання CSRF-токенів, сервісу Captcha, фільтрації вводу користувача або виявлення та блокування зловмисних IP адрес.

Джерелом DDoS volumetric та protocol атак зазвичай є бот-нет мережі, що складаються з персональних комп'ютерів, що інфіковані вірусом, зазвичай типу Trojan, з доступом до них зловмисників для виконання атаки на певну організацію [22]. У випадку application layer атак це можуть бути шкідливі javascript або php файли.

Іншим розповсюдженим типом атак на корпоративні комп'ютерні мережі є фішинг атаки. Відповідно до документу RFC 5901 [23], фішинг атаки полягають у маскуванні зловмисної веб-сторінки під легітимну веб-сторінку, що вимагає введення персональних даних користувача. Таким чином, зловмисники можуть отримати облікові дані користувача, що несе загрозу і персонально користувачу, і безпеці корпоративних даних в цілому. Зазвичай джерелом таких атак є email-повідомлення або повідомлення в соціальній мережі, що мотивує жертву перейти за посиланням. Для виявлення таких атак можуть бути використані чорні списки фішинг сайтів, email фільтри або відслідковування профілю активності користувачів.

Іншим типом загроз є Advanced Persistent Threat (APT), що являють собою довготривалі кібератаки спрямовані проти організацій. Основною властивістю APT загроз є проникнення всередину корпоративної мережі організації та підтримання існування загрози у прихованому вигляді довгий проміжок часу. Таким чином такі загрози можуть наносити максимальної шкоди організації, здійснюючи крадіжку конфіденційних даних, зриваючи внутрішні процеси організації. Для поширення даних загроз зловмисники можуть використовувати композицію методів проникнення, такі як фішинг, соціальну інженерію, експлуатування вразливостей нешкідливого ПЗ, компрометація ланцюгів

постачання та third-party сервісів, фізичний доступ до апаратного забезпечення, різні види шкідливого ПЗ, залучення працівників організації. Прикладом такої загрози є атака групи Sandworm на провайдера мобільної мережі «Київстар» у 2023 році, основою якої стала wiper malware, що інфікувало корпоративну мережу компанії та існувало там декілька місяців перед тим, як отримати повний доступ до віртуальних серверів та вивести з ладу 10000 комп'ютерів, більше ніж 4000 серверів, всю хмарну інфраструктуру та системи бекапу [24]. В залежності від механізму виконання, можна виділити 3 основні типи apt атак: атаки на ланцюги постачання, атаки з використанням вразливостей нульового дня та атаки з використанням шкідливого ПЗ. Механізмами виявлення таких атак є: використання цілей-приманок «honeypot», для заплутування кіберзлочинця; аналіз логів систем на наявність незвичної активності; аналіз профіля активності користувачів на наявність незвичної активності.

Соціальна інженерія є також однією з кіберзагроз для корпоративних комп'ютерних мереж, та використовується для запуску різних видів атак або викрадення персональних даних користувачів. Шляхами потрапляння є фішинг email або соціальні мережі. Дані кіберзагрози виконуються шляхом прямої взаємодії зловмисника з користувачем-жертвою. Механізмами виявлення таких загроз є відслідковування профілю активності користувачів або встановлення email фільтрів.

Надамо порівняльну характеристику найбільш розповсюдженим кіберзагрозам для корпоративної комп'ютерної мережі, що приведена у таблиці 1.1. В таблиці вказано основні типи кожної загрози, а також її джерела, механізми потрапляння в корпоративну комп'ютерну мережу та механізми виявлення. У даному дослідженні, для побудови методів виявлення та прогнозування обрано шкідливе ПЗ та DDoS, як найбільш розповсюджені загрози.

Таблиця 1.1 – Порівняльна характеристика найбільш розповсюджених загроз для корпоративної комп’ютерної мережі

Назва загрози	Джерело загрози	Типи загроз	Механізми потрапляння в корпоративну мережу	Механізми виявлення загрози
1	2	3	4	5
Зловмисне ПЗ	DLL, EXE, ELF, PDF, HTML, DOC, DOCX	Worm, Trojan, Ransomware, Spyware, Adware,Botnet ,Rootkit,Fileless	Завантаження користувачем, Email-прикріплення, Оновлення нешкідливих програм	Сигнатурний метод, Динамічний аналіз, Статичний аналіз
Dos/DDoS	Bot-Net мережі	Volumetric attack, Protocol attack,Application layer attack	Інтернет-трафік, SQL-ін’єкції, XSS-ін’єкції	Виявлення аномалій мережевого трафіку
Фішинг	Email листи, Шкідливі дії користувача	Email-фішинг, Social-фішинг	Веб-сторінки, Соціальні мережі	«Чорні списки» фішинг сайтів, Email фільтри, Відслідковування профілю активності користувача

Продовження таблиці 1.1

1	2	3	4	5
APT-вразливості	Email листи, Шкідлива активність користувача, Оновлення ПЗ, Third-party сервіси, Вороже налаштовані працівники організації	Атаки на ланцюги постачання, атаки з використанням вразливостей нульового дня, атаки з використанням шкідливого ПЗ	Фішинг, соціальна інженерія, шкідливе ПЗ, Оновлення нешкідливого ПЗ, Злам third-party сервісів, залучення працівників організації	використання цілей-приманок «honeypot», аналіз логів систем, аналіз профіля активності
Соціальна інженерія	соціальні мережі, email	-	взаємодія користувача зі зловмисником	аналіз профілю активності користувача

Таким чином, визначено та охарактеризовано основні загрози для корпоративних комп'ютерних мереж, що є основою для формування методів їх виявлення та прогнозування.

1.1.3 Визначення поняття інформаційної безпеки організації

Відповідно до Закону України «Про основні засади забезпечення кібербезпеки України», поняття кібербезпеки визначають як: «захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз

національній безпеці України у кіберпросторі» [1]. У рамках організації дане визначення може бути інтерпольоване на співробітників, вузли, комунікації та дані організації, для яких необхідно бути забезпечене своєчасне виявлення, запобігання та нейтралізація реальних і потенційних загроз кібербезпеці.

Згідно з визначеннями, що надають компанія-виробник мережевого обладнання та сервісів кібербезпеки CISCO, кібербезпека являє собою сукупність дій, що направлені на захист систем, мереж та програм від кібератак. До предметної області кібербезпеки відносяться люди, процеси та технології [25].

Люди представляють собою користувачів систем, мереж та програм та повинні слідувати інструкціям та положенням з використання систем та окремих її компонентів задля мінімізації ймовірності стати жертвою кібератаки: зокрема встановлювати надійні паролі, робити регулярні резервні копії даних, не переходити по підозрілим посиланнях з email-повідомлень та не завантажувати неліцензоване програмне забезпечення невідомих виробників.

Процеси являють собою сукупність та послідовність кроків для організації, що направлені на протидію кібератакам, що плануються та успішно проведеним кібератакам. Однією з загальноприйнятих моделей, що визначає процеси кібербезпеки є програмний каркас (фреймворк) NIST - National Institute of Standards and Technology, що розроблений Національним інститутом стандартів та технологій США [26]. Структура фреймворку NIST складається з шести функцій:

1. Govern – у організації існує стратегія управління ризиками кібербезпеки;
2. Identify – у організації є розуміння поточних ризиків кіберзагроз;
3. Protect – запобіжні заходи для забезпечення кіберзахисту вжиті організацією;
4. Detect – можливі кібератаки та скомпрометовані вузли організації знайдені та проаналізовані;

5. Respond – організацією вжиті заходи відповідно до виявлених інцидентів кібербезпеки;
6. Recover – об’єкти, активи, вузли організації, які стали жертвою інцидентів кібербезпеки, відновлені.

Технологія має важливе значення для надання організаціям і окремим особам інструментів комп’ютерної безпеки, необхідних для захисту від кібератак. Важливо захистити три основні об’єкти: кінцеві пристрої, такі як комп’ютери, та роутери; мережеву та хмарну інфраструктуру. Загальні технології, які використовуються для захисту цих об’єктів, включають брандмауери наступного покоління (Next Generation Firewall), системи виявлення та перешкоджання вторгнень (IDS, IPS), фільтрацію системи доменних імен (DNS), захист від зловмисного програмного забезпечення, антивірусне програмне забезпечення та рішення безпеки електронної пошти.

Таким чином, для забезпечення кібербезпеки організації необхідно зосередитись на забезпеченні людей-користувачів чіткими політиками та рекомендаціями для забезпечення кібербезпеки, мати розуміння поточних ризиків кіберзагроз, створити процеси виявлення, реакції та відновлення після здійснених кібератак та впровадити технології, що дозволяють захищатись та протидіяти кібератакам.

1.2 Аналіз існуючих систем виявлення кіберзагроз для корпоративних комп’ютерних мереж

Існують 4 основні типи систем з кіберзахисту – IPS, IDS, SIEM та EDR [27]. IPS системи працюють автоматично, без втручання людей і спрямовані на найшвидшу нейтралізацію будь-яких виявлених даною системою загроз. Даний підхід має ризики хибних спрацювань, проте його перевагою є швидкість реагування. IDS системи в свою чергу генерують обширний звіт із можливих загроз та надають можливість спеціалісту приймати рішення щодо їх усунення.

SIEM системи пропонують функціонал, що є сукупністю функціоналу систем IPS та IDS – надають звіти з безпеки користувачу, та в той же час, мають функціонал з нейтралізації кіберзагроз. Системи, що поєднують в собі функціонал IPS, IDS та SIEM систем є досить популярні та виступають важливою ланкою в побудові архітектурі домашніх та корпоративних мереж. Системи IPS, IDS та SIEM можуть встановлюватись як на кінцевий пристрій в корпоративній мережі так і на вході до мережі – в якості додаткового ПЗ на роутер або апаратний мережевий екран.

EDR системи (Endpoint Detection and Response) спрямовані на відслідковування можливих кіберзагроз безпосередньо на кінцевих точках мережі – на персональних комп'ютерах, ноутбуках, смартфонах, IoT пристроях та інших фізичних пристроях [28]. Системи EDR поєднують функціонал виявлення кіберзагроз, сповіщення спеціалістів з кібербезпеки, а також активної протидії зі стримування та протидії загрозам.

Далі виконаємо порівняльний аналіз існуючих систем з виявлення кіберзагроз.

1.2.1 Snort

IDS/IPS Snort, розроблений М. Рошем у 1998 році, спочатку був задуманий як легка альтернатива комерційним рішенням для виявлення вторгнень. Спершу це був невеликий проект з відкритим кодом, але з часом він став популярним інструментом серед фахівців із кібербезпеки як у корпоративному, так і в приватному секторі. Станом на 2024 рік кількість завантажень Snort перевищила 5 мільйонів, а активна спільнота користувачів налічує понад 600 тисяч осіб [29].

Snort може працювати в кількох режимах: як сніфер мережевого трафіку, реєстратор пакетів або система виявлення та запобігання вторгненням (IDS/IPS). Для виявлення кіберзагроз Snort використовує набір правил Talos – групи лідерів серед експертів кіберзахисту, що працюють над вивченням та оцінкою

найновіших кіберзагроз та створення правил з кібербезпеки, які використовуються також в антивірусному ПЗ CalmAV, Cisco та інших [30].

Snort правила складаються з заголовку, в якому визначається дія, протокол, ip адреса, порт та оператор, що вказує напрямок мережевого трафіку між двома ip адресами. Тіло правила визначає назву правила та критерії, що характеризують мережевий трафік необхідний для спрацювання правила. Для спрощення написання правил, Snort надає узагальнені три типи правил: правила служби, правила файлів та правила ідентифікації файлів. Правила служби — це новий тип правил в Snort 3, який дозволяє авторам правил зіставляти трафік певної служби за допомогою заголовка правила, який складається з дії та назви служби прикладного рівня. Правила файлів надають змогу аналізувати та реагувати на будь-які файли, що містяться в протоколах HTTP, SMTP, POP3, IMAP, SMB, FTP.

Хоча Snort був створений як окремий інструмент, його часто інтегрують з іншими системами для отримання повного огляду безпекових подій у мережі. Основна функція Snort — моніторинг мережевого трафіку та реакція на потенційно небезпечні події, наприклад, через реєстрацію оповіщень. Проте для зручного управління та аналізу даних його зазвичай використовують разом із іншими платформами моніторингу безпеки.

Цей інструмент також легко поєднується з іншими мережевими рішеннями, такими як iptables або мережеві екрани, що дає змогу автоматично блокувати підозрілі дії певних хостів. На даний момент ПЗ Snort володіє компанія Cisco.

З виходом Snort 3 було додані такі покращення як багатопоточна обробка пакетів. Snort може використовуватися в різних сценаріях залежно від потреб користувача. Основне його завдання — моніторинг мережевого трафіку для виявлення підозрілих активностей. Проте самостійно Snort не надає детального аналізу загроз, оскільки його основна функція — реагувати на вказаний трафік, виконуючи визначені дії. Зазвичай це реєстрація подій у журналі оповіщень.

Тому Snort часто інтегрують з іншими системами моніторингу, які забезпечують зручний доступ до аналітики й перегляд усіх оповіщень.

1.2.2 Suricata

IDS/IPS Suricata була вперше випущена в бета-версії у 2009 році, а офіційний реліз відбувся у 2010 році. Ця система була створена для подолання обмежень одно потокових IDS/IPS, таких як Snort, і запропонувала більш продуктивний підхід для обробки мережевого трафіку [31].

Suricata використовує ті ж бібліотеки, що й Snort, включаючи Libpcap і PF_RING. Остання забезпечує підвищену швидкість прийому й аналізу мережевих пакетів. Як і Snort, Suricata належить до класу IDS/IPS, тобто може працювати в режимах IDS або IPS, аналізуючи трафік на основі попередньо визначених правил. Точність цих правил впливає на ймовірність виникнення хибно позитивних чи хибно негативних спрацювань. У режимі IDS можливі такі дії, як "alert" (сповіщення) та "log" (реєстрація подій), а в IPS додаються "drop" (блокування пакету), "sdrop" (приховане блокування) і "reject" (відмова з повідомленням) [32].

Suricata була розроблена як розширений варіант Snort, з урахуванням сучасних вимог до продуктивності. Suricata використовує багатопотокову архітектуру, що дозволяє ефективно розподіляти обчислювальні ресурси багатоядерних систем і забезпечувати паралельний аналіз трафіку. Завдяки цьому система здатна працювати значно більш масштабовано. У випадку використання на одноядерних пристроях Suricata також підтримує однопотоковий режим, де пакети обробляються послідовно. Механізм виявлення загроз у Suricata побудований на основі кількох потоків, що дає змогу оптимально розподіляти навантаження між різними процесами. Такий підхід покращує точність виявлення загроз та забезпечує стабільну роботу навіть при великому обсязі мережевого трафіку. Однак з виходом Snort3, який є прямим конкурентом IDS

Suricata та теж почав підтримувати багатопоточну обробку мережевих пакетів, продуктивність обох систем вирівнялась.

1.2.3 Cisco Secure Endpoint

EDR система Cisco Secure Endpoint є хмарним програмним забезпеченням, що дозволяє відстежувати, реагувати та відновлюватись після кібератак [33]. Система збирає дані з пристроїв, використовуючи внутрішній фреймворк Orbital. Orbital — це хмарний інструмент для дослідження та реагування на атаки [34]. Він дозволяє користувачеві збирати інформацію про систему та безпеку з мережевих пристроїв клієнта та реагувати на будь-які виявлені загрози. Для цього Orbital дозволяє збирати дані з пристроїв мережі за допомогою SQL, а потім використовувати сценарії Python для відповіді на виявлені загрози. Orbital використовує інструментальну основу ОС osquery [35], щоб дозволити запитам SQL виконуватися на кінцевих точках організації.

Однією з основних функцій Cisco Secure Endpoint є попередження та нейтралізація кіберзагроз на ранніх етапах. До таких функцій відноситься виявлення шкідливого програмного забезпечення, попередження експлойтів, попредження script-based атак, поведінковий аналіз та захист від USB атак.

Процедура виявлення шкідливого ПЗ включає декілька кроків. По-перше виконується пошук інформації про репутацію файлу у внутрішній базі даних, що дозволяє зробити висновки про потенційну загрозу. По-друге, на кожному вузлі мережі розміщується база сигнатур вірусів для ОС Windows, Linux та Mac, що дозволяє виявляти шкідливе ПЗ базі сигнатурного аналізу, навіть при втраті мережевого з’єднання. Для виявлення вірусів, які не знаходяться в базі сигнатур проте є модифікацією існуючих типів вірусів, Cisco Secure Endpoint використовує техніку «loose fingerprint», що порівнює схожість контенту підозрілого файлу з відомими родинами вірусів. По-третє, Cisco Secure Endpoint використовує машинне навчання для виявлення шкідливих файлів та підозрілого профілю активності з використанням набору даних, що містяться в базі Cisco Talos.

Система захищає кінцеві пристрої від експлойтів, що включають в себе memory-based атаки, тобто виконання шкідливих дій напряму в оперативній пам’яті без збереження файлів на жорсткий диск; від script-based атак з використанням скриптових мов програмування, таких як VBScript, JavaScript та HTML Applications (HTA). Також система попереджає виконання несанкціонованих дій, що полягають в запобіганні завантаження процесами визначених DLL файлів в оперативну пам'ять.

Також система здійснює виявлення загроз підвищеної складності, що вже існують на кінцевих приладах. Процедура виявлення загроз здійснюється шляхом побудови поведінкових профілів шкідливої активності кіберзагроз, які також називаються Indicators of Compromise.

1.2.4 Splunk Enterprise Security

Splunk — це американська IT-компанія, яка спеціалізується на розробці програмного забезпечення для пошуку, моніторингу та аналізу великих обсягів різнорідних даних. Компанія позиціонує свою платформу як Data-to-Everything, наголошуючи на можливості роботи з будь-якими даними в реальному часі.

Splunk Enterprise Security — це система класу SIEM [36] (Security Information and Event Management), розроблена для аналізу та прогнозування кіберзагроз. Вона базується на власному фреймворку Adaptive Operations Framework і використовує методи машинного навчання для агрегації даних з різних джерел та адаптації операцій над ними. Цей інструмент допомагає визначати пріоритети інцидентів безпеки залежно від їхньої послідовності та важливості, а також надає можливості для управління подіями й загрозами.

Система дозволяє не лише відстежувати та аналізувати події, але й оцінювати наявні ризики та створювати інтерактивні візуалізації стану мережі в режимі реального часу. Splunk Enterprise Security також підтримує хмарні технології та може працювати у середовищах, побудованих за принципами SoC

(Software on Cloud). Інтерфейс системи досить деталізований, та може надавати відомості про узагальнений рівень ризиків для кібербезпеки мережі.

1.2.5 Порівняльний аналіз систем виявлення кіберзагроз для корпоративних комп'ютерних мереж

На основі аналізу систем виявлення кіберзагроз для корпоративних комп'ютерних мереж, створено порівняльну таблицю 1.2 основних їх характеристик.

Таблиця 1.2 – Порівняльний аналіз систем виявлення кіберзагроз

Критерій	Snort	Suricata	Cisco Secure Endpoint	Splunk Enterprise security
1	2	3	4	5
Тип системи	IDS/IPS	IDS/IPS	EDR	SIEM
Платформа	ПК, сервер	ПК, сервер	Клієнт встановлюється локально, система працює у хмарному середовищі	Клієнт встановлюється локально, система працює у хмарному середовищі
Цільове середовище	Мережі різних типів	Мережі різних типів	Корпоративні мережі	Корпоративні мережі
Методи виявлення загроз	Talos та Snort Community правила	Community правила	Сигнатурний аналіз, порівняльний аналіз, машинне навчання, поведінковий аналіз	Машинне навчання

Продовження таблиці 1.2

1	2	3	4	5
Інтеграції	Iptables, брандмауери	Iptables, мережеві інструменти	Cisco SecureX, інші Cisco продукти	Інтеграція з хмарними платформами та IoT
Open source	+	+	-	-

В результаті порівняння дійшли висновку, що сучасні системи EDR та SIEM системи для захисту корпоративних комп'ютерних мереж адаптують методи машинного навчання, поведінкового аналізу та інші на ряду з класичним сигнатурним підходом. Це підтверджує доцільність пошуку нових та удосконалення існуючих методів виявлення загроз.

1.3 Аналіз методів для виявлення та прогнозування кіберзагроз

У сфері кіберзахисту існують різні підходи до виявлення загроз, що можуть бути застосовані до різних типів загроз та мають свої переваги і недоліки. Серед найпопулярніших підходів можна виділити сигнатурний підхід, підхід з використанням статистичних моделей, підхід з використанням машинного навчання, підхід з використанням експертних систем. В той час як сигнатурний підхід є найпоширенішим, він є й найбільш простим у реалізації та заснований на порівнянні хеш-сум існуючих відомих загроз для визначення типів нових загроз. Окремо також можна виявити підходи з застосуванням Теорії Ігор, адже вони дозволяють моделювати взаємодію двох антагоністичних сторін: спеціалісту з кіберзахисту та кіберзлочинця, або визначати оптимальні стратегії окремо кожної зі сторін шляхом моделювання ігор з природою. Підхід з використанням Теорії Ігор може також бути використаний для прогнозування ймовірностей настання загрози, базуючись на оптимальних стратегіях кіберзлочинця. Для

прогнозування кіберзагроз можуть використовуватись засоби експертних систем та моделі машинного навчання.

Більш детально в даному розділі будуть розглянуті підходи на основі експертних систем, з використанням статистичних методів, з застосуванням моделей машинного навчання, а також з використанням Теорії Ігор.

1.3.1 Виявлення та прогнозування кіберзагроз засобами експертних систем

Для розуміння аспектів застосування засобів експертних систем до виявлення кіберзагроз необхідно надати визначення поняттю експертна система. Експертна система – це програмний засіб, що використовує знання експертів у певній предметній області для імітації процесу міркування людини-експерта. Архітектура експертної системи включає в себе базу знань експертів, рушій генерації висновків, модуль пояснень та користувацький інтерфейс, що надає набір рекомендацій, згенерованих рушієм висновків [37, 38] .

Експертні системи поділяються на 2 основних типи: case base reasoning та rule based reasoning. Для Case base reasoning експертних систем база знань формується на основі попереднього досвіду, отриманого системою та може адаптуватись до нових вхідних даних. Для створення рушію висновків CBR систем можуть застосовуватись різні підходи, зокрема: нечітка логіка, нейронні мережі та машинне навчання, генетичні алгоритми, навчання з підкріпленням, статистичні та графові моделі. Rule based reasoning системи являють собою клас систем, процес генерації висновків яких засновується на використанні заданого набору правил у вигляді if-then визначень. Наприклад, у своїй роботі Wirkuttis, N. та ін. [39] розглянули застосування експертних систем для формування реакції та відповіді на кіберзагрозу. Були проаналізовані 2 типи експертних систем: Case-Based Reasoning та Rule-Based Reasoning. В результаті дослідники дійшли висновку, що CBR системи здатні вивчати нові правила та модифікувати існуючі,

що, на відміну від RBR систем, дає їм змогу адаптуватись під динамічно змінювані середовища, якими є комп'ютерні мережі. Iqbal H. Sarker та ін у своїй роботі дослідили аспекти моделювання експертних систем для кіберзахисту на основі штучного інтелекту. У ході дослідження автори характеризують експертні системи для вирішення проблем кібербезпеки як такі, що можуть бути засновані на знаннях або правилах, та рішення в яких приймаються на основі вказівок щодо кібербезпеки [40]. Автори зазначили, що правила для експертних систем можуть визначатись як експертом людиною, так і з використанням інших технік, зокрема методів класифікації у машинному навчанні.

CBR системи на основі апарату нечіткої логіки використовують нечіткі терміни для оцінки та визначення рішення вхідного запиту до системи. Для порівняння вхідного запиту з існуючими відомими для системи запитами, використовується нечітка функція приналежності для обчислення значень нечітких атрибутів та оцінка схожості між двома випадками на основі значення функції приналежності [41]. Наприклад, Churu, Matida та ін. у своєму дослідженні запропонували експертну систему для аналізу кіберзагроз, що базується на системі нечіткого виводу. Вхідними даними системи стали метрики PC to Server ping time, PC-to-PC ping time, та Download time. Для тестування системи, автори розгорнули локальну мережу та симулювали кібератаки з використанням Kali Linux. В результаті автори дійшли висновку, що зі зростанням PC-to-server time, PC-to-PC time та download time збільшуються й ризики кіберзагрози для досліджуваної мережі [42].

CBR системи можуть інтегрувати нейронні мережі та моделі машинного навчання для дослідження схожості вхідного запиту з існуючою базою знань. У якості прикладу експертних систем з використанням нейронних мереж можна навести модель DeNNeS, представлену у роботі Samaneh Mahdavifar та Ali A. Ghorbani [43]. Автори запропонували архітектуру, що базується на отриманні правил з натренованої Deep Neural Network моделі для заміни бази знань

експертної системи. Автори перевірили створену модель на датасетах UCI phishing websites та датасеті з шкідливим ПЗ для Android, отримавши 97.5% точності та false positive rate 1.8%.

Також CBR експертні системи можуть використовувати статистичні моделі та показники для імплементації рушію висновків. Наприклад, у дослідженні Lakhno, V. та ін. автори запропонували модель адаптивної експертної системи, що базується на обчисленні інформаційного критерію на основі ентропії та критерію Kullback–Leibler для кластеризації атрибутів розпізнавання в комп'ютерних системах, генеруючи вхідну матрицю нечітких правил. Змодельована система показала точність від 76.5% до 99.1% [44].

Дослідники Fred L. Collopy та ін. у своїй роботі [141] стверджують, що засоби експертних системи доцільно використовувати для прогнозування у деяких випадках, зокрема, коли існує необхідність в багаторазовому прогнозуванні певних подій. Наприклад, розглядаючи загрози для комп'ютерних мереж, ми можемо стикатись з різними чинниками, які впливають на критичність загрози і які необхідно оцінювати експерту для кожної окремої загрози. У якості підходу до формування оцінки може виступати стандарт CVSS, який містить в собі декілька різних груп та метрик, в залежності від типу загрози. При кожному випадку виявлення загроз, можна зібрати необхідну інформацію про загрозу та, використовуючи рушій висновків експертної системи, можна отримувати прогнозування ймовірності початкової або повторної експлуатації певних загроз зловмисниками.

Проаналізувавши існуючі дослідження, можемо дійти до висновку, що використання засобів експертних систем для виявлення загроз є проблемою, що активно досліджується. Процес формування набору правил експертної системи для проведення процесу виявлення кіберзагроз може бути заснований як на правилах, заданих експертами, так і з використанням методів машинного навчання та статистичних методів на основі бази знань про відомі загрози. Таким

чином, для нашого дисертаційного дослідження, актуальним науковим завданням визначимо вдосконалення існуючих методів виявлення загроз, як з використанням експертних оцінок людиною-фахівцем, так і з використанням статистичних методів та методів машинного навчання.

1.3.2 Виявлення кіберзагроз з використанням статистичних методів

Підхід до виявлення кіберзагроз з використанням статистичних методів ґрунтується на математичних моделях, що здатні виявляти відхилення від нормальної поведінки. Дані моделі засновані на висуванні гіпотези про певний тип статистичного розподілу даних, що являють собою часові ряди спостережень певних параметрів системи.

Одним з базових підходів до виявлення аномалій є z-score аналіз. Даний метод ґрунтується на обчисленні різниці між значенням окремої точки спостереження та середнім вибірки, що поділена на стандартне відхилення вибірки:

$$z_{score} = \frac{x - \mu}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_I - \mu)^2}}, \tag{1.1}$$

де x – елемент вибірки; μ - середнє значення вибірки; N – кількість елементів вибірки.

У випадку якщо $z_{score} \notin (-1,1)$, точка вважається аномальною. Для використання показнику z-score, необхідно виконання принаймні 2 умов до набору даних: статистичний розподіл даних має бути нормальним, а також процес має бути стаціонарним. Якщо процес має виражені тренди, сезонність, то застосування z-score показника не є доцільним. В той же час, z-score показник може бути використаний для нормалізації вхідних даних для подальшого дослідження іншими моделями. Часто таку нормалізацію використовують для обробки вхідних даних для виявлення DoS та DDoS атак. Наприклад, Kaliyaperumal Prabu та ін. обчислили показник z-score для вхідного набору даних

мережевого трафіку для подальшої його кластеризації з використанням алгоритму DBSCAN [45].

Іншим методом статистичного аналізу для згладжування статистичного процесу та виявлення аномалій в ньому є застосування рухомого середнього та експоненційного згладжування в поєднанні з пороговим коефіцієнтом. Ідея полягає у обчисленні згладженого тренду на основі минулих спостережень для певного вікна, тобто кількості спостережень, та визначення відхилень від даного тренду для виявлення аномальних спостережень. Просте рухоме середнє обчислюється як середнє значення k останніх спостережень. Для зміни ваги певних спостережень в обчисленні рухомого середнього використовують вагові коефіцієнти та обчислюють зважене рухоме середнє:

$$WMA_t = \frac{\sum_{i=t-N+1}^{t} w_i X_i}{\sum_{i=t-N+1}^{t} w_i}, \tag{1.2}$$

де t – час спостереження для якого обчислюється значення WMA; w_i – ваговий коефіцієнт, що застосовується до спостереження X_i; N – розмір вікна спостережень. WMA показник дозволяє, наприклад, надавати більшої ваги найбільш новим спостереженням при визначенні аномалій процесу, що спостерігається. Недоліком такого підходу є необхідність вручну задавати вагові коефіцієнти спостережень. Для усунення даного недоліку застосовується EMA або EWMA рухомі середні, які регулюють міру впливу спостережень за допомогою фактору згладжування α, таким чином присвоюючи вагові коефіцієнти, що зменшуються експоненційно:

$$EMA_t = \alpha X_t + (1-\alpha) EMA_{t-1}, \tag{1.3}$$

де α - фактор згладжування; t – час спостереження, для якого обчислюється EMA; EMA_{t-1} – попередньо обчислене значення показника. Фактор згладжування обчислюється за формулою 1.4.

$$\alpha = \frac{2}{N+1}, \tag{1.4}$$

де N – кількість періодів спостереження. Показник EMA реагує краще на зміни в процесі спостереження, адже присвоює більші вагові коефіцієнти новим спостереженням. Показник EWMA обчислюється так само, проте фактор згладжування α може бути визначений довільно, в залежності від контексту застосування [46]. Для виявлення аномалій методами експоненційного згладжування встановлюється пороговий коефіцієнт, який є мірою перевищень згладжених значень. Якщо згладжене значення перевищує пороговий коефіцієнт, це вказує на наявність аномалії часових спостережень. Зокрема, застосування такого методу може допомагати у процесі прийняття рішень базуючись на динамічно визначених правилах щодо аномальності процесів в комп'ютерних мережах.

Іншим популярним статистичним методом для виявлення аномалій є Seasonal-Trend Decomposition using Loess (STL). Даний підхід виділяє сезонні, трендові та залишкові складові статистичного процесу, тим самим дозволяє уникнути хибних виявлень, що пов'язані зі зміною вище перелічених компонент. Підхід STL працює зі нестаціонарними часовими рядами. Після декомпозиції часового ряду, застосовується певний мінімальний або максимальний поріг для залишкової складової, при перетині якого спостереження вважається аномальним. Даний підхід доцільно використовувати для мережевого трафіку, в якому прослідковується сезонність або тренд. Сезонність мережевого трафіку може бути пов'язана з робочими годинами або днями, а тренд з масштабуванням комп'ютерної мережі, що призводить до збільшення обсягів мережевого трафіку. Основним підходом даного методу є необхідність встановлення чіткого порогу, при перетині якого спостереження вважається аномальним. Це може негативно вплинути на точність виявлення аномалій прі зміні конфігурації мережі або обсягів мережевого трафіку.

Проаналізувавши існуючі статистичні методи для виявлення загроз, можемо зробити висновок, що дані методи можуть бути застосовані для прийняття рішення щодо аномальності процесів у корпоративних комп’ютерних мережах, базуючись на спостереженнях мережевого трафіку.

1.3.3 Виявлення кіберзагроз методами машинного навчання

Застосування методів машинного навчання до виявлення кіберзагроз є добре дослідженою проблемою та має різні сфери використання. Основними напрямками застосування методів машинного навчання є виявлення вторгнень, виявлення шкідливого ПЗ, аналіз поведінки користувача системи, фільтрація spam повідомлень та виявлення DDoS атак. Для кожного з даних напрямків можуть бути застосовані методи керованого та некерованого машинного навчання, машинного навчання з та без підкріплення, методи глибинного навчання, методи кластеризації та методи виявлення аномалій. Загалом всі перелічені методи можна об’єднати за метою, а саме – виявлення та прогнозування кіберзагроз для кінцевих пристроїв або комп’ютерних мереж.

Проведемо порівняльний аналіз застосування методів машинного навчання для створення або удосконалення моделей та методів виявлення та прогнозування кіберзагроз. Виділимо 2 основних варіанти використання методів машинного навчання для виявлення та прогнозування кіберзагроз – виявлення кіберзагроз для кінцевих точок мережі та для мережі в цілому.

Для аналізу використані дослідження різних авторів у сфері захисту комп’ютерних мереж та окремих кінцевих пристроїв, що функціонують на основі операційних системи Windows, Linux та Android. Серед порівняльних характеристик виділено методи, предмет дослідження та набір даних, які використовували автори.

Таблиця 1.3 – Порівняльний аналіз досліджень з виявлення кіберзагроз з використанням методів машинного навчання

Завдання	Використані методи	Автори	Предмет дослідження	Набір даних
1	2	3	4	5
Виявлення кіберзагроз для кінцевих точок мережі	NLP, LogEvent2Vec	Ryciak, Piotr & Wasielewska, Katarzyna & Janicki, Artur [47]	Метод виявлення аномалій в системних логах	лог файли з BlueGene/L (BGL) суперкомп'ютера
	Node2vec, LSTM	Alaca, Yusuf & Celik, Yuksel & Id, Sanjay [48]	Метод виявлення аномалій в системних логах	BGL лог-файли, HDFS лог-файли
	Random Forest , Neural Networks, Decision Tree, KNN, Naïve Bayes, and Support Vector Machine	Alhaidari, Fahd та ін. [49]	Метод виявлення шкідливих Windows PE файлів	VirusShare Windows PE файли
	Logistic Regression, SVM, KNN, word2vec	Parildi, E.S. та ін. [50]		Windows PE опкоди
	Extra Tree, XGBoost, та Stacking models	Yuk, Chang & Seo, Chang. [51]		Метаінформація про заголовки Windows PE файлів
	Нейромережа глибинного навчання	Lad, Sumit & Adamuthe, Amol [52]		Набір даних EMBER 2018
	k-nearest neighbor algorithm (IBk weka)	Koçak, Aynur et al. [53] (IBk algorithm)		Мережеві пакети активності Windows PE файлів
	n-gram, TF-IDF, Logistic regression, SVM, Random Forest	Poudyal, Subash та ін. [54]		NLP аналіз апі викликів Windows DLL файлів
	Support Vector Machine, n-gram	Lin, C.-T та ін. [55]		NLP аналіз логів Windows PE файлів
	LSTM, word2vec	Percílio Azevedo, B.W. та ін.[56]		NLP аналіз Windows PE заголовків
	DNN	N. Visweswaran та ін. [57]		One-hot кодований набір імпортованих функцій Windows PE файлів

Продовження таблиці 1.3

1	2	3	4	5
	SVM, XGBoost, Linear regression	Balram, Neil та ін. [58]		Windows PE strings
	Logistic Regression, Decision Tree, SVM, TF-IDF	Qin B. та ін. [59]		API виклики Windows файлів
	SVM, Random Forest, Naïve Bayes	Kang, B. та ін. [60]	Метод виявлення шкідливих Android файлів	Android APK опкоди
Виявлення кіберзагроз для кінцевих точок мережі	J48, Random Forest, Naive Bayes	A. Ravi та V. Chaturvedi [61]	Метод виявлення шкідливих ELF файлів	Метаінформація про Linux ELF файли
	SVM, KNN, Naïve Bayes, MLP, n-gram	Wan, Tzu-Ling та ін. [62]		NLP оброблені опкоди з точки входу Linux ELF файлів
Виявлення та прогнозування кіберзагроз для мережі в цілому	Naïve Bayes, Decision Tree, Random forest	Bierbrauer, David та ін. [63]	Виявлення загроз в мережевому трафіку	Набір даних UNSW-NB15
	SVM, Random Forest	ANTON, Simon Duque та ін. [64]		Набір TCP пакетів мережі котроллерів SCADA системи
	LSTM, CNN, SVM, KNN	Patil Viay та ін. [65]	Виявлення DDoS атак	CICDDoS2019, CICIDS2017, KDDCUP
	LSTM	Almahmoud, Z. та ін. [66]	Передбачення загроз для мережі	Сайт Hackmageddon та інші джерела

В нашому дослідженні доцільно буде розглянути як можливість виявлення загроз для кінцевих точок мережі, так і для мережі в цілому, оскільки такий підхід дозволяє більше точно виявляти та оцінювати наявні кіберзагрози.

1.3.4 Застосування Теорії Ігор для виявлення загроз

Застосування Теорії Ігор для вирішення проблем кібербезпеки дозволяє досліджувати стратегії кіберзлочинця та спеціаліста з кіберзахисту та приймати

рішення щодо реагування на ту чи іншу кіберзагрозу. Також, засновуючись на вирішенні матричних ігор, даний підхід дозволяє виконувати прогнозування ймовірностей оптимальних стратегій експлуатування вразливостей системи зловмисником. Основними типами ігор, що розглядаються у розрізі проблем кібербезпеки є:

1. Кооперативні ігри. Можуть бути використані для моделювання процесу взаємодії різних підрозділів однієї організації для обміну інформацією про події кібербезпеки.
2. Не кооперативні ігри. Використовуються для моделювання застосування механізмів кіберзахисту проти кібератак.
3. Ігри з нульовою сумою. Використовуються для моделювання змагальних взаємодій між акторами кібератаки та кіберзахисту.
4. Ігри з ненульовою сумою. Можуть бути використані для моделювання ліквідації наслідків кібератак різними підрозділами організації.
5. Статичні ігри. Можуть застосовуватись для статичного задання стратегій захисту, враховуючи можливі стратегії кібератак.
6. Динамічні ігри. Можуть застосовуватись для моделювання прогресу вторгнення та поетапного застосування стратегій кіберзахисту.
7. Баєсові ігри, або ігри з неповною інформацією. Можуть бути застосовані для визначення пріоритету стратегій кіберзахисту.

