Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
# Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження методів протидії змагальним атакам
на системи виявлення вторгнень
(тема)

Виконав:
студент 2 курсу, групи АМСЗІзм-22-1
Кручінін О.В.
(прізвище, ініціали)

Спеціальність: 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми: освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма: Адміністративний менеджмент
у сфері захисту інформації
(повна назва освітньої програми)

Керівник: завідувач кафедри ІКІ ім. В.В. Поповського
Лемешко О.В.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри ___ _____ ____ Лемешко О.В.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інфокомунікацій
(повна назва)
Кафедра Інфокомунікаційної інженерії імені В.В. Поповського
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 125 Кібербезпека
(код і повна назва)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Адміністративний менеджмент у сфері захисту інформації
(повна назва)

ЗАТВЕРДЖУЮ

Зав. кафедри_____

(підпис)

«____»_____2024 р.

**ЗАВДАННЯ**
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Кручініну Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів протидії змагальним атакам на системи виявлення вторгнень
затверджена наказом по університету від 06.11.2023 р. № 246Стз
2. Термін подання студентом роботи до екзаменаційної комісії: 15.01.2024 р.
3. Вихідні дані до роботи: обмежитись аналізом та дослідженням змагальних атак.
4. Перелік питань, які потрібно опрацювати в роботі:
1. Класифікація та аналіз мережних атак.
2. Огляд сучасних систем виявлення вторгнень.
3. Класифікація та характеристика методів протидії атакам на системи виявлення вторгнень.
4. Кількісний порівняльний аналіз методів протидії змагальним атакам на системи виявлення вторгнень.
5. Рекомендації щодо вдосконалення системи виявлення вторгнень.

5. Перелік графічного матеріалу із зазначенням креслень, плакатів, комп'ютерних ілюстрацій: Демонстраційний матеріал у вигляді ppt-презентації (титульний слайд; опис проблеми, об'єкт, предмет і мета дослідження; класифікація мережних атак; узагальнена структура типової системи виявлення вторгнень; класифікація методів протидії атакам на системи виявлення вторгнень; методика та результати порівняльного аналізу методів протидії змагальним атакам на системи виявлення вторгнень.

6. Консультанти розділів роботи

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| Основна частина | завідувач кафедри Лемешко Олександр Віталійович | | |

## КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Отримання завдання | | |
| 2 | Збір матеріалів для дослідження | | |
| 3 | Розробка 1 розділу | | |
| 4 | Розробка 2 розділу | | |
| 5 | Розробка 3 розділу | | |
| 6 | Розробка 4 розділу | | |
| 7 | Оформлення кваліфікаційної роботи | | |

Дата видачі завдання_____

Студент_____Кручінін О .В.__
(підпис) (прізвище та ініціали)

Керівник роботи_____завідувач кафедри ІКІ ім. В.В. Поповського

_____Лемешко О.В.____
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ

Пояснювальна записка: 105 с., 24 рис., 16 табл., 1 додаток, 81 джерел.

СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ, МАШИННЕ НАВЧАННЯ, ЗМАГАЛЬНІ АТАКИ, МЕРЕЖНІ АТАКИ, МЕТОДИ ЗАХИСТУ, ПОКАЗНИКИ ЕФЕКТИВНОСТІ, КІЛЬКІСНЕ ПОРІВНЯННЯ.

Об'єкт дослідження – процес вдосконалення систем виявлення вторгнень.

Предмет дослідження – методи протидії змагальним атакам.

Мета роботи – підвищення ефективності застосування систем виявлення вторгнень за рахунок використання кількісного порівняльного аналізу.

Методи досліджень – емпіричний аналіз, аналіз та порівняння.

Системи виявлення вторгнень є невід’ємною складовою сучасних систем кібербезпеки. Збільшення кількості нових атак потребує розвитку систем на основі виявлення аномалій. Ці системи створюються із застосуванням алгоритмів машинного навчання, які мають вразливості до змагальних атак.

У роботі виконані класифікація та аналіз сучасних систем виявлення вторгнень. Особлива увага приділена застосуванню методів машинного навчання для систем виявлення вторгнень. Наведена класифікація та приклади реалізації змагальних атак на системи виявлення вторгнень на основі машинного навчання, з детальним описом механізмів їх реалізації. Представлена класифікація та характеристика методів протидії цим атака.

Розроблено модель кількісного порівняльного аналізу методів протидії змагальним атакам, а також рекомендації щодо вдосконалення систем виявлення вторгнень.

# ABSTRACT

The report contains: 105 p., 24 fig., 16 tables, 1 annexes, 81 references.

INTRUSION DETECTION SYSTEM, MACHINE LEARNING, ADVERSARIAL ATTACKS, NETWORK ATTACKS, DEFENSE METHODS, PERFORMANCE INDICATORS, QUANTITATIVE COMPARISON.

An object of the study is process of upgrading systems to detect invasion.

A subject of the research is methods of countering adversarial attacks.

An aim of the work is to increase the efficiency of intrusion detection systems by means of quantitative comparative analysis.

Research methods are empirical analysis, formalization, and comparison.

Intrusion detection systems are an integral part of modern cybersecurity systems. The growing number of new attacks requires the development of anomaly detection systems. These systems are created using machine learning algorithms that have vulnerabilities to adversarial attacks.

The paper classifies and analyzes modern intrusion detection systems. Particular attention is paid to the application of machine learning methods for intrusion detection systems. The classification and examples of adversarial attacks on intrusion detection systems based on machine learning are presented, with a detailed description of the mechanisms for their implementation. The classification and characterization of methods for countering these attacks are presented.

A model for quantitative comparative analysis of methods for countering adversarial attacks has been developed, as well as recommendations for improving intrusion detection systems.

ЗМІСТ

# ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

СВВ – система виявлення вторгнень
ACC – Accuracy
AHP – Analytical Hierarchy Process
AIDAE – Anti-Intrusion Detection AutoEncoder
AIDS – Anomaly-based Intrusion Detection System
ANN – Artificial Neural Network
ARP – Address Resolution Protocol
ATT&CK – Adversarial Tactics, Techniques & Common Knowledge
CAPEC – Common Attack Pattern Enumerations and Classifications
$C_{ID}$ – Intrusion Detection Capability
CNN – Convolutional Neural Networks
CRLF – Carriage Return
DBN – Deep Belief Network
DNS – Domain Name System
DoS (DDoS) – Denial of Service (Distributed Denial of Service)
DR – Detection Rate
DT – Detection Time
EIR – Evasion Increase Rate
E-mail – Electronic mail
FAR – False Alarm Rate
FN – False Negative
FNR – False Negative Rate
FP – False Positive
FPR – False Positive Rate
GA – Genetic Algorithm
GANs – Generative Adversarial Networks
HIDS – Host-based Intrusion Detection System
HTTP – Hyper Text Transfer Protocol
ICMP – Internet Control Message Protocol
IDS – Intrusion Detection System
IP – Internet Protocol
KNN – K-Nearest Neighbo

LAN – Local Area Network
LDAP – Lightweight Directory Access Protocol
LR – Logistic Regression
MCC – Matthews Correlation Coefficient
MCUT – Multi-attribute Utility Theory
ML – Machine Learning
MLP – Multi-Layer Perceptron
MTTC – Mean Time-to-Compromise
NIDS – Network Intrusion Detection System
NNIDS – Network Node Intrusion Detection System
OSI – Open System Interconnection
PPP – Point-to-Point Protocol
PPV – Positive Predictive Value
Pr – Precision
PSO – Particle Swarm Optimization
RBM – Restricted Boltzmann Machines
RNN – Recurrent Neural Networks
ROC – Receiver Operating Characteristic
RsT – Response Time
SIDS – Signature-based Intrusion Detection System
SQL – Structured Query Language
SVM – Support Vector Machines
TANTRA – Timing-Based Adversarial Network Traffic Reshaping
TCP – Transmission Control Protocol
Telnet – Teletype Network
TN – True Negative
TNR – True Negative Rate
TP – True Positive
TPR – True Positive Rate
TT – Training Time
UDP – User Datagram Protocol
WEB – World Wide Web
XPath – XML Path Language
XSS – Cross-Site Scripting

# ВСТУП

У сучасному світі, де інформаційні технології пронизують усі сфери людської діяльності, кібербезпека стає одним з пріоритетних напрямків. Зловмисники неухильно вдосконалюють свої методи атак, що змушує фахівців з кібербезпеки постійно шукати нові рішення для захисту інформаційних систем.

Так за результатами звіту оперативного центру реагування на кіберінциденти Державного центру кіберзахисту Держспецзв'язку за III квартал 2023 року, зафіксовано та оброблено безпосередньо аналітиками безпеки 355 кіберінцидентів. При цьому порівняно з II кварталом 2023 року кількість зареєстрованих кіберінцидентів зросла на 46 %. Також протягом III кварталу 2023 року було зафіксовано 202 кібератаки, які були ініційовані хакерськими угрупованнями [1].

Наслідками мережних атак можуть стати фінансові, репутаційні та інші втрати для великої кількісті компаній, установ, окремих осіб та країни в цілому.

Одним із ефективних засобів протидії мережним атакам є системи виявлення вторгнень. Однак, зі збільшенням складності та різноманітності атак, системи виявлення вторгнень стикаються зі складними задачами адаптації та ефективної протидії. Для вирішення цих задач застосовують методи машинного навчання (англ. Machine Learning, ML) для системи виявлення вторгнень. Це відкриває нові можливості для підвищення ефективності виявлення атак, але водночас створює потенційні вразливості. Наявність цих вразливостей створює умови для реалізації змагальних атак на системи виявлення вторгнень на основі машинного навчання. Ці атаки спрямовані маніпулювання вхідними даними або алгоритмами функціонування систем виявлення вторгнень, що створює нові вектори атак, які можуть залишитися непоміченими. Таким чином, є актуальною задача розвитку методів захисту систем виявлення вторгнень від змагальних атак.