Розглянемо існуючі дослідження з застосування Теорії Ігор у сфері кібербезпеки. У своїй оглядовій роботі Cuong, Do та інші навели та дослідили широкий спектр досліджень, пов’язаних із застосуванням теорії ігор для вирішення проблем кібербезпеки. Вивчені роботи були згруповані за ігровою моделлю та проблемою, яка повинна бути вирішена за допомогою теорії ігор. Зокрема, були представлені роботи, в яких за допомогою моделей теорії ігор досліджуються захист приватності, захист кіберфізичних систем, захист від DoS та DDoS атак та інші [67, 68, 69] .

Sayed M.A. та ін. у своїй роботі використовують гру з двома гравцями з нульовою сумою для вирішення питання розміщення “honeypot” пасток в мережі, таким чином, щоб захистити найбільш важливі активи. Автори приділяють увагу атакам нульового дня, тому що такі атаки здатні оминути існуючі пастки. В результаті автори розробили та підтвердили ефективність застосування теоретико-ігрового підходу проти атак нульового дня [70].

M. Major та ін. у своїй роботі застосовують теорії ігор для дослідження кібератак, в яких кібернападник використовує обманні стратегії, щоб відволікти сторону кіберзахисту та, одночасно, спеціаліст з кіберзахисту використовує приманки для приховання та захисту реальних хостів. Автори запропонували модель з множинними деревами ігор для визначення та оцінки стратегій кожного з гравців використовуючи їх неприховані знання про структуру гри та платежі [71].

Іншим прикладом застосування теорії ігор до проблем кібербезпеки є робота Ullah F. та ін., в якій дослідники запропонували покращену IDS для виявлення кібератак на IoT пристрої та перевірили її за допомогою теоретико-ігрового підходу. Автори розробили фреймворк, згідно з яким створюється гра, гравцями якої виступають кіберзлочинець та кіберзахисник. У випадку знаходження рівноваги Неша, вважається що досягнуто стійкість і ефективність різних методів виявлення проти різних типів атак [72].

Gill K.S. та ін. запропонували застосувати теорію ігор до оцінки стратегій кібернападника та кіберзахисника для хмарних середовищ. У своїй роботі автори обчислюють рівновагу Неша з використанням графічного способу. У результаті автори досягли зростання detection rate та зниження false positive rate [73].

Враховуючи проаналізовані дослідження, можемо дійти висновку, що теорія ігор може бути застосована для вирішення питань кібербезпеки, зокрема для оцінки стратегій зловмисника та для прогнозування ймовірності експлуатації вразливостей системи. Також вирішення матричної гри може слугувати для

оцінки існуючих вразливостей та визначення протидії до них, як частина моделі виявлення загроз, представленої в пункті 1.1.1 розділу 1. В нашому дослідженні буде запропоновано використання теорії ігор у якості рушію висновків експертної системи з оцінки та прогнозування ймовірностей експлуатації існуючих загроз та для формування звітів з протидії існуючим загрозам.

1.4 Постановка задачі та логічна структура роботи

Для виявлення та прогнозування кіберзагрози в дисертаційній роботі будемо враховувати сутності та зв'язки, представлені у розробленій моделі кіберзагрози, представленої на рисунку 1.1. Також, скористаємось визначеними політиками та процедурами безпеки, заснованими на моделі CIA Triad для мережевих загроз [8], для побудови власної моделі виявлення та прогнозування загроз для корпоративних комп'ютерних мереж. Перш за все, необхідно визначити активи корпоративної комп'ютерної мережі, що можуть стати ціллю зловмисника. Для корпоративних комп'ютерних мереж такими активами можуть бути сайти, сервіси, бази даних та кінцеві пристрої. По-друге, треба запропонувати та реалізувати методи для виявлення загроз, базуючись на визначених активах та переліку загроз, що є найбільш популярними для корпоративних комп'ютерних мереж. Зокрема, виділимо 2 основні загрози, що були визначені у першому розділі як найбільш розповсюджені для корпоративних комп'ютерних мереж: шкідливе ПЗ та DDoS атаки.

У процесі виявлення загрози та отримання інформації про неї, необхідно оцінити загрозу, з урахуванням її типу, цілі та критичності. Отримавши інформацію про виявлену загрозу, експерти можуть приступити до процесу оцінки загрози. Використовуючи інформацію про виявлений тип та ціль вразливості, експерти визначають значення CVSS загрози – загальної системи оцінки вразливостей. CVSS є відкритим стандартом, запропонованим у індустрії кібербезпеки, що створений для оцінки критичності вразливостей комп'ютерної системи [9]. Експерти задають можливі стратегії протидії загрозі, базуючись на

власному досвіді та інформації про виявлену загрозу. Далі рушій висновків експертної системи, на основі інформації про виявлені та оцінені експертами загрози та методи протидії ним, виконує прогнозування ймовірності нової або повторної експлуатації зловмисником певних загроз і визначає пріоритетність застосування методів протидій загрозам. Запропонована модель процесу виявлення та прогнозування загроз представлена на рисунку 1.2.

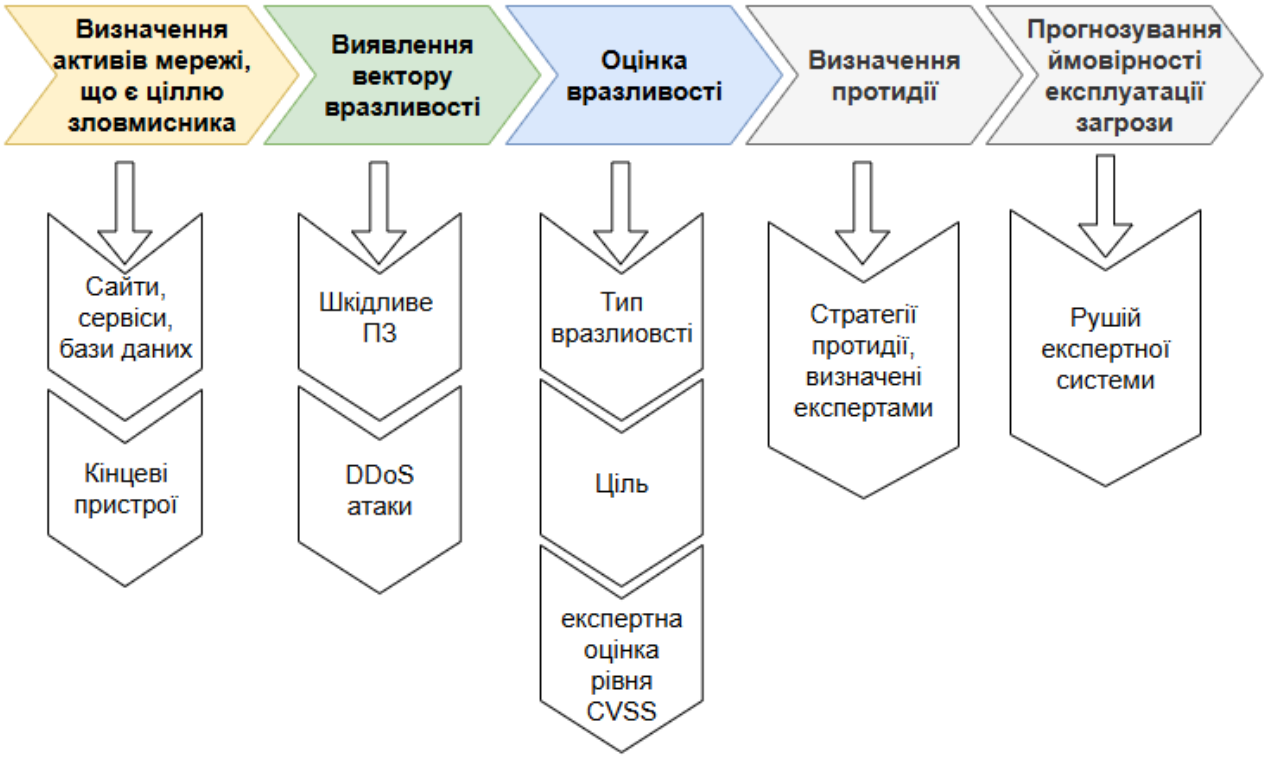


Рисунок 1.2 – Модель процесу виявлення та прогнозування кіберзагроз для корпоративної комп’ютерної мережі

Для реалізації запропонованої моделі, необхідно сформувати набір методів для виявлення та класифікації загроз та запропонувати модель комплексної інформаційної технології виявлення та прогнозування загроз.

Інформаційна система виявлення та прогнозування загроз для корпоративних комп’ютерних мереж засобами експертних систем є частиною загальної системи управління кібербезпекою організації.

З метою побудови інформаційної системи виявлення та прогнозування загроз засобами експертних систем необхідно вирішити наступні задачі:

1) Розробити методи виявлення загроз для корпоративних комп’ютерних мереж з використанням статистичних моделей та методів машинного навчання.
2) Розробити модель інформаційної системи для формування рекомендацій з оптимальних стратегій кіберзахисту та прогнозування ймовірностей експлуатації загроз, яка включає в себе базу знань, рушій висновків та інтерфейс виводу експертної системи.
3) Розробити рушій висновків експертної системи, що функціонує на основі методу експертних оцінок та з застосуванням моделі Теорії Ігор.
4) Розробити функціональну модель інформаційної системи виявлення та прогнозування загроз для корпоративної комп’ютерної мережі засобами експертної системи.
5) Провести експериментальні випробування окремих методів інформаційної системи в штучному середовищі, що імітує корпоративну комп’ютерну мережу.

Висновки до розділу 1

В результаті проведеного аналізу загроз для корпоративної мережі та методів їх виявлення та прогнозування можна зробити такі висновки:

1) В результаті аналізу різного роду кіберінцидентів зроблено висновки, що корпоративні комп’ютерні мережі часто стають ціллю атак, які завдають великих збитків та шкоди організаціям-жертвам. Оскільки велика кількість загроз є новими, для більш ефективного їх виявлення доцільно використовувати технології, засновані на статистичних моделях та методах машинного навчання.

2) В результаті аналізу існуючих систем типів IDS, IPS, SIEM та EDR, дійшли висновку, що гібридні підходи до виявлення загроз з використанням моделей машинного навчання набувають все більшого розвитку. Також, важливу роль відіграє не тільки виявлення та нейтралізація загроз, а й залучення людини-експерта з наданням безпекових звітів для постійного моніторингу та контролю за роботою даних систем. Тому дослідження та створення нових підходів з застосуванням засобів експертних систем для формування звітів та рекомендацій з застосування стратегій захисту є актуальною проблемою.
3) Використання моделей машинного навчання та статистичних методів для виявлення загроз є проблемою, дослідження якої активно розвивається. Удосконалення таких методів є актуальною проблемою для дослідження, адже це дозволяє підвищити точність виявлення та класифікації різних типів загроз.
4) Використання експертних систем, як окремої сутності для вирішення задач забезпечення кібербезпеки є досить детально дослідженим питанням, в той час, як створення інформаційної технології, яка б містила в собі компоненти виявлення та прогнозування кіберзагроз для наповнення бази знань експертних систем, є мало дослідженою проблемою. Моделювання такої інформаційної системи може забезпечити підвищення оперативності виявлення загроз та підтримку прийняття рішень щодо реагування на наявні та можливі вразливості корпоративних комп’ютерних мереж.

Результати досліджень, приведених в розділі, опубліковані в роботах [74,75].

РОЗДІЛ 2
МЕТОДИ ТА МОДЕЛЬ ВИЯВЛЕННЯ ТА ПРОГНОЗУВАННЯ ЗАГРОЗ ДЛЯ КОРПОРАТИВНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

2.1 Метод виявлення DDoS атак з використанням Isolation Forest та EWMA статистики

Аналіз часових рядів мережевого трафіку дозволяє виявляти аномалії та асоціювати їх з відповідними кіберзагрозами для мережі. Такий аналіз допомагає у виявленні атак, що впливають на кількісні показники мережевого трафіку – наприклад DoS та DDoS атаки.

Аномалії часових рядів можна поділити на аномалії окремих спостережень, колективні аномалії, аномалії контексту. Колективні аномалії являють собою випадок, коли спостереження видається нормальним в ізольованому контексті, але ідентифікується як аномалія при отриманні більш ширшого набору даних. Аномалії окремих спостережень дозволяють виявити спостереження, значення яких мають значне відхилення від інших спостережень. Точкові аномалії можуть бути використані для виявлення різких змін обсягу мережевого трафіку, пов'язаного з DDoS атакою, проте вони не враховують контексту спостережень, яким може бути час доби, день тижня або певний хост мережі. Для врахування даних чинників при виявленні DoS та DDoS атак можуть бути використаний метод виявлення аномалій контексту.

В даному дослідженні для виявлення аномалій контексту будемо використовувати алгоритм Isolation Forest. Даний алгоритм працює шляхом розділення даних спостережень з використанням дерева рішень. Аномальні спостереження будуть мати коротший шлях в побудованому дереві, тому що вони відрізняються від інших спостережень. На кожній ітерації алгоритму обирається атрибут q та розділяється на дерева за значенням p. Алгоритм відбувається рекурсивно, поки всі спостереження не стануть ізольованими.

Таким чином, алгоритм IsolationForest враховує контекст спостережень в розрізі значень різних атрибутів та не залежить від типу розподілу даних вибірки. Рівень аномальності обчислюється за формулою:

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}}, \tag{2.1}$$

де *h(x)* – довжина шляху до точки *x* в отриманому дереві; *E(h(x))* – середнє значення довжини шляху до точки *x* в усіх побудованих деревах; *c(m)* – середнє значення довжини шляху до кожної з точок для вибірки розміру m. Основа 2 використовується для нормалізації та масштабування. Якщо рівень аномальності близький до одиниці, то середнє значення довжини шляху до точки є меншим за середнє значення до кожної з точок вибірки, тому точка вважається аномальною. Якщо значення рівня аномальності прямує до нуля, точка вважається нормальною. Алгоритм Isolation Forest використовується для забезпечення якості та виявлення аномалій у сферах обслуговування та виробництві, зокрема дослідниками Wang J. та Liu L. у дослідженні [76].

Застосування алгоритму Isolation Forest для виявлення DDoS атак є дослідженою проблемою. Зокрема MA, Zhaohui та ін. у своєму дослідженні [77] використали композицію алгоритмів Isolation Forest та алгоритму кластеризації K-Means для виявлення DDoS атак. У роботі [78] Rony Chowdhury Ripan та ін. запропонували використовувати алгоритм Isolation Forest для категоризації кібератак в різних датасетах у зв’язці з моделями машинного навчання AdaBoost, Naïve Bayes та K-nearest neighbor. Використання даного алгоритму показало покращення точності отриманих передбачень.

У нашому дослідженні алгоритм Isolation Forest буде використовуватись разом з обчисленням експоненційного зваженого середнього EWMA параметрів мережевого трафіку для виявлення DDoS атак. У своїх дослідженнях І. С. Скітер та ін. [79, 80] пропонують використання EWMA-статистики для виявлення мережевих аномалій та DDoS атак з використанням параметрів вхідного трафіку.

У нашій роботі пропонується використовувати композицію методів Isolation Forest та EWMA-статистики для виявлень DDoS трафіку.

Для виявлення аномальних значень параметрів мережевого трафіку спочатку виконаємо згладжування часових рядів спостережень параметрів мережевого трафіку з використанням EWMA-статистики та знайдемо спостереження, значення яких перевищують поріг в β разів. Таким чином, знаходимо точкові аномалії в часовому ряді спостережень параметрів мережевого трафіку і можемо висунути гіпотезу про наявність DDoS атаки. Проте такі аномалії можуть бути викликані й легітимними діями користувачів корпоративної комп'ютерної мережі, зокрема робота з великими файлами в онлайн форматі, завантаження або пересилання великих обсягів даних тощо, що залежить зокрема й від робочого часу.

Для підтвердження або відкинення гіпотези про наявну DDoS атаку перейдемо до виявлення аномалій в контексті часу їх виникнення з використанням алгоритму IsolationForest. Для цього проведемо масштабування значень спостережень параметру трафіку з використанням *z-score* показнику. Це необхідний крок, адже в алгоритмі Isolation Forest обчислення відстаней до параметрів спостережень відбувається на основі їх значень. Масштабовані значення будуть досліджені в контексті часу спостереження для виявлення контекстних аномалій. З часу спостереження отримуємо годину та хвилину. Далі, з використанням Isolation Forest, серед масштабованих значень параметрів трафіку знаходяться контекстні аномалії. Таким чином, підтверджуємо або спростовуємо гіпотезу про наявність DoS або DDoS атаки в контексті часу її виникнення.

Для виконання EWMA згладжування та виявлення скористаємось формулою 2.2:

$$EWMA_t = \lambda X_t + (1 - \lambda) EWMA_{t-1}, \quad (2.2)$$

де $EWMA_{t-1}$ – середнє зважене значення історичних даних в момент часу *t-1;* X_t – спостереження в момент часу; *t* – кількість спостережень для обчислення; $0 < \lambda \leq 1$ – константа, яка визначає вагу історичних коефіцієнтів EWMA.

Значення константи λ впливає на ступінь врахування часових спостережень параметрів трафіку – чим більше значення параметру, тим більшу вагу мають більш нові спостереження. Значення λ розрахуємо за формулою 2.3:

$$\lambda = \frac{2}{N+1}, \tag{2.3}$$

де, *N* – кількість спостережень параметру.

Для спостереження X_t порівняємо його значення зі значенням середнього зваженого помноженого на «пороговий коефіцієнт» β (формула 2.4).

$$X_t \geq \beta * EWMA_{t-1} \tag{2.4}$$

де X_t – значення параметру мережі; $EWMA_{t-1}$ – середнє зважене значення параметру за методикою EWMA для масиву даних до момент часу *t-1* .

Для виявлення аномалій мережевого трафіку у певний момент часу – а саме DoS та DDoS атак, нам необхідно сформувати часові ряди параметрів трафіку, що проходить через корпоративну мережу. Значення цих параметрів мають бути кількісною мірою трафіку, що проходить за одиницю часу та надавати інформацію як про зовнішню взаємодію з мережею, так і відповіді мережі на здійснені взаємодії. Також, обрані параметри мають бути неперервними величинами та надавати можливість проаналізувати тенденції мережевого трафіку та його аномалії.

Базовим параметром було визначено кількість пакетів, що надійшла до комп'ютерної мережі $X_t = \left| S_t^{pkt} \right|$.

Даний параметр підходить для визначення DoS та DDoS атак. При здійсненні DoS або DDoS атаки також треба дослідити здатність мережі реагувати на вхідні запити. Для цього в якості параметру трафіку використано

значення X_t^{ddos}, що відповідає відношенню кількості вихідних мережевих пакетів відносно кількості вхідних пакетів (формула 2.5).

$$X_t^{ddos} = \frac{\left|S_t^{out}\right|}{\left|S_t^{in}\right|}, \tag{2.5}$$

де $|S_t^{out}|$ – розмір набору вихідних мережевих пакетів за проміжок спостережень *t*; $|S_t^{in}|$ – розмір набору всіх вхідних мережевих пакетів за проміжок спостережень *t*.

Були проведені експерименти зі здійснення послідовних DDoS атак на WordPress сервер, що знаходиться в корпоративній комп'ютерній мережі з використанням інструментів metasploit та hping, що входять до стандартного набору пакетів ОС Kali Linux. В результаті експериментів було визначено, що для параметру X_t підхід виявлення аномалій EWMA виділив на 2 аномалії більше ніж алгоритм IsolationForest. Загалом при здійсненні 2 послідовних DDoS атак, кожна з яких тривала близько 5 хвилин, розробленим методом було виявлено 2 спільні аномалії, що вказують на коректність спрацювання алгоритму та достатню точність виявлення.

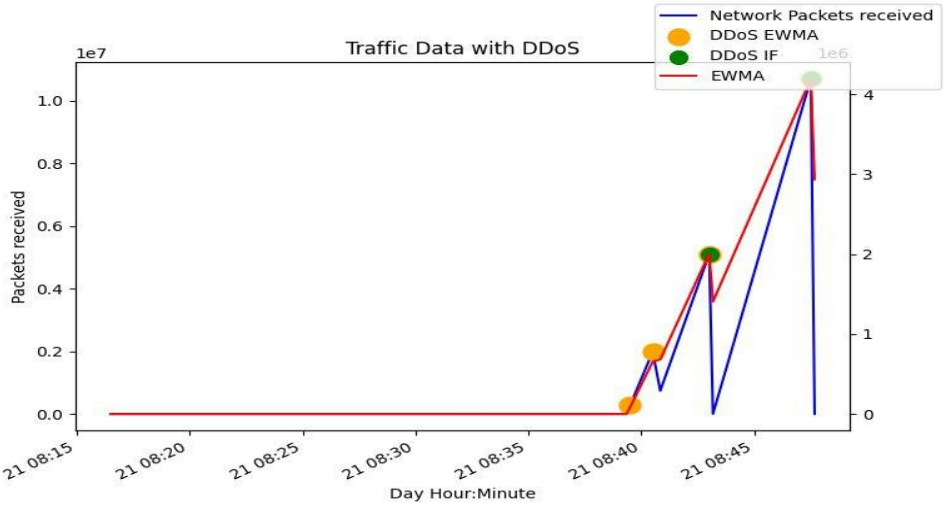


Рисунок 2.1 – Процес виявлення DDoS атак розробленим алгоритмом з використанням EWMA показнику та Isolation Forest для параметру X_t

Для параметру X_t^{ddos} алгоритм виявив 4 аномалії, 3 з яких лежать в часовому проміжку здійснення атаки. Даний показник підтверджує доцільність аналізу параметру X_t^{ddos} для підкріплення гіпотези про здійснення DDoS атаки.

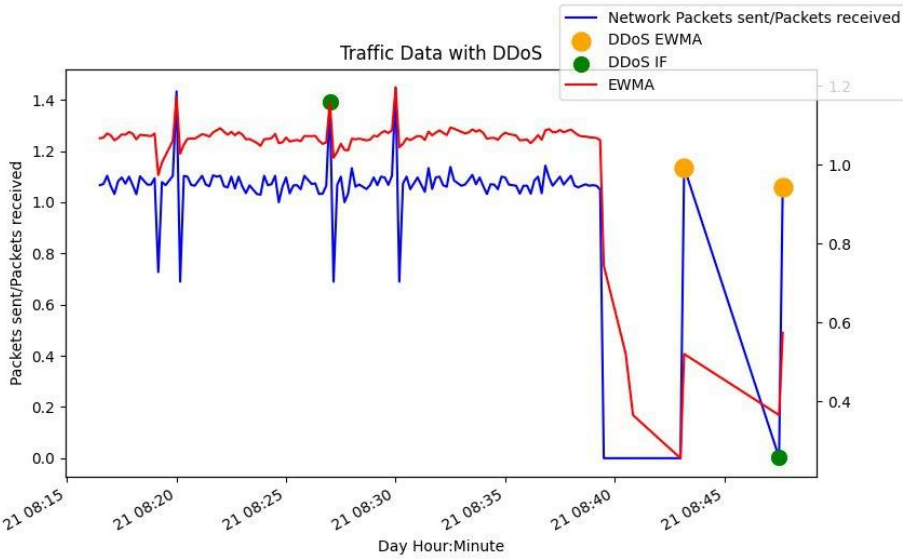


Рисунок 2.2 – Результати виявлення DDoS атаки з використанням EWMA показника та Isolation Forest алгоритму для параметру X_t^{ddos}.

Отже, запропонований метод дозволяє виявляти DoS та DDoS атаки враховуючи аномалії часового контексту параметрів спостереження мережевого трафіку та точкові аномалії, з урахуванням ковзного зваженого середнього трафіку та його перевищень.

2.2 Метод визначення секції Linux ELF файлу для ідентифікації шкідливого ПЗ

Виявлення та дослідження шкідливого програмного забезпечення узагальнено можна поділити на два підходи: статичний аналіз та динамічний аналіз [19]. До статичного аналізу відносяться методи дослідження байт-коду, асемблерних бінарних команд або імпортованих динамічних бібліотек (DLL). Динамічний аналіз передбачає дослідження поведінки шкідливого ПЗ в часі.

Статичний аналіз може бути застосований як до цілого бінарного файлу, так і для окремих секцій.

Наприклад, Ian Shiel та інші у своїй роботі [81] запропонували метод вдосконалення алгоритму нечіткого хешування, шляхом застосування його до кожної із секцій бінарного PE файлу. Своїм дослідженням автори вирішують проблему виявлення шкідливого ПЗ для ОС Windows, в якому, за дизайном, є спільні для всіх файлів секції, розробники можуть змінювати порядок програмних секцій або вставляти додаткові секції для ускладнення його ідентифікації. В результаті автори домоглись на 92% більше істинно позитивних (TP) виявлень на не обфускованих файлах та на 88% більше TP для запакованого шкідливого ПЗ, в порівнянні з нечітким хешуванням цілого файлу.

Добре дослідженим є використання згорткових нейронних мереж (CNN) до бінарних файлів для їх класифікації. Наприклад, Edward Raff та ін. [82] у своєму дослідженні використали цілий бінарний файл у вигляді послідовності байтів, що подавалась на вхід згорткової нейронної мережі для подальшої класифікації. Однією з проблем роботи з необробленим потоком байтів став факт, що байти мають різне значення в залежності від контексту. Завдяки застосуванню до потоку байтів алгоритму вкладання слів, що дозволяє виділити слова зі схожим значенням та контекстом, вдалось вирішити цю проблему. Розроблена архітектура MalConv показала гарні узагальнені результати на великих датасетах даних та може застосовуватись до бінарних файлів, без зосередження на їх внутрішній структурі. Значним недоліком запропонованого рішення є великий час та обчислювальні потужності, необхідні для проведення операцій згортки на дуже довгих даних, таких як вхідний потік байтів бінарного файлу.

Іншим прикладом статистичного аналізу бінарних файлів з використанням CNN є робота Fangtian Zhong та ін. [83], в якій автори запропонували перетворити вхідний бінарний файл в зображення, що оброблюється та класифікується CNN. Отримана модель показала гарні результати та ефективність.

Основною перевагою підходів з використанням згорткових нейронних мереж є відсутність необхідності доменних знань з кібербезпеки та детального дослідження структури файлу. В той же час, такі підходи не враховують особливості побудови бінарних файлів та мають великі ресурсні затрати на тренування та калібрацію моделей машинного навчання.

Іншим підходом до аналізу шкідливого ПЗ, є застосування техніки вкладання слів та семантичного аналізу до текстового або байтового вмісту файлу. Наприклад, BooJoong Kang та ін. у своєму дослідженні [60] застосували метод n-gram для аналізу та обробки кодів операцій APK файлів. Отримані n-грами були використані для моделей машинного навчання. Автори досягнули найкращих показників класифікації, використовуючи n=3 та n=4 n-грами.

Семантичний аналіз застосовують і у поєднанні з динамічним аналізом шкідливого програмного забезпечення, вивчаючи та аналізуючи результати виконання шкідливих файлів. Наприклад, у своєму дослідженні Bin Qin та інші застосували метод TF-IDF до послідовності викликів API в лог-файлах програми та використали цю інформацію як датасет для визначення та класифікації шкідливого та безпечного ПЗ [59] Результати дослідження довели ефективність застосування семантичного аналізу та методу TF-IDF для покращення результатів класифікації та виявлення шкідливого ПЗ.

Проблема класифікації шкідливого програмного забезпечення активно досліджується. Як об’єкт дослідження зазвичай беруться PE файли, які є виконуваними файлами для операційної системи Windows, або APK файли, які є виконуваними файлами для операційної системи Android. Причиною цьому, серед інших, є великий обсяг типів вірусів, розроблених для цих платформ. Файли формату ELF, які є виконуваними файлами для UNIX-подібних ОС, беруться рідше в якості об’єкта для дослідження та класифікації ПЗ. Однак, на даний час, існують досить загрозливі та руйнівні за своєю дією типи шкідливого ПЗ для UNIX-подібних операційних систем, включаючи найпоширеніші

Ransomware, Worm, Trojan та BotNet. Відносно невелика кількість робіт з ідентифікації шкідливих файлів для Linux систем [61, 62], та робіт з дослідження сучасних загроз для UNIX-подібних ОС [84, 85] [86, 87] [143], є причиною для пошуку та удосконалення методів виявлення загроз для UNIX-подібних ОС.

У якості вхідних даних для статичного аналізу файлів, було вирішено брати послідовність асемблерних команд, що знаходяться у різних секціях бінарного ELF файлу. На виході необхідно отримати оброблений текст команд у вигляді числових векторів для подальшої класифікації файлів.В якості методологій для векторизації текстових даних було розглянуто n-gram методику та алгоритм TF-IDF.

Одним з базових підходів до обробки природньої мови (NLP) є метод n-gram [88]. Даний метод полягає у створенні імовірнісної моделі для передбачення наступної фрази на основі статистичних показників. Моделі n-gram передбачають w_i слово базуючись на послідовності $\{w_{i-(n-1)},...,w_{n-1}\}$. Для обчислення ймовірності послідовності $P(w_1 w_2,...,w_n)$ користуються формулою 2.6.

$$P(w_{1:n}) = \prod_{k=1}^{n} P(w_k \mid w_{1:k-1}), \tag{2.6}$$

де $P(w_k \mid w_{1:k-1})$ - умовна ймовірність появи слова w_k в послідовності слів $w_{1:k}$.

Метод n-gram можна застосовувати, як для обробки програмного коду так і для асемблерних команд, тому що він може згрупувати разом семантично важливі асемблерні команди, що представляють певну логіку виконання програми. Однак, застосування методу n-gram не враховує довжину документа – в нашому випадку деасембльованої файлової секції. В довших секціях n-грами можуть зустрічатись частіше, ніж в коротших, в той час, як різне за довжиною наповнення однакових секцій різних файлів може мати спільний контекст. Тому в нашому дослідженні n-gram буде використаний як основа для методу TF-IDF, який враховує довжину документів та частоту слів в них. TF-IDF є статистичним показником, який дозволяє визначити яке слово вживається найчастіше в

конкретному документі, та рідше у всіх інших документах колекції [89]. Показник складається з двох частин: TF (term frequency) – частота слова, що обчислюється відношенням 2.7.

$$TF = \frac{n_i}{\sum_{k=1}^{N} n_k}, \tag{2.7}$$

де n_i - кількість включень даного слова в документ; n_k - кількість включень k - слова в документ; N - загальна кількість слів в документі.

IDF (inverse document frequency) - інвертоване значення частоти, з якою слово зустрічається в усіх документах колекції, обчислюється рівністю 2.8.

$$IDF = \log \frac{|D|}{|d_i \supset t_i|}, \tag{2.8}$$

де $|D|$ - кількість документів в колекції; $|d_i \supset t_i|$ - кількість документів d_i, в яких зустрічається слово t_i .

В результаті добутку двох показників, отримаємо значення TF-IDF (2.9).

$$\text{TF-IDF} = \text{TF} \cdot \text{IDF}, \tag{2.9}$$

де TF - term frequency; IDF – inverted document frequency.

Отже, значення TF-IDF прямо пропорційне кількості вживань обраного слова у обраному документі, та обернено пропорційне кількості документів, в яких міститься обране слово.

В нашій роботі досліджено кількість n-gram від 1 до 5 та обрано таку, яка дає найбільшу точність для базового порогу класифікації. Сформовані n-gram для обраної секції файлу були використані для подальшої векторизації методом TF-IDF.

Для встановлення базового порогу класифікації використано поліноміальний Баєсовий класифікатор. Основною ідеєю даного метода є пошук класу, до якого документ належить з найвищою ймовірністю, яка обчислюється за формулою Баєса (2.10) [90].

$$P(c \mid t_i) = \frac{P(c)P(t_i \mid c)}{P(t_i)},\ c \in C \qquad (2.10)$$

де, C - множина класів документа; t_i - документ колекції.

Висунувши гіпотезу, що слова в документах розподілені за допомогою певної параметричної моделі, можемо визначити ці параметри скориставшись Поліноміальним Баєсовим класифікатором. Ґрунтуючись на даному твердженні, Jiang Su та інші вирішують проблему класифікації тексту у своїй роботі [91]. У якості основних методів класифікації, були обрані моделі опорних векторів, метод градієнтного спуску та метод градієнтного бустингу.

Моделі засновані на методі опорних векторів – support vector machine або SVM, часто використовують для класифікації тексту, тому що вони оптимізовані для виявлення нелінійних шаблонів в багатовимірному просторі особливостей, який являє собою векторизований текст [92]. Точність класифікації SVM залежить від обраної функції ядр. Для порівняння було досліджено 4 функції: лінійну, поліноміальну, радіальну та сигмоїд.

Для покращення результатів тренування обрано два методи: градієнтний спуск та градієнтний бустинг. Метод градієнтного спуску полягає в знаходженні локального мінімуму диференційованої функції [93, 94]. Даний метод використовується для мінімізації функції втрат при тренуванні моделі шляхом поетапного переналаштування параметрів моделі. Для моделі опорних векторів використано стохастичний градієнтний спуск - SGD, який направлений на мінімізацію функції втрат при тренуванні моделі. На відміну від звичайного градієнтного спуску, стохастичний градієнтний спуск на кожній ітерації використовує випадково обрані екземпляри тренувальної вибірки, замість знаходження мінімуму для всіх екземплярів вибірки, що оптимізує його роботу [94]. Метод градієнтного бустингу полягає в використанні композиції моделей [95]. В процесі знаходження локального мінімуму функції втрат, обирається найкраща модель з композиції. В якості методу градієнтного бустингу було

використано XGBoost [96] який використовує композицію моделей дерев прийняття рішень.

У ході дослідження запропоновано метод, що дозволяє ідентифікувати шкідливе та безпечне програмне забезпечення, базуючись на визначенні секції Linux ELF файлу, з використанням семантичного аналізу та вибору моделі класифікації, що дає найбільшу точність та F1-міру. Результатом методу є набір моделей машинного навчання, натренованих на розпізнавання шкідливого та безпечного програмного забезпечення для UNIX-подібних ОС. Схема створеного алгоритму зображена на рисунку 2.3.

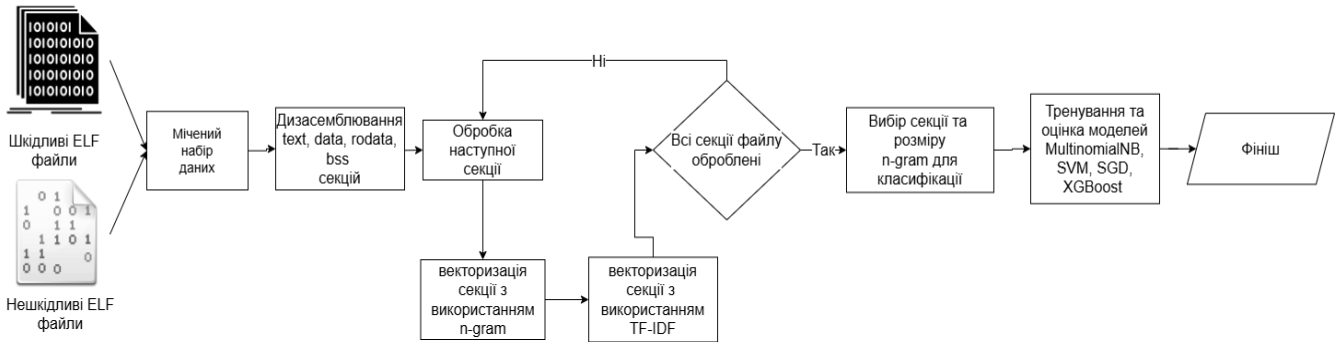


Рисунок 2.3 – Схема методу визначення секції Linux ELF файлу для ідентифікації шкідливого ПЗ

У якості вхідних даних для алгоритму було вирішено використати набір бінарних ELF файлів, що були ідентифіковані як шкідливі та безпечні. Кожний ELF файл складається з двох частин: ELF header та file data. Секція ELF header визначає формат конкретного файлу та завжди має нульовий відступ. Секція file data може містити таблицю заголовків програми та таблицю секцій програми [97].

Для дослідження використовується частина file data з секціями програми – базова частина файлу, в якій знаходяться дані та код файлу. Кожна з секцій, має різне призначення та наповнення. Основні секції ELF файлів та їх призначення наведені нижче:

- .text – містить код програми;
- .data – ініціалізовані дані програми;
- .rodata – містить ініціалізовані дані тільки для читання;
- .bss – містить не ініціалізовані змінні програми, які мають бути обнулені.

Для семантичного аналізу було досліджено кожну з перелічених секцій програмного файлу. Для використання в розробленому алгоритмі обирається секція, яка показала найкращі результати в комбінації з відповідною кількістю n для n-gram при визначенні порогового значення класифікації.

Бінарні ELF файли було завантажено з сайту VirusShare [98]. Для кожного з файлів отримана інформація про наявні загрози, що несе файл. Для цього було використано безкоштовне публічне API сайту VirusTotal [99]. За допомогою MD5 хеш суми кожного з файлів було завантажено та оброблено звіт про файл та наявні в ньому загрози, виявлені різним антивірусним ПЗ. Після дослідження кількості виявлених загроз для кожного з антивірусних ПЗ, обраний антивірус від Microsoft як такий, що має найбільше виявлених загроз, які можна згрупувати у окремі сімейства. Для дослідження були обрані загрози, які є найбільш поширеними. Різні модифікації загроз, які належать до одного сімейства, були згруповані та позначені назвою їх сімейства. Таким чином вдалось отримати набір даних з 5 різних типів вірусів, що є поширеними для платформи Linux.

- Gafgyt - вірус, який заражає операційні системи Linux для запуску на них DDoS атак [84].
- Mirai - вірус, який заражає UNIX-подібні операційні системи, утворюючи з них мережу bot net. Зазвичай запускається на пристроях інтернет речей [86].
- Lightaidra - вірус що проникаючи у UNIX-подібні системи, утворює з них мережу bot net [100].
- Trojan – сімейство вірусів «троянський кінь», що маскується під корисне ПЗ.

– Backdoor – сімейство вірусів, що дозволяють зламувати звичний процес автентифікації до системи.

Для формування набору нешкідливих файлів було вирішено взяти ELF файли з ОС Linux, що знаходяться в директорії /usr/bin, /usr/sbin/, /bin; стандартні утиліти набору coreutils-8.30, низькорівневі утиліти з пакету util-linux 2.33. Таким чином вдалось отримати набір з 3872 нешкідливих ELF файлів.

Після збору даних отримали датасет, що складається зі 6804 бінарних ELF файлів. В датасеті представлені 3872 нешкідливих файлів, 62 загроз типу Mirai, 163 загроз типу Gafgyt, 197 загрози типу Lightaidra, 91 загроз типу Trojan, 49 вірус типу Backdoor та 2370 вірусів, що не належать до жодного з перелічених сімейств. Розподіл загроз у створеному датасеті зображено на рисунку 2.4.

Для проведення класифікації отриманий набір даних було вирішено розділити на 2 класи – клас «вірус» та клас «не вірус». Таким чином, було поставлено завдання бінарної класифікації. Причиною цьому стало те, що тренувальний набір даних є незбалансованим, оскільки різні сімества вірусів представлені в різній кількості.

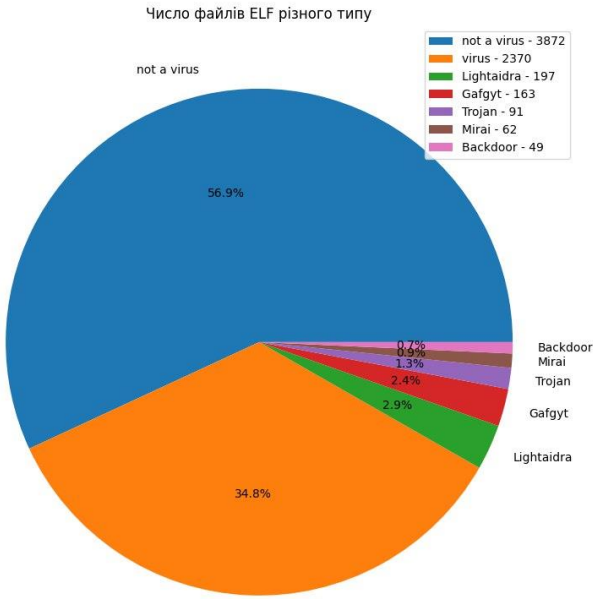


Рисунок 2.4 – Графік розподілу типів ELF файлів в датасеті.

Наступним кроком дослідження стало деасемблювання та посекційне зчитування бінарних файлів. Для секції кожного з файлів послідовно зчитуються асемблерні команди.

Набір зчитаних файлових асемблерних команд наведено на рисунку 2.5.

```
adddwordptr[rax],eax addal,byteptr[rax] addbyteptr[rax],al insddwordptr[rdi],dx
adddwordptr[rax],eax addal,byteptr[rax] jo imulebp,dwordptr[rsi+0x74], jae
cmpdh,byteptr[rip+] addbyteptr[rax],bh cmpbyteptrcs:[rsi],ch cmpbyteptr[rsi],ch cmpbyteptr[rax],al
xordh,byteptr[rbx] addbyteptr[rax],bh cmpbyteptrcs:[rsi],ch cmpbyteptr[rsi],ch cmpbyteptr[rax],al
ax],al addcl,dl subeax,dwordptr[rax] addbyteptr[rax],al xorbyteptr[rdx+0x40],dl addbyteptr[rax],al
```

Рисунок 2.5 – Асемблерні команди деасембльованих програмних секцій.

Отримавши набір асемблерних команд для кожної секції файлів, можна переходити до їх векторизації та тренування моделей машинного навчання. Оброблені вхідні дані були передані на вхід до наступного кроку – утворення n-gram. Отримані n-gram були передані для векторизації за допомогою алгоритму TF-IDF. За допомогою TF-IDF список асемблерних команд з програмних файлів було векторизовано та перетворено на розріджену матрицю розмірністю <*n, n_features*>, де *n* – кількість документів, *n_features* – кількість виділених алгоритмом значущих особливостей. Розріджена матриця, отримана після векторизації TF-IDF моделлю вхідного датасету команд, передається для класифікації. Вхідний датасет розділений на тренувальний та тестовий: 80% на тренування та 20% на тести.

Для оцінки результатів кожного з методів, обчислюється середня точність класифікації та зважене значення F1-міри [101]. Середня точність класифікації обчислюється як пропорція правильно передбачених класів до загальної кількості передбачень. Для обчислення F1-міри, спочатку необхідно обчислити точність передбачень Precision та повноту передбачень Recall (2.11-2.14).

$$\text{Precision}=\frac{TP}{TP+FP}, \tag{2.11}$$

де TP - кількість істинно позитивних передбачень класу; FP - кількість хибно позитивних передбачень класу (формула 2.12).

$$\text{Recall}=\frac{TP}{TP+FN}, \tag{2.12}$$

де TP - кількість істинно позитивних передбачень класу; FN - кількість хибно негативних передбачень класу.

Обчисливши precision та recall, можна обчислити F1 (формула 2.13).

$$F1=2*\frac{\text{Precision*Recall}}{\text{Precision+Recall}}, \tag{2.13}$$

де Precision – точність передбачень; Recall - повнота передбачень.

Зважене значення F1 обчислюється для всіх класів вибірки, враховуючи пропорцію кількості екземплярів кожного класу відносно всіх екземплярів вибірки, що визначається як вага класу (формула 2.14).

$$\text{F1}_{\text{зважене}}=\sum_{i=1}^{N}\text{F1}_i*W_i, \tag{2.14}$$

де N - кількість класів у вибірці; F1_i - F1-міра для класу i; W_i - вага класу i.

Проведено тренування з визначенням базового порогу класифікації для усіх можливих програмних секцій та відповідної кількості n для n-gram. Результати оцінки базового порогу класифікації наведено в таблиці 2.1. Після отримання результатів, було визначено сукупність найвищих значень показників середньої точності та F1, що стало основою для вибору n-gram та програмної секції.

Для покращення результатів, в Баєсовому класифікаторі використано згладжування Лапласа. Даний метод дозволяє уникнути отримання нульових ймовірностей появи слова в тексті шляхом додавання до обчисленої ймовірності параметра α [102]. Використовуючи налаштування гіперпараметрів, встановлено оптимальне значення аддитивного згладжувального параметру α=0,01.

Таблиця 2.1. Оцінка базового порогу класифікації для різних значень n-gram та програмних секцій файлу

Модель	Розмір n-gram	Секція ELF файлу	Середня точність класифікації, %	Значення $F1_{зважене}$	Час класифікації, с
1	2	3	4	5	6
Поліноміальний Баєсовий класифікатор	1	text	91	0.91	39.3
		data	88	0.88	0.19
		rodata	92	0.92	0.49
		bss	66	0.55	0.07
Поліноміальний Баєсовий класифікатор	2	text	90	0.89	69.3
		data	84	0.83	0.25
		rodata	86	0.85	0.67
		bss	58	0.37	0.07
Поліноміальний Баєсовий класифікатор	3	text	91	0.90	91.9
		data	85	0.83	0.28
		rodata	87	0.86	0.78
		bss	58	0.37	0.06
Поліноміальний Баєсовий класифікатор	4	text	91	0.90	103.36
		data	84	0.82	0.35
		rodata	86	0.85	0.77

Продовження таблиці 2.1

1	2	3	4	5	6
Поліноміальний Баєсовий класифікатор	4	Bss	46	0.29	0.03
Поліноміальний Баєсовий класифікатор	5	text	89	0.88	118
		data	83	0.80	0.37
		rodata	85	0.84	0.82
		bss	46	0.29	0.03

Як таку, що має найкращі показники середньої точності та значення $F1_{зважене}$ обрано секцію rodata та кількість n для моделі n-gram рівною 1. Для отриманого класифікатора побудовано криві навчання. Для побудови кривих навчання ініціалізуємо модель на обраних гіперпараметрах, та виконаємо перехресну валідацію на тренувальному наборі даних, розділивши його на тренувальні дані та тестувальні дані. Для перехресної валідації виконано 10 експериментів на різних поділах тренувальних даних в пропорції 80% на навчання та 20% на перевірку моделі. В якості метрик для кривої навчання обрана точність передбачень та $F1_{зважене}$. На рисунку 2.6 приведений графік кривих навчання з використанням метрики $F1_{зважене}$. Отримані криві навчання вказують на зміну точності моделі зі зміною розміру вибірки та демонструють збіжність зі збільшенням розміру вибірки. Криві зближаються зі збільшенням розміру тренувальної вибірки. Це може свідчити про досягнення балансу між зсувом та дисперсією в результаті тренування моделі. Також це дає підставу стверджувати, що модель не перенавчена на тренувальних даних.

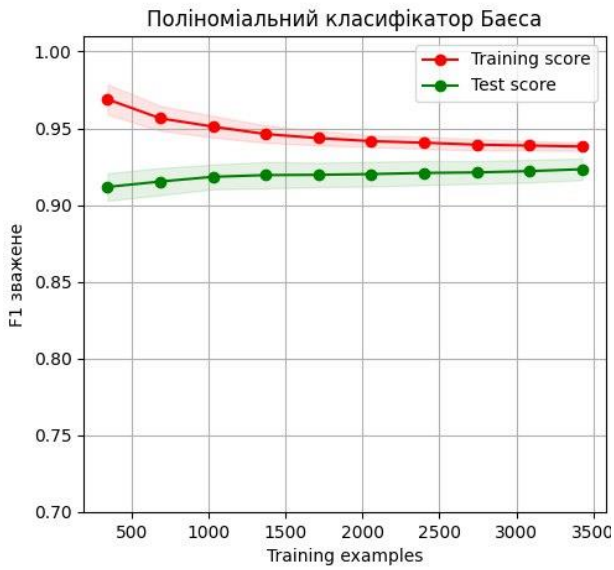


Рисунок 2.6 – Криві навчання для моделі поліноміальний класифікатор Баєса.

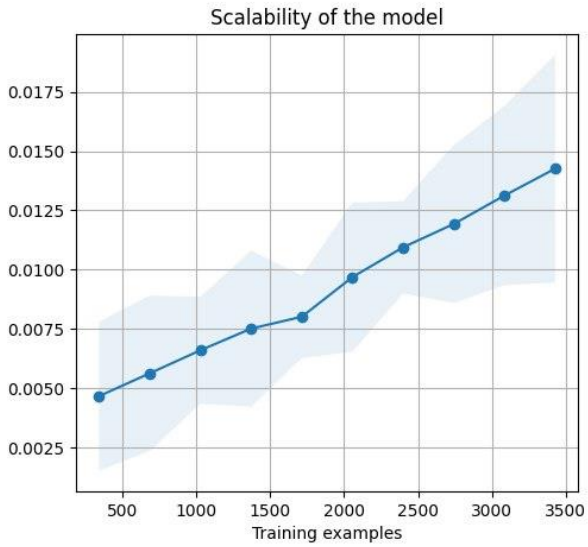


Рисунок 2.7 – Графік залежності середнього часу тренування від розміру вибірки для моделі поліноміальний класифікатор Баєса.

На основі результатів оцінки базового порогу класифікації отримали значення середньої точності 92% та $F1_{зважене}$ 0.92 для розміру n-gram=1 та секції “rodata”. Експериментальне підтвердження статистичної значущості середньої точності та $F1_{зважене}$ отриманих експериментальних результатів з використанням t-тесту Стьюдента для рівня $\alpha=0.01$ приведено в додатку В, в таблиці В.1. Зі збільшенням вибірки можливе подальше зближення кривих навчання та зростання тестової точності моделі. Для оцінки обсягу обчислень, витрачених на тренування моделі, було побудовано графік залежності середнього часу в секундах, затраченого на тренування, від розміру тренувальної вибірки. Отриманий графік для моделі зображено на рисунку 2.7. Зі зростанням розміру вибірки час на тренування лінійно зростає. Для візуальної оцінки виявлених класів бінарних файлів з тестової вибірки, побудовано матрицю невідповідностей, що зображена на рисунку 2.8. Проаналізувавши її, можна побачити, що більш точно модель виявила нешкідливі файли.

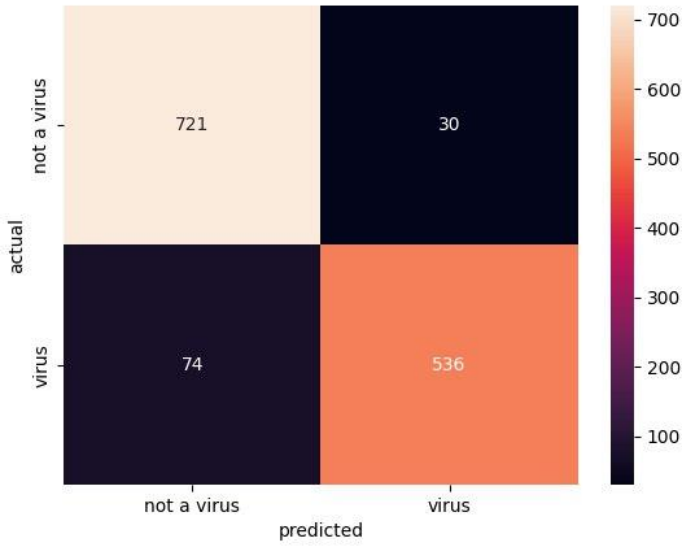


Рисунок 2.8 – Матриця невідповідностей для моделі поліноміальний класифікатор Баєса.

Після отримання базових результатів було вирішено використати алгоритм GridSearchCV для пошуку найкращих гіперпараметрів для моделі опорних векторів. Алгоритм GridSearchCV робить перехресну валідацію моделі на декартовому добутку множин значень гіперпараметрів та обирає найкращу множину значень гіперпараметрів, базуючись на найвищій отриманій точності моделі [103].

Для моделі опорних векторів були проаналізовані гіперпараметри:

- C – міра регуляризації. При низьких значеннях C гіперплощина має великий відступ до точок, при високих значеннях відступ мінімальний. Множина досліджених значень: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
- kernel – функція ядра, що приймає дані та трансформує їх у необхідний вигляд, змінюючи їх вимірність. Використані три функції: лінійна (linear), поліноміальна (poly), радіальний базис (rbf) та сигмоїд (sigmoid).

Графік залежності показника точності перехресної валідації моделі опорних векторів від значень досліджених гіперпараметрів наведено на рисунку 2.9. Проаналізувавши графік визначили, що найвищий показник точності на

тренувальних даних отримуємо для значення параметрів моделі $C = 0,9$ та kernel=rbf .

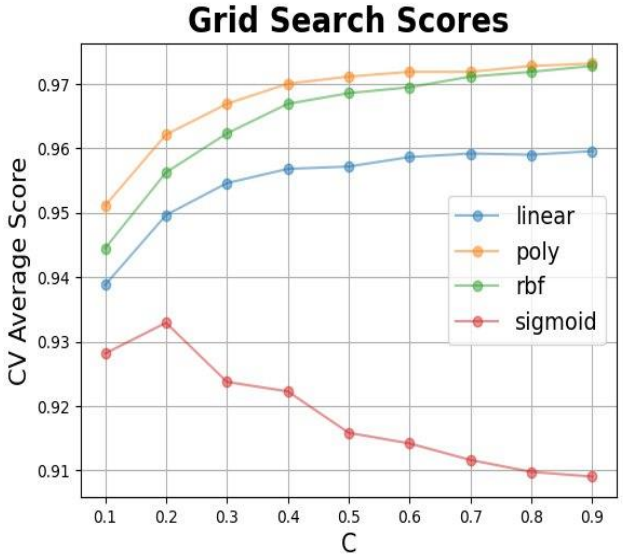


Рисунок 2.9 - Криві показників середньої точності моделі в залежності від значень гіперпараметрів C та kernel для моделі опорних векторів.

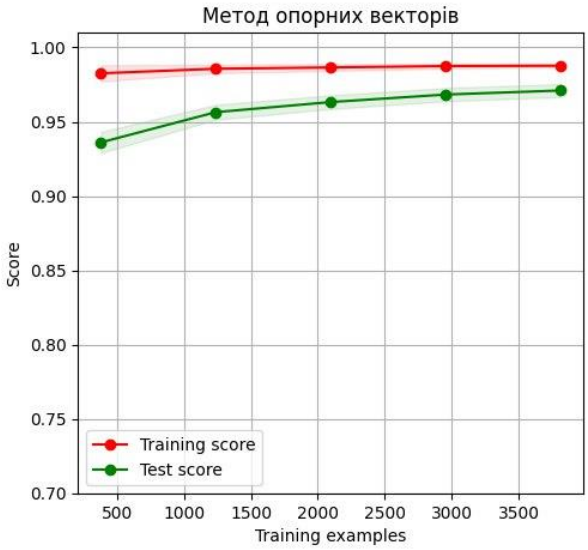


Рисунок 2.10 – Крива навчання моделі опорних векторів.

Для оцінки здатності до узагальнення створеної моделі, побудуємо криву навчання. Отримана крива зображена на рисунку 2.10. З графіку видно, що точність передбачень на тестовій вибірці зростає зі збільшенням її розміру. Це означає, що модель здатна коректно класифікувати дані, які є новими для неї. Точність моделі на тренувальних та тестових даних зближаються зі збільшенням вибірки, проте не збігаються на кінцевому значені. Це може свідчити про те, що зі збільшенням кількості екземплярів в тренувальній вибірці, точність на тестових даних може зростати. Точність тестування та тренування є доволі високою. На тестових даних, натренована модель SVM з обраними гіперпараметрами показала середню точність 97% та $F1_{зважене}$=0.97.

Отримана матриця невідповідностей зображена на рисунку 2.11.

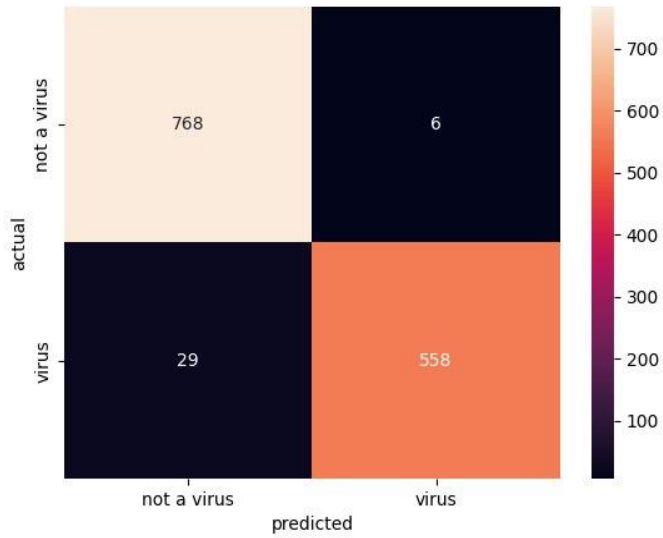


Рисунок 2.11 - Матриця невідповідностей для моделі опорних векторів.

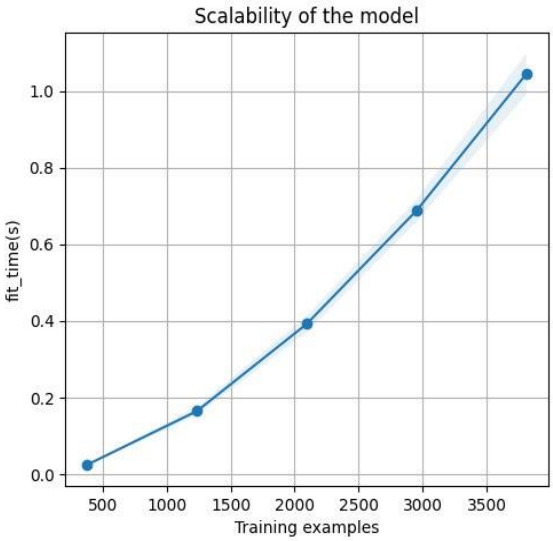


Рисунок 2.12 – Графік залежності середнього часу тренування моделі опорних векторів від розміру тренувальної вибірки

Також було побудовано графік залежності середнього затраченого часу в секундах на тренування моделі від розміру тренувальної вибірки. Даний графік зображено на рисунку 2.12.

Для порівняння результатів передбачень, побудовано модель стохастичного градієнтного спуску на основі моделі опорних векторів. Для моделі стохастичного градієнтного спуску були проаналізовані наступні гіперпараметри:

– eta0 – початкове значення швидкості навчання. Параметр, який використовується для обчислення швидкості навчання.
– learning_rate (lr) – алгоритм обчислення швидкості навчання. Досліджені алгоритми: “constant” обчислюється за формулою 2.15; “optimal” - алгоритм, запропонований Леон Ботту у роботі [104]; “invscale” за формулою 2.16; “adaptive” - алгоритм, за яким швидкість

навчання ділиться на 5, при досягненні значення втрат, яке не міняється 5 ітерацій підряд.

$$lr_{cons} = eta0, \tag{2.15}$$

де $eta0$ - початкове значення швидкості навчання.

$$lr_{inv} = eta0 \ / \ \sqrt{t} \tag{2.16}$$

де $eta0$ - початкове значення швидкості навчання; t - значення елементу тренувальної вибірки.

Графік залежності точності моделі від значень гіперпараметрів зображений на рисунку 2.13. З графіку видно, що найвища точність отримана для $lr_{optimal}$ та $eta0 = 0,01$. Побудовані криві навчання, для моделі з відібраними гіперпараметрами, зображені на рисунку 2.14. З рисунку видно, що криві для тренувальної та тестової точності мають тенденцію до наближення, проте не збігаються. Схожий результат був отриманий і для моделі опорних векторів.

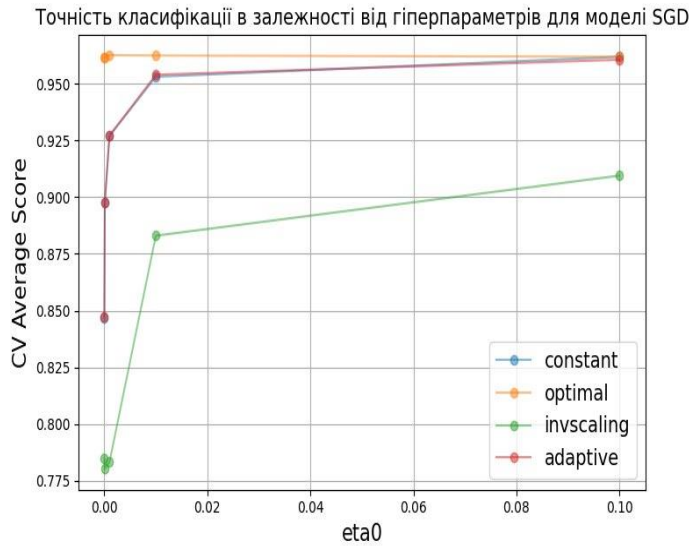


Рисунок 2.13 – Криві показників середньої точності моделі опорних векторів зі стохастичним градієнтним спуском в залежності від значень гіперпараметрів lr та $eta0$.

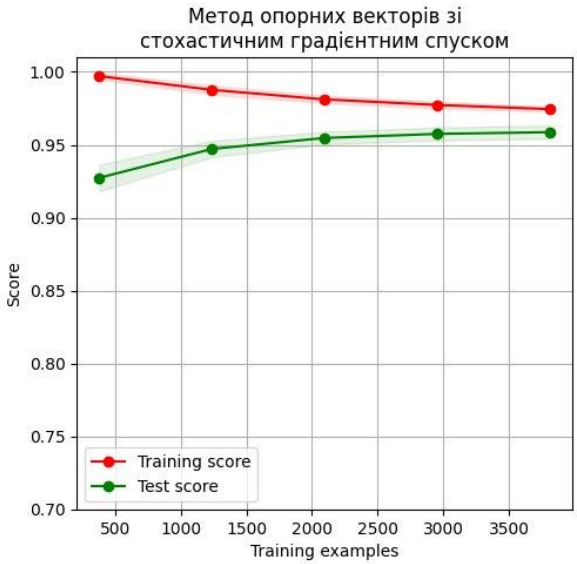


Рисунок 2.14 – Криві навчання для моделі опорних векторів зі стохастичним градієнтним спуском.

Можемо зробити висновок, що криві навчання збігаються, що є ознакою досягнення компромісу між зсувом та дисперсією для моделі опорних векторів зі стохастичним градієнтним спуском.

Графік залежності середнього часу тренування моделі стохастичного градієнтного спуску від розміру вибірки зображений на рисунку 2.15. Матриця невідповідностей для методу опорних векторів зі стохастичним градієнтним спуском зображена на рисунку 2.16.

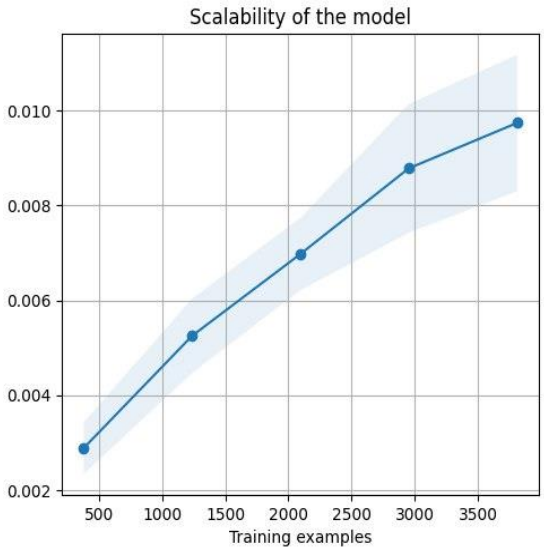


Рисунок 2.15 – Графік залежності середнього часу тренування моделі опорних векторів зі стохастичним градієнтним спуском від розміру тренувальної вибірки

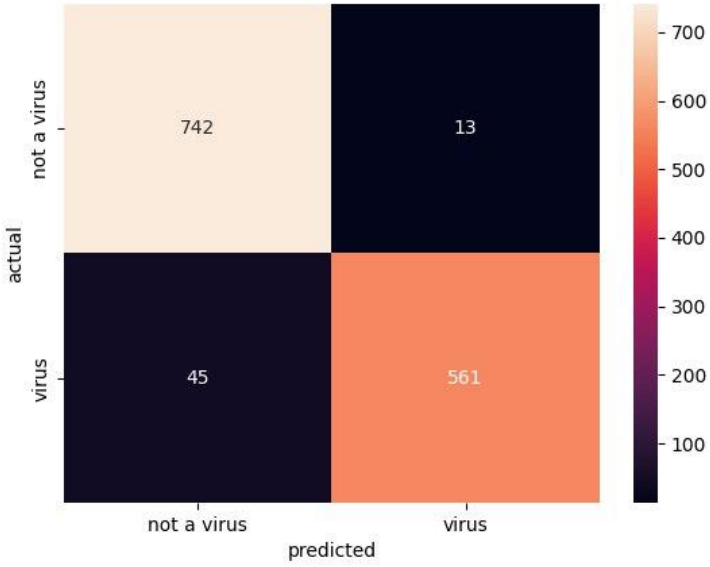


Рисунок 2.16 – Матриця невідповідностей для моделі опорних векторів зі стохастичним градієнтним спуском.

На тестовій вибірці модель опорних векторів з застосуванням стохастичного градієнтного спуску показала середню точність в 96% та $F1_{зважене}$=0.95. Результати точності погіршились на 1%, однак модель показала менший середній час тренування моделі порівняно з методом опорних векторів без використання стохастичного градієнтного спуску. Можна зробити висновок, що використання

стохастичного градієнтного спуску дійсно пришвидшує процес тренування шляхом випадкового вибору екземплярів для мінімізації функції втрат на кожній ітерації алгоритму замість обчислення функції втрат для всіх екземплярів вибірки.

Для доповнення оцінки результатів, отриманих з використанням моделей опорних векторів та стохастичного градієнтного спуску, проведено класифікацію за допомогою моделей градієнтного бустингу. У якості бустерів, обрано моделі дерева рішень, ансамбль яких використовується для вибору найоптимальнішої моделі. Для побудови моделі градієнтного бустингу проаналізовані наступні гіперпараметри:

– max_depth – максимальна глибина дерева рішень.

– learning_rate (lr) – швидкість навчання.

Графік залежності середньої точності моделі від значень гіперпараметрів зображений на рисунку 2.17. Найкращі показники точності для моделі були отримані для $lr = 0.1$ та $\text{max\_depth}=7$.

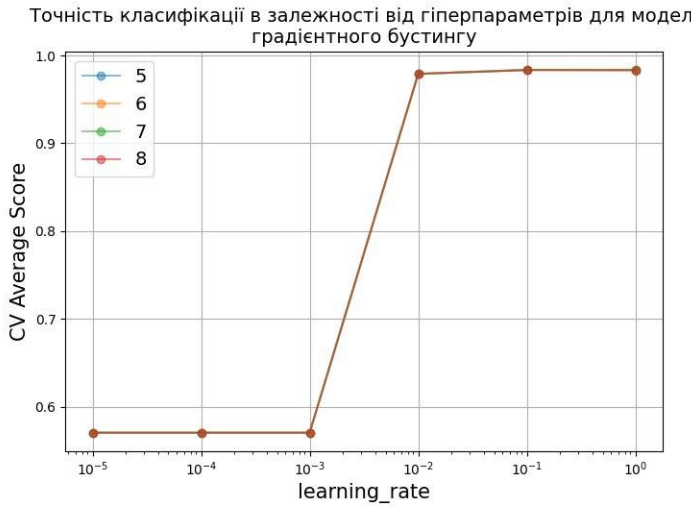


Рисунок 2.17 – Криві значень середньої точності моделі в залежності від значень гіперпараметрів lr та max_depth для моделей градієнтного бустингу.

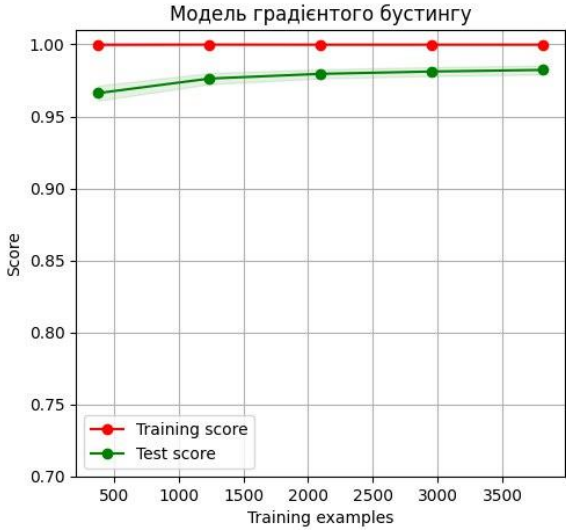


Рисунок 2.18 – Криві навчання для моделей градієнтного бустингу.

Для перевірки наявності перенавчання та дисперсії, на рисунку 2.18 наведені криві навчання. Проаналізувавши криві навчання, бачимо, що вони мають тенденцію до збіжності, але не збігаються в кінці вибірки. Таку ж саму поведінку спостерігали і для попередньо досліджених моделей. Можемо дійти висновку, що на даній вибірці спостерігається наявність дисперсії, яка може бути усунена тренуванням на більшій вибірці. Модель не має ознак перетренування, оскільки отримана точність на тестовій вибірці не є нижчою за точність отриману на тренувальних даних. Отримані результати тестування для моделей градієнтного бустингу середня точність 98% та $F1_{зважене}$=0,98.

Матриця невідповідностей для моделей градієнтного бустингу зображена на рисунку 2.19. Залежність часу, витраченого на тренування моделей градієнтного бустингу від розміру вибірки зображено на рисунку 2.20.

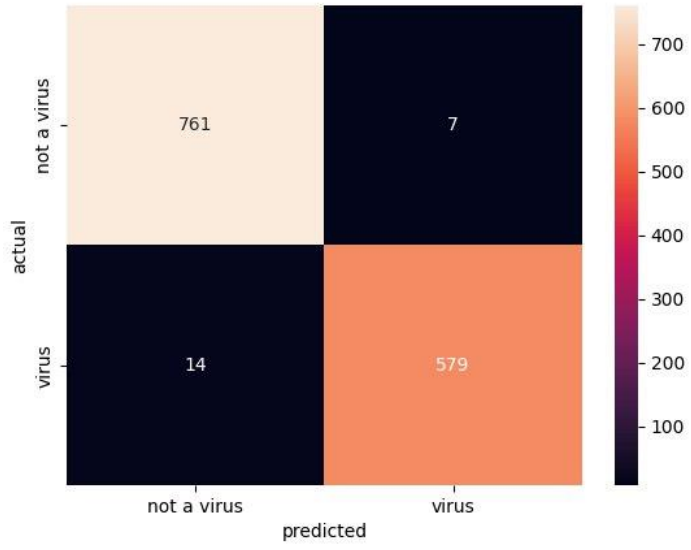


Рисунок 2.19 – Матриця невідповідностей для моделей градієнтного бустингу.

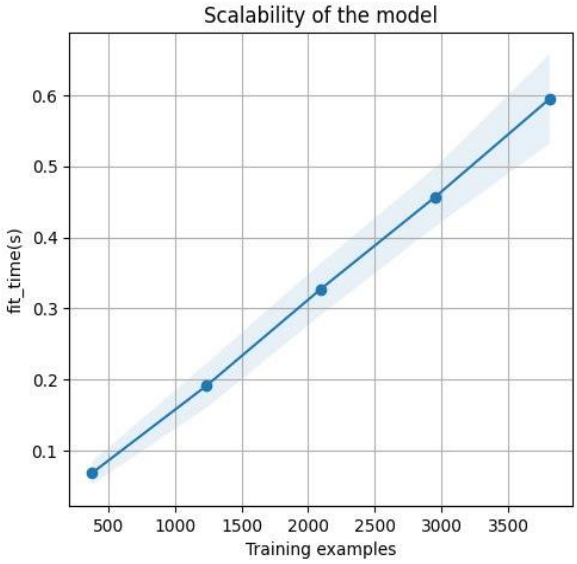


Рисунок 2.20 – Графік залежності часу тренування моделей градієнтного бустингу від розміру вибірки.

Для порівняння отриманих результатів класифікації шкідливих та безпечних ELF файлів, була створена таблиця 2.2, в якій для кожного класифікатора наведено його середню точність, $F1_{\text{зважене}}$ та середній час, витрачений на тренування на тестовій вибірці.

Таблиця 2.2. Результати класифікації шкідливих та безпечних ELF файлів моделями поліноміального Баєсового класифікатора, опорних векторів, опорних векторів зі стохастичним градієнтним спуском та градієнтного бустингу

Модель	Середня точність (%)	Міра $F1_{\text{зважене}}$	Загальний час тренування, с
1	2	3	4
Поліноміальний Баєсовий класифікатора	92	0,92	0,51
Метод опорних векторів	97	0,97	1,71
Метод опорних векторів зі стохастичним градієнтним спуском	96	0,95	0,45
Модель градієнтного бустингу	98	0,98	1,55

В загальний час тренування, вказаний в таблиці 2.2, був врахований час, витрачений на векторизацію асемблерних команд секції rodata. Отримані результати тренування моделей машинного навчання показали високу середню точність та зважений F1. Базовий поріг класифікації в 94%, отриманий для моделі

поліноміального Баєсового класифікатора, вдалось покращити, отримавши максимальне значення на тестовій вибірці у 98% для моделі градієнтного бустингу, що свідчить про правильний вибір моделей машинного навчання. Необхідно відмітити зменшення часу, затраченого на тренування моделі опорних векторів з використанням стохастичного градієнтного спуску - 0.45 секунд, порівняно з моделлю опорних векторів без стохастичного градієнтного спуску – 1.71 секунд. Це пов’язано зі стохастично обраними екземплярами на кожній ітерації алгоритму для обчислення вагових коефіцієнтів, порівняно з використанням всього набору даних для моделі без стохастичного градієнтного спуску. Також більший час тренування показала модель градієнтного бустингу, що пояснюється застосуванням ансамблю моделей до вирішення задачі класифікації. Найкращий час, показаний поліноміальним класифікатором Баєса пояснюється швидкістю обчислень для ймовірнісних моделей.

Ефективність запропонованого методу була порівняна з ефективністю методу ідентифікації шкідливих ELF файлів, запропонованого в дослідженні [143], що підтвердило статистично значущі покращення середньої точності та $F1_{зваженого}$ запропонованого в даній роботі методу, порівняно з існуючим методом. Результати порівняльного дослідження приведені в додатку В, таблиці В.2. На основі отриманих результатів можна зробити висновок, що розроблений алгоритм семантичного аналізу та класифікації шкідливого програмного забезпечення для UNIX-подібних операційних систем є дієвим та показує високу точність класифікації після вибору найоптимальнішої секції ELF файлу, розмірності n-gram та моделі класифікації.

Отже, запропоновано метод визначення секції Linux ELF файлу для ідентифікації шкідливого ПЗ, який містить процес семантичного аналізу та вибору моделі класифікації. Експериментально визначено секцію Linux-файлу яка дає найвищу точність класифікації серед досліджених секцій – rodata.

Визначено, що модель опорних векторів та модель градієнтного бустингу показала найкращі результати класифікації шкідливих та безпечних ELF файлів.

2.3 Метод ідентифікації шкідливих Windows PE файлів

Для ідентифікації шкідливих Windows Portable Executable файлів будемо використовувати статичний аналіз. Для аналізу використовується набір імпортованих бібліотек в файлі. Таким чином, завдання аналізу зводиться до вирішення проблеми класифікації тексту. Для вирішення цієї задачі були проаналізовані існуючі техніки natural language processing (NLP).

Серед існуючих архітектур нейромереж для класифікації текстових даних можна виділити Encoder-Decoder архітектуру та Transformers. Архітектура Encoder-Decoder створена на основі архітектури рекурентних нейронних мереж та використовується для виконання машинного перекладу [105], сентиментального аналізу тексту [106] та перетворення зображення в текстовий опис [107]. Архітектура transformers була запропонована в 2017 році Ashish Vaswani, Noam Shazeer та ін. та показала кращі результати: менший час на тренування та більшу здатність до паралелізації у вирішенні проблеми машинного перекладу тексту в порівнянні з моделями encoder-decoder [108]. Архітектура transformers набула популярності і в дослідженнях, що спрямовані на аналіз та класифікацію шкідливого ПЗ. Наприклад Khan S. та Nauman M у своєму дослідженні [109] створили модель на основі архітектури transformers, що змогла досягнути точності від 90% до 97% у класифікації різних типів шкідливих Windows PE файлів. У своїй роботі автори використовували архітектуру transformers для обробки опкодів PE-файлу. Дослідники Ye et. al. [125] використовували базу сигнатур та API виклики Windows PE файлів для генерації правил виявлення та хі-квадрат для оптимізації набору згенерованих правил. Автори досягли 88.09% точності. У нашій роботі, з іншого боку,

використовується лише розділ таблиці імпорту з секції “Optional Headers” PE-файлу.