Метою роботи є підвищення ефективності застосування систем виявлення вторгнень за рахунок використання кількісного порівняльного аналізу.

Для вирішення поставленої задачі, в першому розділі кваліфікаційної роботи проведено аналіз класифікацій мережних атак. Виконано огляд сучасних систем виявлення вторгнень та методів машинного навчання для цих систем. Обґрунтовано вплив змагальних атак на машинне навчання.

У другому розділі роботи виконана класифікація атак на системи машинного навчання та виконано аналіз механізмів реалізації змагальних атак на системи

виявлення вторгнень на основі машинного навчання. За результатами аналізу характеристик методів протидії атакам на системи виявлення вторгнень, було зроблено висновок про необхідність розробки методики кількісної оцінки ефективності цих методів.

В третьому розділі роботи, за результатами аналізу впливу змагальних атак та методів протидії на показники систем виявлення вторгнень, запропонована система показників для оцінки ефективності методів протидії. Розроблена модель кількісного порівняльного аналізу методів протидії змагальним атакам на базі методу визначення вагових коефіцієнтів на основі функції втрати ефективності систем.

Четвертий розділ роботи присвячений питанням вдосконалення систем виявлення вторгнень.

Окремі результати роботи доповідались на XVIII Міжнародній конференції «Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості» [2, 3, 4].

# 1 АНАЛІЗ МЕРЕЖНИХ АТАК ТА ОГЛЯД СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ

## 1. Класифікація мережних атак

Згідно з НД ТЗІ 1.1-003-99 [5], атака – це спроба реалізації загрози. У свою чергу, загроза — це будь-які обставини або події, що можуть бути причиною порушення політики безпеки інформації та/або нанесення збитків автоматизованій системі. Таким чином, мережна атака – це спроба несанкціонованого доступу до мережі комп'ютерів, системи чи іншого мережного ресурсу, що може бути причиною отримання незаконного доступу, руйнування, модифікації або крадіжки інформації, завдання шкоди або перешкоджання нормальному функціонуванню системи чи мережі. В деяких джерелах окремо визначають віддалену мережеву атаку, як інформаційний руйнівний вплив на розподілену обчислювальну систему, що здійснюється програмно через канали зв'язку.

В різних джерелах є декілька варіантів визначення мережних атак:

– це спроба впливати на видалений комп'ютер з використанням програмних методів;

– це спроба несанкціонованого доступу до мережі комп'ютерів, системи чи іншого мережного ресурсу з метою отримання незаконного доступу, руйнування, модифікації або крадіжки інформації, завдання шкоди або перешкоджання нормальному функціонуванню системи чи мережі;

– це дія, метою якої є захоплення контролю (підвищення прав) над віддаленою локальною обчислювальною системою [6].

Класифікація мережних атак – це процес групування мережних атак у певні підгрупи. Класифікація мережних атак необхідна для можливості чіткого розуміння механізмів реалізації атак, аналізу можливих наслідків та розробки методів та засобів захисту, а також визначення схожих типів атак у майбутньому.

Класифікація мережних атак є неодмінною умовою для вироблення чіткого розуміння атак. Існує достатньо велика кількість класифікаторів мережних атак, які схожі між собою, проте базуються на різних таксономіях атак. Всі вони мають свої переваги та недоліки, які треба враховувати при виборі відповідної класифікації мережних атак [2].

Класифікація мережних атак повинна відповідати певним вимогам [7, 8].

1) Прийнятність. Таксономія повинна бути розроблена таким чином, щоб вона стала загальноприйнятою та ґрунтуватися на попередніх роботах, які є загальновизнаними.

2) Зрозумілість. Класифікація повинна бути легкою для розуміння тими, хто працює у сфері комп'ютерних мереж, безпеки або суміжних галузях.

3) Повнота. Для того, щоб класифікація була повною, всі мережні атаки повинні бути включені в цю класифікацію і мати певну категорію.

4) Взаємовиключення. Ця вимога відносить кожну загрозу до одного класу.

5) Повторюваність. Класифікація повинна бути повторюваною.

6) Однозначність. Групування має бути чітко визначене таким чином, щоб не виникало жодних сумнівів щодо того, до якої категорії слід віднести мережну атаку.

7) Корисність. Корисна класифікація може бути використана в сфері комп'ютерних мереж, кібербезпеки або в інших суміжних сферах.

Одна з перших класифікацій мережних атак була запропонована Пітером Меллом (Peter Mell) [9]. По суті, це спрощена класифікація, яка відображає найбільш типові атаки.

1) Віддалене проникнення (remote penetration). Атаки, які дають змогу реалізувати віддалене керування комп'ютером через мережу.

2) Локальне проникнення (local penetration). Атаки, що призводять до отримання несанкціонованого доступу до вузлів, на яких вони ініційовані.

3) Віддалена відмова в обслуговуванні (remote denial of service). Атаки, що дають можливість порушити функціонування системи або перенавантажити комп'ютер через мережу (зокрема, через Інтернет).

4) Локальна відмова в обслуговуванні (local denial of service). Атаки, що дають змогу порушити функціонування системи або перенавантажити комп'ютер, на якому їх ініційовано.

5) Сканування мережі (network scanning). Аналіз топології мережі та активних сервісів, доступних для атаки.

6) Використання сканерів уразливостей (vulnerability scanning). Сканери вразливостей призначені для пошуку вразливостей на локальному або віддаленому комп'ютері.

7) Злам паролів (password cracking). Для цього використовують програмні засоби, що підбирають паролі користувачів.

8) Пасивне прослуховування мережі (sniffing). Пасивна атака, спрямована на розкриття конфіденційних даних, зокрема ідентифікаторів і паролів доступу.

Перші чотири класи атак розрізняють переважно за кінцевим результатом (або метою реалізації), а решту — за способом їх здійснення.

Така класифікація не дає змоги визначати елементи мережі, схильні до впливу тієї чи іншої атаки, а також наслідки, до яких може призвести успішна реалізація атак.

Говард [10] пропонує таксономію інцидентів, яка класифікує атаки за подіями, тобто атакою, спрямованою на певну ціль, що має призвести до зміни стану. Подія включає в себе дію та ціль. Цей підхід дозволяє описати всі кроки, які охоплює атака, і те, як вона розвивається. Атака складається з п'яти частин, які зловмисник виконує для досягнення несанкціонованого результату. Цими кроками є: інструменти, вразливість, дія, ціль і несанкціонований результат:

– інструмент – це механізм, який використовується для здійснення атаки;
– вразливість – це тип експлойту, який використовується для здійснення атаки;
– дія – це метод, використаний зловмисником для здійснення атаки;
– ціль – це намір, який зловмисник намагається реалізувати;
– несанкціонований результат – це зміна стану, спричинена атакою.

Такий підхід забезпечує достатньо інформативний опис атак, але недостатньо деталізований.

Гансман і Хант [11] запропонували таксономію з чотирма унікальними вимірами, які забезпечують цілісну класифікацію, що охоплює мережні та комп'ютерні атаки. Їх таксономія забезпечує узгодженість у формулюваннях опису атак. Перший вимір – це вектор атаки, що використовується для класифікації атаки. Другий вимір класифікує ціль атаки. Третій вимір складається з класифікаційного номера вразливості або критеріїв з таксономії Говарда. Четвертий вимір визначає корисне навантаження або наслідки, до яких призводить атака. У межах кожного виміру надаються різні рівні інформації для надання деталей атаки. Ця таксономія не дозволяє виконувати класифікацію змішаних атак, а також не визначає вразливості.

Іншу таксономію під назвою VERDICT (Validation Exposure Randomness Deallocation Improper Conditions Taxonomy) запропонував Daniel Lough [12].

Замість деревоподібної таксономії тут використовуються чотири характеристики атак:

– неправильна перевірка;
– неправильна вразливість;
– неправильна випадковість;
– неправильний розподіл.

В цій таксономії відсутня класифікація за типом атаки та її важко застосовувати для нових атак.

Більш повна таксономія кібератак має назву AVOIDIT (Attack Vector, Operational Impact, Defense, Information Impact, and Target) [13]. В цій таксономії використовується п'ять основних класифікаторів для характеристики атаки:

– за вектором атаки;
– за метою атаки;
– за оперативним впливом;
– за інформаційним впливом;
– за захистом.

Ця п'ята категорія, класифікація за захистом, може використовуватися для пом'якшення або усунення атаки. Ця таксономія ефективно класифікує змішані атаки та має прикладний характер.

В класичних джерелах, мережні атаки класифікують за наступними ознаками (рис 1.1) [14].

1) За характером впливу:

– пасивні, під час яких атакуючий лише спостерігає за мережею та намагається здобути інформацію без активної взаємодії;
– активні, під час яких атакуючий активно взаємодіє з цільовою системою.

2) За метою впливу:

– порушення конфіденційності інформації або ресурсів системи;
– порушення цілісності інформації (приклад: впровадження хибного об’єкта);
– порушення працездатності (доступності), направлені на призупинення або обмеження доступу до ресурсів або послуг.

3) За умовою початку здійснення впливу. Віддалений вплив, також як і будь-який інший, може здійснюватися тільки за певних умов. У розподілених автоматизованих системах існують три види умов початку здійснення віддаленої атаки:

– напад за запитом від об’єкта, що атакується. У цьому випадку атакуючий очікує передачі від потенційного об’єкта атаки запиту певного типу, який і буде умовою початку здійснення впливу;

– напад за настанням очікуваної події на об’єкті, що атакується. У цьому випадку атакуючий здійснює постійне спостереження за станом операційної системи потенційного об’єкта атаки і при виникненні певної події в цій системі починає вплив;

– безумовний напад. У цьому випадку початок здійснення атаки безумовно стосовно потенційного об’єкту нападу, тобто напад здійснюється негайно і безвідносно до стану системи об’єкту, що атакується.