Модель Bidirectional Encoder Representations from Transformers (BERT) була побудована на основі архітектури transformers. BERT отримує послідовність цифрових представлень токенів або слів і генерує відповідну послідовність семантично закодованих векторів, що представляють текст. Модель використовує Masked Language Model (MLM), яка робить передбачення випадково замаскованих слів на основі контексту документа. BERT можна використовувати не тільки для векторизації тексту, але і для вирішення завдань NLP в цілому, за умови додавання в модель відповідних кінцевих нейронних шарів. Особливістю BERT є те, що він може приймати тексти довжиною не більше 512 токенів [110]. Як вдосконалення моделі були розроблені RoBERTa [111] та ELECTRA [112]. У моделі RoBERTa представлений модифікований MLM, що полягає в динамічній зміні шаблону маскування, застосованого до вхідного тексту. У моделі ELECTRA було модифіковано MLM таким чином, що слова не приховуються, а замінюються синтетично згенерованими альтернативами. Таким чином, завдання моделі полягає в прогнозуванні не пропущених, а замінених слів. Значним проривом стала побудова моделі Generative Pre-trained Transformer (GPT) [113], автори якої запропонували підхід до навчання на великому обсязі немаркованих даних, у якому тонке налаштування моделі виконується окремо для кожного завдання. Це дозволило забезпечити ефективне передавальне навчання без істотної перебудови моделі.

У своєму дослідженні [114] Jan Sawicki та ін. зробили порівняльний огляд багатьох сучасних технік NLP. Вони зазначили, що статистичні підходи, такі як bag of words, вимагають введення ретельно очищеного тексту. З іншого боку, більш сучасні методи вбудовування слів можуть працювати з необробленим оригінальним текстом. Однією з технік вбудовування слів є техніка word2vec. Техніка word2vec була запропонована Tomas Mikolov et al. в [115] і [116] як

найновіша архітектура для векторизації слів і як альтернатива вже існуючим моделям та моделям n-gram. У word2vec було запропоновано 2 архітектури: continuous bag of words (CBOW) і contionious skip-gram. Результати порівняння створених моделей показали перевагу в точності над уже існуючими мовними моделями нейронних мереж (NNLM) і рекурентними мовними моделями нейронних мереж (RNNLM). Суть методу CBOW полягає у використанні контексту, що оточує слово, щоб передбачити слово в середині контексту. Модель skip-gram, з іншого боку, передбачає навколишні слова на основі існуючого слова. Враховуючи послідовність слів, метою моделі Skip-gram є максимізація середньої логарифмічної ймовірності оточуючих слів. Отже, метою функції витрат буде мінімізація від'ємної логарифмічної ймовірності, що те саме, що й максимізація позитивної логарифмічної ймовірності:

$$J_0 = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c,\, j\neq 0} \log p(w_{t+j} \mid w_t)\,, \tag{2.17}$$

де J_0 – функція витрат для моделі skip-gram; T – розмір тексту, c - розмір навчального контексту (чим більший контекст, тим більше екземплярів для навчання, що підвищує точність); w_t – центральне слово; log $p(w_{t+j}|w_t)$ – логарифмічна ймовірність оточуючих слів w_{t+j} , маючи слово w_t.

Файли PE складаються з кількох структур, основними з яких є заглушка MS-DOS, PE signature, заголовок файлу COFF, додатковий заголовок і заголовок розділу [117]. Типова структура PE-файлу показана на рисунку 2.21. Структура Optional header data directories, яка зберігає імпортовані та експортовані бібліотеки та функції, що використовуються для виконання PE-файлу, була обрана для статичного аналізу зловмисного програмного забезпечення. Для цієї роботи буде використано секцію "Import Table". Враховуючи те, що в деяких файлах секція «Import table» може бути відсутньою, під час роботи запропонований метод позначає такі файли як потенційно небезпечні. Такі файли

не будуть задіяні в навчальному процесі. У наборі даних для навчання, тестування та перевірки всі файли мають секцію таблиці імпортів.

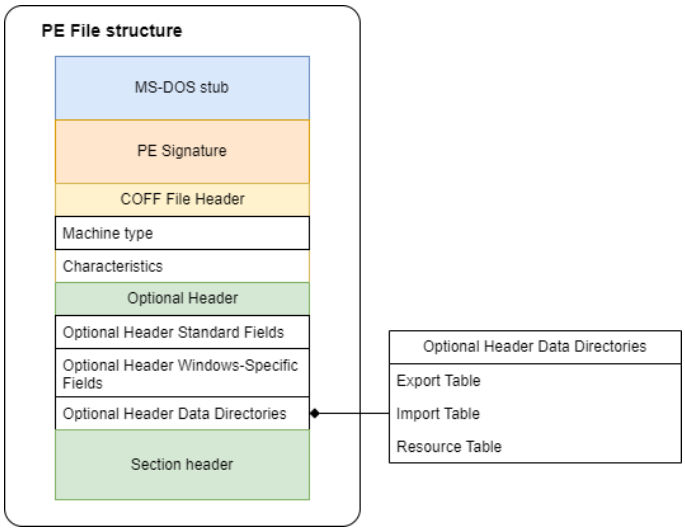


Рисунок 2.21 – Ілюстративна структура файлу PE

У вхідному наборі файлів для аналізу береться послідовність функцій, імпортованих PE-файлами. Для кожної функції було згенеровано токен виду (формула 2.18):

$$w = library_i \ \& \ function_{ij}\,, \tag{2.18}$$

де $library_i$ - бібліотека, що імпортується в секції таблиці імпортів PE файлу; $function_{ij}$ - функція, що належить $library_i$ та використовується в PE файлі. На рисунку 2.22 представлений приклад обробленої секції таблиці імпортів для PE файлу, який позначений як вірус типу Trojan.

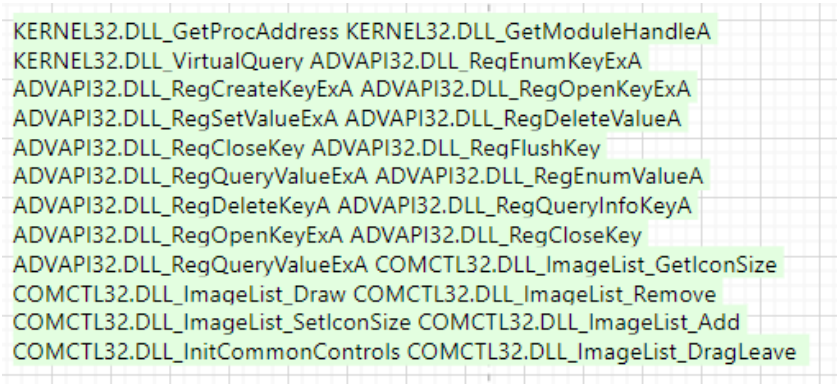


Рисунок 2.22 – Фрагмент тексту із обробленим вмістом секції таблиці імпортів PE-файлу Trojan

Таким чином, вхідний набір даних – це набір текстових документів, послідовність імпортованих бібліотек для кожного файлу. Кожен із цих документів має бути класифікований за типом вірусу, який містить файл, або за відсутністю вірусу, якщо файл безпечний. Вирішено використати техніку word2vec на основі моделі skip-gram для векторизації PE-файлів. Модель word2vec використовує неглибокі нейронні мережі, що прискорює навчання моделі та потребує менше ресурсів для класифікації файлів у реальному часі. У випадку, коли за мету ставиться прискорення процесу виявлення загроз в файлах, це є одним з ключових факторів при виборі моделі.

Окрему увагу приділено вибору та обробці набору даних для навчання та перевірки моделі класифікації файлів PE. Було проаналізовано такі набори даних, як Mal-API-2019 [118], BODMAS [119] і VirusShare [120]. Серед проаналізованих наборів даних обрано набір даних VirusShare, який представляє інформацію про шкідливі файли, отриману з веб-сайту VirusShare. А саме, були включені такі набори: “VirusShare_00000”, “VirusShare_Zeus_20190213”, “VirusShare_InstallCore_000”, “VirusShare_x86-64_WinEXE_20130711”, “VirusShare_Mediyes_000”. Набір даних включає 3 типи шкідливих файлів EXE та DLL та окремий тип безпечних файлів:

- Trojan - вірус, який маскується під безпечне програмне забезпечення. Згідно зі статистичними даними корпорації Майкрософт про найпоширеніші загрози, станом на травень 2024 року 2 із 5 найпоширеніших загроз належать до троянських програм [121].
- Adware - вірус, який встановлюється у прихований спосіб і замінює результати пошуку, додає рекламні банери на робочий стіл ОС тощо. Також відомий як PUA – потенційно небажана програма.
- Worm - вірус, який заражає систему та поширюються на інші комп’ютери через мережу. Хробаки можуть шифрувати або видаляти файли, розсилати спам, запускати DDoS-атаки тощо.

- Benign - не вірусна програма. Представлено 4328 екземплярами, які були отримані з системних каталогів «Windows» і каталогу «Program Files» у операційних системах Windows 7 і Windows 10.

Кількість вірусів кожного типу у наборі даних Windows PE файлів наведено в таблиці 2.3.

Таблиця 2.3. Кількість вірусів кожного типу у наборі даних Windows PE файлів

Тип загрози	Кількість файлів в наборі даних
1	2
Trojan	4628
Adware	4031
Worm	3875
Benign (not a virus)	4328

Відсотковий розподіл типів показано на рисунку 2.23. Дані були розділені – для навчання використовується 80% набору даних, для перевірки – 20%.

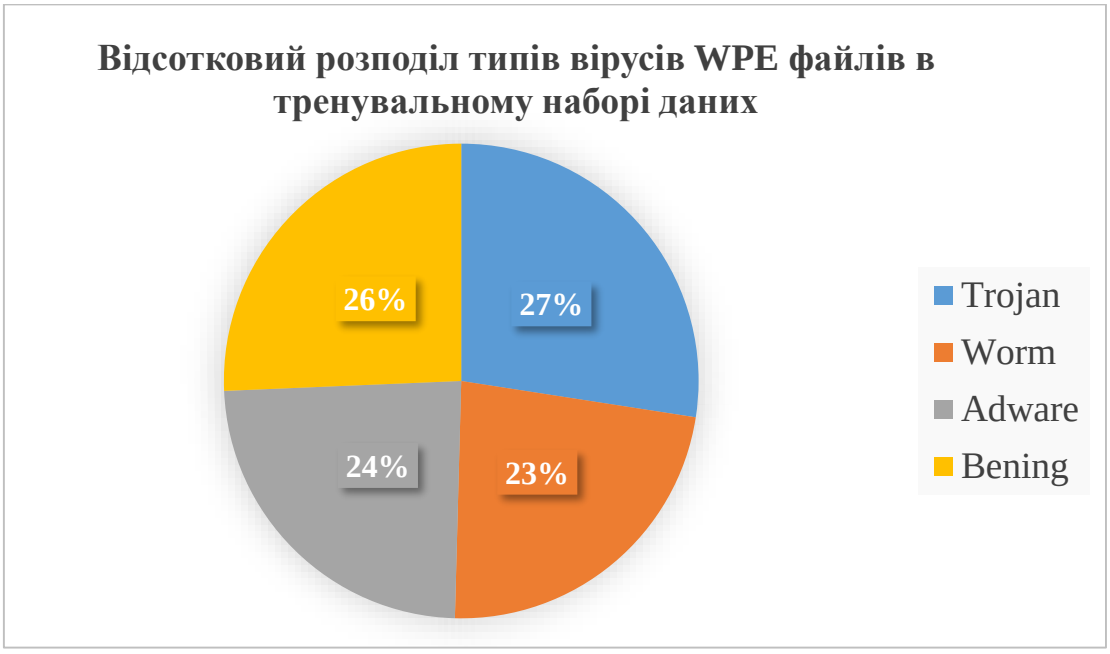


Рисунок 2.23 – Діаграма відсоткового розподілу кожного з типів вірусів в наборі PE файлів

Щоб виконати класифікацію файлів, векторизований текст передається як вхідні дані для моделей машинного навчання. Для отримання найкращого результату виконано порівняння ефективності моделей опорних векторів, ансамблю дерев рішень та багатошаровго перцептрону.

Метод опорних векторів полягає в побудові гіперплощини для розділення точок різних класів у багатовимірному просторі. Метод можна розділити на 2 підходи: лінійне розділення та нелінійне розділення. При лінійному розділенні вхідні дані не зазнають жодних перетворень – класи можна відразу розділити, побудувавши гіперплощину. Нелінійний підхід полягає у використанні функції ядра, яка перетворює вхідні дані у простір з більшою розмірністю, що полегшує розрізнення подібних даних, які належать до різних класів. У нашій роботі використовуємо функцію ядра Radial Basis Function (RBF) для побудови гіперплощин, яка представлена у формулі 2.19:

$$k(x,z) = e^{-\gamma\|x-z\|^2}, \tag{2.19}$$

де $\gamma > 0$ - параметр, який контролює вплив кожного екземпляра з вибірки на межу рішення; x, z - екземпляри вибірки [122]. Чим більше евклідова відстань між екземплярами вибірки, тим ближче значення функції до нуля. Це означає, що такі екземпляри, швидше за все, належать до різних класів.

Модель ансамблю дерев рішень – це метод машинного навчання, який полягає у створенні ансамблю дерев рішень під час навчання. У разі класифікації клас елемента визначається найбільш частим результатом навчання дерева рішень. Ансамбль дерев рішень часто використовується для класифікації шкідливих файлів за допомогою статичної текстової інформації файлу. Наприклад, Trung Kien Tran і Hiroshi Sato у своїй роботі [123] використовували векторизований набір команд API як вхідні дані для моделі ансамблю дерев рішень, отримавши найкращу точність 98% для набору даних 2 класів і 95% для 4-класового набору даних.

У нашому дослідженні використовуються наступні параметри для моделі ансамблю дерев рішень:

– number_of_trees = 100 – кількість дерев

– criterion = ‘Gini’ – критерій домішок Gini, формула 2.20. Функція, яка вимірює кількість домішок у екземплярах, віднесених до одного класу (до вузла дерева). Ближче значення до 0 означає менше домішок і кращу якість класифікації.

$$Gini = 1 - \sum_{i=1}^{n} p(i)^2 , \tag{2.20}$$

де $p(i)$ ймовірність, що екземпляр належить вузлу дерева, що є класом i; n – кількість класів.

Алгоритм багатошарового перцептрона є нейронною мережею з різними можливими функціями активації. У даній роботі використана функція активації ReLU (формула 2.21).

$$f(x) = max(0, x) , \tag{2.21}$$

де x - вхідний сигнал. Було побудовано модель багатошарового перцептрона з першим прихованим шаром із 100 нейронів, другим прихованим шаром із 50 нейронів і третім прихованим шаром із 25 нейронів. Як метод оптимізації багатошарового перцептрона використано алгоритм Adam [124], який полягає в розрахунку швидкості навчання для кожної ваги нейронної мережі та її коригування під час навчання.

Щоб оцінити та порівняти результати класифікації моделей, будемо використовувати криву Reciever Operating Chracteristics (ROC) - графік, що показує ефективність моделі класифікації за всіма пороговими значеннями класифікації. Крива показує відношення дійсного позитивного результату до хибного позитивного результату – чим ближче це значення до одиниці, тим краще працює модель.

Було використано такі показники для оцінки ефективності класифікації файлів PE:

1. *Pr* (формула 2.22) – влучність класифікації, вимірює частку істинно позитивних результатів серед усіх класифікованих позитивних результатів.

2. *Rec* (формула 2.23) – повнота класифікації, вимірює частку істинно позитивних результатів серед хибно негативних і істинно позитивних результатів.

3. F1-міра (формула 2.24) – середнє гармонійне *Pr* та *Rec*.

4. *Acc* (формула 2.25) – середня точність для всіх прогнозів, зроблених моделлю.

$$\mathrm{Pr}(\mathrm{class}=a)=\frac{\mathrm{TP}(\mathrm{class}=a)}{\mathrm{TP}(\mathrm{class}=a)+\mathrm{FP}(\mathrm{class}=a)} \tag{2.22}$$

де, TP – кількість істинно позитивних результатів; FP – кількість хибно позитивних результатів.

$$\mathrm{Rec}(\mathrm{class}=a)=\frac{\mathrm{TP}(\mathrm{class}=a)}{\mathrm{TP}(\mathrm{class}=a)+\mathrm{FN}(\mathrm{class}=a)} \tag{2.23}$$

де TP – кількість істинно позитивних результатів, FN – кількість хибно негативних результатів.

$$\mathrm{F1}(\mathrm{class}=a)=\frac{2\times\mathrm{Pr}(\mathrm{class}=a)\times\mathrm{Rec}(\mathrm{class}=a)}{\mathrm{Pr}(\mathrm{class}=a)+\mathrm{Rec}(\mathrm{class}=a)} \tag{2.24}$$

де Precision(class=a) – точність для класу a; Recall(class=a) – повнота передбачень для класу a.

$$\mathrm{Acc}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{2.25}$$

де TP – кількість істинно позитивних результатів; TN – кількість істинно негативних результатів; FP – кількість хибно позитивних результатів; FN – кількість хибно негативних результатів.

Щоб оцінити загальну якість класифікації серед всіх класів, було розраховано макросереднє та середньозважене значення для показників *acc*, *pr*, *rec* та F1-міри. Макросереднє – середнє значення метрики для кожного класу. Середньозважене – середнє значення метрики для кожного класу, помножене на частку екземплярів для класу.

ROC криву було побудовано для оцінки якості навчання моделей. Крива будує залежність двох параметрів: True positive rate (формула 2.26) і False positive rate (формула 2.27).

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{2.26}$$

де TP – кількість істинно позитивних результатів; FN – кількість хибно негативних результатів.

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{2.27}$$

де FP – кількість хибно негативних результатів; TN – кількість істинно негативних результатів.

Щоб адаптувати криву для випадку небінарної класифікації, було використано техніку OVR – One vs rest, за якої крива будується для кожного класу, беручи до уваги, що правильно класифікований конкретний клас є позитивним, а всі інші класи є негативними.

Також були побудовані додаткові криві: крива micro ROC, для якої TPR і FPR розраховуються з використанням суми всіх TP, FP, TN, FN для всіх класів; макро-крива ROC, для якої TPR і FPR розраховуються окремо для кожного з класів і діляться на кількість класів.

Для кожної з запропонованих моделей класифікації, була проведена перехресна валідація різних значень гіперпараметрів і вибрано найефективніші гіперпараметри. Для методу опорних векторів було проведено перехресну валідацію значень параметра регуляризації C на 5 тестових розбивках, які

показали найкращий середній результат точності 0,926 для C=600. Результати перехресної валідації представлені в таблиці 2.4.

Таблиця 2.4. Перехресна валідація параметра регуляризації C для методу опорних векторів

Значення параметру *C*	Середня точність	Середній час тренування (секунди)	Середній час тестування (секунди)
1	2	3	4
0.1	0.838	77.66	4.7
0.5	0.873	31.71	2.12
1	0.876	26.83	2.07
4	0.903	22.14	1.56
10	0.912	16.07	1.21
50	0.92	15.72	1.09
100	0.922	17.28	1.14
500	0.925	24.56	1.18
600	**0.926**	**30.39**	**1.29**
1000	0.925	28.07	1.13
2000	0.924	31.22	1.17

У результаті перехресної валідації для методу опорних векторів були обрані такі параметри: C=600, kernel='rbf' - радіальна базисна функція.

Результати оцінки методу опорних векторів наведені в таблиці 2.5. З таблиці бачимо, що модель найточніше класифікує ПЗ класу adware – точність 0,97, F1-міра – 0,97. Найменш точно класифікований ПЗ класу trojan – точність 0,87, F1-міра 0,88. Криві ROC, побудовані для методу опорних векторів, які показано на рисунку 2.24, показують велику площу під кривою (AUC) – від 0,97 для

нешкідливих файлів і до 1,00 для adware файлів. Це означає, що метод опорних векторів здатний правильно класифікувати кожен з класів вибірки.

Таблиця 2.5 – Отримані показники ідентифікації зловмисного програмного забезпечення та безпечних файлів PE за допомогою моделі опорних векторів.

	Точність	Повнота	F1-міра
1	2	3	4
adware	0.97	0.97	0.97
benign	0.96	0.98	0.97
trojan	0.87	0.89	0.88
Worm	0.92	0.87	0.90
Макро значення	**0.93**	**0.93**	**0.93**
Зважене значення	**0.93**	**0.93**	**0.93**
Середня точність, %	**93**		
Середній час класифікації WPE файлу, секунди	**0.03**		

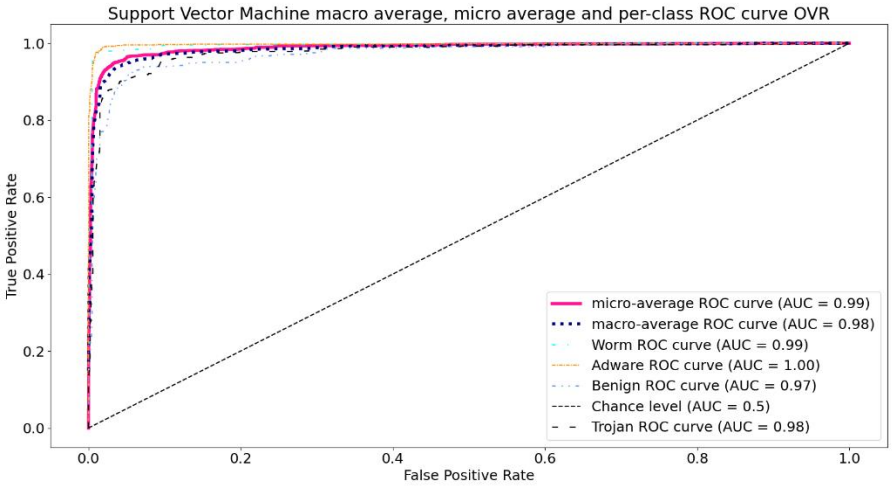


Рисунок 2.24 – Криві ROC ідентифікації шкідливих PE файлів з використанням методу опорних векторів

Далі використаємо класифікатор на основі ансамблю дерев рішень. Результати даного класифікатора приведені у таблиці 2.6.

Таблиця 2.6 – Отримані показники ідентифікації зловмисного програмного забезпечення та безпечних файлів PE за допомогою моделі ансамблю дерев рішень.

Тип	Точність	Повнота	F1-міра
1	2	3	4
adware	0.99	0.98	0.99
benign	0.94	0.99	0.96
Trojan	0.92	0.89	0.90
worm	0.92	0.89	0.91
Макро значення	**0.94**	**0.94**	**0.94**
Зважене значення	**0.94**	**0.94**	**0.94**
Середня точність, %	**94**		
Середній час класифікації WPE файлу, секунди	**0.04**		

Модель ансамблю дерев рішень показує середню точність – 94%, макро F1-міру – 0,94, зважену F1-міру – 0,94. Шкідливі файли типу adware ідентифікуються найкраще – 0,99 точності, 0,98 повноти передбачень. Криві ROC для моделі ансамблю дерев рішень показані на рисунку 2.25. Після аналізу кривих ROC для моделі ансамблю дерев рішень, можна зробити висновок, що здатність моделі правильно ідентифікувати PE-файли підтверджується значенням AUC для micro ROC – 0,99 , макро ROC – 0,99 та для ROC кожного окремого класу.

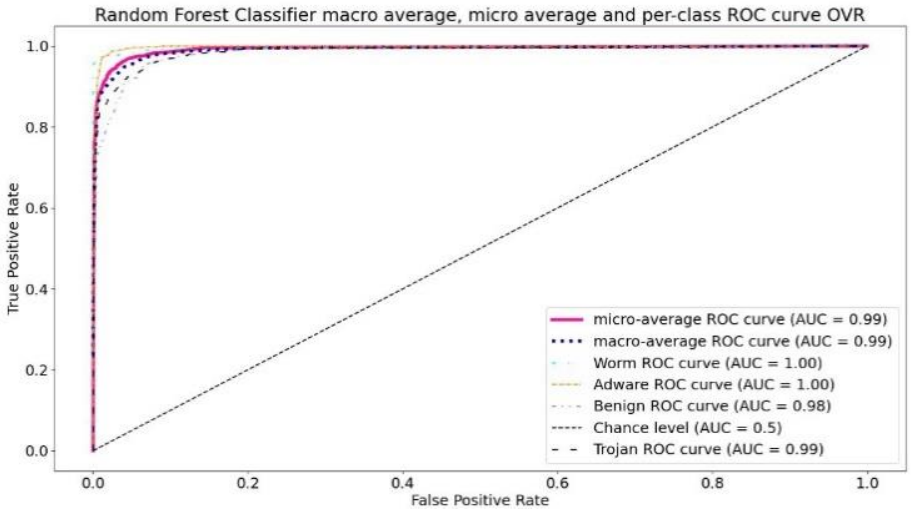


Рисунок 2.25 – Криві ROC для ідентифікації шкідливих PE файлів з використанням моделі анасамблю дерев рішень

Третьою проаналізованою моделлю є багатошаровий перцептрон, який використовує функцію активації ReLU та оптимізатор adam. Результати тренування моделі представлено в таблиці 2.7.

Таблиця 2.7 - Отримані показники ідентифікації зловмисного програмного забезпечення та безпечних файлів PE за допомогою моделі багатошарового перцептрону.

	Точність	Повнота	F1-міра
adware	0.99	0.96	0.97
benign	0.96	0.98	0.97
trojan	0.83	0.86	0.86
worm	0.91	0.85	0.88
Макро показник	**0.92**	**0.92**	**0.92**
Зважений показник	**0.92**	**0.92**	**0.92**
Середня точність, %	**92**		
Середній час класифікації WPE файлу, секунди	**0.1**		

Криві ROC для моделі багатошарового перцептрона показані на рисунку 2.26.

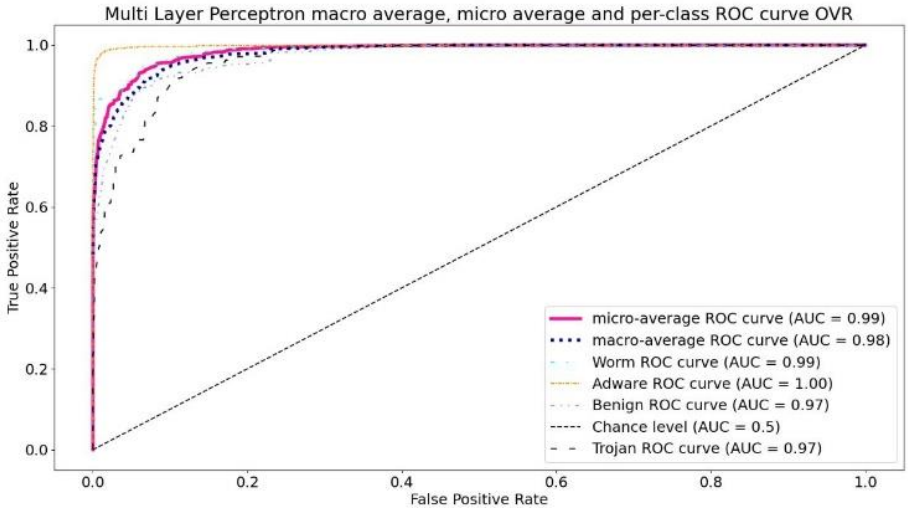


Рисунок 2.26 – Криві ROC для ідентифікації шкідливих PE-файлів з використанням багатошарового перцептрона

Як бачимо з таблиці 2.7, модель багатошарового перцептрону найкраще впоралася з класифікацією файлів шкідливого ПЗ типу adware – точність 99%, повнота передбачень 96%, оцінка F1 0,97, середня точність моделі становила 92%. Макро та середнє зважене значення F1 для моделі становить 0,92.

Кожна з моделей показала нижче значення F1-міри для вірусів типу Trojan, ніж для інших типів вірусів. Це може бути пов’язано з тим, що сімейство троянських програм містить багато різних підтипів загроз, наприклад програм-вимагачів, шпигунських програм тощо. Відповідно, шкідливі файли PE типу Trojan можуть мати більше відмінностей, ніж шкідливі файли інших типів, що ускладнює процес класифікації таких файлів моделями машинного навчання.

Порівняння запропонованого в нашому дослідженні методу ідентифікації шкідливих Windows PE файлів з існуючими моделями класифікації файлів WPE, запропонованими в роботах [51, 52] [53], наведено в таблиці 2.8.

Для порівняння ефективності запропонованого в роботі методу з існуючими методами було виконано t-test з одним хвостом, в якому визначено критерії: μ0 –

середнє значення точності та F1 класифікації шкідливих Windows PE файлів для методів, запропонованих в роботах [51, 52] [53, 125], μ - середнє значення точності та F1 класифікації шкідливих Windows PE файлів 10 експериментів, проведених для методу, що запропонований у нашому дослідженні. Визначена гіпотеза H_0="$\mu \leq \mu 0$", гіпотеза H_1= "$\mu > \mu 0$", для рівня значущості α=0.01. Результати порівняльного дослідження з застосуванням t-тесту Стьюдента, що підтверджують статистично значуще покращення ефективності запропонованого в роботі методу порівняно з існуючими дослідженнями, приведені в додатку В, таблиці В.3. Для методу [52] не вдалось відхилити гіпотезу H_0 для значень середньої точності, однак вдалось відхилити H_0 та прийняти H_1 для значення F1-міри.

У роботі Lad Sumit et. al. [52] використовують модель DNN на наборі даних EMBER 2018 із 5 типами атрибутів: загальна інформація про файл (віртуальний розмір, кількість імпортованих і експортованих функцій тощо), інформація заголовка, імпортовані функції, експортовані функції, інформація про розділ. Автори досягли 94,09% точності та значення F1-міри у 0,8866.

Дослідники Koçak, Aynur та ін. використовували метод найближчих сусідів (IBk weka) для кластеризації зловмисного програмного забезпечення WPE шляхом аналізу мережевих пакетів активності WPE і досягли точності 90,47% та F1-міри 0,904 [53].

У дослідженні [51] Yuk Chang та ін. використовували метадані заголовків PE файлів, з використанням підходу one hot кодування для класифікації шкідливих PE файлів моделями машинного навчання. Автори досягли точності класифікації 92.4 %.

Можна зробити висновок, що запропонований в даному дисертаційному дослідженні метод ідентифікації шкідливих PE файлів з використанням методів word2vec і ансамблю дерев рішень показав покращення порівняно з проаналізованими існуючими методами щодо точності та оцінки F1.

Таблиця 2.8. Порівняння існуючих методів ідентифікації шкідливих PE файлів з запропонованим у дослідженні методом

Метод	Вхідні дані	Розмір вибірки	Точність, %	F1-score
1	2	3	4	5
Lad, Sumit & Adamuthe, Amol [52] (DNN)	EMBER 2018 dataset	1000000	94,09	0,8866
Koçak, Aynur et al. [53] (IBk weka algorithm)	Мережеві пакети активності Windows PE файлів	77184	90,47	0,9040
Yuk, Chang та Seo, Chang [51]	Метаінформація заголовків Windows PE файлів	933	92,40	0,9260
Запропонований метод (word2vec та Random forest classifier)	WPE import table	**16862**	**94,00**	**0,9400**

В результаті проведеного дослідження:

1. Удосконалено метод ідентифікації Windows PE файлів, шляхом використання техніки word2vec та ансамблю дерев рішень (Random forest classifier) для бібліотек та функцій з секції файлу «Import Table», що показав збільшення точності від 2 до 4 % та F1-міри від 0.02 до 0.06 порівняно з існуючими методами.

2. Практичною цінністю є можливість виявляти шкідливі Windows PE файли з відомими або модифікаціями існуючих загроз, базуючись на секції «Import Table».

2.4 Метод прогнозування загроз з використанням Мережі Баєса

Мережа Баєса являє собою спрямований ациклічний граф, що відображає зв'язки між змінними та їхніми умовними ймовірностями. Використання мереж Баєса досліджується для виявлення кореляції між сповіщеннями систем безпеки [126]. Також методи з використанням Мереж Баєса використовуються для удосконалення процесу прогнозування та виявлення загроз [127]. Мережі Баєса параметризуються з використанням умовного розподілу ймовірностей та ланцюгового правила ймовірностей, за формулою 2.28.

$$P(A \cap B) = P(B \mid A)P(A), \tag{2.28}$$

де P(A) – ймовірність змінної A; P(B) – ймовірність змінної B; P(B|A) – ймовірність змінної B залежно від змінної A.

Для динамічної побудови мережі Баєса використовують евристичний алгоритм «Hill Climb Search». Даний алгоритм полягає у ініціалізації випадковим чином початкової структури мережі Баєса, визначення якості побудованої мережі шляхом обчислення критерію інформативності Баєса (формула 2.29) у якості функції оцінки та застосування модифікацій до попередньо визначеною структури з метою отримання покращених результатів функції оцінки мережі.

$$BIC = -2\ln(L) + k\ln(n), \tag{2.29}$$

де L – функція правдоподібності моделі враховуючі вхідні дані; k – кількість параметрів моделі; n – кількість точок спостережень.

Для проведення експериментів з прогнозування ймовірностей виникнення загроз з використанням Мережі Баєса, скористаємось набором даних CSE-CIC-IDS2018 в якому містяться атаки різних типів та відповідний ним мережевий

трафік. На рисунку 2.27 зображена мережа, побудована на основі обраного набору даних.

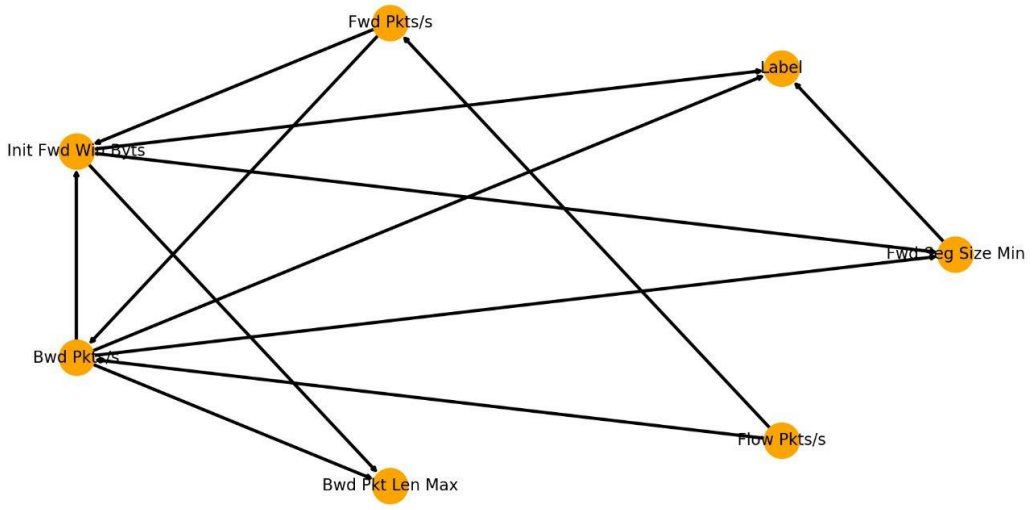


Рисунок 2.27 – Мережа Баєса побудована на основі набору даних CSE-CIC-IDS2018

В результаті побудови мережі Баєса та застосування її до прогнозування ймовірностей на тестовому наборі даних, отримали точність, приведену у таблиці 2.9.

Таблиця 2.9 – Оцінка F1-міри прогнозування ймовірностей загроз з використанням мережі Баєса

Тип загрози або її відсутність	F1-міра	Кількість екземплярів в тестовій вибірці
FPT BruteForce	0,99	6607
SSH BruteForce	0,98	6696
Нормальний трафік	0,99	6696

Таким чином, визначено, що метод прогнозування ймовірності загроз з використанням мережі Баєса показує високу точність та F1-міру та може бути використаний як частина інформаційної технології виявлення та прогнозування загроз.

2.5 Модель комплексної інформаційної технології виявлення та прогнозування загроз для корпоративної комп'ютерної мережі засобами експертних систем

Використання експертних систем, як окремої сутності для вирішення проблем кібербезпеки є досить детально дослідженим питанням, яке розглянуто нами у розділі 1.3.1. Створення композиції методів, які б містили в собі компоненти виявлення та прогнозування кіберзагроз для наповнення бази знань експертних систем є новим підходом. Розроблена модель комплексної інформаційної системи (рисунок 2.28), що містить модулі експертної системи, модулі ідентифікації шкідливого ПЗ, DDoS атак та рушій висновків на основі Теорії Ігор, які дозволяють визначити рівень критичності загрози та виконати прогнозування ймовірності реалізації визначених векторів загроз.

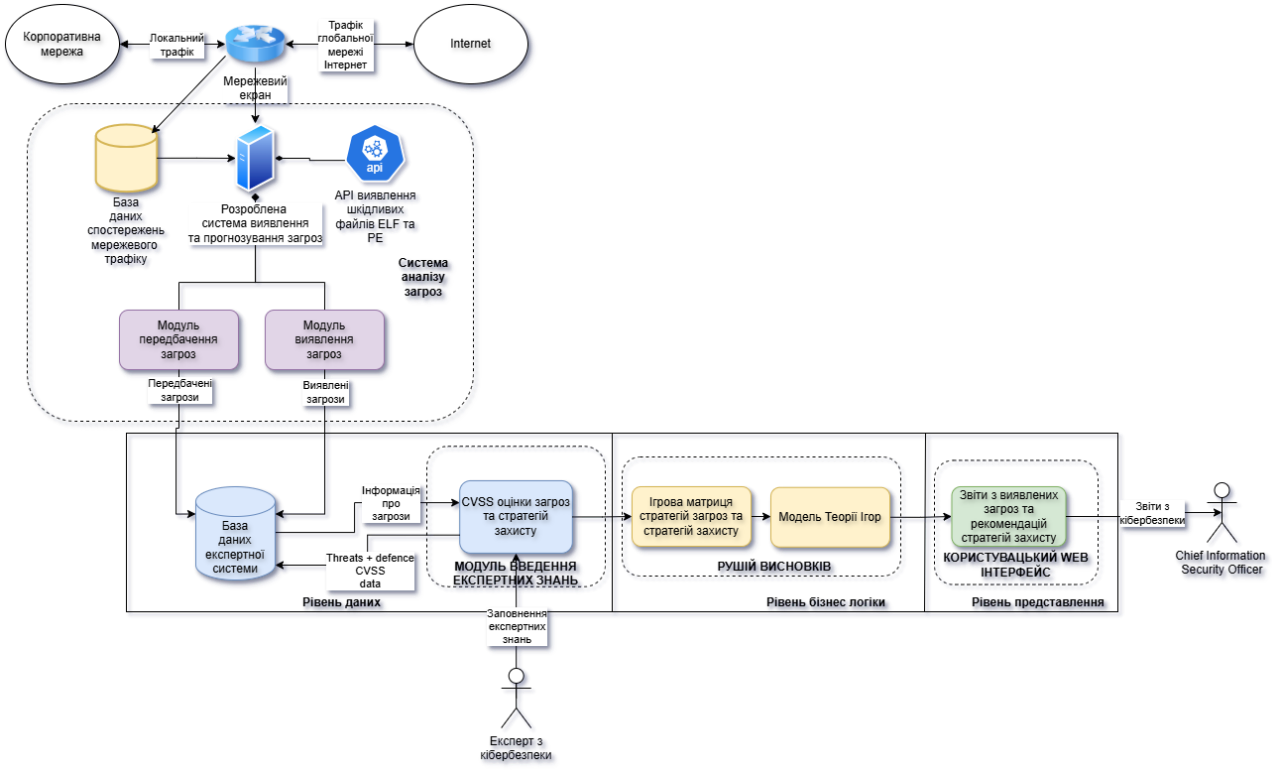


Рисунок 2.28 – Модель комплексної інформаційної технології виявлення та прогнозування загроз для корпоративних комп'ютерних мереж

Наведена модель містить заповнення бази знань експертами, яке відбувається шляхом оцінки рівня CVSS та визначення стратегій протидії фіксованому набору кіберзагроз, що надходять до бази даних з модулів виявлення та прогнозування загроз, розроблених та описаних в розділах 2.1 – 2.4 дисертаційного дослідження. База знань експертів наповнюється шляхом визначення показнику CVSS для певної атаки та можливої протидії атаці у вигляді зв'язки *<атака>-<протидія>-<рівень cvss>*. Common Vulnerability Scoring System (CVSS) – загальна система оцінки вразливостей, відкритий стандарт індустрії кібербезпеки. Вона створена для оцінки рівня критичності вразливостей комп'ютерної системи [9]. Найновіша специфікація CVSS 4.0 складається з 4 груп метрик: Base, Threat, Environmental та Supplemental. Групи та відповідні їм метрики представлені на рисунку 2.29.

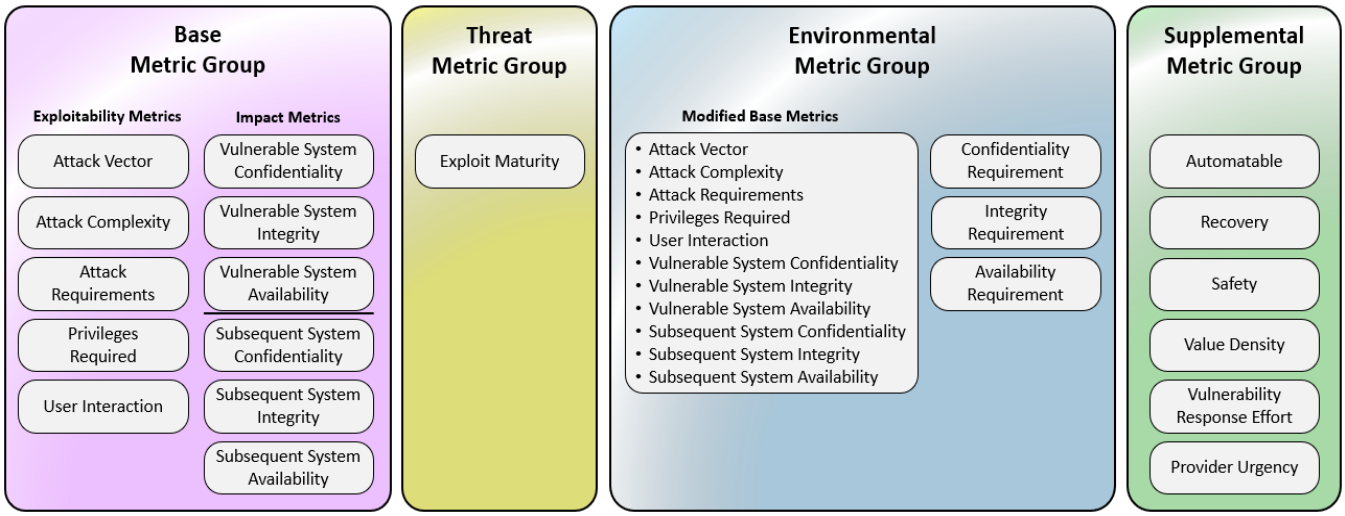


Рисунок 2.29 – Групи та метрики стандарту CVSS 4.0

Джерело: **first.org [119]**

Відповідно до нової специфікації, загальне значення CVSS обчислюється за допомогою створення текстового вектору (формула 2.30).

$$CVSS{:}\,4.0/< metric\_name >:< metric\_value >, \qquad (2.30)$$

де *metric_name* – умовне позначення метрики; *metric_value* – значення метрики.

Значення CVSS може бути від 0 до 10, де 10 – максимальна міра ступеню загрози кібератаки. Наприклад, текстовий вектор *CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:H/SC:N/SI:N/SA:H* - описує DDoS атаку на локальну мережу і містить наступні метрики: Attack Vector (AV): Network (N), Attack Complexity (AC): Low (L), Attack Requirements (AT): None (N), Privileges Required (PR): None (N), User Interaction (UI): None (N), Confidentiality (VC): None (N), Integrity (VI): None (N), Availability (VA): High (H); Subsequent system impact metrics: Confidentiality (SC): None (N), Integrity (SI): None (N), Availability (SA): High (H). Відповідна атака має значення CVSS=9.3, що відповідає критичному рівню загрози. У таблиці 2.10 приведено значення метрик CVSS для досліджених в роботі загроз та кориговані метрики, якщо були вжиті стратегії з кіберзахисту.

Алгоритм оцінки текстового вектору CVSS не надається у відкритому доступі, проте на офіційному сайті [128] представлено калькулятор, за яким для текстового вектору обчислюється значення CVSS та може бути використаний експертами.

У нашому дослідженні експертам пропонується оцінювати рівень CVSS за базовою групою метрик. Метрика CVSS є стандартом, що досліджується, вдосконалюється та застосовується у різних сферах – в тому числі і в сфері кібербезпеки. Наприклад, у своїй роботі D.T. Vasireddy та ін. запропонували передбачення значення метрики CVSS для вразливостей в системах електропостачання з використанням Doc2Vec моделі та нейромереж. Автори векторизують текстовий опис вразливості та виконують регресійний аналіз векторизованого тексту. Розроблений метод вдосконалює CVSS, замінюючи необхідність розрахунку значення метрики вручну експертами [129].

Таблиця 2.10. Метрики CVSS для досліджених кіберзагроз та стратегії кіберзахисту

Кібер-загроза	Текстовий вектор CVSS	Рівень CVSS	Стратегія кіберзахисту	Текстовий вектор CVSS, якщо вжито стратегію кіберзахисту	Рівень CVSS, якщо вжито стратегію кіберзахисту
1	2	3	4	5	6
DDoS атака	CVSS:4.0/AV:N/ AC:L/AT:N/PR:N/ UI:N/VC:N/VI:N/ VA:H/SC:N/SI:N/SA: H	9.3	Блокування IP адрес, з яких надходить атака	CVSS:4.0/AV:N/ AC:L/AT:N/PR:N/UI:N /VC:N/VI:N/VA:N/SC:N/SI:N/SA:N	0
Шкідливий PE файл класу Trojan	CVSS:4.0/AV:L/AC: L/AT:P/PR:N/UI:P/V C:H/VI:H/VA:H/SC:L /SI:L/SA:L	7.3	Тренування з кібербезпеки для користувачів мережі	CVSS:4.0/AV:L/AC:H/AT:P/PR:H/ UI:A/VC:H/VI:H/VA:H/SC:L/SI:L/ SA:L	5.4
Шкідливий PE файл класу adware	CVSS:4.0/AV:L/AC: L/AT:P/PR:N/UI:A/V C:N/VI:L/VA:N/SC:N /SI:N/SA:N	5.7	Використати ПЗ CCleaner на ПК < IP ПК>	CVSS:4.0/AV:L/AC:L/AT:P/PR:N/ UI:A/VC:N/VI:H/VA:H/SC:N/SI:L/ SA:L	1.8
Шкідливий ELF файл	CVSS:4.0/AV:N/AC: L/AT:N/PR:N/UI:A/V C:L/VI:N/VA:H/SC:L /SI:N/SA:H	8.3	Заблокувати й видалити файл <filename>, заблокувати IP <source ip> звідки завантажено файл	CVSS:4.0/AV:N/AC:L/AT:N/PR:N/ UI:A/VC:N/VI:N/VA:N/SC:N/SI:N/ SA:N	0

Автори Maghrabi L. та ін. у своїй роботі [130] розробили методику вдосконалення стратегій усунення кібервразливостей шляхом знаходження рівноваги Неша для матричної гри, гравцями якої виступають кіберзлочинець та кіберзахисник. Для обчислення значень цільової функції автори використовували Confidentiality, Integrity та Availability метрики стандарту CVSS.

Робота авторів Maghrabi L. та ін. стала частиною удосконалення існуючого ПЗ CAESAIR [131], що є системою аналізу кіберінцидентів та спрямована на надання аналітичної підтримки експертам з кібербезпеки на основі NLP аналізу звітів з кіберінцидентів, виявлення кореляції між інцидентами, визначення ключових особливостей та представлення аналітичних відомостей користувачу. На відміну від запропонованої авторами системи, модель інформаційної системи, представленої нами у дослідженні, базується на використанні методу експертних оцінок виявлених загроз, що являють собою ідентифіковані шкідливі бінарні файли, інформацію про DoS та DDoS атаки. Також в запропонованій нами моделі вирішення матричної гри відбувається з використанням ітераційного методу фіктивного розігрування.

S. Zhang та ін. провели дослідження найбільш схильних до помилок метрик CVSS, проаналізувавши ідентичні або достатньо схожі записи в National Vulnerability Database, що мають достатньо віддалені значення CVSS. Це дозволило встановити більш досконалу відповідність між значеннями метрик та реальними кіберзагрозами [132].

У якості рушію висновків інформаційної системи вирішено використовувати теоретико-ігрову модель для вирішення стратегічної антагоністичної гри, акторами якої виступають кіберзлочинець та спеціаліст з кіберзахисту. Для кожного гравця його стратегія представлена як вектор, де кожен елемент вектора відповідає значенню CVSS, у разі вибору певної стратегії – кіберзагрози або кіберзахисту. Елементи матриці гри представлені відповідним значенням CVSS. Проте якщо певну загрозу можна повністю нейтралізувати

відповідною стратегією кіберзахисту, це вважатиметься програшем для кіберзлочинця, тому відповідне значення CVSS в матрицю буде зараховуватись зі знаком мінус. Задача кіберзлочинця максимізувати CVSS для стратегії експлуатації загрози, експерта з кіберзахисту – мінімізувати це значення.

Експерт встановлює початкове значення CVSS для загрози, яке актуальне, якщо жодних заходів з протидії не було вжито. Далі експерт може визначити заходи з протидії та відповідне кориговане значення CVSS. Для відповідності загроза-протидія, що не були визначені експертом, встановлюється початкове значення CVSS загрози. Таким чином, гра буде визначена як система множин:

$$G = \left\langle 2, \{S_i\}_{i\in\{1,2\}}, \{H_i\}_{i\in\{1,2\}} \right\rangle, \tag{2.33}$$

де S_i-стратегія гравця *i,* що являє собою кібератаку або кіберзахист, H_i-виграш гравця *i,* що являє собою CVSS score – позитивне значення означає виграш кіберзлочинця, а негативне значення – виграш спеціаліста з кібербезпеки. Для вирішення матричної гри буде застосовуватись ітераційний метод фіктивного розігрування [133]. На рис. 2.29 схематично зображено метод формування рекомендацій в експертній системі.

Ітераційний метод полягає у знаходженні змішаних стратегій для кожного з гравців шляхом фіктивного багаторазового розігрування гри. Метод починається з вибору першим гравцем своєї максимінної стратегії. У відповідь на це другий гравець розігрує стратегію, що приносить йому найменший програш. В подальших ітераціях кожний гравець обирає стратегії відповідно до попередньої зіграної стратегії протилежним гравцем. Змішані стратегії кожного з гравців обчислюються шляхом вирахування частоти використання кожної зі стратегій. Програмний код запропонованого метода приведений у додатку Г даної дисертаційної роботи.

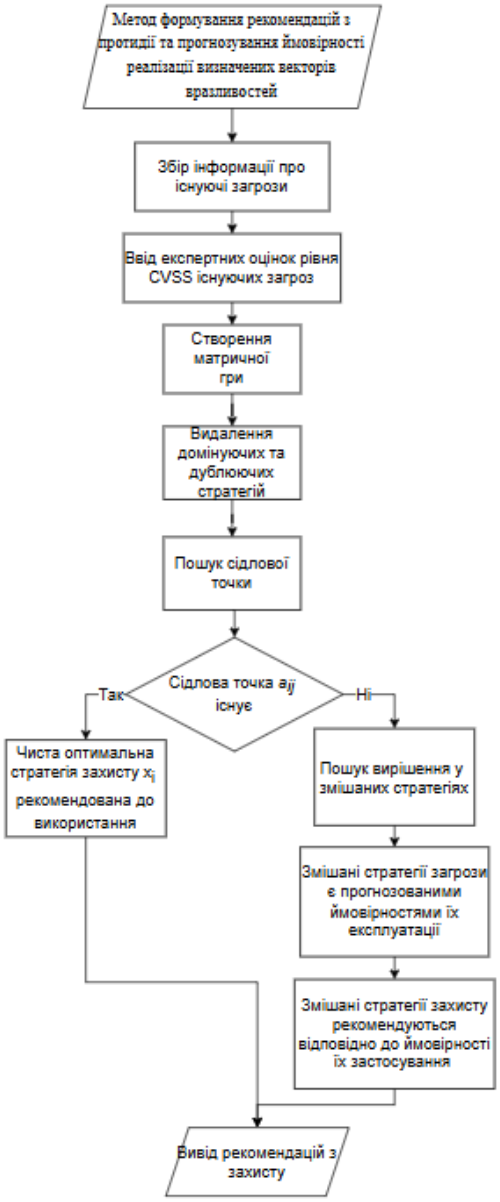


Рисунок 2.29 – Метод формування рекомендацій з застосування оптимальних стратегій кіберзахисту в інформаційній системі

Створена інформаційна система генерує рекомендації відповідно до рішення матричної гри. Першим кроком процесу формування рекомендацій з кіберзахисту є пошук вирішення матричної гри в чистих стратегіях. Відбувається пошук нижньої ціни гри $\underline{V} = \max_i \min_j a_{ij}$, верхньої ціни гри $\bar{V} = \min_j \max_i a_{ij}$ та сідлової точки $\bar{V} = \underline{V}$

Якщо сідлова точка знайдена – то вирішенням гри є стратегії (x_{i_0}, y_{j_0}), що є парою оптимальних чистих стратегій. Якщо не знайдена, необхідно перейти до вирішення в змішаних стратегіях. Формування рекомендацій із застосування стратегій кіберзахисту залежить від вирішення матричної гри. Якщо рішення в чистих стратегіях було отримано, то чиста оптимальна стратегія експерта з кіберзахисту буде рекомендована до застосування. Якщо гра має рішення в змішаних стратегіях, стратегії експерта з кіберзахисту рекомендуються до застосування з пріоритетом, що визначається шляхом сортування ймовірності їх застосування. Рекомендації виводяться за спаданням пріоритетності застосування тих чи інших стратегій захисту корпоративної комп'ютерної мережі.

Акторами системи виступають експерт з кібербезпеки та Chief Information Security Officer – CISO. Експерт з кібербезпеки відповідає за заповнення бази знань, а CISO отримує та аналізує звіти та відповідає за прийняття рішень щодо реагування на наявні та можливі загрози.

Система аналізу загроз включає в себе:

– методи ідентифікації шкідливого ПЗ для файлів Linux ELF та Windows Portable Executable;

– метод виявлення мережевих аномалій із використанням Isolation Forest та EWMA статистики;

– модуль передбачення загроз із використанням мереж Баєса.

Список загроз оновлюється в реальному часі методами виявлення та прогнозування загроз. При ініціалізації системи експерти оцінюють задану множину загроз та визначають задану множину стратегій захисту. Під час роботи системи може виникати можливість коригування експертних знань відповідно до зміни модулів виявлення та прогнозування атак. Рушій висновків експертної системи включає в себе матрицю гри, сформовану шляхом поєднання інформації

про виявлені та прогнозовані кіберзагрози та експертних знань з рівня CVSS для загроз та стратегій протидії загрозам.

Висновки до розділу 2

Результатом моделювання та розробки методів виявлення та прогнозування загроз для корпоративних комп'ютерних мереж є:

1) Розроблений метод виявлення загроз з використанням EWMA-статистики та Isolation Forest. Застосування даного методу дозволяє виявляти як контекстні так і точкові аномалії. Дієвість методу експериментально перевірено та підтверджено в ході симуляцій декількох послідовних DDoS атак.
2) Запропоновано метод визначення секції Linux ELF файлу UNIX-подібних операційних систем, що містить процес семантичного аналізу та ідентифікації шкідливих Linux ELF фалів. Метод полягає у вибору секції файлу та розміру n-gram, що дає найвищу точність класифікації. Метод протестовано з використанням набору шкідливих та безпечних файлів розміром 6804 файлів та отримано найкращу точність 98% та F1-score 0.98 для моделі XGBoost.
3) Розроблений метод ідентифікації шкідливих Windows PE файлів з використанням секції Import Table та методу word2vec. Векторизовані дані передаються для класифікації моделям опорних векторів, ансамблю дерев рішень та багатошаровому перцептрону. Виконано тестування методу на наборі даних шкідливих та нешкідливих файлів розміром 16862 файли, що був розділений на тренувальний набір та тестувальний в пропорції 80% та 20%. В результаті найкращу точність, F1-міру та час класифікації показала модель ансамблю дерев рішень. Отримані результати були порівняні з існуючими дослідженнями класифікації Windows PE файлів. Порівняльний аналіз виявив покращення показників F1-міри з використанням

запропонованого методу у порівнянні з існуючими дослідженими методами.

4) Запропонований метод прогнозування загроз з використанням мережі Баєса дозволяє прогнозувати загрози спираючись на атрибути мережевого трафіку. Точність методу була експериментально досліджена на наборі даних, що містить 562767 агрегованих спостережень за мережевим трафіком, 2-ма типами загроз та нормальним трафіком. Отримали 99% точності та 0.99 F1-score, що є високим показником.
5) Розроблена модель комплексної інформаційної технології виявлення та прогнозування загроз для корпоративної комп'ютерної мережі з використанням засобів експертних систем. Інформаційна технологія використовує модулі виявлення та прогнозування загроз з пунктів 2.1-2.4 для наповнення бази знань. Експерти вносять оцінки критичності загроз за методикою CVSS та протидії загрозам. У результаті застосування моделі Теорії Ігор, система визначає прогнозовані ймовірності експлуатації виявлених загроз зловмисником та пріоритети застосування стратегій протидій загрозам.

Результати досліджень, приведених в розділі, опубліковані в роботах [74, 134, 142].

РОЗДІЛ 3
ФУНКЦІОНАЛЬНА МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВИЯВЛЕННЯ ТА ПРОГНОЗУВАННЯ КІБЕРЗАГРОЗ ДЛЯ КОРПОРАТИВНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

3.1 Загальна функціональна модель

В ході дослідження розроблено функціональну модель процесу «Виявлення та прогнозування кіберзагроз засобами експертних систем». Дана модель описує комплексну інформаційну технологію виявлення та прогнозування загроз для корпоративної комп'ютерної мережі з використанням експертних оцінок. Розроблена функціональна модель зображена на рисунку 3.1.

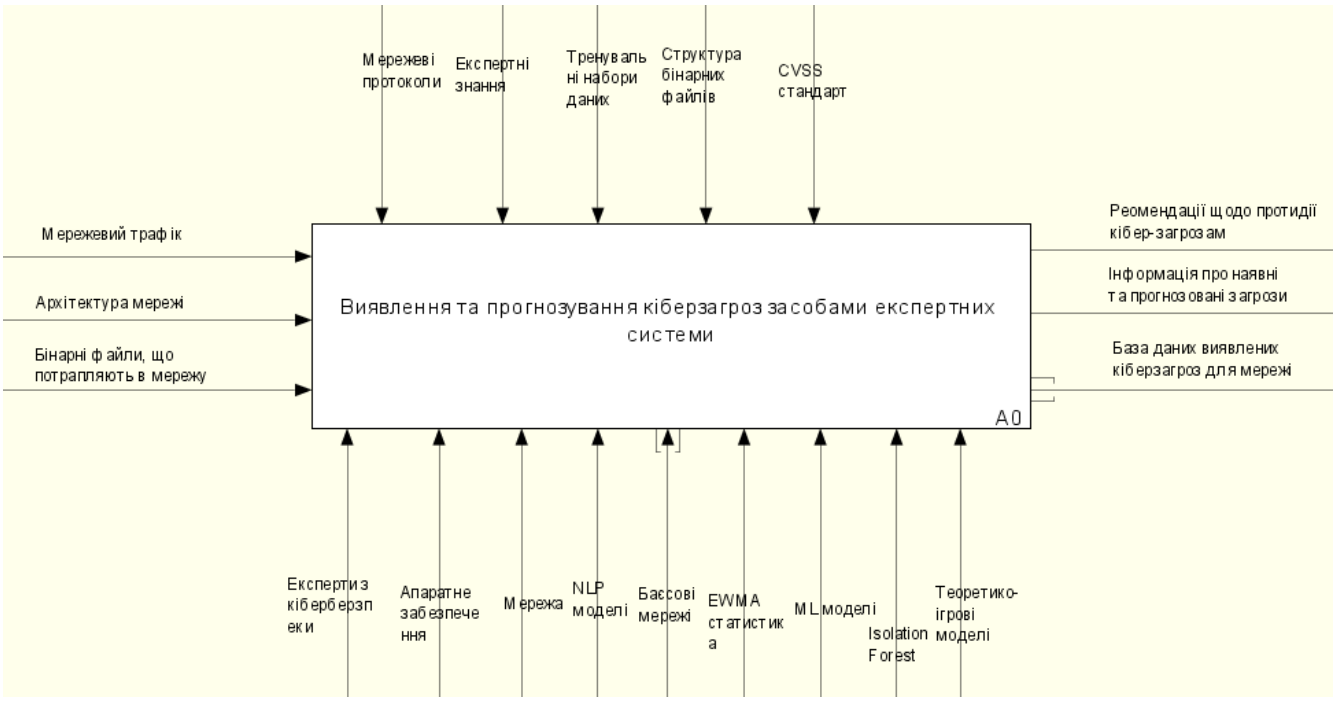


Рисунок 3.1 – Функціональна модель процесу «Виявлення та прогнозування кіберзагроз засобами експертних систем»

Вхідною інформацією для процесу виводу експертної системи визначено мережевий трафік, конфігурацію мережі та бінарні файли, що потрапляють в мережу. Керуючими елементами виступають: експертні знання спеціалістів з

кібербезпеки, що подаються на вхід рушію висновків експертної системи; мережеві протоколи - використовуються для визначення атак та аномалій трафіку; тренувальні набори даних – використовуються для тренування моделей машинного навчання та статистичних моделей для ідентифікації шкідливих бінарних файлів та прогнозування аномалій мережевого трафіку; структура бінарних файлів слугує для вхідною інформацією для проведення класифікації та виявлення зловмисного ПЗ; CVSS стандарт слугує для оцінки рівня критичності виявлених загроз та побудови ігрової матриці, що подається на вхід рушію висновків експертної системи. До механізмів керування відносяться експерти з кібербезпеки, апаратне забезпечення, мережа, NLP моделі, EWMA-статистика, Isolation Forest, ML моделі, Баєсові мережі та теоретико-ігрові моделі. На виході отримуємо інформацію про виявлені та прогнозовані кіберзагрози, рекомендації з протидії кіберзагрозам та базу даних виявлених загроз для мережі.

3.1.1 Вхідна інформація

Мережевий трафік є найбільшим джерелом кіберзагроз для комп’ютерної мережі. Необроблений мережевий трафік подається на вхід до модулів виявлення та прогнозування загроз. Модулі виконують декомпозицію мережевих пакетів, та виділяють параметри, необхідні для методів виявлення та прогнозування загроз, запропонованих в 2 розділі даного дисертаційного дослідження. Виявлення загроз відбувається шляхом ідентифікації шкідливих бінарних файлів та статистичного аналізу трафіку, зокрема виявлення аномалій його кількісних компонент, таких як кількість вхідних пакетів або співвідношення кількості вихідних до кількості вхідних пакетів.

Архітектура мережі є одним з вхідних компонентів, що використовується як для виявлення, так і для генерації рекомендацій з протидії загроз. Зокрема, в залежності від архітектури мережі, визначається кількість та конфігурація підмереж та місце розміщення розробленої інформаційної системи. Також

виконано розгортання API для ідентифікації шкідливих файлів, для роботи якого необхідно буде знати інформацію про архітектуру мережі та виконати налаштування файерволу для забезпечення можливості ПК кожної з підмереж комунікувати з API. Володіючи інформацією про адреси певних ПК в мережі та приналежність їх до певних підмереж, можна визначати чи коригувати експертні оцінки та рекомендації з протидії виявленим загрозам.

3.1.2 Керуючі елементи

Базовим керуючим елементом методу виявлення та прогнозування загроз є мережеві протоколи. Для кожної з вразливостей зловмисники використовують окремі конкретні протоколи, тому знання протоколів є основою для виявлення загроз. Оскільки технологія, що розроблюється, здатний працювати з обмеженою множиною загроз, то відповідні обмеження накладаються й на протоколи, з якими буде працювати технологія.

Тренувальні набори даних є керуючим елементом, що використовується моделями машинного навчання та статистичними моделями для попереднього тренування та валідації отриманих результатів. Такі набори даних представлені датасетами шкідливих та безпечних бінарних файлів та датасетами нормального та аномального мережевого трафіку.

Окремим керуючим елементом є структура бінарних файлів. Розроблювана система здатна виявляти шкідливі Linux ELF та Windows PE файли за допомогою моделей машинного навчання. У якості вхідних даних для виявлення подаються різні програмні секції файлів, що векторизуються за допомогою NLP технік та моделей та подаються на вхід до класифікатору.

Також у якості керуючого елементу використовується CVSS стандарт,що дозволяє експертам з кібербезпеки кількісно оцінити критичність загрози, яку несе та чи інша кібератака.

3.1.3 Елементи і механізми виконання

Експерти з кібербезпеки в розроблюваній системі представляють собою людей, що мають знання в предметній області кіберзахисту та оцінюють ризики тих чи інших кібератак з застосуванням кількісної метрики CVSS. Також експерти задають набір протидій атакам, застосовуючи які, рівень CVSS може бути зменшено або нейтралізовано.

Апаратне забезепечення включає в собі компонент корпоративної мережі, на якому розміщується інформаційна система. Дане апаратне забезпечення представлене у вигляді окремого серверного обладнання. До апаратного забезпечення також відноситься web-сервер, який представляє інтерфейс експертної системи та серверне обладнання для тренування ML моделей.

Мережа є сукупністю елементів та механізмів, що забезпечують передачу та отримання трафіку. Вхідна інформація для системи виявлення та прогнозування загроз формується виходячи зі стану мережі та мережевого трафіку. Також для формування рекомендацій експертної системи важлива інформація про кожен вузол та загальну архітектуру мережі, що дає основу для конкретизації рекомендацій щодо виявлення та протидії загрозам.

NLP моделі використовуються для векторизації та класифікації байткодів та окремих секцій бінарних файлів, що дозволяє отримати висновок про наявність загроз в тому чи іншому файлі та інформацію про тип вірусу в файлі, якщо він присутній.

EWMA-статистика та Isolation Forest модель дозволяють виявляти DoS та DDoS атаки. Інформація про виявлені атаки передається на вхід інформаційної системи, де експерти оцінюють їх за шкалою CVSS та встановлюють можливі протидії.

Мережа Баєса здійснює прогнозування загроз відповідно до дискретизованих значень параметрів трафіку. ML моделі використовуються для ідентифікації шкідливих бінарних Windows PE та Linux ELF файлів. Результати

виявлення загроз передаються на вхід до інформаційної системи, де проходять експертну оцінку та забезпечують формування рекомендацій з протидії загрозам.

Теоретико-ігрові моделі використовуються у якості рушію висновків експертної системи. Базуючись на виявлених та прогнозованих загрозах, а також на експертних знаннях про ці загрози, формується матрична антагоністична стратегічна гра, гравцями якої виступають кіберзлочинець та експерт з кібербезпеки. Вирішуючи гру та знаходячи оптимальні стратегії, експертна система формулює звіт з доцільності застосування тої чи іншої стратегії експертом з кібербезпеки та рекомендації щодо пріоритетності застосування стратегій з кіберзахисту.

3.1.4 Вихідна інформація

Інформація щодо виявлених та прогнозованих кіберзагроз повертається системою у вигляді звіту з деталізацією про час виявлення, ймовірності виникнення та інформацією про вузли мережі, для яких виявлено або прогнозовано кіберзагрози. Дана інформація базується на результатах роботи модулей з виявлення та прогнозування загроз. Вона також може бути використана системним адміністратором або працівниками департаменту з кіберзахисту у якості підкріплення прийняття рішень щодо протидії загрозам.

Рекомендації щодо протидії кіберзагрозам виводяться системою у вигляді звіту з відсортованими за пріоритетом застосування стратегій з протидії загрозам. Звіт формується на основі висновків про оптимальні стратегії протидії, отриманих рушієм висновків експертної системи. Експертна система використовує знання експертів та результати модулів виявлення і прогнозування кіберзагроз як базу знань, а теоретико-ігрові моделі у якості математичного апарату рушію висновків. Проаналізувавши дану інформацію, системний адміністратор або працівники департаменту з кіберзахисту можуть приймати рішення щодо застосування рекомендованих стратегій.

3.2 Декомпозиція загальної функціональної моделі

Для більш детального огляду функціональної моделі виконано її декомпозицію, що зображена на рисунку 3.2 .

В результаті декомпозиції процесу виявлення та прогнозування загроз з використанням експертних систем, виділено 6 робіт:

1. Розбір мережевого трафіку;
2. Виявлення та передбачення загроз;
3. Заповнення бази знань експертами;
4. Формування стратегічної матричної гри;
5. Вирішення матричної гри;
6. Формування звітів з кіберзахисту.

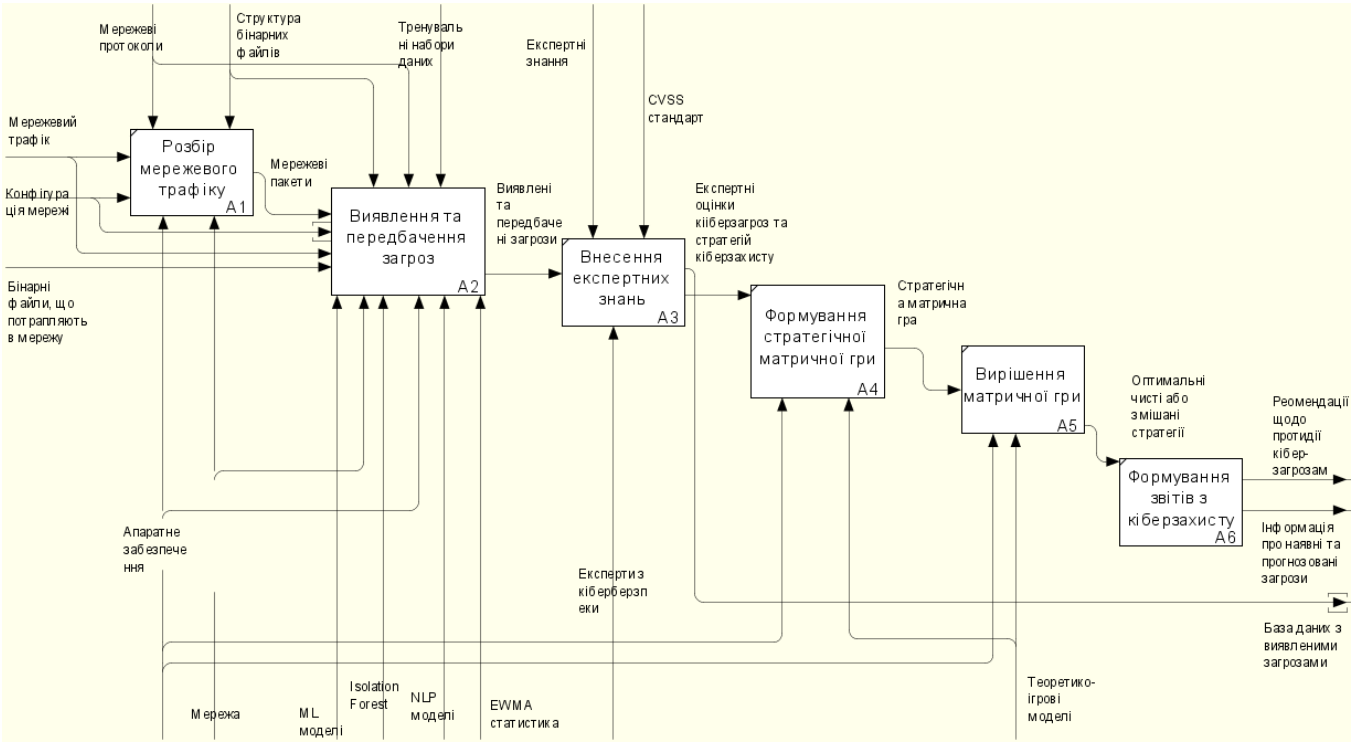


Рисунок 3.2 – Декомпозиція функціональної моделі «Виявлення та прогнозування кіберзагроз з використанням експертних системи»

Першою роботою процесу виявлення та прогнозування кіберзагроз та мережевих аномалій є розбір мережевого трафіку. Для виявлення мережевих аномалій використовуються часові ряди, сформовані з кількісних показників мережевого трафіку, таких як кількість отриманих та відправлених пакетів.

В представленій інформаційній технології для виявлення DoS та DDoS атак використовується кількість вхідних та вихідних пакетів, а для передбачення загроз з використанням мережі Баєса, обрано такі параметри як максимальний розмір вихідних пакетів, кількість вхідних пакетів за секунду, середній розмір вхідних пакетів, кількість пакетів за секунду. Парсер мережевого трафіку розміщується на роутері, що слугує вхідною точкою для всього мережевого трафіку. В результаті виконання даної роботи отримуємо корисні дані мережевого трафіку, що використовуються для виконання наступної роботи – виявлення та передбачення кіберзагроз.

Отримавши необхідні елементи мережевого трафіку, система виконує виявлення мережевих загроз. Процес виявлення мережевих загроз відбувається в реальному часі та без втручання людини. Однак, даний процес вимагає попереднього тренування та тестування моделей машинного навчання, побудови мережі Баєса та використанні статистичних моделей. Натреновані моделі для класифікації трафіку та виявлення аномалій розгортаються на окремому сервері, разом з REST API сервісом для класифікації та збору інформації про шкідливе ПЗ, експертною системою та користувацьким інтерфейсом. Виявлені кіберзагрози та аномалії передаються на вхід до роботи «Внесення експертних знань».

Отримавши набір виявлених та передбачених загроз, відбувається заповнення бази знань експертної системи генерації звітів з виявлених та прогнозованих загроз та рекомендацій щодо протидії наявним загрозам. Для оцінки рівня критичності кіберзагроз використовується метрика CVSS. Механізмом виконання даного процесу є експерти з кібербезпеки – саме вони

визначають значення CVSS для кожної загрози та можливі протидії. Зібравши необхідну експертну інформацію, система створює або оновлює платіжну матрицю гри, що використовується для визначення оптимальних стратегій кіберзлочинця та спеціаліста з кіберзахисту. Елементами платіжної матриці гри є значенням CVSS для відповідної стратегії загрози та протидії загрозі.

Вирішення матричної гри дозволяє отримати чисті або змішані оптимальні стратегії для кіберзлочинця та спеціаліста з кіберзахисту. Механізмом виконання даного процесу є теоретико-ігрові моделі, а саме ітераційний метод фіктивного розігрування стратегічних ігор.

Отримавши вирішення матричної гри в чистих або змішаних стратегіях, експертна система може сформулювати рекомендації з кіберзахисту. Також система виводить звіти по виявлених та прогнозованих загрозах на даний момент часу. При кожному виявлені нових загроз, система обчислює матричну гру та оновлює рекомендації та звіти.

Оскільки процес виявлення та передбачення кіберзагроз складається з декількох окремих робіт, доцільно виконати декомпозицію даного процесу. Декомпозиція процесу «Виявлення та передбачення кіберзагроз» представлена на рисунку 3.3.

Для передбачення загроз на основі мережевого трафіку використовуємо Мережу Баєса, натреновану на датасеті мережевого трафіку з наявними кіберзагрозами та безпечним трафіком. Керуючим елементом даного процесу є датасет мережевого трафіку. Механізмами виконання є апаратне забезпечення у вигляді серверного обладнання для тренування та розгортання моделі, а також Мережа Баєса.