4. За наявності зворотного зв’язку з об’єктом, що атакується:

– зі зворотним зв’язком;

– без зворотного зв’язку (односпрямована атака).

Віддалена атака, яка здійснюється при наявності зворотного зв’язку з об’єктом, що атакується, характеризується тим, що на деякі запити, передані на об’єкт, що атакується, потрібно одержати відповідь, а, отже, між атакуючим і об’єктом нападу існує зворотний зв’язок, який дозволяє атакуючому адекватно реагувати на всі зміни, що відбуваються на об’єкті, що атакується. Подібні віддалені атаки найбільш характерні для розподілених автоматизованих систем.

На відміну від атак зі зворотним зв’язком, віддаленим атакам без зворотного зв’язку не потрібно реагувати на зміни, які відбуваються на об’єкті, що атакується. Атаки даного виду звичайно здійснюються передачею одиночних запитів на об’єкт, що атакується, відповіді на які атакуючому не потрібні (приклад: відмова в обслуговуванні).

5. За розташуванням суб’єкта атаки щодо об’єкта, що атакується:

– внутрішньо-сегментні;

– міжсегментні.

З погляду віддаленої атаки надзвичайно важливо, як по відношенню одне до одного розташовуються суб’єкт і об’єкт атаки, тобто чи вони перебувають в одному або в різних сегментах вони перебувають. Міжсегментний віддалений напад представляє набагато більшу небезпеку, ніж внутрішньо-сегментний. Це пов’язане з тим, що у випадку міжсегментного нападу об’єкт та безпосередньо атакуючий можуть перебувати на відстані багатьох тисяч кілометрів один від одного, що може суттєво перешкодити заходам щодо відбиття атаки.


Рисунок 1.1 – Класифікація мережних атак [14]

6. За рівнем еталонної моделі ISO/OSI, за яким здійснюється вплив.

Будь-який мережний протокол обміну, як і будь-яку мережну програму, можна з тим або іншим ступенем точності віднести до відповідного рівня еталонної моделі OSI.

Така класифікація дуже узагальнено описує мету атаки та можливі наслідки.

Одною з альтернативних класифікацій мережних атак є класифікації на основі послідовних запитань [15]. Сутність цієї класифікації побудована на послідовних запитаннях: "Хто", "Де", "Як" і "Що". До одного типу атак можна віднести атаки, які мають однаковий тип зловмисників (Хто), однакові місця, де були розпочаті атаки (Де), використання схожих інструментів для атаки (Як), а також ступінь і тип впливу атаки (Що).

До переліку зловмисників відносяться.

1) Joke – здійснює мережну атаку в першу чергу для навчання та/або самоствердження.

2) White-hat hackers – здійснюють мережну атаку з метою з'ясування вразливостей мережі, яку атакують, і повідомляють про це мережному адміністратору.

3) Black-hat hackers – здійснюють мережну атаку, використовуючи певні вразливості мережі та пошкоджуючи або викрадаючи інформацію з атакованої мережі.

4) Little sisters – організації або компанії, які здійснюють атаки на мережі конкурентів з метою отримання фінансової вигоди.

5) Big brothers – уряди або організації, пов'язані з урядом.

Місця, де були ініційовані атаки, поділяють:

– на основі хоста, коли атака запускається з комп'ютера або будь-якого пристрою, який має мережне підключення;

– на основі мережі, коли атака може бути запущена з декількох пристроїв, з'єднаних між собою.

За масштабом атаки визначають:

– об'єктні – об'єктом атаки є окремий об'єкт у реальному житті, який має підключення до мережі;

– хост-орієнтовані – ціль атаки знаходиться на комп'ютерному терміналі;

– локальні сегментні – ціль атаки знаходиться в сегменті мережі, який має багато хостів, з'єднаних між собою;

– сегментно-орієнтовані - цей тип намагається атакувати ядро глобальної мережі;

– на основі бездротової мережі - ціль атаки знаходиться в мобільній мережі.

Інструменти для атаки визначаються:

– платформою: Software, Hardware, Embedded hardware, Mobile;

– каналами, які використовуються: Legacy network equipment ports, Undefined network equipment ports, Virtualization channel, User-to-network channel, Network-to-network channel.

При визначенні типу атаки аналізують параметри, які можуть мати аномальні значення: активність системи, обсяг трафіку, запити.

Відстежуючи процес мережної атаки від початку до кінця, цей підхід дозволяє проаналізувати ланцюг здійснення атаки, але теж є достатньо узагальненим.

## 1.2 Аналіз мережних атак

Аналіз мережних атак є важливою складовою в області розробки та вдосконалення систем кібербезпеки. Це обумовлено зростанням кількості та складності мережних загроз. Аналіз мережних атак дозволяє виявити та класифікувати типи загроз, розкривати їхні характеристики для вдосконалення захисту.

Атака типу «відмова в обслуговуванні» (англ. Denial of Service, DoS) — це атака, спрямована на блокування комп'ютера або мережі, що робить їх недоступними для цільових користувачів. DoS-атаки досягають цього, створюючи для цілі надмірний трафік або надсилаючи їй інформацію, яка викликає збій. В обох випадках DoS-атака позбавляє авторизованих користувачів доступу до послуги чи ресурсу [16].

DoS-атаки часто націлені на веб-сервери відомих організацій, таких як банківські, комерційні та медіа-компанії, або урядові та торгівельні організації. Хоча DoS-атаки зазвичай не призводять до крадіжки або втрати значної інформації чи інших активів, але вони можуть призводити до репутаційних та фінансових втрат. Крім цього, DoS-атаки можуть бути тільки початком більш складної багаторівневої атаки.

Існує два основних методи DoS-атак: перенасичення служб трафіком (flood-атаки) або збій служб. Flood-атаки виникають, коли система отримує занадто

забагато трафіку для буферизації сервера, що призводить до їх уповільнення та зрештою зупинки. До розповсюджених flood-атак належать:

Buffer overflow attacks (атаки переповнення буфера) – найпоширеніша DoS-атака. Сутність цієї атаки полягає в тому, щоб надсилати більше трафіку на мережну адресу, ніж система може обробити. Можливі наступні варіанти реалізації:

ICMP flood – використовує неправильно налаштовані мережні пристрої, надсилаючи підроблені пакети, які виконують команду ping для не лише однієї конкретної системи, а для великої кількості, які знаходяться у цільовій мережі. За рахунок цього збільшується трафік в мережі.

SYN flood – зловмисник надсилає запит на підключення до сервера, але так і не завершує сеанс. Продовжується до тих пір, поки всі відкриті порти не будуть перенасичені запитами, і жоден з них не буде доступний для підключення авторизованих користувачів.

Інші DoS-атаки використовують вразливості, які спричиняють збій цільової системи або служби. Під час цих атак надсилаються вхідні дані, які використовують помилки в цілі, які згодом призводять до збою або серйозної дестабілізації системи, тому до неї неможливо отримати доступ або використовувати.

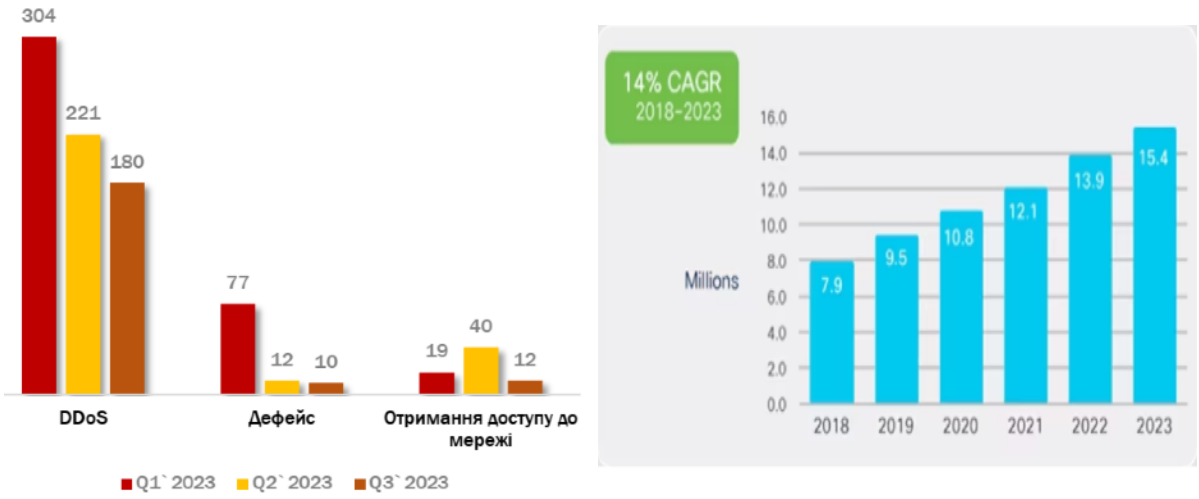
Різновидом DoS-атаки є розподілена атака типу «відмова в обслуговуванні» (англ. Distributed Denial of Service, DDoS). DDoS-атака відбувається, коли кілька систем організовують синхронізовану DoS-атаку на одну ціль. При цьому, як правило, використовуються заздалегідь скомпрометовані ботнети. DDoS-атака дозволяє зловмиснику значно збільшити ефективність атаки та створити додаткові труднощі для його ідентифікації. Слід зазначити, що незважаючи на існування великої кількісті механізмів захисту від більшості форм DoS-атак, насамперед саме DDoS-атаки залишаються актуальними.

За результатами звіту оперативного центру реагування на кіберінциденти Державного центру кіберзахисту Держспецзв'язку за ІІІ квартал 2023 року [1] та за результатами досліджень одними із розповсюджених типом атаки є DDoS (рис 1.2).

Наступною групою атак є атаки ін’єкцій, які здебільш направлені на атаки веб-додатків. До цієї групи можна віднести:

– введення коду;
– SQL ін’єкція;
– введення команди;

– міжсайтовий скриптинг (XSS-атаки);
– ін’єкція XPath;
– ін’єкція поштової команди;
– ін’єкція CRLF;
– ін’єкція заголовка хосту;
– ін’єкція LDAP.