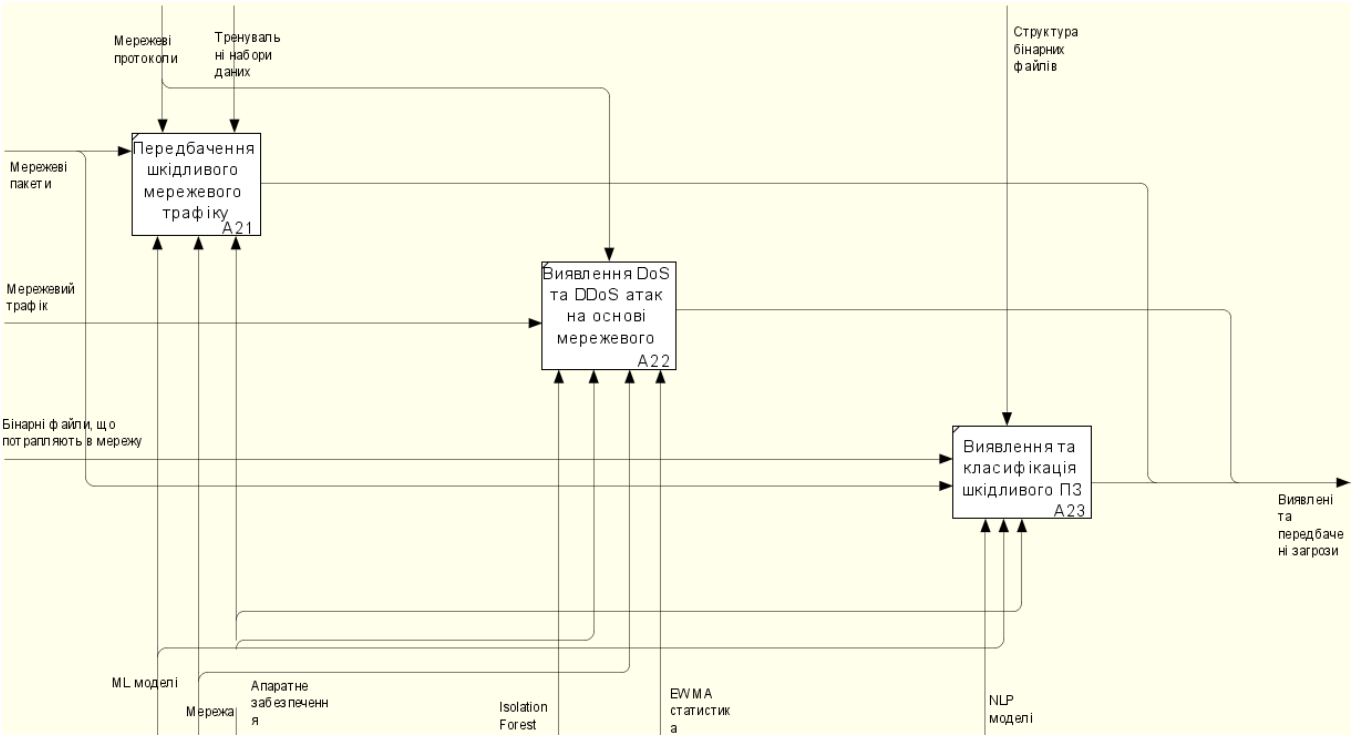


Рисунок 3.3 – Декомпозиція процесу «Виявлення та передбачення кіберзагроз»

Для виявлення аномалій мережевого трафіку, а саме DoS та DDoS атак використовується EWMA-статистика та модель Isolation Forest. Вхідними даними для цього процесу є мережевий трафік, керуючим елементом є мережеві протоколи.

Ідентифікація шкідливого ПЗ відбувається на основі класифікації бінарних файлів з використанням алгоритмів та методів, розроблених у розділі 2, пунктах 2.2 та 2.3 даного дослідження. На кожен з вузлів мережі встановлюється програма «агент», яка збирає інформацію про файли та надсилає їх на сканування розробленою інформаційною системою за допомогою запитів до REST API.

На виході процесу отримуємо інформацію про виявлені та передбачені кіберзагрози. Дана інформація зберігається до бази даних розробленої інформаційної системи, та, в подальшому, буде використана для формування звітів з кібербезпеки корпоративної комп’ютерної мережі.

3.3 Варіанти використання системи

Для технічної реалізації експертної системи з виявлення та прогнозування загроз необхідно визначити вимоги до її функціоналу а також варіанти її використання. Розроблена експертна система повинна реалізувати наступний функціонал:

1. Можливість виявлення у реальному часі кіберзагроз для корпоративних комп’ютерних мереж, що включають в себе:
   - потенційно шкідливий мережевий трафік, що може свідчити про здійснення кібератаки, такої як DoS або DDoS атака, тощо;
   - потенційно шкідливе ПЗ, що являє собою виконувані файли для операційних систем Windows та Linux;
2. Можливість прогнозування загроз мережевого трафіку з використанням мережі Баєса.
3. Можливість оцінки рівня CVSS для виявлених або прогнозованих кіберзагроз та внесення можливих протидій кіберзагрозам експертами з кібербезпеки.
4. Відображення звітів по виявлених та передбачених загрозах.
5. Відображення інформації про стратегії протидії наявним та передбаченим загрозам, відсортованих за пріоритетністю їх застосування.

Діаграма варіантів використання експертної системи з виявлення та прогнозування загроз для корпоративної комп’ютерної мережі представлена на рисунку 3.4 .

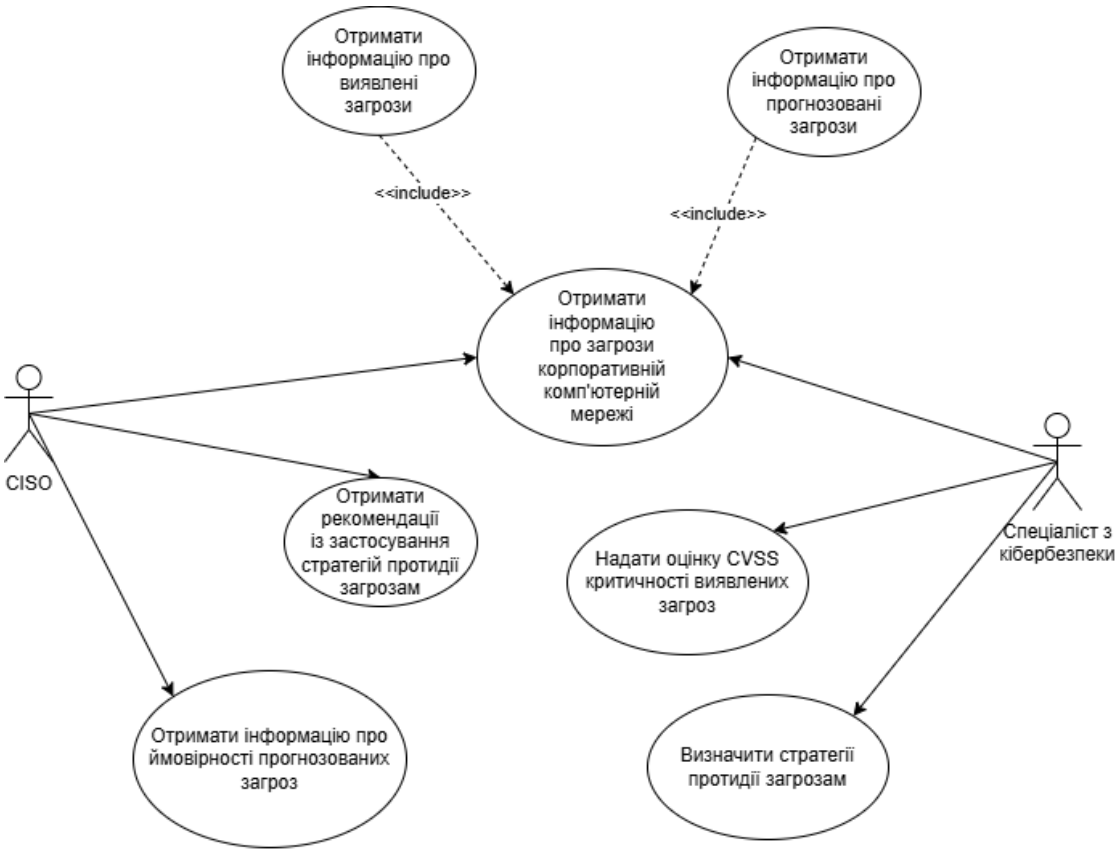


Рисунок 3.4 – Діаграма варіантів використання експертної системи з виявлення та прогнозування загроз для корпоративної комп’ютерної мережі

Кінцевими користувачами системи виступають голова відділу інформаційної безпеки (Chief Information Security Officer або CISO) та експерт з кібербезпеки. CISO відповідає за розробку опрацювання стратегій з забезпечення захисту корпоративної комп’ютерної мережі. Експерт з кібербезпеки оцінює наявні та виявлені кіберзагрози, вносить інформацію про можливі стратегії кіберзахисту. Відповідно до вимог, поставлених до функціоналу експертної системи, були визначені сценарії використання системи:

1. Перегляд експертом з кібербезпеки інформації про загрози корпоративній комп’ютерній мережі.
2. Внесення експертом з кібербезпеки оцінок рівня CVSS кіберзагроз.
3. Внесення експертом з кібербезпеки можливих стратегій протидії загрозам.

4. Перегляд CISO кібербезпеки інформації про загрози корпоративній комп’ютерній мережі.
5. Перегляд CISO інформації про прогнозовані загрози корпоративній комп’ютерній мережі
6. Перегляд CISO рекомендацій з застосування протидій загрозам.

3.4 Діаграма бази даних

Для запропонованої інформаційної системи будуть використовуватись дві окремі бази даних: реляційна база даних для розробленої інформаційної системи, що містить інформацію про кіберзагрози та оцінки CVSS та база даних часових рядів, в якій зберігаються часові ряди мережевого трафіку. База даних часових рядів має лише одну таблицю, яка містить часові спостереження мережевого трафіку та не має інших таблиць та зв’язків. База даних для розробленої інформаційної системи містить декілька таблиць та зв’язки між ними. Тому визначимо доцільним завданням виконати проектування реляційної бази даних запропонованої інформаційної системи.

Для проектування бази даних запропонованої інформаційної системи, визначимо сутності та зв’язки, необхідні для нашої системи. Скористаємось моделлю кіберзагрози, що була запропонована нами у розділі 1 пункті 1.1.1 (рисунок 1.1):

1. Сутність «Вектор вразливості». Дана сутність узагальнює всі можливі вектори вразливостей, що є загрозами для корпоративної комп’ютерної мережі та які здатна виявляти система. Вона містить інформацію про назву вразливості, час виявлення, опис, ip адресу вузла мережі, для якого виявлена вразливість, ip адресу джерела вразливості, додаткову інформацію (наприклад, назву шкідливого файлу, тощо). Для визначення можливих стратегій протидії кіберзагрозі, дана сутність

повинна мати зв’язок “many-to-many” з сутністю «Протидія кіберзагрозі».

2. Сутність «Вразливість». Дана сутність має ідентифікатор вразливості, рівень критичності вразливості за значенням метрики CVSS, та індикатор протидії вразливості.
3. Сутність «Протидія вразливості». Дана сутність визначає протидію виявленим вразливостям системи та містить в собі назву протидії та ідентифікатор протидії.

Розроблена діаграма «сутність-зв’язок» бази даних інформаційної системи з виявлення та передбачення кіберзагроз для корпоративної комп’ютерної мережі відображена на рисунку 3.5.

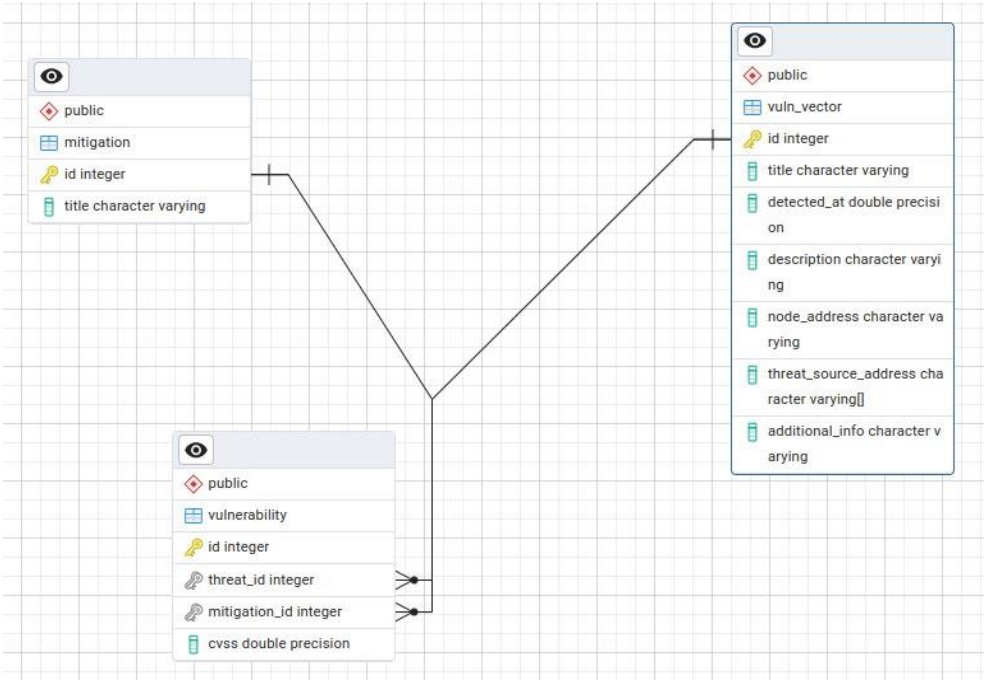


Рисунок 3.5 – Діаграма «сутність-зв'язок» бази даних інформаційної системи виявлення та прогнозування кіберзагроз для корпоративної комп’ютерної мережі

Як бачимо, система має досить просту структуру бази даних, проте дана структура дозволяє зберігати всю необхідну інформацію для функціонування

системи та використовувати технологію ORM для представлення сутностей бази даних в програмному коді.

Висновки до розділу 3

В результаті проектування інформаційної системи виявлення та прогнозування загроз для корпоративних комп’ютерних мереж, визначено основні вимоги до функціональної моделі, варіантів використання системи та бази даних.

1) Створена загальна функціональна модель інформаційної системи виявлення та прогнозування кіберзагроз для корпоративних комп’ютерних мереж з деталізацією вхідної інформації, керуючих елементів, елементів та механізмів виконання та вихідна інформація.
2) В результаті декомпозиції загальної функціональної моделі виділено 6 основних робіт, а також деталізовано процес виявлення та передбачення кіберзагроз.
3) В результаті моделювання створеної системи з використанням Use-Case діаграми виділено основні варіанти використання та основний функціонал.
4) Спроектовано діаграму баз даних, що задовольняє функціональним вимогам системи та дозволяє використовувати технології ORM для програмної розробки системи.

Результати досліджень, приведених в розділі, опубліковані в роботі [75].

РОЗДІЛ 4
ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ВИЯВЛЕННЯ ТА ПРОГНОЗУВАННЯ ЗАГРОЗ ДЛЯ КОРПОРАТИВНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

4.1 Проектування корпоративної комп'ютерної мережі

Корпоративні комп'ютерні мережі можуть мати різну архітектуру, в залежності від потреби організацій, що володіють такими мережами. Зазвичай при проектуванні архітектури корпоративної комп'ютерної мережі розглядають 2 підходи - "top-down" та "bottom-up" [135]. Обидва підходи використовують за основу модель OSI при проектуванні комп'ютерної мережі. Підхід "top-down" полягає у розробці архітектури мережі, починаючи з верхнього шару моделі OSI – прикладного рівня, спускаючись до нижнього фізичного рівня. Ключовою особливістю цього підходу є розробка архітектури мережі, відштовхуючись від бізнес-цілей організації, а не від наявних фізичних пристроїв чи технологій. Спершу враховуються потреби та вимоги організації до бізнес-логіки, такі як: необхідні веб-сервіси, застосунки, механізми авторизації, типи пристроїв в мережі тощо. На цьому етапі також виконується вибір та застосування мережевих екранів та систем виявлення вторгнень. Наступним кроком є аналіз транспортного рівня, що зв'язує мережевий рівень з прикладним рівнем. На даному рівні визначаються протоколи обміну інформацією між сервісами та пристроями мережі, такі як TCP та UDP, здійснюється контроль за помилками обміну інформацією. Далі виконується моделювання мережевого рівня. При цьому визначаються механізми маршрутизації в мережі, створення сегментів мережі, підмереж, DMZ тощо. На цьому етапі робиться вибір роутерів та світчів, які здатні виконувати маршрутизацію між різними сегментами мережі. На даному рівні вирішальним аспектом є структура організації, кількість співробітників-користувачів мережі, наявність серверного обладнання. Наприклад, в залежності від структури компанії виділяється окрема підмережа

для кожного структурного підрозділу з метою полегшення управління та моніторингу мережі. Серверне обладнання, яке слугує для розміщення mail, dns, web серверів зазвичай роміщується в демілітаризованій зоні для ізоляції та обмеження доступу з локальної мережі. Останнім кроком є планування канального та фізичного рівня. На цьому етапі визначаються з використанням технології зв'язку, серед яких може бути PON – Passive Optical Network, Ethernet, WiFi, тощо. Відповідно до обраної технології зв'язку, використовується різне мережеве обладнання (модеми, світчі 2-го рівня тощо) та кабелі. На відміну від описаного підходу "top-down", підхід "bottom-up" зосереджений в першу чергу на виборі найбільш сучасних технологій та пристроїв для планування архітектури мережі, максимізуючи пропускну здатність та розроблюючи архітектуру мережі, в яку закладено не тільки поточні потреби організації, а й можливе розширення та масштабування таких потреб. Кожен з описаних методів має свої переваги та недоліки. Основною перевагою методу «top-down» є можливість економії грошових, обчислювальних та апаратних ресурсів. Недоліком даного підходу є підвищені часові витрати на планування та аналіз бізнес-цілей організації, а також обмежені можливості до масштабування в майбутньому. Перевагою підходу «bottom-up» є швидкість імплементації та закладений запас до масштабування бізнес-потреб організації. Недоліком «bottom-up» підходу є перевитрата грошових та апаратних ресурсів.

В результаті аналізу процесу планування та моделювання мережі, а також порівняння підходів до планування «top-down» та «bottom-up» зроблено декілька висновків. По-перше, було виділено основні компоненти мережі, які забезпечують її функціонування на кожному з рівнів OSI. Визначено, що сучасні корпоративні комп'ютерні мережі мають включати в себе такі основні компоненти: мережевий екран, DMZ, роутери та світчі, системи виявлення вторгнень, механізми сегментації мережі та обладнання канального рівня. По-друге, для проектування архітектури корпоративної мережі з метою проведення

експериментальних досліджень алгоритмів та методів, розроблених у розділі 2 даного дослідження, вирішено застосувати метод «top-down» для оптимізації використання обчислювальних та апаратних ресурсів, що дозволить виконати моделювання мережі у віртуальному середовищі. Наступним кроком є підбір компонентів та визначення архітектури мережі, що буде використовуватись для експериментальних досліджень.

У якості експериментального середовища вирішено змоделювати мережу для невеликої організації з двома підрозділами, 10 співробітниками та web-сервером, на якому розміщуються сайти WordPress, що має бути ізольованим від локальної мережі у зоні DMZ-1. При цьому, співробітники можуть використовувати операційні системи Windows або Linux на персональних комп'ютерах. Модулі ідентифікації загроз та експертної системи, запропоновані у 2-му розділі, вирішено виокремити в окрему підмережу, організувавши таким чином зону DMZ-2. Також виділено сутність зловмисника, що знаходиться поза корпоративною мережею, та користується операційною системою Kali Linux для здійснення атак. Таким чином, для виділення 2-х підмереж для підрозділів організації, 2-х DMZ зон для експертної системи та WordPress серверу, використано 4 світчі. Для з'єднання локальної мережі з глобальною мережею встановлений роутер. Роутери розділяють на типи в залежності від типу передачі даних – провідні або безпровідні, місцем встановлення – core або edge та за способом встановлення – фізичні або віртуальні. Провідні роутери забезпечують маршрутизацію пакетів, використовуючи технології Ethernet, зокрема Gigabit Ethernet, а безпровідні роутери забезпечують маршрутизацію на основі технології WiFi. В той час як технологія WiFi 7 номінально забезпечує більшу швидкість передачі даних, ніж Gigabit Ethernet, останній є більш стабільним та менш залежним від середовища та фізичних перепон, що більше підходить для корпоративних комп'ютерних мереж. Edge роутери встановлюються на межі комп'ютерної мережі та забезпечують зв'язок з глобальною мережею через

інтернет-провайдерів (ISP). Core роутери встановлюються всередині мережі та слугують з'єднувальним елементом всередині великих підприємств або кампусів. Фізичні роутери являють собою апаратне забезпечення, в той час як віртуальні роутери є програмним забезпеченням, яке може бути встановлене на різні апаратні платформи з використанням технології віртуалізації. Перевагою віртуальних роутерів є можливість горизонтального масштабування, оскільки вони не мають прив'язки до певної апаратної платформи та можуть бути адаптовані для використання в корпоративних мережах організацій різних розмірів.

Проаналізувавши наведені вище особливості різних типів роутерів та мережевих технологій, вирішено використовувати віртуальний провідний edge роутер для виконання експериментальних досліджень та технологію Ethernet, як таку, що забезпечує більш стабільне з'єднання.

Порівняльний аналіз віртуальних роутерів Cisco Catalyst 8000V Edge Software та pfSense показав, що роутер Cisco Catalyst 8000V Edge Software є платним продуктом, та вимагає придбання підписки Cisco DNA software. Даний роутер побудований на основі пропрієтарної операційної системи "Cisco IOS XE". Відповідно до сайту-виробника зазначається, що ротуер підтримує максимальну пропускну здатність до 40 Гбіт/с у разі локального встановлення з використанням мінімум 2-х ядерного CPU для звичайного функціонування та 16-ти ядерного CPU для використання технології IPsec. Також даний роутер містить в собі Next Generation Firewall, VPN, NAT [136].

Роутер PfSense є безкоштовним програмним забезпеченням з відкритим кодом. PfSense створений компанією NetGate, побудований на основі відкритої операційної системи FreeBSD. У документації роутера pfSense вказано пропускну здатність у 5.59 Гбіт/с при розмірі мережевого пакету у 1500 байт. Також вказано, що роутер може забезпечувати до 4000000 одночасних з'єднань при використанні приблизно 7800 Мб оперативної пам'яті та 500000 одночасних

з’єднань при обсягу RAM 976 Мб. Роутер pfSense може бути використаний також в якості мережевого екрану, для налаштування VPN, NAT, IPSec [137]. Перевагою pfSense є можливість встановлення додаткових відкритих програмних пакетів для збору мережевої інформації, виявлення загроз, наприклад Telegraf, Snort та ін.

Проаналізувавши специфікації кожного з роутерів, вирішено використовувати роутер PfSense, як такий, що є достатнім для експериментального середовища та може задовольняти потреби корпоративних комп’ютерних мереж, а також має можливість встановлення додаткових програмних модулів, що дозволяють збирати важливу мережеву інформацію для виявлення загроз з використанням методів, представлених у розділі 2. Для виявлення мережевих аномалій з використанням Isolation Forest та EWMA статистики для роутеру PfSense буде встановлено модуль Telegraf, який буде збирати та зберігати часові ряди мережевого трафіку у базі даних InfluxDB.

Для емуляції роботи корпоративної комп’ютерної мережі було використано програмне забезпечення GNS3. Програмне забезпечення GNS3 дозволяє створювати віртуальні середовища, що симулюють мережі різної складності з використанням компонентів, які розповсюджуються компаніями CISCO, IBM та ін., так і open-source компонентів, які створені та підтримуються спільнотою розробників. До переваг середовища GNS3 можна віднести простоту налаштування мережевої маршрутизації, віртуалізацію всіх ресурсів та гнучкість налаштування. Також перевагою даного середовища є можливість конфігурації різних типів вузлів на основі Docker контейнерів, що значно розширює види доступних вузлів. Серед недоліків середовища є його ресурсоємність.

Для проведення експерименту було створено корпоративну комп’ютерну мережу, що складається з 10-ти персональних комп’ютерів, 4-х пристроїв switch, 1 роутера PfSense, окремо виділеного серверу на базі ОС Linux для інформаційної системи, розробленої в розділі 2 та серверу WordPress. Корпоративна

комп'ютерна мережа підключена до глобальної мережі інтернет – WAN. Для симуляції кіберзагроз та кібератак до IPS роутера мережі WAN був під'єднаний персональний комп'ютер на базі ОС Kali Linux, з якого буде здійснюватись атаки на корпоративну комп'ютерну мережу. Структура мережі наведена на рисунку 4.1

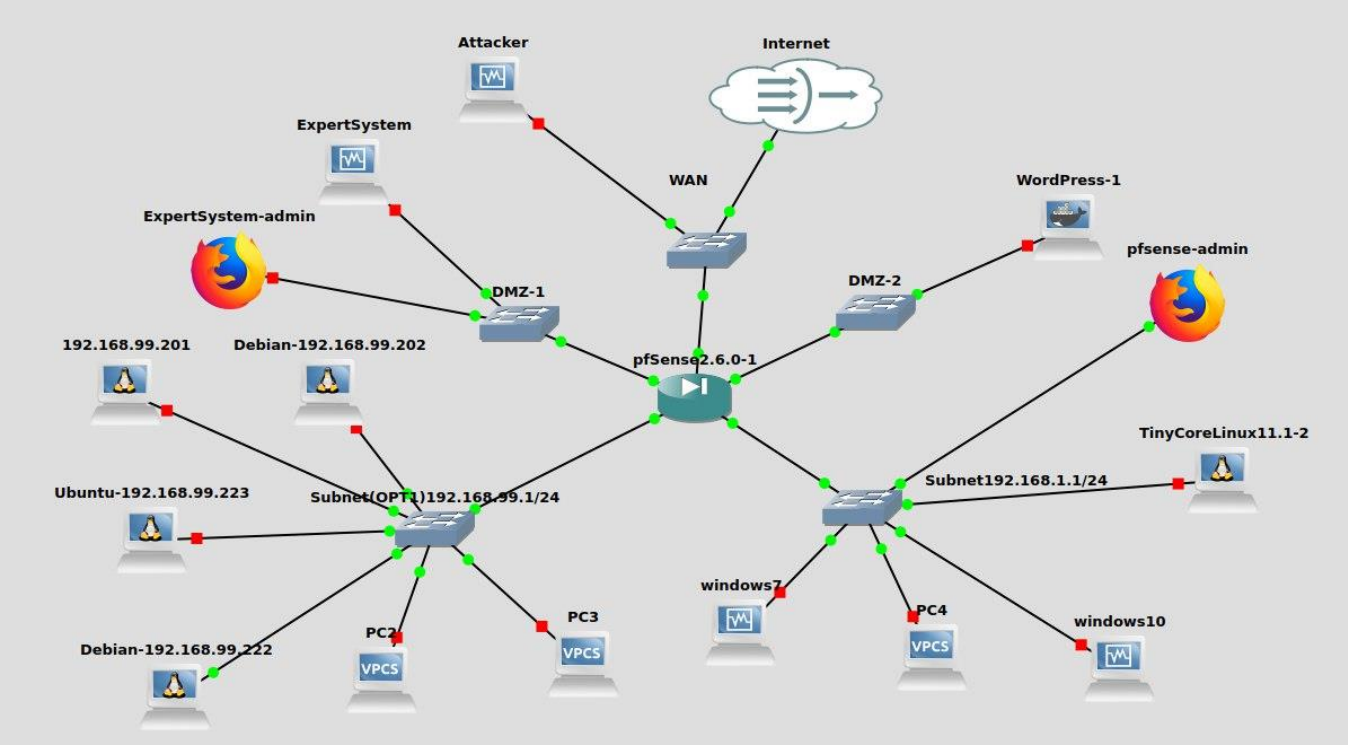


Рисунок 4.1 – Корпоративна комп'ютерна мережа створена в середовищі GNS3 для експериментального аналізу створеної інформаційної системи

Всі експерименти були проведені на ПК, що має наступні технічні характеристики: ОС Ubuntu 22.04.5 LTS 64-bit, ОЗП 16 ГіБ, ЦП Intel Core i7-1065G7, графічний адаптер Mesa Intel Iris(R) Plus Graphics (ICL GT2).

4.2 Проведення експериментів з виявлення загроз у ELF та PE файлах

Для оцінки ефективності виявлення шкідливих файлів за допомогою розробленої інформаційної системи, на вузлі «Expert System» було запущено REST API сервіс, що приймає POST запити на класифікацію файлів та повертає мітку-клас файлу. Для класифікації PE файлів, модель Ансамблю дерев рішень та word2vec була натренована та перевірена на тренувальному датасеті, після

чого збережена в бінарні файли з використанням бібліотеки pickle. При запуску API відбувається десеріалізація моделей у відповідні змінні в API, а у разі відсутності серіалізованих файлів, попередньо відбувається тренування моделі на тренувальному наборі даних. Після цього, при надходженні запиту на виявлення загрози в бінарному PE файлі, модель класифікує надісланий файл та повертає мітку класу файлу. Код для тренування, валідації та серіалізації моделі з використанням бібліотеки scikit-learn та pickle приведений у лістингу 4.1.

Лістинг 4.1 – Програмний код методу класифікації шкідливих Windows PE файлів

```
    import pandas as pd
    from sklearn.model_selection import train_test_split
    import gensim
    import nltk
    import pickle
    nltk.download('punkt')
    nltk.download('punkt_tab')
    from nltk.corpus import stopwords
    from gensim.models import Word2Vec
    from nltk.tokenize import word_tokenize
    import string
    from sklearn import metrics
    from sklearn.ensemble import RandomForestClassifier
    def preprocess_pe_import_table(text):
        if isinstance(text, float):
            text = str(text)
        text = text.lower()
        text = ''.join([word for word in text if word not in string.punctuation])
        tokens = word_tokenize(text)
        tokens = [word for word in tokens]
        return ' '.join(tokens)
    df = pd.read_csv("../train_parsed.csv").dropna()
    train, test = train_test_split(df, test_size=0.2)
    train_x = train['imports']
    train_y: pd.Series = train['type']
    test_x = test['imports']
    test_y: pd.Series = test['type']
    X_train = train_x.apply(preprocess_pe_import_table)
    X_test = test_x.apply(preprocess_pe_import_table)
    sentences = [sentence.split() for sentence in X_train]
    w2v_model = Word2Vec(sentences, hs=1, vector_size=100, window=5, min_count=1, workers=4, epochs=10, sg=1)
    X_train = np.array([vectorize(sentence, w2v_model) for sentence in X_train])
    X_test = np.array([vectorize(sentence, w2v_model) for sentence in X_test])
    model = RandomForestClassifier(n_estimators=400, min_samples_split=10, min_samples_leaf=1, max_features='sqrt', max_depth=60, bootstrap=False)
```

```
model.fit(X_train, train_y)
y_pred = model.predict(X_test)
print(metrics.classification_report(test_y,y_pred,target_names=np.unique(test_y)))
with open("../../w2v.pkl", "wb") as f:
    pickle.dump(w2v_model, f)
with open("../../r_forest.pkl", "wb") as f:
    pickle.dump(model, f)
```

Набір даних для оцінки ефективності підсистеми з класифікації шкідливих PE файлів містить загалом 1706 файлів, що представляють собою 2 набори по 853 файли. Кожен файл позначається типом вірусу або відсутністю вірусу. Типи вірусів розподілилися відповідно: для першого набору Trojan – 222, Worm – 176, Adware – 237, безпечні файли - 218; для другого набору: Trojan – 215, Worm – 232, Adware – 199, безпечні файли – 287. Для оцінки створеного модулю виявлення шкідливих PE файлів було виконано порівняння точності і швидкодії з вбудованим антивірусним ПЗ Windows Defender. Windows Defender представляє собою вбдоване антивірусне ПЗ, що може як забезпечувати виявлення загроз в реальному часі так і сканування на вимогу користувача [138]. Для сканування було виконано команду у командному рядку: `"C:\Program Files\Windows Defender\MpCmdRun.exe" -Scan -ScanType 3 -File file_path`. Для порівняння результатів виявлення шкідливих файлів, було розраховано точність та F1-міру виявлення загроз та середній час на сканування одного файлу.

Для проведення експериментів з виявлення шкідливих ELF файлів також було створено 2 набори файлів: для першого набору кількість файлів virus становила 62, безпечних файлів – 90, для другого кількість вірусів була 53, а безпечних файлів – 97. Нешкідливі файли були представлені більшою кількістю зразків у кожному наборі, що відповідає ситуації у реальному житті. Для оцінки створеного модулю з виявлення шкідливих ELF файлів було проведено порівняння ефективності його роботи з ефективністю антивірусного програмного продукту з відкритим кодом ClamAV [139], що призначений для використання з

UNIX-подібними системами. Для сканування шкідливих ELF фалів з використанням ClamAV була виконана команда: `clamscan --recursive --no-summary dir_path`. Код реалізації методу виявлення шкідливих ELF фалів з використанням моделі XGBoost, а також векторизацією секції rodata ELF файлу з використанням методів n-gram та TF-IDF, запропонованих у 2-му розділі даного дослідження, приведено на лістину 4.2.

Лістинг 4.2 – Програмний код методу класифікації шкідливих Linux ELF файлів

```
import pickle
from sklearn.feature_extraction.text import TfidfTransformer, CountVectorizer
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import pandas as pd
df = pd.read_csv("elf_virus_total.csv")
train_total, test_total = train_test_split(df, test_size=0.1)
train, test = train_test_split(train_total, test_size=0.2)
train_x = train["processed_opcodes_rodata"]
train_y: pd.Series = train["virus"]
test_x = test["processed_opcodes_rodata"]
test_y: pd.Series = test["virus"]
count_vect = CountVectorizer(ngram_range=(1, 1))
X_train_counts = count_vect.fit_transform(train_x)
X_test_counts = count_vect.transform(test_x)
tf_idf_vectorizer = TfidfTransformer()
X_train_tfidf = tf_idf_vectorizer.fit_transform(X_train_counts)
X_test_tfidf = tf_idf_vectorizer.transform(X_test_counts)
le = LabelEncoder()
train_y_nums = le.fit_transform(train_y)
test_y_nums = le.fit_transform(test_y)
model = XGBClassifier(learning_rate=0.1, max_depth=7)
model.fit(X_train_tfidf, train_y_nums, eval_set=[(X_train_tfidf, train_y_nums), (X_test_tfidf,
test_y_nums)])
predictions_xgb = model.predict(X_test_tfidf)
predicted = le.inverse_transform(predictions_xgb)
from sklearn.metrics import classification_report
print(classification_report(test_y, predicted))
with open("../../tfidf_vect.pkl", "wb") as tf:
    pickle.dump(tf_idf_vectorizer, tf)
with open("../../xgb_model.pkl", "wb") as f:
    pickle.dump(model, f)
with open("../../xgb_label_encoder.pkl", "wb") as le_f:
    pickle.dump(le, le_f)
with open("../../count_vect.pkl", "wb") as cv:
    pickle.dump(count_vect, cv)
```

Для кожного ПК в корпоративній комп'ютерній мережі було проведено експеримент з виявлення файлових загроз на окремому наборі файлів. Швидкість інтернету для експериментального середовища становила 100 Мбіт/с. На кожному ПК було запущено скрипт, що відслідковував додавання нових файлів у директоріях /tmp для ОС Linux та C:\Users\user\App Data\Local\Tmp для OS Windows, та робив запити до REST API на вузлі «Expert System» для визначення файлових загроз. У якості операційних систем було обрано дистрибутиви Linux Debian, Linux Ubuntu, Windows 7 та Windows 10. Усі показники продуктивності були розраховані як середнє арифметичне для запитів від кожного ПК, окремо для кожного типу файлів – ELF та PE файлів. Результати порівняння ефективності розроблених у 2-му розділі методів виявлення файлових загроз у Linux ELF та Windows PE файлах з існуючим антивірусним ПЗ Microsoft Defender та ClamAV в умовах корпоративної мережі змодельованої в середовищі GNS3 приведені у таблиці 4.1.

Таблиця 4.1 – Результати експерименту з виявлення шкідливих та безпечних Linux ELF та Windows PE файлів в умовах корпоративної мережі

	Ансамбль дерев рішень + word2vec, PE файли	Microsoft defender, PE файли	Модель XGBoost + TF-IDF, ELF файли	ClamAV, ELF файли
1	**2**	3	**4**	5
Середня точність класифікації файлів, %	**94**	93	**99**	96

Продовження таблиці 4.1

1	**2**	3	**4**	5
F1-score класифікації файлів	**0.94**	0.93	**0.98**	0.94
Середній час класифікації одного файлу, сек	**0.17**	9.58	**0.03**	1.07

Як бачимо, в результаті експериментального дослідження створених у 2-му розділі алгоритмів та методів виявлення та класифікації шкідливого ПЗ, запропоновані рішення мають конкурентну спроможність з існуючими антивірусними програмними продуктами. Так, метод виявлення файлових загроз для PE файлів на 2-х однакових експериментальних наборах даних показав покращену точність та F1-міру та набагато менший час, витрачений на класифікацію одного PE файлу, порівняно з антивірусним ПЗ Microsoft Defender. Алгоритм виявлення загроз в ELF файлах показав кращу точність порівняно з антивірусним ПЗ ClamAV, краще значення F1-міри та менший середній час класифікації одного файлу на 2-х однакових експериментальних наборах даних.

4.3 Проведення експериментів з виявлення DDoS атак

Для проведення експериментів з виявлення DDoS атак до мережі був доданий вузол з WordPress сервером, на якому було створено та розгорнуто сайт. З метою надання доступу до сайту з глобальної мережі Інтернет на мережевому екрані pfSense було налаштовано NAT Port Forwarding правило, згідно з яким запити з інтерфейсу WAN на порт 80 та 443 перенаправляються на локальну адресу WordPress серверу на відповідні порти. Після чого сайт став доступним з адреси інтерфейсу WAN корпоративної комп’ютерної мережі.

Для симуляції процесу DDoS атаки було використано утиліту metasploit, що входить до дистрибутиву Kali Linux. Для того, щоб дослідити роботу методів EWMA та Isolation Forest в залежності від шаблонів мережевого трафіку, атаки було проведено у двох режимах: атаки з однаковими та різними інтервалами. Періодичні атаки були розділені на часові проміжки по 30 секунд з відповідним інтервалом затримки, та були здійснені протягом 10 хвилин. Код для проведення атак з однаковим інтервалом приведено у лістингу 4.3, а результати на рисунку 4.2 . Інші атаки були проведені протягом однієї хвилини, з розділенням на інтервали різної довжини від 5 до 16 хвилин. Код для проведення атак з різним інтервалом приведено у лістингу 4.4, а результати на рисунку 4.3.

Лістинг 4.3 – Код для проведення DDoS атак з однаковим інтервалом

```
ddos.rc
set SessionExpirationTimeout 20
set SessionCommunicationTimeout 20
use auxiliary/dos/tcp/synflood
set RHOSTS 192.168.122.215
exploit
exit
run_ddos_experiment.sh
while true
do
echo "Start exploit"
sudo timeout 20s  msfconsole -r ddos.rc
echo "Done exploit"
sleep 30
done
```

Лістинг 4.4 – Код для проведення DDoS атак з різним інтервалом

```
run_ddos_experiment_periodic.sh
sudo timeout 60s  msfconsole -r ddos.rc
sleep 960
sudo timeout 60s  msfconsole -r ddos.rc
sleep 300
sudo timeout 60s  msfconsole -r ddos.rc
sleep 500
sudo timeout 60s  msfconsole -r ddos.rc
```

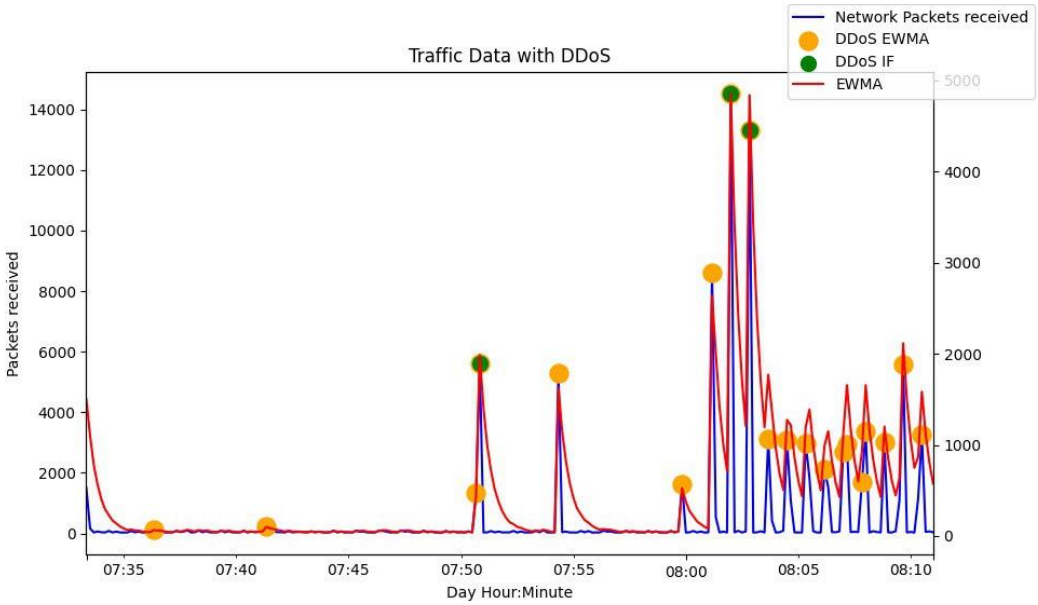


Рисунок 4.2 – Виявлення атаки з використанням параметру X_t при проведення DDoS атаки з однаковим інтервалом

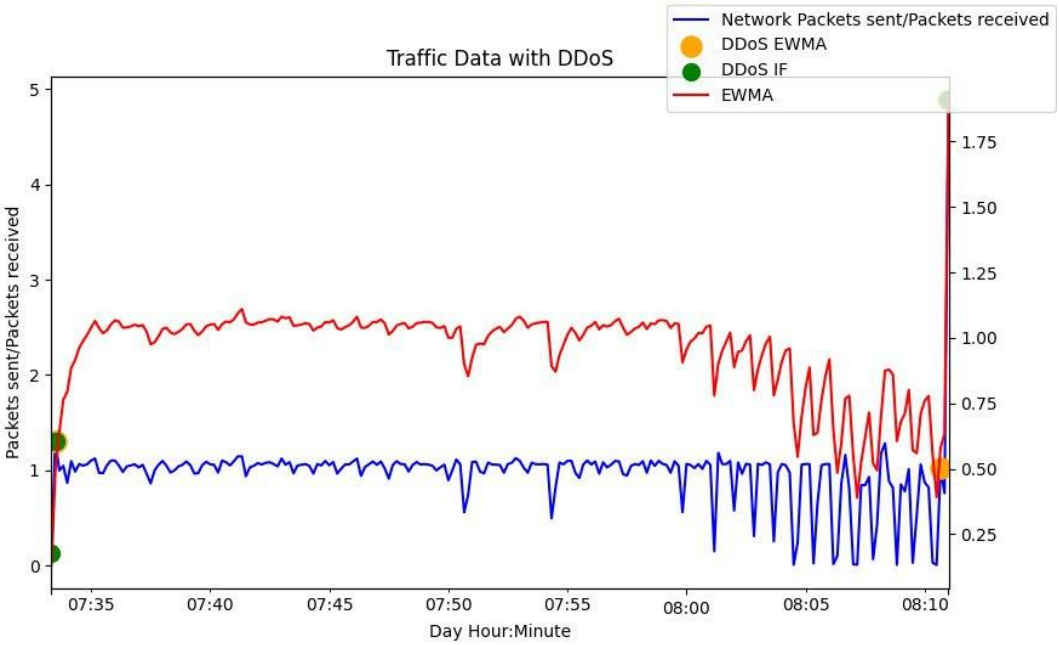


Рисунок 4.3 – Виявлення атаки з використанням параметру X_t^{ddos} при проведенні DDoS атаки з різним інтервалом

З результатів можна побачити, що при проведенні DDoS атаки з однаковим інтервалом, обидва алгоритма Isolation Forest та EWMA реагують на початок атаки, але при її продовженні алгоритм Isolation Forest перестає реагувати, адже

з урахуванням часового контексту вважає, що дана поведінка є періодичною та не є аномальною. В той час як алгоритм EWMA продовжує фіксувати точкові аномалії, тому що за проведений період спостережень параметрів трафіку – 50 хвилин, атака триває лише 10 хвилин, і значною мірою перевищує середньозважені значення.

При проведенні атаки з різним інтервалом, обидва алгоритми виявляють DDoS атаку, адже значення параметру значною мірою перевищують середньозважені та мають різні інтервали прояву. Таким чином, було виявлено що запропонований метод виявлення DDoS атак підходить як до виявлення атак з однаковим інтервалом, так і для виявлення атак з різним інтервалом і враховує часовий контекст спостережень мережевого трафіку.

4.4 Реалізація інформаційної системи виявлення та прогнозування загроз засобами експертних систем

В якості рушія висновків в модулі експертної оцінки використано ітераційний алгоритм фіктивного розігрування. Для оцінки збіжності ітераційного алгоритму фіктивного розігрування та визначення оптимальної кількості ітерацій гри було завантажено набір даних [140] з проаналізованими вразливостями за 2022 рік на базі звітів агенції CISA – Cybersecurity and Infrastructure Security Agency.

Набір даних містить список вразливостей, призначений їм рівень CVSS та рекомендації з протидії цим вразливостям. З набору даних були видалені дублюючі та домінуючі стратегії, в результаті чого отримали матрицю, що містить 42 загроз та 9 протидій. Частина матриці гри розмірністю 9 на 9 приведена в таблиці 4.2

Таблиця 4.2 – Частина платіжної матриці гри, сформованої на основі датасету агенції CISA

attack/defence	Apply updates per vendor instructions.	Update Adobe Acrobat and Reader or Delete Adobe Flash Player	Update or Disconnect device	Upgrade Apache Log4j to 2.15+	Upgrade Confluence server to 7.18.1	impacted product is end-of-life and should be disconnected	multiple impacted products are end-of-life and should be disconnected	patch D-Link KEV entry CVE-2018-6530	upgrade product from version 6 to 7
"sigred" - microsoft windows domain name system (dns) server remote code execution	**10.0**	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0
adobe acrobat and reader, flash player unspecified	-9.2	**9.2**	-9.2	-9.2	-9.2	-9.2	-9.2	-9.2	-9.2
adobe flash player and air integer overflow	-9.8	-9.8	-9.8	-9.8	-9.8	-9.8	**9.8**	-9.8	-9.8
adobe flash player arbitrary code execution	-9.8	-9.8	-9.8	-9.8	-9.8	**9.8**	-9.8	-9.8	-9.8
apache log4j2 remote code execution	-10.0	-10.0	-10.0	**10.0**	-10.0	-10.0	-10.0	-10.0	-10.0
atlassian confluence server and data center remote code execution	-9.2	-9.2	-9.2	-9.2	**9.2**	-9.2	-9.2	-9.2	-9.2
checkbox survey deserialization of untrusted data	-9.8	-9.8	-9.8	-9.8	-9.8	-9.8	-9.8	-9.8	**9.8**
d-link multiple routers os command injection	-9.2	-9.2	-9.2	-9.2	-9.2	-9.2	-9.2	**9.2**	-9.2
netgear multiple devices exposure of sensitive information	-9.2	-9.2	**9.2**	-9.2	-9.2	-9.2	-9.2	-9.2	-9.2

Проведений експеримент з дослідження збіжності середнього виграшу V, отриманого для кількості ітерацій від 10 до 3000 з кроком 10 – загалом 300 розіграшів. Отриманий результат зображений на рисунку 4.4. З графіку видно

побачити тенденцію до збіжності середнього значення виграшу гри. Після 1000 ітерацій можна відмітити зменшення коливання середнього значення V в межах від 5.8 до 6.0, що вказує на доцільність застосування показника в 1000 ітерацій для вирішення заданої матричної гри.

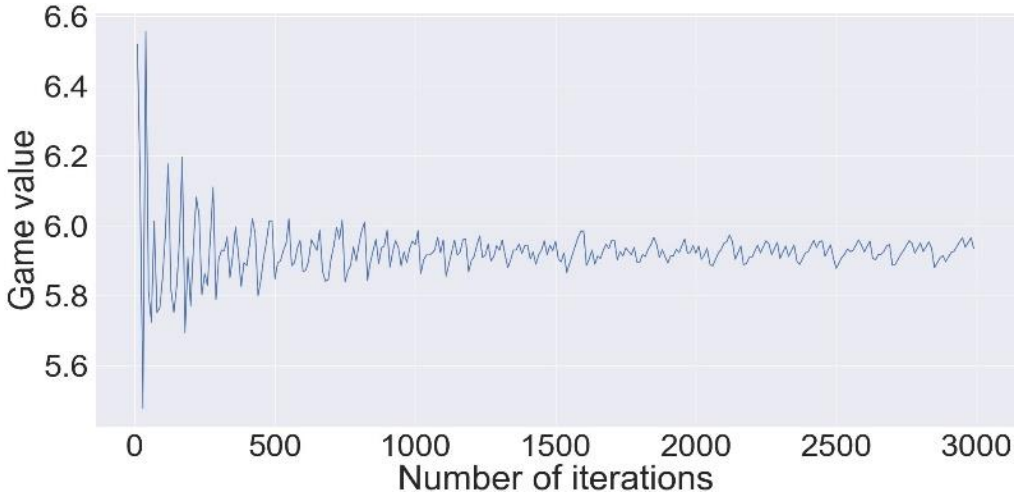


Рисунок 4.4 – Результати експериментів з дослідження збіжності середнього виграшу V в грі з використанням набору даних CISA від кількості ітерацій алгоритму фіктивного розігрування

У ході дослідження було створено MVP інформаційної системи виявлення та прогнозування загроз з використанням експертних оцінок, що представляє собою WEB додаток з 4 вкладками. На рисунках 4.5 – 4.8 представлено основні екрани web-додатку інформаційної системи.

Дана система побудована на основі методів та моделі, запропонованих в даному дисертаційному дослідженні. При розробці інформаційної системи застосована інформаційна технологія, запропонована у розділі 2.5 (рисунок 2.27). Система була інтегрована з корпоративною комп’ютерною мережею, що була спроектована у розділі 4.1. База знань була заповнена на основі експериментів, проведених раніше у даному розділі та експертних оцінок.

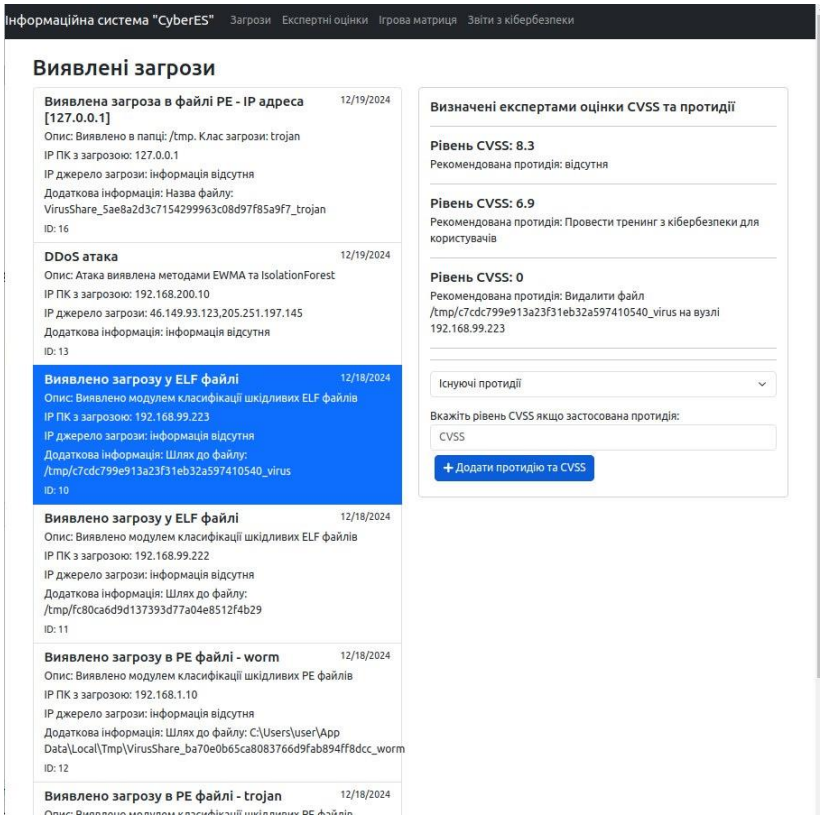


Рисунок 4.5 – Інтерфейс з інформацією про виявлені загрози

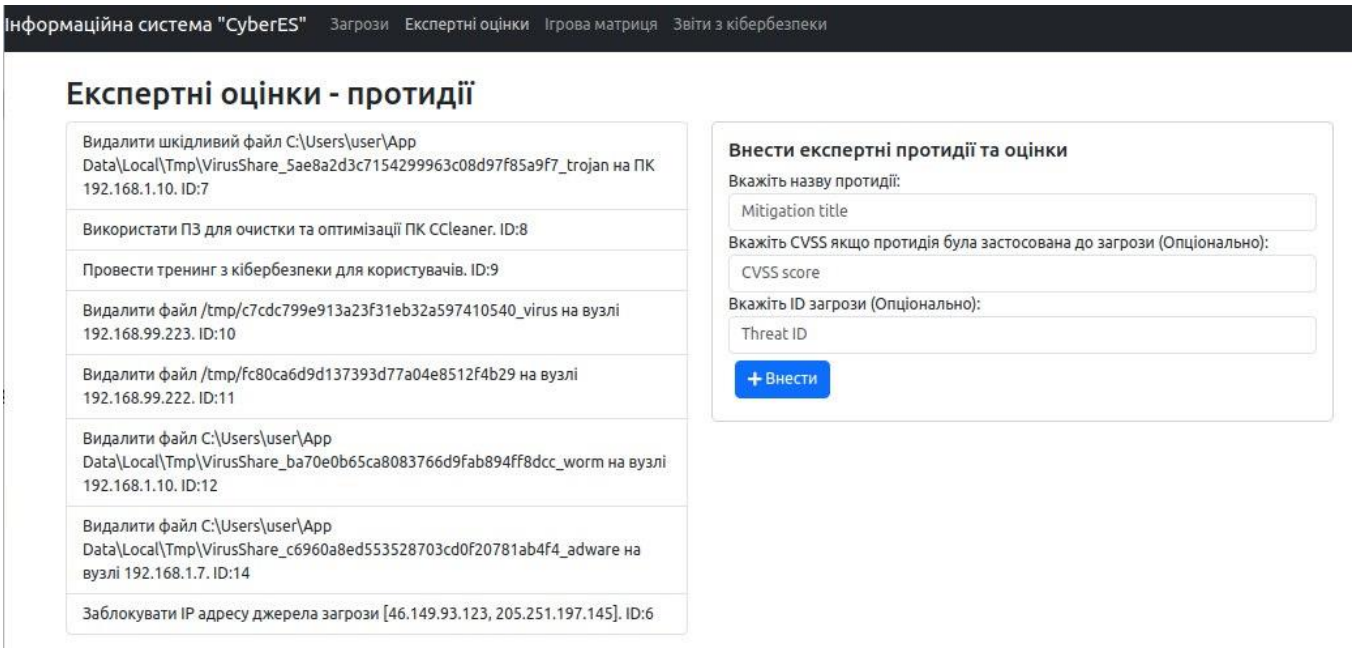


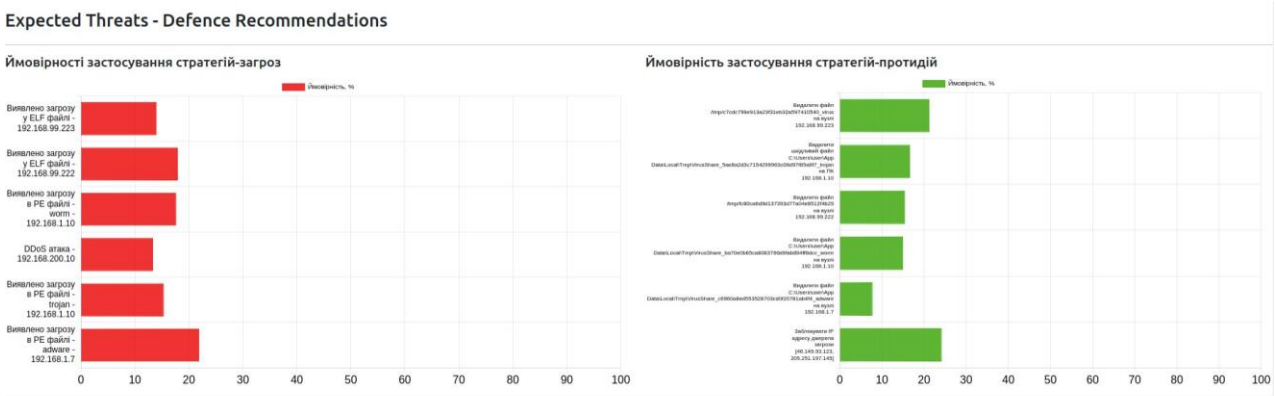
Рисунок 4.6 – Інтерфейс внесення експертних знань до системи

Matrix Visualization

Attack type	Видалити файл /tmp/c7cdc799e913a23f31eb32a597410540_virus на вузлі 192.168.99.223	Видалити файл /tmp/fc80ca6d9d137393d77a04e8512f4b29 на вузлі 192.168.99.222	Видалити файл C:\Users\user\App Data\Local\Tmp\VirusShare_ba70e0b65ca8083766d9fab894ff8dcc_worm на вузлі 192.168.1.10	Видалити файл C:\Users\user\App Data\Local\Tmp\VirusShare_c6960a8ed553528703cd0f207 на вузлі 192.168.1.7
Виявлено загрозу у ELF файлі - 192.168.99.223	-8.3	8.3	8.3	8.3
Виявлено загрозу у ELF файлі - 192.168.99.222	6.9	-6.9	6.9	6.9
Виявлено загрозу в PE файлі - worm - 192.168.1.10	7	7	-7	7
DDoS атака - 192.168.200.10	9.3	9.3	9.3	9.3
Виявлено загрозу в PE файлі - trojan - 192.168.1.10	7.3	7.3	7.3	7.3
Виявлено загрозу в PE файлі - adware - 192.168.1.7	5.7	5.7	5.7	-5.7

Рисунок 4.7 – Інтерфейс з ігровою матрицею

Можна побачити, що система здатна виводити інформацію про виявлені загрози, надає можливість експертам оцінювати рівень CVSS загроз та визначати можливі стратегії захисту, а також аналізувати звіти з кібербезпеки в реальному часі. Система розгорнута на вузлі “Expert System”.



Рекомендації щодо захисту:

Стратегія захисту	Пріоритет
Заблокувати IP адресу джерела загрози [46.149.93.123, 205.251.197.145]	1
Видалити файл /tmp/c7cdc799e913a23f31eb32a597410540_virus на вузлі 192.168.99.223	2
Видалити шкідливий файл C:\Users\user\App Data\Local\Tmp\VirusShare_5ae8a2d3c7154299963c08d97f85a9f7_trojan на ПК 192.168.1.10	3
Видалити файл /tmp/fc80ca6d9d137393d77a04e8512f4b29 на вузлі 192.168.99.222	4
Видалити файл C:\Users\user\App Data\Local\Tmp\VirusShare_ba70e0b65ca8083766d9fab894ff8dcc_worm на вузлі 192.168.1.10	5
Видалити файл C:\Users\user\App Data\Local\Tmp\VirusShare_c6960a8ed553528703cd0f20781ab4f4_adware на вузлі 192.168.1.7	6

Рисунок 4.8 – Інтерфейс з прогнозованими ймовірностями експлуатації виявлених загроз зловмисником та пріоритетом застосування стратегій протидій загрозам

Користуючись запропонованим інтерфейсом, можна отримати детальні рекомендації протидії загрозам, що забезпечують підтримку прийняття рішень захисту корпоративної комп’ютерної мережі.

Висновки до розділу 4

У ході практичної реалізації інформаційної технології виявлення та прогнозування загроз для корпоративної комп’ютерної мережі, змодельовано корпоративну комп’ютерну мережу у віртуальному середовищі GNS3 та проведено ряд експериментів з виявлення загроз.

1) У результаті проведення експериментального дослідження виявлення шкідливих ELF та PE файлів, визначено високу точність класифікації та задовільний середній час запиту. Для виявлення шкідливого ПЗ використано централізований підхід, в якому кожний вузол мережі має доступ до REST API для виявлення шкідливих файлів. Вузол надсилає запит з файлом та отримує результат ідентифікації загрози в файлі. Таким чином файл маркується шкідливим або безпечним. Виконано експеримент з порівняння запропонованих методів з методами існуючих антивірусних ПЗ. В результаті зроблено висновок про кращі показники точності, F1-міри та часу класифікації створених методів порівняно з існуючими.
2) У результаті проведення експериментів з виявлення DDoS атак підтверджено здатність запропонованого у розділі 2 методу виявляти DDoS атаки як з однаковим, так і з різним інтервалом, з урахуванням часового контексту спостережень мережевого трафіку.
3) Визначено оптимальну кількість ітерацій для ітераційного методу фіктивного розігрування гри для прогнозування ймовірності експлуатації загроз та оптимальних стратегій протидій в рушії висновків експертної системи.

4) Розроблена MVP модель системи, що дає можливість практичного її застосування для апробації в реальних умовах корпоративних комп’ютерних мереж.

Результати досліджень, приведених в розділі, опубліковані в роботі [74].

Висновки

У ході проведеного дисертаційного дослідження виконано аналіз моделей і методів виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж та запропоновано нові підходи, ефективність яких експериментально доведена та реалізовано в моделі інформаційної технології з елементами експертної системи.

Проведений аналіз існуючих проблем кіберзахисту корпоративних комп'ютерних мереж показав, що зловмисники вдаються до модифікації існуючих атак шляхом додавання нової логіки, зміни порядку команд, пошуку нових експлойтів, що зумовлює актуальність виявлення загроз з використанням статистичних моделей та моделей машинного навчання. Також є важливим вирішення питання підтримки прийняття рішень при розподілі ресурсів щодо захисту корпоративних мереж. Це потребує визначення рівня критичності загрози, що робить актуальним використання методів експертних систем в інформаційних технологіях виявлення загроз.

Розроблена модель комплексної інформаційної технології виявлення та прогнозування загроз для корпоративної комп'ютерної мережі, враховує використання модулей ідентифікації шкідливого ПЗ та рушій висновків, що дозволяють виконати прогнозування ймовірності реалізації визначених векторів вразливостей для підтримки прийняття рішень щодо реагування на загрози.

Для наповнення модуля ідентифікації шкідливого ПЗ запропоновано метод визначення секції Linux ELF файлу. Запропонований метод ідентифікації шкідливого програмного забезпечення для UNIX-подібних операційних систем містить процес семантичного аналізу секції бінарного файлу та вибору моделі класифікації. При проведенні експерименту отримали підвищення точності класифікації на 5% порівняно з існуючим методом. Також було запропоновано метод ідентифікації шкідливих Windows PE файлів, який базується на використанні секції таблиці імпортів в поєднанні з техніками word2vec та

ансамблю дерев рішень. Для порівняння з існуючими методами, визначена точність та F1-міра, які показали покращення значень цих показників при застосуванні запропонованого методу.

На основі методів машинного навчання та статистичних моделей було запропоновано метод виявлення мережевих аномалій на основі поєднання моделей Isolation Forest та EWMA-статистики, який дозволяє виявляти DDoS атаки з урахуванням часового контексту рядів спостережень мережевих параметрів. Розроблений метод прогнозування ймовірностей виникнення загроз з використанням мереж Баєса для корпоративної комп'ютерної мережі забезпечує підтримку прийняття рішень щодо стратегій реагування на можливі кіберзагрози.

Виконано проектування інформаційної системи з використанням діаграм IDEF0, розробкою функціональної моделі та її декомпозиції. Визначено варіанти використання з застосуванням Use-Case діаграм, структуру бази даних та загальну архітектуру інформаційної системи, що може бути основою для розробників інформаційних систем захисту від кіберзагроз.

Проведені експерименти роботи системи з виявлення кіберзагроз в умовах, що моделюють середовище корпоративної комп'ютерної мережі. Експерименти показали зменшення часу виявлення загроз в ELF файлах до 0.03 секунд порівняно з антивірусним ПЗ ClamAV, час сканування яким одного файлу складає 1.07 секунд, а також збільшення точності до 99% та F1-міри до 0.98 для запропонованого методу порівняно з 96% та 0.94 для ClamAV. Також було досягнуто зменшення середнього часу виявлення загроз у Windows PE файлах, порівняно з антивірусним ПЗ Microsoft Defender – 0.17 секунди для сканування одного файлу порівняно з 9.58 секундами для сканування Microsoft Defender, та збільшено точність та F1-міру: 94% та 0.94 відповідно порівняно з 93% та 0.93 для Windows Defender. Розроблена MVP модель системи, що дає можливість практичного її застосування для апробації в реальних умовах корпоративних

комп’ютерних мереж. Таким чином, всі задачі, встановлені на початку дослідження були виконані в повному обсязі і представлені в роботі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] ЗАКОН УКРАЇНИ «Про основні засади забезпечення кібербезпеки України» від 5 жовтня 2017 року № 2163-VIII [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/2163-19#Text.

[2] “What Is a Cyberattack?” [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html

[3] Aslan, Ömer & Aktug, Semih & Ozkan, Merve & Yılmaz, Abdullah & Akin, Erdal. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. *Electronics*. 12. 1-42. DOI: 10.3390/electronics12061333.

[4] “Types of cyberthreats” [Електронний ресурс] – 2024 – Режим доступу до ресурсу: https://www.ibm.com/think/topics/cyberthreats-types

[5] Current Operational Security Practices in Internet Service Provider Environments. RFC 4778 [Електронний ресурс] – 2007 – Режим доступу до ресурсу: https://datatracker.ietf.org/doc/html/rfc4778#page-3

[6] “Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX™)” [Електронний ресурс] – 2012 – Режим доступу до ресурсу: https://www.mitre.org/sites/default/files/publications/stix.pdf

[7] Yeboah-Ofori, A.; Islam, S. Cyber Security Threat Modeling for Supply Chain Organizational Environments. *Future Internet* 2019, 11, 63. DOI: 10.3390/fi11030063

[8] CIA Triad and Network Threats [Електронний ресурс] – 2024 – Режим доступу до ресурсу: https://www.howtonetwork.com/free-ccna-study-guide-ccna-book/cia-triad/

[9] Common Vulnerability Scoring System: Specification Document [Електронний ресурс] / First.org // 1.2. – 2024. – Режим доступу до ресурсу: https://www.first.org/cvss/specification-document.

[10] Bahuguna, Ashutosh & Bisht, Raj & Pande, Jeetendra. (2018). Roadmap Amid Chaos: Cyber Security Management for Organisations. 1-6. DOI: 10.1109/ICCCNT.2018.8493977.

[11] Mehmet KARAKAYA, Abdullah SEVİN “A Survey of Cyber-Threats for the Security of Institutions”. *SETSCI Conference Proceedings 13, pp. 93-99, 2022* DOI: 10.36287/setsci.5.1.018

[12] Malik, Annas & Abid, Adnan & Farooq, Shoaib & Abid, Irfan & Nawaz, Naeem & Ishaq, Kashif. (2022). Cyber threats: taxonomy, impact, policies, and way forward. *KSII Transactions on Internet and Information Systems.* 16. DOI: 10.3837/tiis.2022.07.017 .

[13] Anton Cherepanov, “Analysis of TeleBots’ cunning backdoor”, Eset, [Електронний ресурс] – 2017. – Режим доступу до ресурсу: https://www.welivesecurity.com/2017/07/04/analysis-of-telebots-cunning-backdoor/

[14] “What are Petya and NotPetya?”, Cloudfare, [Електронний ресурс] – Режим доступу до ресурсу: https://www.cloudflare.com/learning/security/ransomware/petya-notpetya-ransomware/

[15] “Industroyer2: Industroyer reloaded”, Eset, [Електронний ресурс] – 2017. – Режим доступу до ресурсу: https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/

[16] Vicente Diaz, “VirusTotal Malware Trends Report: Emerging Formats and Delivery Techniques”, [Електронний ресурс] – 2023. – Режим доступу до ресурсу: https://blog.virustotal.com/2023/07/virustotal-malware-trends-report.html

[17] Aslan, Ömer & Samet, Refik. (2020). A Comprehensive Review on Malware Detection Approaches. *IEEE Access. 8. 1-1*. DOI: 10.1109/ACCESS.2019.2963724.

[18] Shane Molinari. “Malware Science: A comprehensive guide to detection, analysis, and compliance”. *Pakt Publishing* - pp 4 – 5, 15 December 2023.

[19] Yusirwan, S., Prayudi, Y. & Riadi, I.. “Implementation of Malware Analysis using Static and Dynamic Analysis Method.” *International Journal of Computer Applications.* 117. 975-8887. 2015. pp.11-15 DOI: 10.5120/20557-2943 .

[20] Kornblum, Jesse. (2006) Identifying Almost Identical Files using Context Triggered Piecewise Hashing. *Digital Investigation 3(suppl.) - pp. 91-97*. DOI: 10.1016/j.diin.2006.06.015 .

[21] Lazo E. G. Combing through the fuzz: Using fuzzy hashing and deep learning to counter malware detection evasion techniques [Електронний ресурс] / Lazo // Microsoft Security. – 2021. – Режим доступу до ресурсу: https://www.microsoft.com/en-us/security/blog/2021/07/27/combing-through-the-fuzz-using-fuzzy-hashing-and-deep-learning-to-counter-malware-detection-evasion-techniques/.

[22] Reiher, Peter. (2004). A taxonomy of DDoS attack and DDoS Defense mechanisms. *ACM SIGCOMM Computer Communication Review*. 34. DOI: https://doi.org/10.1145/997150.997156.

[23] RFC 5901 [Електронний ресурс] – 2010. – Режим доступу до ресурсу: https://www.ietf.org/rfc/rfc5901.txt

[24] Major Cyber Attack Paralyzes Kyivstar - Ukraine's Largest Telecom Operator [Електронний ресурс] – Режим доступу до ресурсу: https://thehackernews.com/2023/12/major-cyber-attack-paralyzes-kyivstar.html

[25] “What is cybersecurity?” [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/site/us/en/learn/topics/security/what-is-cybersecurity.html

[26] “The NIST Cybersecurity Framework (CSF) 2.0”, National Institute of Standards and Technology, 2024. DOI: 10.6028/NIST.CSWP.29 .

[27] Jason Miller. IDS VS IPS VS SIEM: WHAT YOU SHOULD KNOW [Електронний ресурс] / Jason Miller // bitlyft.com. – 2020. – Режим доступу до ресурсу: https://www.bitlyft.com/resources/ids-vs-ips-vs-siem.

[28] Що таке протидія загрозам у кінцевих точках (EDR)? [Електронний ресурс] // microsoft.com. – 2020. – Режим доступу до ресурсу: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-edr-endpoint-detection-response

[29] Snort - Network Intrusion Detection System [Електронний ресурс] – 2024 – Режим доступу до ресурсу: https://snort.org .

[30] Intelligence Center // Cisco Talos [Електронний ресурс] – 2024 – Режим доступу до ресурсу: https://www.talosintelligence.com/ .

[31] Suricata. [Електронний ресурс] –2024 – Режим доступу до ресурсу: https://suricata.io/

[32] Suricata User Guide. [Електронний ресурс] – Режим доступу до ресурсу: https://docs.suricata.io/en/latest/ .

[33] Cisco Secure Endpoint Data Sheet. [Електронний ресурс] – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/collateral/security/fireamp-endpoints/datasheet-c78-733181.html .

[34] What is Orbital? // orbital.amp.cisco.com [Електронний ресурс] – Режим доступу до ресурсу: https://orbital.amp.cisco.com/help/Content/What%20is%20Orbital.htm

[35] Osquery. [Електронний ресурс] – Режим доступу до ресурсу: https://osquery.readthedocs.io/en/stable/ .

[36] Splunk Enterprise Security [Електронний ресурс] – Режим доступу до ресурсу: https://www.splunk.com/en_us/products/enterprise-security.html .

[37] Hu, S. D. (2013). Expert Systems for Software Engineers and Managers. *Switherland: Springer US.* DOI: 10.1007/978-1-4613-1065-5

[38] Liebowitz, J. (2019). The Handbook of Applied Expert Systems. *Great Britain: Taylor & Francis Group, p. (2-1).* DOI: 10.1201/9780138736654

[39] Wirkuttis, N. and Klein, H. (2017) Artificial Intelligence in Cybersecurity. *Cyber, Intelligence, and Security, 1, pp. 103-119.*

[40] Sarker, Iqbal & Furhad, Md & Nowrozy, Raza. (2021) AI-Driven Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. *SN Computer Science, 2.* DOI: 10.1007/s42979-021-00557-0.

[41] Aly, Shady & Vrana, Ivan. (2018). Toward efficient modeling of fuzzy expert systems: a survey. *Agricultural Economics (Zemědělská ekonomika). 52, pp. 456-460*. DOI: 10.17221/5051-AGRICECON.

[42] Churu, Matidaa & Blaauw, Dewalda & Watson, Brucea. (2024). A Review and Analysis of Cybersecurity Threats and Vulnerabilities, by Development of a Fuzzy Rule-Based Expert System. *Communications in Computer and Information Science, 2069 CCIS, pp. 151-168.* DOI: 10.1007/978-3-031-57639-3_7

[43] Mahdavifar, Samaneh & Ghorbani, Ali. (2020). DeNNeS: deep embedded neural network expert system for detecting cyber attacks. *Neural Computing and Applications, 32.* DOI: 10.1007/s00521-020-04830-w.

[44] Lakhno, V., Tkach, Y., Petrenko, T., Zaitsev, S., & Bazylevych, V. (2016). Development of adaptive expert system of information security using a procedure of clustering the attributes of anomalies and cyber attacks. *Eastern-European Journal of Enterprise Technologies, 6(9 (84)), pp. 32–44.* DOI: 10.15587/1729-4061.2016.85600

[45] Kaliyaperumal, Prabu & Periyasamy, Sudhakar & Thirumalaisamy, Manikandan & Balamurugan, Balamurugan & Benedetto, Francesco. (2024). A Novel Hybrid Unsupervised Learning Approach for Enhanced Cybersecurity in the IoT. *Future Internet. 16. 253.* DOI: 10.3390/fi16070253.

[46] Lotysh, Volodymyr & Gumeniuk, Larysa & Humeniuk, Pavlo. (2023). COMPARISON OF THE EFFECTIVENESS OF TIME SERIES ANALYSIS METHODS: SMA, WMA, EMA, EWMA, AND KALMAN FILTER FOR DATA ANALYSIS. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska. 13, pp. 71-74.* DOI: 10.35784/iapgos.3652.

[47] Ryciak, Piotr & Wasielewska, Katarzyna & Janicki, Artur. (2022). Anomaly Detection in Log Files Using Selected Natural Language Processing Methods. *Applied Sciences. 12. 5089.* DOI: 10.3390/app12105089.

[48] Alaca, Yusuf & Celik, Yuksel & Id, Sanjay. (2023). Anomaly Detection in Cyber Security with Graph-Based LSTM in Log Analysis. *Chaos Theory and Applications. 5*. DOI: 10.51537/chaos.1348302.

[49] Alhaidari, Fahd & Shaib, Nouran & Alsafi, Maram & Alharbi, Haneen & Alawami, Majd & Aljindan, Reem & Rahman, Atta & Zagrouba, Rachid. (2022). ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques. *Computational Intelligence and Neuroscience.* DOI: 10.1155/2022/1615528.

[50] Parildi, E.S., Hatzinakos, D. & Lawryshyn, Y. (2021) Deep learning-aided runtime opcode-based Windows malware detection. *Neural Computing & Applications 33, 11963–11983.* DOI: 10.1007/s00521-021-05861-7

[51] Yuk, Chang & Seo, Chang. (2022). Static Analysis and Machine Learning-based Malware Detection System using PE Header Feature Values. *International Journal of Innovative Research and Scientific Studies. 5., pp. 281-288.* DOI: 10.53894/ijirss.v5i4.690.