а) б)

Рисунок 1.2 – Динаміка DDOS атак
а) – динаміка активності хакерських угруповань за типами атак,
б) – кількість DDoS-атак за результатами досліджень Cisco

XSS-атаки та SQL-ін’єкції є одними із найпоширеніших видів атак на веб-додатки на веб-сайти. Вразливості до цих атак радикально знижують рівень захищеності інформації в системі або в додатку [17].

XSS-атаки розділяють на два види: пасивні та активні. Пасивні атаки відбуваються, коли жертва переходить за спеціальним чином згенерованим посиланням, яке містить в собі зловмисний код (часто зашифрований). Для реалізації атак цього виду зазвичай використовується соціальна інженерія. Активні XSS-атаки є більш небезпечними. При такій атаці зловмисний код впроваджується в базу даних або який-небудь файл на сервері. Таким чином, всі відвідувачі сайту автоматично стають жертвами. Інтеграція може бути здійснена зокрема за допомогою SQL-ін'єкції.

SQL-ін'єкція – це тип уразливості, що виникає у веб-додатках, які керуються базами даних. Атаки SQL-ін'єкцій можливі через відсутність належної перевірки вхідних даних та заходів безпеки у веб-додатках, які використовують бази даних на мові структурованих запитів (англ. Structured Query Language, SQL).

Зловмисник користується відсутністю належної перевірки вхідних даних, впроваджуючи шкідливий код, в який потім виконується як частина SQL-запиту до бази даних.

Таким чином, атаки SQL-ін'єкції виконують процес ін'єкції в цільову базу даних, тоді як XSS-атаки впроваджують код зі шкідливими функціями, який вводиться в систему у вигляді JavaScript.

Слід зазначити, що згідно висновків, які зроблені в роботі [18], саме атаки типу «відмова в обслуговуванні» та ін’єкції є найбільш актуальними.

Ще одна група атак – це spoofing атаки. У загальному сенсі, спуфінг – це маскування повідомлення або ідентифікаційної інформації таким чином, щоб вона виглядала пов’язаною з надійним авторизованим джерелом. Спуфінг може приймати різноманітні форми [19]:

– IP: підробка IP-адреси джерела, щоб замаскувати відправника;
– ARP: пов'язує MAC-адресу зловмисника з цільовою IP-адресою;
– E-mail: підробляє заголовки, щоб імітувати відправника;
– WEB: створює підроблені WEB-сайти, які імітують законні WEB-сайти для викрадення облікових даних;
– DNS: перенаправляє трафік на підроблені WEB-сайти.

Spoofing атаки можуть використовувати для маскування джерела інших атак.

В окрему групу можна виділити атаки, які базуються на використанні спеціалізованих програм. До них відносяться Sniffing-атаки [20]. Ці атаки реалізуються шляхом перехоплення мережного трафіку за допомогою аналізаторів пакетів, які можуть незаконно отримати доступ і зчитувати дані, які не зашифровані.

Існують різні типи Sniffing-атак.

1) LAN Sniff – сніффер атакує внутрішню локальну мережу та сканує всю IP-адресу, отримуючи доступ до активних хостів, відкритих портів, сканування сервера тощо.

2) Protocol Sniff – атаки сніффера відбуваються на основі мережного протоколу, який використовується. Можуть використовуватися різні протоколи, наприклад ICMP , UDP, Telnet, PPP, DNS або інші протоколи.

3) ARP Sniff – атаки ARP Poisoning або атаки підробки пакетів відбуваються на основі даних, отриманих для створення карти IP-адрес і пов’язаних MAC-адрес.

4) TCP Session stealing (викрадання сеансу TCP) – викрадення сеансу TCP використовується для моніторингу та отримання деталей трафіку (як номер порту, тип служби, порядкові номери TCP, дані) між IP-адресою джерела та призначення.

5) Application-level sniffing – програми, що працюють на сервері, піддаються атаці з метою планування подальшої атаки.

6) Web password sniffing (перегляд WEB-паролів) – HTTP-сеанси, створені користувачами, перехоплюються сніферами, щоб отримати ідентифікатор користувача, пароль та іншу конфіденційну інформацію.

Представлений перелік, безумовно, не є повним, але включає основні мережні атаки. Слід зазначити, що всі вони мають різні механізми реалізації та прояви, що ускладнює задачу їх виявлення.

## 1.3 Огляд сучасних систем виявлення вторгнень

Система виявлення вторгнень (СВВ) (англ. Intrusion Detection System, IDS) -програмний або апаратний засіб, призначений для виявлення фактів несанкціонованого доступу до комп'ютерної системи чи мережі або несанкціонованого управління ними.

Існує три основні типи засобів для виявлення вторгнень які можуть бути складовими частинами однієї системи [21]:

– система виявлення вторгнень у мережу (англ. Network Intrusion Detection System, NIDS);

– система виявлення вторгнень мережного вузла (англ. Network Node Intrusion Detection System, NNIDS);

– система виявлення вторгнень на хост (англ. Host-based Intrusion Detection System, HIDS).

Система виявлення вторгнень у мережу зазвичай розгортається або розміщується в стратегічних точках по всій мережі, призначена для охоплення тих місць, де трафік найбільш імовірно вразливий для атак. Вона пасивно переглядає мережний трафік, що надходить через точки мережі, на якій вона розгорнута. Їх можна відносно легко захистити та зловмисникам їх важко виявити. Ці системи аналізують велику кількість мережного трафіку, тому вони іноді мають низьку точність.

Система виявлення вторгнень мережного вузла схожа на NIDS, але вона одночасно застосовується лише до одного хоста, а не до всієї підмережі.

Система виявлення вторгнень працює на всіх пристроях у мережі з доступом до Інтернету та інших частинах корпоративної мережі. HIDS має певні переваги перед NIDS завдяки своїй здатності ретельніше контролювати внутрішній трафік, а також працює як друга лінія захисту від шкідливих пакетів, які NIDS не вдалося виявити.

На сьогодні існує достатньо велика кількість IDS, в більшості випадках у вигляді програмних засобів [22 – 24]. Перелік та порівняльна характеристика найбільш популярних із них наведена в таблиці 1.1.

За результатами аналізу характеристик ISD можна зробити висновок, що основні відмінності полягають у наявності відкритого коду та безкоштовної версії системи. Крім того, не всі IDS забезпечують хмарну інтеграцію.

Таблиця 1.1 – Порівняльна характеристика IDS

| № | Функція / IDS | Моніторинг у реальному часі | Керування журналами | Виявлення на основі сигнатур | Виявлення на основі аномалій | Відкритий код | Хмарна інтеграція | Доступна безкоштовна версія |
|---|---|---|---|---|---|---|---|---|
| 1 | SolarWinds Security Event Manager | так | так | так | так | ні | так | ні |
| 2 | ManageEngine EventLog Analyzer | так | так | так | так | ні | так | ні |
| 3 | ManageEngine Log360 | так | так | так | так | ні | так | ні |
| 4 | ESET Protect | так | так | так | так | ні | так | ні |
| 5 | Snort | так | так | так | так | так | ні | так |
| 6 | OSSEC | так | так | так | так | так | ні | так |
| 7 | CrowdSec | так | так | так | так | так | так | так |
| 8 | Suricata | так | так | так | так | так | ні | так |
| 9 | Zeek | так | так | так | так | так | ні | так |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 10 | Security Onion | так | так | так | так | так | ні | так |

Безумовно, кожна із представлений IDS має свої додаткові переваги та недоліки. Окремо слід виділити систему ESET Protect, яка сертифікована Державною службою спеціального зв'язку та захисту інформації України. Більш розгорнутий аналіз цих та інших систем виявлення вторгнень наведено в [25].

Слід зазначити, що всі представлені IDS можуть виявляти атаки на основі аналізу як сигнатур, так і аномалій.

IDS на основі сигнатур (англ. Signature-based Intrusion Detection System, SIDS) полягає у виявленні атак за допомогою бази даних, яка заздалегідь сформована на базі вже відомих атак. Тобто, насамперед, SIDS призначені для виявлення вже відомих атак. Узагальнена структура IDS на основі сигнатур наведена на рис. 1.3.

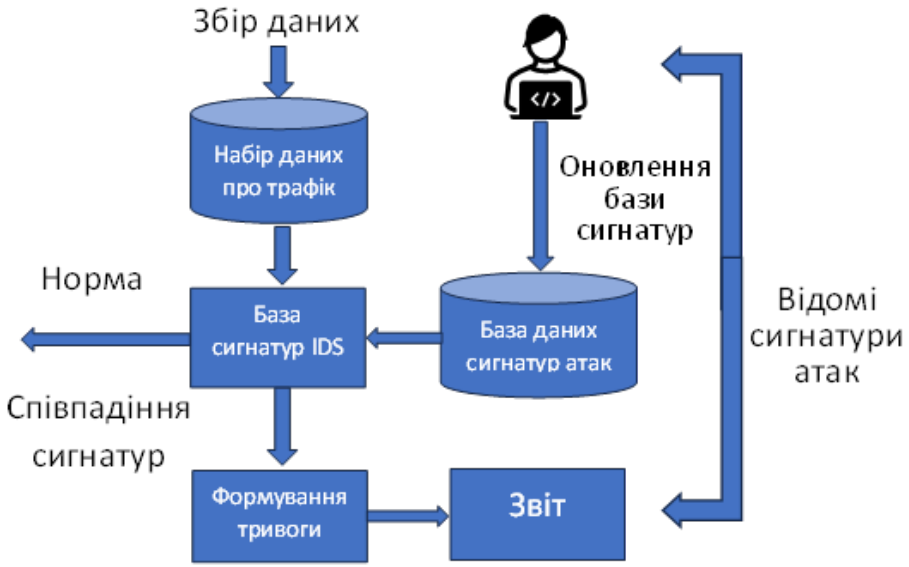



Рисунок 1.3 – Узагальнена структура IDS на основі сигнатур

Основними перевагами SIDS є:

– швидкодія;
– мала кількість хибних тривог;
– порівняно проста реалізація.

Основними недоліками SIDS є:

– необхідність регулярно оновлювати базу сигнатур;
– не ефективні для виявлення нових атак;
– не ефективні для багатоетапних атак.

IDS на основі аномалій (англ. Anomaly-based Intrusion Detection System, AIDS) може забезпечити детектування невідомих вторгнень за допомогою аналізатора. Цей аналізатор здатен навчатися і виявляти вторгнення за

відхиленням від нормальної поведінки. Узагальнена структура IDS на основі аномалій наведена на рис. 1.4.

Основними перевагами AIDS є:

– здатність виявляти нові види атак;
– результати роботи можна використовувати для створення сигнатур.

Основними недоліками AIDS є:

– потребує навчання;
– висока вірогідність хибних спрацювань;
– мають місце некласифіковані оповіщення;
– не може обробляти зашифрований трафік.

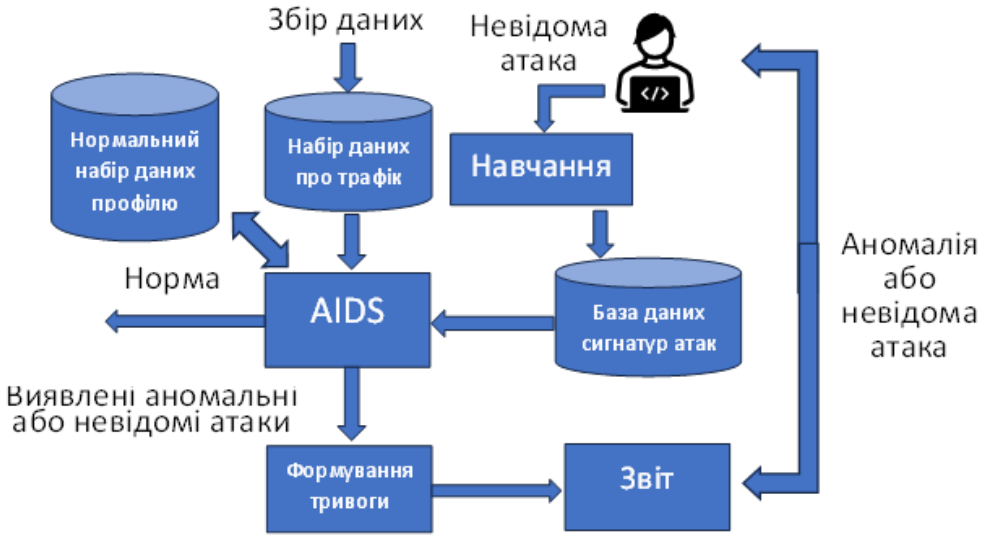



Рисунок 1.4 – Узагальнена структура IDS на основі аномалій

Підходи, які засновані на аномаліях, поділяються на три методи, а саме:

– на основі інтелектуального аналізу даних;
– на основі статистики;
– на основі машинного навчання.

В останній час саме використання машинного навчання вважається найбільш перспективним напрямом для вдосконалення IDS.

До основних методів машинного навчання відносяться:

– дерева рішень;
– мережі Баєса;
– нейронні мережі;
– імунні мережі;
– генетичні алгоритми;
– метод опорних векторів.

Таблиця 1.2 – Характеристика методів виявлення атак

| | Аномалії/ | Верифікованість | Адаптивність | Стійкість |
|---|---|---|---|---|
| Поведінкові методи | | | | |
| Статичний аналіз | +/- | - | + | - |
| Вайвлет-аналіз | +/- | - | + | - |
| Кластерний аналіз | +/+ | - | + | - |
| Спектральний аналіз | +/- | - | + | - |
| Фрактальний аналіз | +/- | - | + | - |
| Аналіз ентропії | +/- | + | + | - |
| Біометрія поведінки | +/- | - | + | - |
| Методі які засновані на знаннях | | | | |
| Експертні системи | +/+ | + | + | + |
| Методи на | -/+ | + | - | - |
| Аналіз систем станів | -/+ | + | - | + |
| Сигнатурний аналіз | -/+ | + | - | + |
| Мережі Петрі | -/+ | + | - | + |
| Методи на основі машинного навчання | | | | |
| Дерева рішень | +/+ | + | - | - |
| Мережа Баєса | +/+ | - | + | + |
| Генетичні алгоритми | +/+ | - | + | + |
| Нечітка логіка | +/+ | + | + | + |
| Нейроні мережі | +/+ | - | + | - |
| Імунні мережі | +/+ | - | + | - |
| Метод опорних | +/+ | - | + | - |
| Роєві алгоритми | +/+ | + | + | - |
| Регресивний аналіз | +/+ | - | + | - |

Згідно результатів порівняльного аналізу методів виявлення кібератак (таблиця 1.2) саме ці методи мають слабку стійкість [26]. Це обумовлено різними факторами, у тому числі із вразливістю методів машинного навчання до змагальних атак.

## 1.4 Огляд методів машинного навчання для систем виявлення вторгнень

Оскільки мережне середовище швидко змінюється, постійно з’являються нові атаки. Таким чином, необхідно розробляти IDS, які можуть виявляти невідомі атаки [27]. Щоб вирішити цю проблему, створюються IDS за на базі методів машинного навчання. Машинне навчання – це тип технології штучного інтелекту, який може автоматично виявляти корисну інформацію з масивних наборів даних [28]. IDS на основі машинного навчання можуть досягти задовільних рівнів

виявлення, коли доступні достатні навчальні дані, а моделі машинного навчання мають достатню можливість узагальнення для виявлення різних варіантів атак і нових атак.

Глибоке навчання – це перспективний напрям розвитку машинного навчання. Порівняно з традиційними методами машинного навчання, методи глибокого навчання краще справляються з великими даними. Крім того, методи глибокого навчання можуть автоматично навчатися представленням функцій з необроблених даних, а потім виводити результати. Однією з характеристик глибокого навчання є глибока структура, яка містить кілька прихованих шарів. У порівнянні, традиційні моделі машинного навчання, такі як метод опорних векторів (англ. Support Vector Machines, SVM) і метод k-найближчих сусідів (англ. K-Nearest Neighbo, KNN), не містять або містять лише один прихований шар. Тому ці традиційні моделі машинного навчання також називають неглибокими моделями.

Як відомо, існує два основних типи машинного навчання: кероване та некероване. Кероване навчання спирається на корисну інформацію в маркованих даних. Класифікація є найпоширенішим завданням у керованому навчанні (і також найчастіше використовується в IDS). Однак маркування даних вручну є дорогим і трудомістким. Отже, відсутність достатньої кількості маркованих даних є головним слабким місцем керованого навчання. Навпаки, некероване навчання добуває цінну інформацію про функції з немаркованих даних, що значно полегшує отримання навчальних даних. Однак ефективність виявлення при некерованих методах навчання зазвичай гірша, ніж при керованих методах навчання. Загальні алгоритми машинного навчання, які використовуються в IDS, показані на рис. 1.5 [28].

Слід зазначити, що традиційні моделі машинного навчання (неглибокі моделі) для IDS передусім включають штучну нейронну мережу (ШНМ, англ. Artificial Neural Network, ANN), SVM, KNN), наївний Байєс (англ. Naive Bayes), логістичну регресію (англ. Logistic Regression, LR), дерево рішень, кластеризацію, а також комбіновані та гібридні методи. Деякі з цих методів вивчаються протягом кількох десятиліть, і їхня методологія є зрілою. Вони зосереджені не лише просто на виявленні, але й на практичних проблемах, наприклад, ефективності виявлення та управлінні даними. Переваги та недоліки різних неглибоких моделей наведено в таблиці 1.3 [28].