[52] Lad, Sumit & Adamuthe, Amol. (2022). Improved Deep Learning Model for Static PE Files Malware Detection and Classification. *International Journal of Computer Network and Information Security. 14, pp. 14-26.* DOI: 10.5815/ijcnis.2022.02.02.

[53] Koçak, Aynur & Söğüt, Esra & Alkan, Mustafa & Erdem, Ayhan. (2023). Detection of Different Windows PE Malware Using Machine Learning Methods – Makine Öğrenimi Metotları Kullanılarak Farklı Windows PE Kötü Amaçlı Yazılımların Tespiti. *Journal of Polytechnic.* DOI: 10.2339/politeknik.1207704.

[54] Poudyal, Subash & Dasgupta, Dipankar & Akhtar, Zahid & Gupta, Kishor Datta. (2019). A Multi-Level Ransomware Detection Framework using Natural Language Processing and Machine Learning. *International Conference on Malicious and Unwanted Software (MALCON 2019)*

[55] Lin, C.-T & Wang, N.-J & Xiao, Han & Eckert, Claudia. (2015). Feature Selection and Extraction for Malware Classification. *Journal of Information Science and Engineering. 31. pp. 965-992.*

[56] Percílio Azevedo, B.W., Oliveira Albuquerque, R.d., García Villalba, L.J. (2025). Method to Automate the Classification of PE32 Malware Using Word2vec and LSTM. *In: Gritzalis, D., Choo, KK.R., Patsakis, C. (eds) Malware. Advances in Information Security, vol 91. Springer, Cham.* DOI: 10.1007/978-3-031-66245-4_9

[57] N. Visweswaran, M. Jeevanantham, C. Thamarai Kani, P. Deepalakshmi., S. Sathiyandrakumar (2019). Automated PE32 Threat Classification using Import Table and Deep Neural Networks. *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES).* DOI: 10.1109/INCCES47820.2019.9167732

[58] Balram, Neil & Hsieh, George & McFall, Christian. (2019). Static Malware Analysis Using Machine Learning Algorithms on APT1 Dataset with String and PE Header Features. *2019 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 90-95.*. DOI: 10.1109/CSCI49370.2019.00022.

[59] Qin, B., Junpeng Zhang and Hongyu Chen. (2021).Malware detection based on TF-(IDF&ICF) method. *Journal of Physics: Conference Series* 2024 (2021). DOI: 10.1088/1742-6596/2024/1/012030

[60] Kang, B., Yerima, S.Y., Mclaughlin & Sezer S. (2016). N-opcode analysis for android malware classification and categorization. *International Conference On Cyber Security And Protection Of Digital Services (Cyber Security), 2016, pp. 1-7.* DOI: 10.1109/CyberSecPODS.2016.7502343

[61] A. Ravi and V. Chaturvedi (2022). Static Malware Analysis using ELF features for Linux based IoT devices. *35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID), Bangalore, India, 2022, pp. 114-119.* DOI: 10.1109/VLSID2022.2022.00033.

[62] Wan, Tzu-Ling & Ban, Tao & Cheng, Shin-Ming & Lee, Yen-Ting & Sun, Bo & Isawa, Ryoichi & Takahashi, Takeshi & Inoue, Daisuke. (2020). Efficient Detection and Classification of Internet-of-Things Malware Based on Byte Sequences from Executable Files. *IEEE Open Journal of the Computer Society. 1, pp. 262-275.* DOI: 10.1109/OJCS.2020.3033974.

[63] Bierbrauer, David & Chang, Alexander & Kritzer, Will & Bastian, Nathaniel. (2021). Cybersecurity Anomaly Detection in Adversarial Environments. DOI: 10.48550/arXiv.2105.06742.

[64] Duque Anton, Simon & Kanoor, Suneetha & Fraunholz, Daniel & Schotten, Hans. (2018). Evaluation of Machine Learning-based Anomaly Detection Algorithms on an Industrial Modbus/TCP Data Set. *The 13th International Conference on Availability, Reliability and Security (ARES 2018), pp.1-9.* DOI: 10.1145/3230833.3232818.

[65] Patil, Vinay & Deore, Shailesh. (2023). A STUDY OF DDOS ATTACK DETECTION METHODS. *Shu Ju Cai Ji Yu Chu Li/Journal of Data Acquisition and Processing. 38, pp. 3583-3591.* DOI: 10.5281/zenodo.98549840.

[66] Almahmoud, Z., Yoo, P.D., Alhussein, O., Ilyas Farhat & Ernesto Damiani (2023). A holistic and proactive approach to forecasting cyber threats. *Sci Rep 13, 8049* DOI: 10.1038/s41598-023-35198-1

[67] Cuong, Do & Tran, Nguyen & Hong, Choong Seon & Kamhoua, Charles & Kwiat, Kevin & Blasch, Erik & Ren, Shaolei & Pissinou, Niki & Iyengar, Sundararaj. (2017). Game Theory for Cyber Security and Privacy. *ACM Computing Surveys. 50. pp. 1-37.* DOI: 10.1145/3057268.

[68] Zhu, Quanyan & Başar, Tamer (2015). Game-Theoretic Methods for Robustness, Security, and Resilience of Cyberphysical Control Systems: Games-in-Games Principle for Optimal Cross-Layer Resilient Control Systems. *IEEE Control Systems Magazine 35(1), pp. 46-65.* DOI: 10.1109/MCS.2014.2364710.

[69] Wang, Kun & Du, Miao & Yang, Dejun & Zhu, Chunsheng & Shen, Jian & Zhang, Yan. (2016). Game-Theory-Based Active Defense for Intrusion Detection in Cyber-Physical Embedded Systems. *ACM Transactions on Embedded Computing Systems. 16. pp.1-21.* DOI: 10.1145/2886100.

[70] Sayed, M.A., Anwar, A.H., Kiekintveld, C., Bosansky, B., Kamhoua, C. (2023). Cyber Deception Against Zero-Day Attacks: A Game Theoretic Approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 13727 LNCS, pp. 44-63.* DOI: 10.1007/978-3-031-26369-9_3

[71] M. Major, S. Fugate, J. Mauger and K. Ferguson-Walter (2019) Creating Cyber Deception Games. *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI), Los Angeles, CA, USA, 2019 pp. 102-111.* DOI: 10.1109/CogMI48466.2019.00023

[72] Ullah, F.; Turab, A.; Ullah, S.; Cacciagrano, D.; Zhao, Y. (2024) Enhanced Network Intrusion Detection System for Internet of Things Security Using Multimodal Big Data Representation with Transfer Learning and Game Theory. *Sensors 2024, 24, 4152.* https://www.scopus.com/record/display.uri?eid=2-s2.0-85198342880&origin=resultslist DOI: 10.3390/s24134152

[73] K. S. Gill, S. Saxena and A. Sharma (2024) NCGTM: A Noncooperative Game-Theoretic Model to Assist IDS in Cloud Environment. *IEEE Transactions on Industrial Informatics, vol. 20, no. 3, pp. 3124-3132, March 2024.* DOI: 10.1109/TII.2023.3300452

[74] Mishchenko, M.V.. і Dorosh, M.S.. (2024) An Expert System of Recommendations for Combating Cyber Threats Using CVSS Metrics and Game Theory. *Herald of Advanced Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2024, Vol.7, No.3, pp. 284–295.* DOI: 10.15276/hait.07.2024.20

[75] Міщенко, М.. (2024). Функціональна модель системи виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж з

використанням експертних оцінок. *Технічні науки та технології, (3 (37)), с. 143–152.* https://doi.org/10.25140/2411-5363-2024-3(37)-143-152

[76] Wang, J., & Liu, L. (2023). A new multivariate control chart based on the isolation forest algorithm. *Quality Engineering, 36(2), pp. 390–406.* DOI: 10.1080/08982112.2023.2220773

[77] MA, Zhaohui; LI, Zhuojie. (2020) RESEARCH ON DDOS ATTACK DETECTION BASED ON ISOLATION FOREST AND KMeans ALGORITHM IN SDN. *International Journal of Computer Science and Information Technology(IJCSIT), 2020, 2.1: pp.1-25.*

[78] RIPAN, Rony Chowdhury, Md. Moinul Islam, Hamed Alqahtani, Iqbal H. Sarker. (2022) Effectively predicting cyber-attacks through isolation forest learning-based outlier detection. *Security and Privacy, 2022, 5.3: e212.* DOI: 10.1002/spy2.212

[79] Скітер, І.С., Скітер, А.І «ВИЯВЛЕННЯ DDOS АТАК НА ОСНОВІ АНАЛІЗУ ПРОФІЛЮ КОМП'ЮТЕРНОЇ МЕРЕЖІ З ВИКОРИСТАННЯМ EWMA-СТАТИСТИКИ» // *Друга всеукраїнська науково-практична конференція «Перспективні напрями захисту інформації». Одеса, 2016. – С. 67-70.*

[80] Скітер І. С. Ідентифікація аномальної поведінки трафіку комп'ютерної мережі на основі EWMA-статистики / І. С. Скітер, І. В. Бальченко // *Перша Міжнародна конференція «Проблеми виведення з експлуатації об'єктів ядерної енергетики та відновлення оточуючого середовища» INUDECO'16 25-27 квітня 2016 : зб. матеріалів. – Славутич : СФ НТУУ «КПІ», 2016. – С. 171–178.*

[81] Shiel, I. & O'Shaughnessy, S. (2019) Improving file-level fuzzy hashes for malware variant classification. *Digital Investigation. 28. 2019. S88–S94.* DOI: 10.1016/j.diin.2019.01.018

[82] Raff, E., et al. "Malware detection by eating a whole exe." *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018. DOI: 10.48550/arXiv.1710.09435

[83] Zhong, Fangtian & Chen, Zekai & Xu, Minghui & Zhang, Guoming & Yu, Dongxiao & Cheng, Xiuzhen. (2021). Malware-on-the-Brain: Illuminating Malware Byte Codes with Images for Malware Classification. DOI: 10.48550/arXiv.2108.04314.

[84] Sahota, J. & Vlajic, N. (2021) Mozi IoT Malware and Its Botnets: From Theory To Real-World Observations. *2021 International Conference on Computational Science and Computational Intelligence, CSCI 2021, pp. 698 - 703* DOI: 10.1109/CSCI54926.2021.00181

[85] Cozzi, E., Graziano, M., Fratantonio, Y. & Balzarotti, D. (2018) "Understanding Linux Malware," *2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 161-175,* DOI: 10.1109/SP.2018.00054

[86] Margolis, J., Oh, T. T., Jadhav, S., Kim, Y. H. & Kim J. N. (2017) "An In-Depth Analysis of the Mirai Botnet," *2017 International Conference on Software Security and Assurance (ICSSA), 2017, pp. 6-12.* DOI: 10.1109/ICSSA.2017.12.

[87] McNulty, Leona & Vassilakis, Vassilios. (2022). IoT Botnets: Characteristics, Exploits, Attack Capabilities, and Targets. DOI: 10.1109/CSNDSP54353.2022.9908039.

[88] Jurafsky, D., Martin, J. H. ”N-gram Language Models”. *Speech and Language Processing. 2021. Chapter 3. Draft.*

[89] Schaetti, N. (2017) UniNE at CLEF 2017: TF-IDF and Deep-Learning for Author Profiling. *CLEF, Dublin, 2017.* DOI: 10.13140/RG.2.2.14902.60482.

[90] Kibriya, A. M., Frank, E., Pfahringer, B. & Holmes, G. (2004) Multinomial naive Bayes for text categorization revisited. *Australasian Joint Conference on Artificial Intelligence, AI 2004: Advances in Artificial Intelligence pp 488–499.* DOI: 10.1007/978-3-540-30549-1_43

[91] Su, Jiang & Shirab, Jelber. (2011) Large Scale Text Classification using Semisupervised Multinomial Naive Bayes. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA.* 2011.

[92] Hearst, M. A. “Support vector machines”. *IEEE Intelligent Systems*. 1998. pp. 18-28. DOI: 10.1109/5254.708428 .

[93] Aatila, M., Mohamed, L.& Kartit A. (2020) An Overview of Gradient Descent Algorithm Optimization in Machine Learning: Application in the Ophthalmology Field. *Smart Applications and Data Analysis. pp.349-359. 2020*. DOI: 10.1007/978-3-030-45183-7_27 .

[94] Lei, Yunwen, Ting Hu and Ke Tang. “Generalization Performance of Multi-pass Stochastic Gradient Descent with Convex Loss Functions.” J. Mach. Learn. Res. 22 (2021): 25:1-25:41.

[95] Friedman J., “Stochastic gradient boostig” *Computational Statistics & Data Analysis. 2002. Volume 38, Issue 4, pp. 367-378* DOI: 10.1016/S0167-9473(01)00065-2

[96] Chen, T. & Guestrin, C. “XGBoost: A Scalable Tree Boosting System”. *Proceedings of the 22nd {ACM} {SIGKDD} International Conference on Knowledge Discovery and Data Mining.* 2016. DOI: 10.1145/2939672.2939785

[97] “Portable Formats Specification”. Tool Interface Standards. *TIS Committee.* 1993.

[98] All new ELF binaries collected since the previous release in 2019 [Електронний ресурс] // VirusShare. – 2020. – Режим доступу до ресурсу: https://virusshare.com/torrents.

[99] Get a file report [Електронний ресурс] // VirusTotal. – 2021. – Режим доступу до ресурсу: https://developers.virustotal.com/reference/file-info..

[100] McNulty, Leona & Vassilakis, Vassilios. (2022). IoT Botnets: Characteristics, Exploits, Attack Capabilities, and Targets. DOI: 10.1109/CSNDSP54353.2022.9908039.

[101] Grandini, Margherita & Bagli, Enrico & Visani, Giorgio. (2020). Metrics for Multi-Class Classification: an Overview. DOI: 10.48550/arXiv.2008.05756.

[102] Alfons, J. & Hermann, N. “Reversing and Smoothing the Multinomial Naive Bayes Text Classifer” *Pattern Recognition in Information Systems, Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems, PRIS 2002, In conjunction with ICEIS 2002, Ciudad Real. 2002. pp. 200-212.*

[103] Adnan, M., Alarood, A., Uddin, M. I., & Rehman, I. “Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models.” *PeerJ Computer Science.* 2022. 8. e803. DOI: 10.7717/peerj-cs.803.

[104] Bottou, L. “Stochastic gradient descent tricks” *Neural Networks: Tricks of the Trade. 2012. p. 421–436.* DOI: 10.1007/ 978-3-642-35289-8_25.

[105] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio, “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”, *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, October 2014, pp. 103-111.* DOI: 10.48550/arXiv.1409.1259

[106] Q. Qi, L. Lin, R. Zhang and C. Xue, "MEDT: Using Multimodal Encoding-Decoding Network as in Transformer for Multimodal Sentiment Analysis", in *IEEE Access, vol. 10, pp. 28750-28759, 2022.* DOI: 10.1109/ACCESS.2022.3157712

[107] X. Xiao, L. Wang, K. Ding, S. Xiang and C. Pan, "Deep Hierarchical Encoder–Decoder Network for Image Captioning", in *IEEE Transactions on Multimedia, vol. 21, no. 11, pp. 2942-2956, Nov. 2019.* DOI: 10.1109/TMM.2019.2915033

[108] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, “Attention is all you need”, *Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2017.* DOI: 10.48550/arXiv.1706.03762

[109] Khan S., Nauman M., “Interpretable Detection of Malicious Behavior in Windows Portable Executables Using Multi-Head 2D Transformers”, *Big Data Mining and Analytics, vol. 7, pp. 485 – 499, 2024.* DOI: 10.26599/BDMA.2023.9020025

[110] D. Tsirmpas, I. Gkionis, G Th. Papadopoulos, Mademlis I., “Neural natural language processing for long texts: A survey on classification and summarization”, *Engineering Applications of Artificial Intelligence, vol. 133, 2024.* DOI: 10.1016/j.engappai.2024.108231

[111] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach”, 2019. DOI: 10.48550/arXiv.1907.11692

[112] K. Clark, M. Luong, Q. V. Le, C. D. Manning, “ELECTRA: pre-training text encoders as discriminators rather than generators” *Proceedings of the International Conference on Learning Representations (ICLR), 2020.* DOI: 10.48550/arXiv.2003.10555

[113] Radford, Alec and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training.” (2018).

[114] Jan Sawicki, Maria Ganzha, Marcin Paprzycki, “The State of the Art of Natural Language Processing—A Systematic Automated Review of NLP Literature Using NLP Techniques”, *Data Intelligence vol. 5 (3), pp. 707–749,2023.* DOI: 10.1162/dint_a_00213

[115] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient Estimation of Word Representations in Vector Space”, *Proceedings of Workshop at ICLR, January 2013.* DOI: 10.48550/arXiv.1301.3781

[116] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, “Distributed Representations of Words and Phrases and their Compositionality”, *Advances in Neural Information Processing Systems, vol. 26, October 2013.* DOI: 10.48550/arXiv.1310.4546

[117] PE Format [Електронний ресурс] // Microsoft Learn. – 2024. – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/windows/win32/debug/pe-format.

[118] Catak Ferhat Ozgur, Ahmed Javed, Sahinbas Kevser, Khand Zahid Hussain, “Data augmentation based malware detection using convolutional neural networks”, *PeerJ Computer Science, vol. 7, 2021.* DOI: 10.48550/arXiv.2010.01862

[119] Yang Limin, Ciptadi Arridhana, Laziuk Ihar, Ahmadzadeh Ali, Wang Gang, “BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware”, *4th Deep Learning and Security Workshop, San Francisco, CA, USA, 2021, pp. 78-84.* DOI: 10.1109/SPW53761.2021.00020

[120] VirusShare [Електронний ресурс] // GitHub - seifreed. – 2024. – Режим доступу до ресурсу: https://github.com/seifreed/VirusShare.

[121] Global Threat Activity [Електронний ресурс] // Microsoft. – 2024. – Режим доступу до ресурсу: https://www.microsoft.com/en-us/wdsi/threats.

[122] Cao, Hui & Naito, Takashi & Ninomiya, Yoshiki. (2008). Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification. DOI: 10.1007/978-1-4020-8450-8_1.

[123] Tran Kien, Sato Hiroshi, “NLP-based approaches for malware classification from API sequences”, *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), pp. 101-105, 2017.* DOI: 10.1109/IESYS.2017.8233569

[124] Kingma Diederik, Ba Jimmy, “Adam: A Method for Stochastic Optimization”, *International Conference on Learning Representations, December 2014.* DOI: 10.48550/arXiv.1412.6980

[125] Ye, Yanfang & Li, Tao & Jiang, Qingshan & Wang, Youyu. (2010). CIMDS: Adapting Postprocessing Techniques of Associative Classification for Malware Detection. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on. 40. pp. 298 - 307.* DOI: 10.1109/TSMCC.2009.2037978.

[126] Ahmadian Ramaki, Ali & Khosravi-Farmad, Masoud & Bafghi, Abbas. (2015). Real time alert correlation and prediction using Bayesian networks. DOI: 10.1109/ISCISC.2015.7387905.

[127] Devarakonda, Nagaraju & Pamidi, Srinivasulu & Vatsavayi, Valli Kumari & Govardhan, Dr. (2012). Intrusion Detection System using Bayesian Network and Hidden Markov Model. *Procedia Technology. 4. pp. 506–514.* DOI: 10.1016/j.protcy.2012.05.081.

[128] Common Vulnerability Scoring System Version 4.0 Calculator [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://www.first.org/cvss/calculator/4.0.

[129] D. T. Vasireddy, D. S. Dale and Q. Li, "CVSS Base Score Prediction Using an Optimized Machine Learning Scheme," *2023 Resilience Week (RWS), National Harbor, MD, USA, 2023, pp. 1-6.* DOI: 10.1109/RWS58133.2023.10284627.

[130] Maghrabi, Louai & Pfluegel, Eckhard & al-Fagih, Luluwah & Graf, Roman & Settanni, Giuseppe & Skopik, Florian. (2017). Improved software vulnerability patching techniques using CVSS and game theory. pp. 1-6. DOI: 10.1109/CyberSecPODS.2017.8074856.

[131] Settanni, Giuseppe & Skopik, Florian & Graf, Roman & Wurzenberger, Markus & Fiedler, Roman. (2016). Correlating cyber incident information to establish situational awareness in Critical Infrastructures. pp. 78-81. DOI: 10.1109/PST.2016.7906940.

[132] S. Zhang, M. Cai, M. Zhang, L. Zhao and X. d. C. de Carnavalet, "The Flaw Within: Identifying CVSS Score Discrepancies in the NVD," *2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Naples, Italy, 2023, pp. 185-192.* DOI: 10.1109/CloudCom59040.2023.00039.

[133] Akyar, Emrah & Akyar, Handan & Duzce, Serkan. Brown–Robinson method for interval matrix games. *Soft Comput. (2011). 15. 2057-2064.* DOI: 10.1007/s00500-011-0703-6.

[134] Mishchenko M. V., Dorosh M. S. “Semantic analysis and classification of malware for UNIX-like operating systems with the use of machine learning methods”.

Applied Aspects of Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2022; Vol.5 No.4: 371–386. DOI: 10.15276/aait.05.2022.25

[135] Top-down vs. Bottom-up Network Design [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://www.cbtnuggets.com/blog/technology/networking/top-down-vs-bottom-up-network-design

[136] Cisco Catalyst 8000V Edge Software Data Sheet [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://www.cisco.com/c/en/us/products/collateral/routers/catalyst-8000v-edge-software/catalyst-8000v-edge-software-ds.html

[137] Hardware Sizing Guidance [Електронний ресурс] // pfSense. – 2024. – Режим доступу до ресурсу: https://docs.netgate.com/pfsense/en/latest/hardware/size.html.

[138] Configure and manage Microsoft Defender Antivirus with the mpcmdrun.exe command-line tool [Електронний ресурс]. – 2024 – Режим доступу до ресурсу: https://learn.microsoft.com/uk-ua/defender-endpoint/command-line-arguments-microsoft-defender-antivirus

[139] ClamAV Documentation [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: https://docs.clamav.net/

[140] Cybersecurity Risk (2022 CISA Vulnerability) [Електронний ресурс] // kaggle.com. – 2022. – Режим доступу до ресурсу: https://www.kaggle.com/datasets/thedevastator/exploring-cybersecurity-risk-via-2022-cisa-vulne.

[141] Collopy, Fred & Adya, Monica & Armstrong, J.. (2011). Expert Systems for Forecasting. 30. DOI: 10.1007/978-0-306-47630-3_14.

[142] Mishchenko, M., & Dorosh, M. (2025). Detection of Windows Portable Executable Malware using NLP Techniques and Proxy-server. International Journal of Computing, 23(4), 663-672. https://doi.org/10.47839/ijc.23.4.3765

[143] Nghi, Tran & Dang, Kien & Quoc, Dung & Tho, Nguyen & Binh, Nguyen. (2019). A Novel Framework to Classify Malware in MIPS Architecture-Based IoT Devices. Security and Communication Networks. 2019. 1-13. 10.1155/2019/4073940.

ДОДАТКИ

ДОДАТОК А
СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. А.Г. Гребенник, О.В. Трунова, В.В. Казимир, М.В. Міщенко «Виявлення та прогнозування загроз для корпоративної комп'ютерної мережі.» *Технічні науки та технології, 2020. - № 2 – с.175–184.* (0,5 ум. друк. арк.) (Особистий внесок здобувача: загальна архітектура системи прогнозування та виявлення загроз ККМ, аналіз методу виявлення кіберзагроз з використанням EWMA-статистики) (0,3 ум. друк. арк.)
2. Mishchenko M.V., Dorosh M.S.. "Semantic analysis and classification of malware for UNIX-like operating systems with the use of machine learning methods". Applied Aspects of Information Technology. 2022; Vol. 5, No. 4: 371-386, DOI: https://doi.org/10.15276/aait.05.2022.25. (0,6 ум. друк. арк.) (Особистий внесок здобувача: метод семантичного аналізу та класифікації шкідливого програмного забезпечення для UNIX-подібних систем) (0,5 ум. друк. арк.)
3. Mishchenko, M. V., Dorosh, M. S.. (2024). «An expert system of recommendations for combating cyber threats using CVSS metrics and game theory.» Вісник сучасних інформаційних технологій, 7(3), 284–295. https://doi.org/10.15276/hait.07.2024.20 (0,6 ум. друк. арк.) (Особистий внесок здобувача: експертна система рекомендації протидії загрозам з використанням метрик CVSS та теорії ігор) (0,5 ум. друк. арк.)
4. Міщенко, М. «Функціональна модель системи виявлення та прогнозування кіберзагроз для корпоративних комп'ютерних мереж з використанням експертних оцінок.» Технічні науки та технології – 2024. № 3 (37), - с. 143–152. https://doi.org/10.25140/2411-5363-2024-3(37)-143-152 (0,6 ум. друк. арк.)
5. Mishchenko, M., & Dorosh, M. (2024). Detection of Windows Portable Executable Malware using NLP Techniques and Proxy-server. International Journal of Computing, 23(4), 663-672. https://doi.org/10.47839/ijc.23.4.3765 (0,6

ум. друк. арк.) (Особистий внесок здобувача: метод ідентифікації шкідливих Windows PE файлів) (0,5 ум. друк. арк.)

6. Міщенко М.В., Гребенник А.Г., Трунова О.В.. “Прогнозування рівня загроз з використанням мереж Байєса” XV міжнародна науково-практична конференція математичне та імітаційне моделювання систем МОДС 2020. С. 120-123. Тези доповідей. (0,2 ум. друк. арк.) (Особистий внесок здобувача: дослідження існуючих методів прогнозування загроз з використанням мережі Баєса) (0,1 ум. друк. арк.)
7. Міщенко М.В. «Створення сервісу для виявлення шкідливих elf файлів за допомогою машинного навчання з використанням хмарних технологій aws» Всеукраїнська науково-практична конференція студентів, аспірантів та молодих учених НОВІТНІ ТЕХНОЛОГІЇ У НАУКОВІЙ ДІЯЛЬНОСТІ І НАВЧАЛЬНОМУ ПРОЦЕСІ. – 2023 – с.107-108. Тези доповідей. (0,1 ум. друк. арк.)
8. Міщенко М.В. «Створення експертної системи генерації рекомендацій з протидії кібератакам з використанням теорії ігор.» XIV Міжнародна науково-практична конференція студентів, аспірантів і молодих вчених ЮНІСТЬ НАУКИ – 2024 – с. 1175-1176. Тези доповідей. (0,1 ум. друк. арк.)

ДОДАТОК Б
ДОВІДКИ ПРО ВПРОВАДЖЕННЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

вул. Шевченка, 95, Чернігів, 14030, Україна

тел. +38(0462) 665-103;
факс +38(0462) 665-105
E-mail: cstu@stu.cn.ua
www.stu.cn.ua
Код СДРПОУ 05460798

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

CHERNIHIV POLYTECHNIC NATIONAL UNIVERSITY

95, Shevchenko str., Chernihiv, 14030, Ukraine

13.01.2025 № 202/22-121/ЕС

На № _____ від _____

Довідка
про впровадження результатів дисертаційної роботи
МІЩЕНКА Максима Валерійовича
на тему «Пронозування та виявлення загроз для корпоративних комп'ютерних мереж засобами експертних систем»
на здобуття наукового ступеня доктора філософії за спеціальністю
122 «Комп'ютерні науки»

Результати дисертаційного дослідження МІЩЕНКА Максима Валерійовича були впроваджені в навчальному процесі та проектній діяльності Національного університету «Чернігівська політехніка»:

1. Методи класифікації та дослідницький аналіз даних в курсі лекцій та лабораторній роботі №5 «Ознайомлення з бібліотеками машинного навчання в мові Python» з дисципліни «Системи штучного інтелекту» для здобувачів вищої освіти за освітнім ступенем бакалавр спеціальності 121 – «Інженерія програмного забезпечення».

2. При виконанні проекту «Cyber Rapid Analisys for Defense Awareness of Real-time Situation - CyRADARS» за грантом NATO SPS, (grant agreement number: G5286). Результати дослідження з удосконалення методу прогнозування ймовірностей виникнення загроз засобами Баєсових мереж були включені до фінального звіту.

Проректор з наукової роботи А.Л. Приступа

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
"ДАТЧИКОВЕ ПІДПРИЄМСТВО "ЗАВОД РАПІД"
(ТОВ "ДП "ЗАВОД РАПІД")
код ЄДРПОУ 35779941
14030, м. Чернігів, вул. Захисників України, 25 А
т./ф. (0462) 66-58-80, 66-58-82
Код IBAN UA303223130000026003000012285
ФІЛІЯ АТ «УКРЕКСІМБАНК»
в м. Чернігові, МФО 322313
Індивідуальний податковий номер 357799425265
№ свідоцтва платника податку 100101389
E-mail: rapid@zavod-rapid.com
http://zavod-rapid.com

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
"ДАТЧИКОВОЕ ПРЕДПРИЯТИЕ "ЗАВОД РАПИД"
(ООО "ДП "ЗАВОД РАПИД")
код ЕГРПОУ 35779941
14030, г. Чернигов, ул. Защитников Украины, 25 А
т./ф. (0462) 66-58-80, 66-58-82
Код IBAN UA303223130000026003000012285
ФИЛИАЛ АО «УКРЭКСИМБАНК»
в г. Чернигове, МФО 322313
Индивидуальный налоговый номер 357799425265
№свидетельства налогоплательщика 100101389
E-mail: rapid@zavod-rapid.com
http://zavod-rapid.com

№ 1154 " 27 " 12 2024 р.

Проректору з наукової роботи
Національного університету
«Чернігівська політехніка»
Анатолію ПРИСТУПІ
14035, м. Чернігів, в. Шевченка, 95

ДОВІДКА ПРО ВПРОВАДЖЕННЯ
наукових результатів дисертаційної роботи
МІЩЕНКА Максима Валерійовича на тему: «Прогнозування та виявлення загроз для корпоративних комп'ютерних мереж засобами експертних систем»,
представленої на здобуття доктора філософії
зі спеціальності 122 – «Комп'ютерні науки»

Запропоновані в дисертаційному дослідженні МІЩЕНКА Максима Валерійовича методи ідентифікації шкідливого ПЗ для операційних систем Windows та Linux в комп'ютерній мережі підприємства дозволили:

- виявити існуючі файлові загрози на комп'ютерах в корпоративній мережі;
- покращити швидкість виявлення шкідливого ПЗ.

З залученням співробітників інформаційно-технічного відділу підприємства для внесення експертних оцінок в розроблену інформаційну систему, вдалось виконати прогнозування ймовірності реалізації зловмисниками визначених векторів вразливостей комп'ютерної мережі.

З повагою
Директор ТОВ «ДП «ЗАВОД РАПІД» Володимир ТИТЕНОК

Товариство з обмеженою відповідальністю "ДАТЧИКОВЕ ПІДПРИЄМСТВО "ЗАВОД РАПІД" Ідентифікаційний код 35779941 Україна, м. Чернігів

ДОДАТОК В
ДОСЛІДЖЕННЯ СТАТИСТИЧНОЇ ЗНАЧУЩОСТІ РЕЗУЛЬТАТІВ ПРОВЕДЕНИХ ЕКСПЕРИМЕНТІВ

Таблиця В.1 – Результати дослідження статистичної значущості середніх значень точності та $F1_{зваженого}$ ідентифікації шкідливих Linux ELF файлів з використанням класифікатору Naïve Bayes

Метод	1	2	3	4	5	6	7	8	9	10
$F1_{зважене}$ запропонованого методу	0,910	0,914	0,916	0,917	0,918	0,920	0,920	0,920	0,920	0,920
Точність запропонованого методу	0,908	0,913	0,916	0,918	0,919	0,920	0,920	0,921	0,921	0,922
H_0 гіпотеза	Середнє значення точності та $F1_{зважене}$ статистично значуще не відрізняється від μ_0									
Рівень значущості α	0,010									
Гіпотетичне середнє точності μ_0	0,920									
Гіпотетичне середнє $F1_{зважене}$ μ_0	0,920									
t-test оцінки середньої точності	-1,512									

Продовження таблиці В.1

p-значення оцінки середньої точності	0,165
t-test оцінки середньої $F1_{зваженого}$	-1,963
p-значення оцінки середнього $F1_{зваженого}$	0,081

Таблиця В.2 – Результати дослідження статистичної значущості збільшення середніх значень точності та $F1_{зваженого}$ ідентифікації шкідливих Linux ELF файлів з використанням моделі XGBoost порівняно з існуючим методом

№	1	2	3	4	5	6	7	8	9	10
$F1_{зважене}$ (μ) запропонованого методу	0,987	0,980	0,986	0,980	0,980	0,984	0,986	0,988	0,986	0,982
Точність (μ) запропонованого методу	0,988	0,980	0,986	0,980	0,980	0,984	0,986	0,988	0,987	0,982
Гіпотеза H_0	$\mu \leq \mu_0$									
Гіпотеза H_1	$\mu > \mu_0$									
Рівень значущості α	0,01									
$t\text{-test}_{crit}$, df=9	2,821									
Nghi, Tran та ін. [143]	$F1_{зважене}$ μ_0	t-test оцінки середньої $F1_{зваженого}$		p-значення оцінки середнього $F1_{зваженого}$		p-значення середньої точності		Середня точність μ_0		t-test середньої точності
	0,9686	47.4262		$4,12 * 10^{-12}$		$1,71 * 10^{-7}$		0,9334		14,2940
Прийнята гіпотеза	H_1					H_1				

Таблиця В.3 – Результати дослідження статистичної значущості середніх значень точності та $F1_{зваженого}$ ідентифікації шкідливих Windows PE файлів порівняно з існуючими методами

Метод	1	2	3	4	5	6	7	8	9	10
$F1_{зважене}$ (μ) запропонованого методу	0,943	0,943	0,940	0,942	0,936	0,946	0,940	0,948	0,933	0,940
Точність (μ) запропонованого методу	0,945	0,949	0,939	0,941	0,939	0,935	0,940	0,948	0,939	0,935
Рівень значущості α	0,01									
$t\text{-test}_{crit}$, df=9	2,821									
Гіпотеза H_0	$\mu \leq \mu_0$									
Гіпотеза H_1	$\mu > \mu_0$									
Lad, Sumit & Adamuthe, Amol [52]	$F1_{зважене}$ μ_0	t-test оцінки середньої $F1_{зваженого}$	p-значення оцінки середнього $F1_{зваженого}$	Середня точність μ_0	p-значення середньої точності	t-test середньої точності				
	0,887	35,947	$4,939 * 10^{-11}$	0,941	0,999	0,000				
Прийнята гіпотеза	H_1			H_0						
Ye et al. [125]	$F1_{зважене}$ μ_0	t-test оцінки середньої $F1_{зваженого}$	p-значення оцінки середнього $F1_{зваженого}$	Середня точність μ_0	t-test середньої точності	p-значення середньої точності				
	-	-	-	0.881	38,910	$2,431 * 10^{-11}$				

Продовження таблиці В.3

Прийнята гіпотеза	-			H_1		
Kocak, Anyur et al. [53]	$F1_{зважене}$ μ_0	t-test оцінки середньої $F1_{зваженого}$	p-значення оцінки середнього $F1_{зваженого}$	Середня точність μ_0	t-test середньої точності	p-значення середньої точності
	0,904	24,588	$1,459 * 10^{-9}$	0,905	23,346	$2,311* 10^{-9}$
Прийнята гіпотеза	H_1			H_1		
Yuk, Chang та Seo, Chang [51]	$F1_{зважене}$ μ_0	Середня точність μ_0	t-test середньої точності	p-значення середньої точності	t-test оцінки середньої $F1_{зваженого}$	p-значення оцінки середнього $F1_{зваженого}$
	0,926	0,924	11,025	$1.580 * 10^{-6}$	9,889	$3,927 * 10^{-6}$
Прийнята гіпотеза	H_1			H_1		

ДОДАТОК Г
ПРОГРАМНИЙ КОД РУШІЮ ВИСНОВКІВ ЕКСПЕРТНОЇ СИСТЕМИ НА ОСНОВІ ТЕОРІЇ ІГОР

```
import csv
import random
from copy import copy
import numpy as np
from itertools import count


def iterative_game(matrix_strat: np.ndarray, iterations_number):
    rows, cols = matrix_strat.shape
    maxmin = {(i, np.argmin(matrix_strat[i, :])): np.min(matrix_strat[i, :]) for i in range(rows)}
    minmax = {(np.argmax(matrix_strat[:, j]), j): np.max(matrix_strat[:, j]) for j in range(cols)}
    maxmin_idx = max(maxmin, key=maxmin.get)
    minmax_idx = min(minmax, key=minmax.get)
    for i in range(rows):
        min_el_row_j = np.argmin(matrix_strat[i])
        if i == np.argmax(matrix_strat[:, min_el_row_j]):
            print("Saddle point found: ", i, min_el_row_j)
            return matrix_strat[i, min_el_row_j], i, min_el_row_j
    strat_index_player_i = maxmin_idx[0]
    print("starting strategy is ", strat_index_player_i)
    strats_player_i = []
    strats_player_j = []
    cum_sum_i = np.zeros(matrix_strat.shape[1])
    cum_sum_j = np.zeros(matrix_strat.shape[0])
```

```
        for i in range(iterations_number):
            cum_sum_i = np.add(cum_sum_i, matrix_strat[strat_index_player_i, :])
            new_strat_j = np.argmin(cum_sum_i)
            cum_sum_j = np.add(cum_sum_j, matrix_strat[:, new_strat_j])
            strats_player_i.append(strat_index_player_i)
            strats_player_j.append(new_strat_j)
            strat_index_player_i = np.argmax(cum_sum_j)
            v_min = np.min(cum_sum_i) / (i+1)
            v_max = np.max(cum_sum_j) / (i+1)
            v_avg = (v_min + v_max) / 2
        return v_avg, count_percent(strats_player_i, matrix_strat.shape[0]),
count_percent(strats_player_j, matrix_strat.shape[1])

    def count_percent(strats, count):
        return [len([s for s in strats if s == i]) / len(strats) for i in range(count)]
```