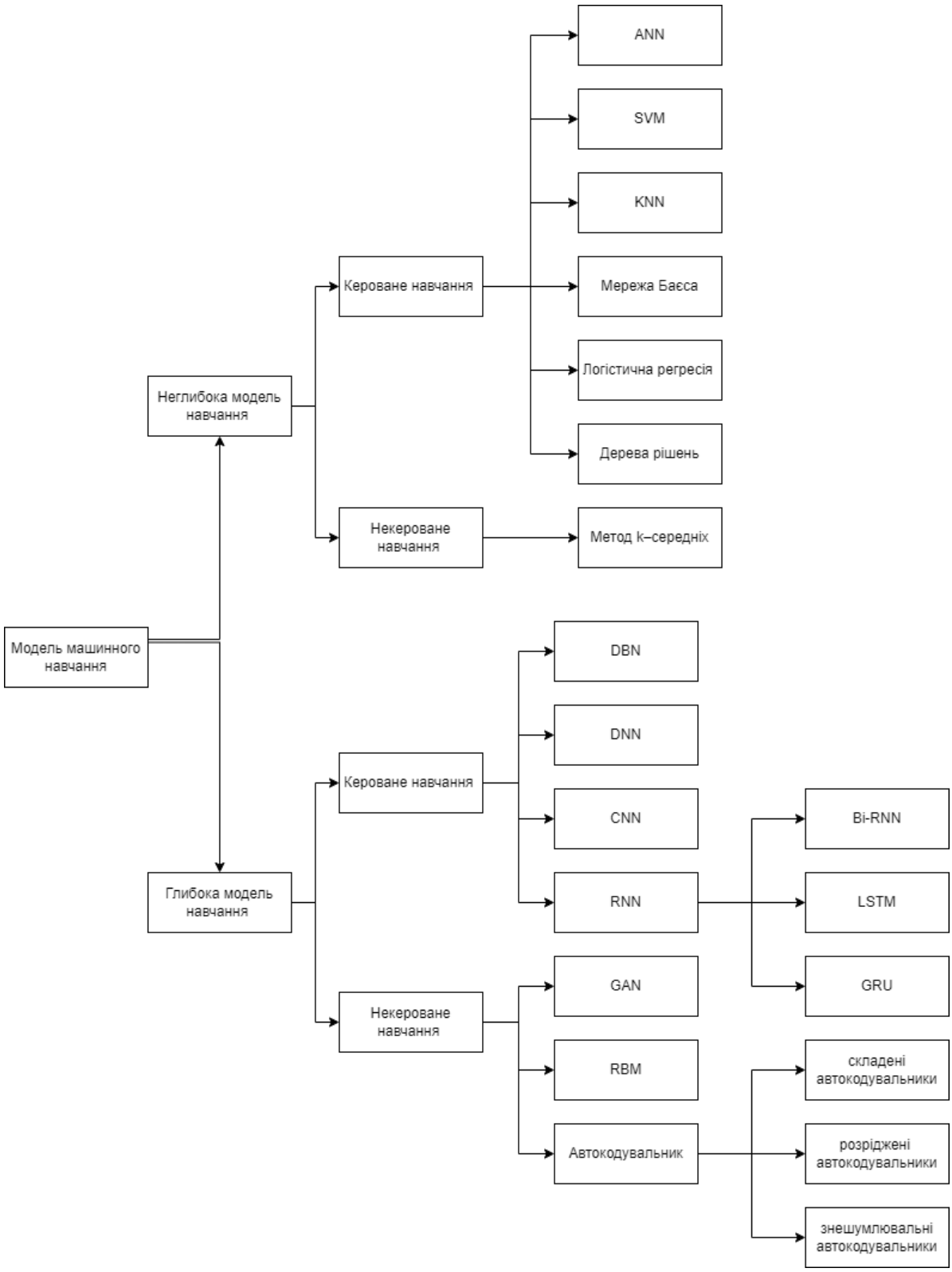



Рисунок 1.5 – Таксономія алгоритмів машинного навчання [28]

Таблиця 1.3 – Характеристика неглибоких моделей

| Алгоритм | Переваги | Недоліки | Заходи щодо покращення |
|---|---|---|---|

| | | | |
|---|---|---|---|
| ANN | Вміє працювати з нелінійними даними.<br>Сильна здатність до підгонки. | Схильний до переобладнання.<br>Схильність заст рягти в локальному оптимумі.<br>Навчання моделі займає багато часу. | Прийняти покращені оптимізатори, функції активації та функції втрати. |
| SVM | Отримує корисну інформацію з невеликого навчального набору.<br>Сильна генеруюча здатність. | Погана робота з великими даними або кількома завданнями класифікації.<br>Чутливий до функціональних параметрів ядра. | Оптимізовані параметри за допомогою оптимізації рою частинок (PSO). |
| KNN | Може бути застосований до великих об’ємів даних.<br>Підходить для нелінійних даних.<br>Здатний швидко тренуватися.<br>Стійкий до шуму. | Низька точність щодо класу меншості.<br>Тривалий час тестування.<br>Чутливий до параметра К. | Зменшення часу порівняння через тригонометричну нерівність.<br>Оптимізовані параметри шляхом оптимізації рою частинок (PSO).<br>Збалансовані набори даних з використанням техніки передискретизації синтетичної меншості (SMOTE). |

Продовження таблиці 1.3

| Алгоритм | Переваги | Недоліки | Заходи щодо покращення |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Мережа Баєса | Стійкий до шуму. Здатний навчатися поступово. | Погано працює з даними, пов’язаними з атрибутами. | Імпортовані приховані змінні, щоб пом'якшити незалежне припущення. |
| Логістична регресія | Простий, піддається швидкому навчанню. Автоматичне масштабування функцій. | Погана робота з нелінійними даними. Схильний до надмірного оснащення. | Імпортована регулярізація для уникнення надмірного пристосування. |
| Дерева рішень | Автоматичний вибір функцій. Здатність до якісної інтерпретації. | Тенденції класифікації результату до класу більшості. Ігнорування кореляції даних. | Збалансовані набори даних за допомогою SMOTE. Введення латентних змінних. |
| Метод k–середніх | Простий, піддається швидкому навчанню. Сильна масштабованість. Підходить для великих даних. | Погано працює на неопуклих даних. Чутливий до ініціалізації. Чутливий до параметра К. | Покращений метод ініціалізації. |

Таким чином, до основних методів машинного навчання для систем виявлення вторгнень відносяться наступні [28].

1) Штучна нейронна мережа (англ. Artificial Neural Network, ANN). Ідея дизайну ANN полягає в імітації роботи людського мозку. ANN містить вхідний рівень, кілька прихованих шарів і вихідний рівень. Блоки в суміжних шарах повністю з’єднані. ANN містить величезну кількість одиниць і теоретично може апроксимувати довільні функції, а отже, має сильну здатність підгонки, особливо для нелінійних функцій. Через складну структуру моделі навчання ANN займає багато часу. Слід зазначити, що моделі ANN навчаються за допомогою алгоритму зворотного поширення, який не можна використовувати для навчання глибоких

мереж. Таким чином, ANN належить до неглибоких моделей і відрізняється від моделей глибокого навчання.

2) Метод опорних векторів (SVM). Стратегія SVM полягає в тому, щоб знайти гіперплощину поділу з максимальним запасом у n-вимірному просторі ознак. SVM можуть досягти якісних результатів навіть із невеликими навчальними наборами, оскільки гіперплощина поділу визначається лише невеликою кількістю опорних векторів. Однак SVM чутливі до шуму поблизу гіперплощини. SVM здатні добре вирішувати лінійні задачі. Для нелінійних даних зазвичай використовуються функції ядра. Функція ядра відображає вихідний простір у новий простір, щоб вихідні нелінійні дані можна було розділити. Ядровий трюк широко поширений як серед SVM, так і серед інших алгоритмів машинного навчання.

3) Метод k-найближчих сусідів (KNN). Основна ідея KNN базується на гіпотезі різноманітності. Якщо більшість сусідів вибірки належать до одного класу, вибірка має високу ймовірність приналежності до класу. Таким чином, результат класифікації пов’язаний лише з верхніми-k найближчими сусідами. Параметр k сильно впливає на продуктивність моделей KNN. Чим менше k, тим складніша модель і тим вищий ризик переобладнання. І навпаки, чим більше k, тим простіше модель і тим слабша здатність підгонки.

4) Мережа Баєса (алгоритм). Наївний алгоритм Байєса базується на умовній ймовірності та гіпотезі незалежності атрибутів. Для кожної вибірки наївний класифікатор Байєса обчислює умовні ймовірності для різних класів. Вибірку класифікують за класом максимальної ймовірності. Формула умовної ймовірності обчислюється, як показано у формулі (1.1).

$$P\left(X = x \middle| Y = c_k\right) = \prod_{i=1}^{n} P\left(X^{(i)} = x^{(i)} \middle| Y = c_k\right). \qquad (1.1)$$

Коли гіпотеза про незалежність атрибутів задовольняється, алгоритм Наївного Байєса досягає оптимального результату. На жаль, цю гіпотезу важко задовольнити повністю, тому наївний алгоритм Байєса погано працює з даними, пов’язаними з атрибутами.

5) Логістична регресія (LR). LR є типом логарифмічної лінійної моделі. Алгоритм LR обчислює ймовірності різних класів за допомогою параметричного логістичного розподілу, як показано у формулі (1.2).

$$P\left(Y = k \middle| x\right) = \frac{e^{w_k \bullet x}}{1 + \sum_k^{K-1} e^{w_k \bullet x}}, \quad (1.2)$$

де k = 1, 2 ... K − 1.

Вибірка *x* класифікується в клас максимальної ймовірності. Модель LR легко побудувати, а навчання моделі є ефективним. Однак LR не може добре працювати з нелінійними даними, що обмежує її застосування.

6) Дерево рішень. Алгоритм дерева рішень класифікує дані за допомогою серії правил. Модель схожа на дерево, що робить її інтерпретованою. Алгоритм дерева рішень може автоматично виключати нерелевантні та зайві функції. Процес навчання включає вибір функцій, створення дерева та обрізання дерева. Під час навчання моделі дерева рішень алгоритм вибирає найбільш підходящі функції окремо та генерує дочірні вузли з кореневого вузла. Дерево рішень є основним класифікатором.

7) Кластеризація (метод k–середніх). Кластеризація базується на теорії подібності, тобто групування дуже схожих даних в однакові кластери та групування менш схожих даних у різні кластери. На відміну від класифікації, кластеризація є типом некерованого навчання. Для алгоритмів кластеризації не потрібні попередні знання чи позначені дані, тому вимоги до набору даних відносно низькі. Однак при використанні алгоритмів кластеризації для виявлення атак необхідно посилатися на зовнішню інформацію.

Метод k–середніх – це типовий алгоритм кластеризації, де k – це кількість кластерів, а середнє – це середнє значення атрибутів. Метод k–середніх використовує відстань як критерій міри подібності. Чим менша відстань між двома об’єктами даних, тим імовірніше, що вони будуть розміщені в одному кластері. Метод k–середніх добре адаптується до лінійних даних, але його результати для неопуклих даних не ідеальні. Крім того, метод k–середніх чутливий до умови ініціалізації та параметра K. Отже, потрібно виконати багато повторюваних експериментів, щоб встановити правильне значення параметра.

8) Ансамблі та гібриди. Кожен окремий метод має сильні сторони та недоліки. Природним підходом є поєднання різних слабких класифікаторів для реалізації сильного класифікатора. Методи ансамблю навчають декілька класифікаторів; потім класифікатори голосують для отримання остаточних результатів. Гібридні методи розроблені як набір етапів, у яких кожен етап використовує модель класифікації. Оскільки ансамблеві та гібридні класифікатори зазвичай працюють краще, ніж одиночні класифікатори, все більше дослідників

почали вивчати ансамблеві та гібридні класифікатори. Ключові моменти полягають у виборі класифікаторів для поєднання та способу їх поєднання.

Окремо слід виділити моделі глибокого навчання, які складаються з різноманітних глибоких мереж. До них відносяться: глибокі мережі переконань (англ. Deep Belief Network, DBN), глибокі нейронні мережі (англ. Deep Neural Networks, DNN), згорткові нейронні мережі (англ. Convolutional Neural Networks, CNN) і рекурентні нейронні мережі (англ. Recurrent Neural Networks, RNN), що є керованими моделями навчання, тоді як автокодувальники, обмежені машини Больцмана (англ. Restricted Boltzmann Machines, RBM) і генеративні змагальні мережі (англ. Generative Adversarial Networks, GANs) – це некеровані моделі навчання.

Моделі глибокого навчання безпосередньо вивчають представлення функцій з вихідних даних, не вимагаючи ручної розробки функцій. Таким чином, методи глибокого навчання можуть виконуватися наскрізним способом. У вивченні глибинного навчання основними акцентами є мережна архітектура, вибір гіперпараметрів і стратегія оптимізації. Порівняння різних моделей глибокого навчання показано в таблиці 1.4.

Таким чином, до основних моделі глибокого навчання відносяться.

1) Автокодувальник містить два симетричних компоненти, кодувальник і декодер, як показано на рис. 1.6. Кодувальник добуває ознаки з необроблених даних, а декодер реконструює дані з витягнутих функцій. Під час навчання розбіжність між входом кодувальника і виходом декодера поступово зменшується. Коли декодеру вдається реконструювати дані за допомогою вилучених ознак, це означає, що функції, витягнуті кодувальником, представляють суть даних. Важливо зазначити, що весь цей процес не вимагає контрольованої інформації. Існує багато відомих варіантів автокодувальників, таких як автокодувальники з усуненням шумів [29] і розріджені автокодувальники [30].

Таблиця 1.4 – Порівняння різних моделей глибокого навчання [28]

| Алгоритми | Відповідні типи даних | Керований чи некерований | Функції |
|---|---|---|---|
| Автокоду-вальник | Необроблені дані. Вектори ознак. | Некерований | Вилучення ознак. Зменшення ознак. Знешумлення. |
| RBM | Вектори ознак. | Некерований | Вилучення ознак. Зменшення ознак. Знешумлення. |
| DBN | Вектори ознак. | Керований | Вилучення ознак. Класифікація. |
| DNN | Вектори ознак. | Керований | Вилучення ознак. Класифікація. |
| CNN | Необроблені дані. Вектори ознак. Матриці. | Керований | Вилучення ознак. Класифікація. |
| RNN | Необроблені дані. Вектори ознак. Послідовність даних. | Керований | Вилучення ознак. Класифікація. |
| GAN | Необроблені дані. Вектори ознак. | Некерований | Збільшення даних. Навчання змагальності. |

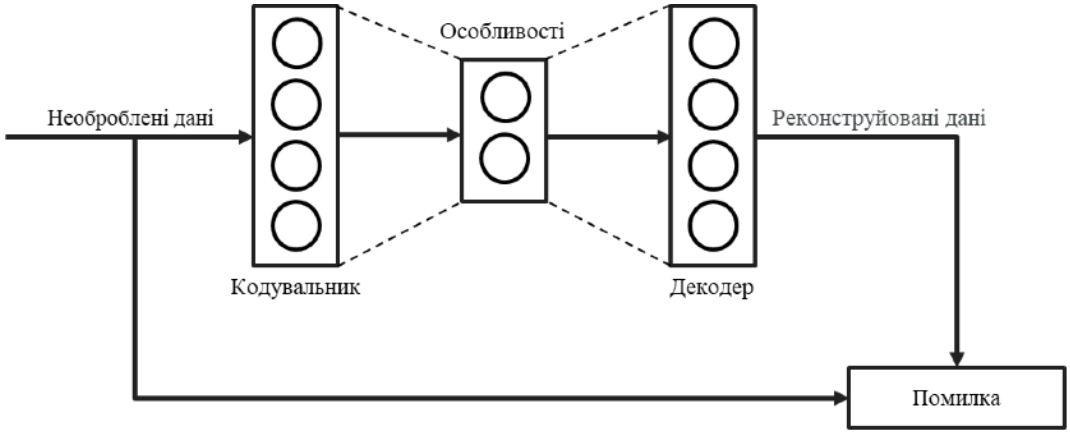



Рисунок 1.6 – Структура автокодувальника [28]

2) Обмежена машина Больцмана (RBM). – це рандомізована нейронна мережа, у якій одиниці підкоряються розподілу Больцмана. RBM складається з

видимого та прихованого шарів. Блоки в одному шарі не з'єднані, однак елементи на різних рівнях повністю з'єднані, як показано на рис. 1.7, де $v_i$ – видимий шар, а $h_i$ – прихований. RBM не розрізняють напрямок вперед і назад, таким чином, ваги в обох напрямках однакові. RBM – це моделі неконтрольованого навчання, навчені алгоритмом контрастної дивергенції [31], і вони зазвичай застосовуються для виділення ознак або усунення шумів.

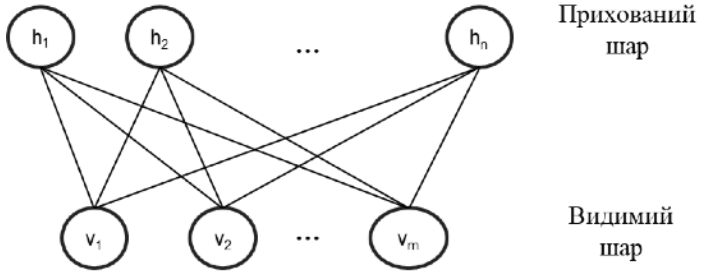



Рисунок 1.7 – Структура RBM [31]

3) Глибокі мережі переконань (DBN) складаються з кількох рівнів RBM і шару класифікації softmax (нормована експоненційна функція), як показано на рис. 1.8 [28]. Навчання DBN включає два етапи: некероване попереднє навчання та кероване точне налаштування. Спочатку кожен RBM навчається за допомогою жадібного пошарового попереднього навчання. Потім вага шару softmax вивчається за позначеними даними. У виявленні атак DBN використовуються як для визначення ознак, так і для їх класифікації.

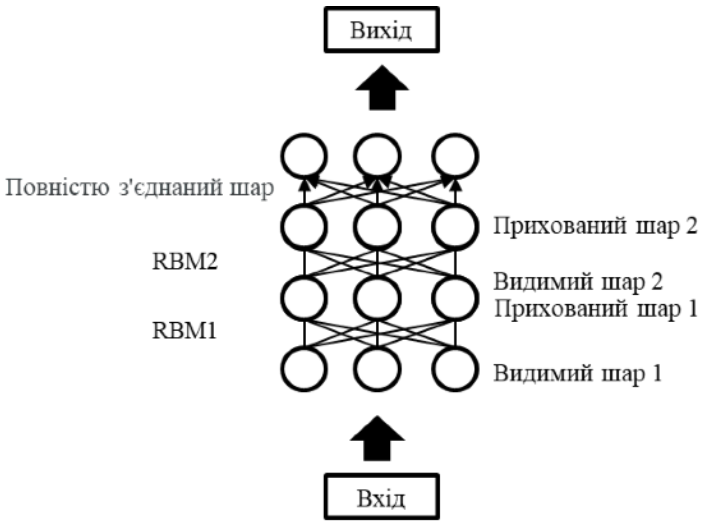



Рисунок 1.8 – Структура DBN [28]

4) Глибока нейронна мережа (DNN) реалізує стратегію пошарового попереднього навчання та тонкого налаштування, і дає змогу створювати DNN з кількома рівнями, як показано на рис. 1.9. Під час навчання DNN параметри

вивчаються спочатку з використанням немаркованих даних, що є етапом некерованого навчання функцій, а потім мережа налаштовується за допомогою мічених даних, що є етапом керованого навчання. Досягнення DNN головним чином пов’язані з некерованим етапом вивчення функцій [28].

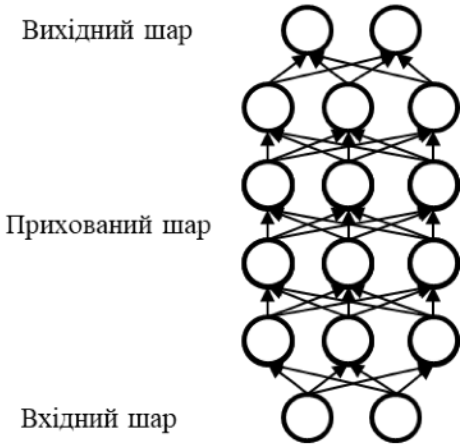



Рисунок 1.9 – Структура DNN [28]

5) Згорткова нейронна мережа (CNN) розроблена для імітації зорової системи людини, таким чином CNN досягли великих досягнень у сфері комп’ютерного зору. CNN складається з альтернативних шарів згортки та об’єднання, як показано на рис. 1.10. Згорткові шари використовуються для виділення функцій, а шари об’єднання – для покращення узагальнення ознак. CNN працюють з двовимірними даними, тому вхідні дані мають бути переведені в матриці для виявлення атак [28].

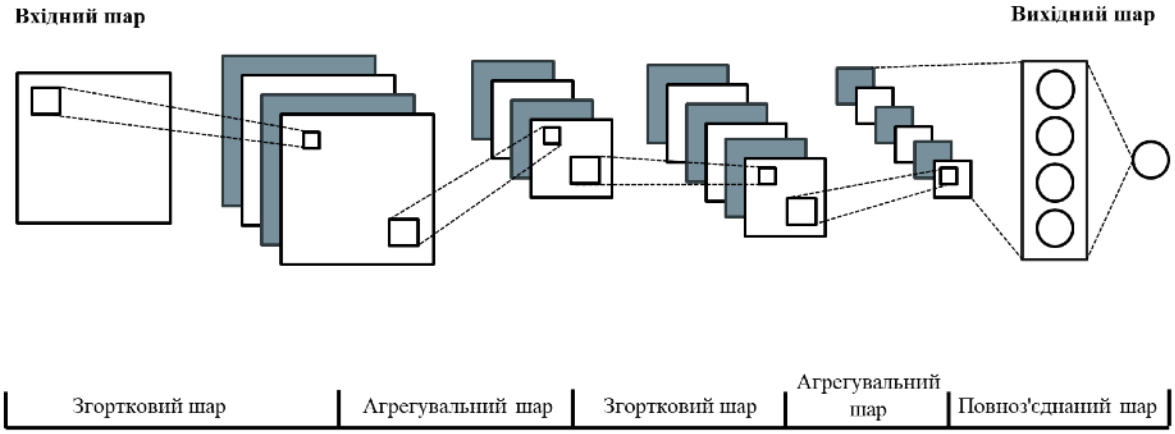



Рисунок 1.10 – Структура CNN [28]

6) Рекурентна нейронна мережа (RNN) це мережа, яка розроблена для послідовних даних і широко використовуються в обробці природної мови. Характеристики послідовних даних є контекстними, тому аналіз ізольованих даних із послідовності не має сенсу. Для отримання контекстної інформації

кожен блок у RNN отримує не лише поточний стан, але й попередні стани. Структура RNN показана на рис.1.11, де всі елементи W однакові. Ця характеристика призводить до того, що RNN часто страждають від зникнення або вибуху градієнтів. Насправді стандартні RNN мають справу лише з послідовностями обмеженої довжини. Для вирішення проблеми довготривалої залежності було запропоновано багато варіантів RNN, таких як довга короткочасна пам'ять (LSTM, long short-term memory), вентильний рекурентний вузол (GRU, gated recurrent unit) і двонаправлені рекурентні нейронні мережі (bi-RNN) [28].

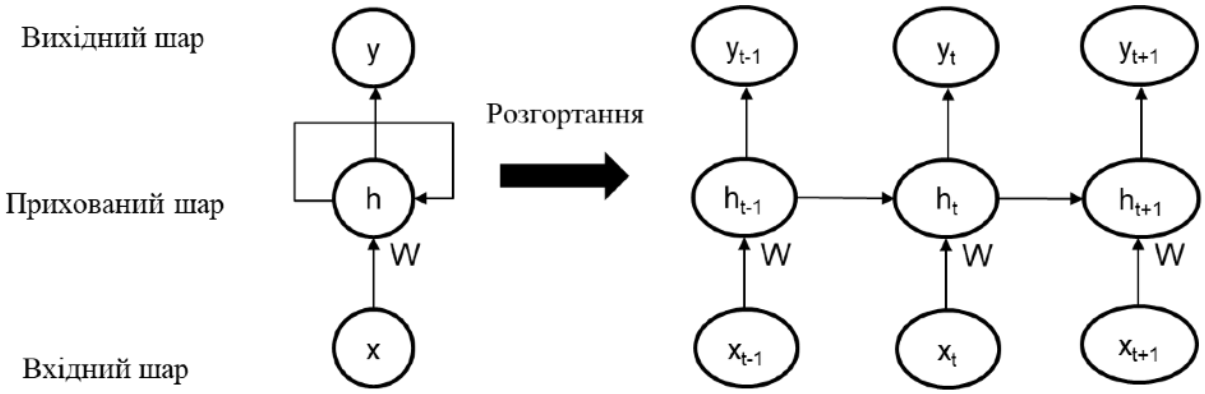



Рисунок 1.11 – Структура RNN [28]

7) Генеративні змагальні мережі (GAN) – це клас алгоритмів неконтрольованого машинного навчання. Схема генеративної змагальної мережі наведена на рис 1.12. Вони складаються з двох нейронних мереж: генератора і дискримінатора. Розглядаючи набір даних, генератор генерує нові екземпляри даних, подібні до тих, що містяться в наборі даних. Дискримінатор по суті є класифікатором, який розрізняє згенеровані дані як оригінальні або підроблені. Дискримінатор приймає дві форми даних: оригінальні дані та дані, згенеровані генератором.

Під час навчання дискримінатор використовує вихідні дані як позитивний приклад, а згенеровані дані – як негативні/протилежні приклади. $L_D$ – це покарання для дискримінатора, коли дискримінатор не може виявити або правильно розрізнити дані, при цьому штраф збільшується, а в іншому випадку – зменшується. Для оновлення ваг дискримінатора використовується зворотне поширення. Ще одна втрата $L_G$ – це втрата генератора [31].

Генератор створює синтетичний набір даних, отримуючи зворотній зв'язок від дискримінатора, і вчиться створювати дані так, щоб дискримінатор класифікував синтетичні дані як оригінальні.

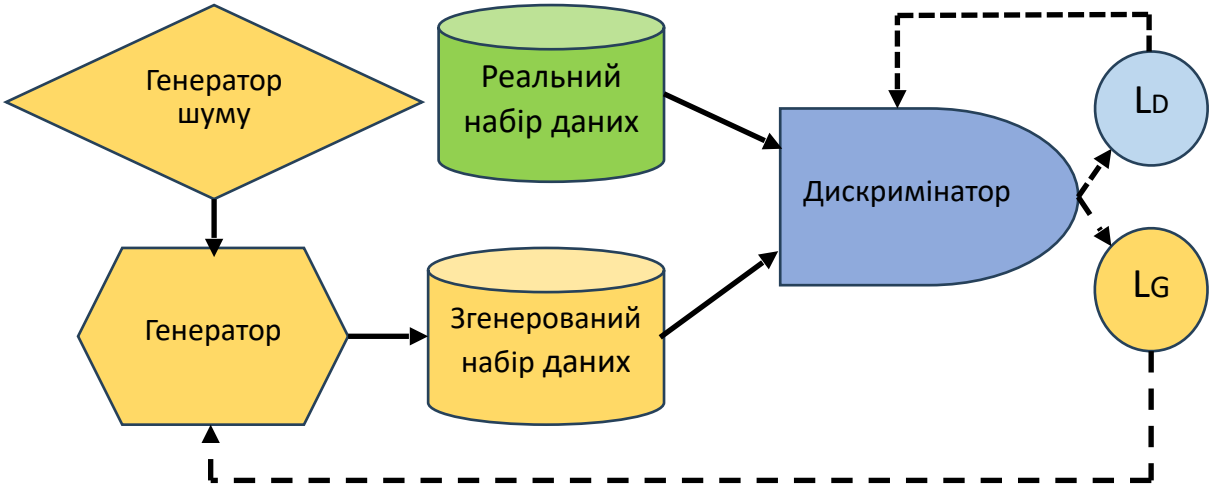



Рисунок 1.12 – Схема генеративної змагальної мережі

До інших моделей навчання відносяться наступні.

1) Генетичні алгоритми (англ. Genetic Algorithm, GA) є однією областей комп'ютерної безпеки, особливо IDS. GA базуються на генетиці, особливо на теорії Дарвіна: виживання найбільш пристосованих. Це свідчить про те, що слабші представники виду мають тенденцію відмирати, залишаючи сильніших і здоровіших. Члени, що вижили, створюють потомство і забезпечують продовження виживання виду. Ця концепція разом із концепцією природного відбору використовується в інформаційних технологіях для підвищення продуктивності комп'ютерів. Підхід GA використовує навчені мережні дані. Коли необхідно виявити невідомі типи атак, такий метод має перевагу, щоб автоматично перенавчати моделі виявлення на вхідних даних. Зазвичай для цієї мети використовується великий обсяг мережних даних. Існуючі IDS також значною мірою покладаються на людей-аналітиків для розрізнення нормальних і ненормальних мережних з'єднань. Використання GA в IDS допомагає уникнути цього величезного завдання аналітиків шляхом створення правил для ID. GA також надає кілька рішень проблеми, таким чином, він зможе виявити багато вторгнень [32]. GA пропонує певні переваги перед іншими методами машинного навчання, а саме:

– GA за своєю суттю є паралельними, оскільки вони мають кілька нащадків, то вони можуть досліджувати простір рішень у кількох напрямках одночасно. Якщо один шлях виявиться тупиковим, то його легко усунуть і продовжать роботу над більш перспективними напрямками;

– завдяки паралелізму, який дозволяє їм неявно оцінювати багато схем одночасно, генетичні алгоритми особливо добре підходять для розв'язання

проблем, де простір усіх потенційних рішень справді величезний – надто великий, щоб здійснювати вичерпний пошук за будь-який розумний проміжок часу;

– робота з популяціями потенційних рішень, а не з одним рішенням, і використання стохастичних операторів для керування процесом пошуку дозволяє GA добре справлятися з взаємодіями атрибутів і уникати застрягання в локальних максимумах, що разом робить їх дуже придатними для роботи з такими класами, як вторгнення;

– систему, засновану на GA, можна легко перенавчити. Ця властивість забезпечує адаптивність системи на основі GA, що є обов’язковою якістю системи виявлення вторгнень, враховуючи високий рівень нових атак.

2) Нечітка логіка (Fuzzy logic) є розширенням булевої логіки, яка часто використовується для комп’ютерного прийняття складних рішень. У той час як елемент класичної булевої логіки може бути або повним членом, або не бути членом булевого (іноді його називають «чітким») набору, належність елемента до нечіткого набору може бути будь-яким значенням у межах інтервалу [0, 1] , що дозволяє також часткове членство елемента в наборі.

Нечітка експертна система складається з трьох різних типів сутностей: нечітких наборів, нечітких змінних і нечітких правил. Приналежність нечіткої змінної до нечіткої множини визначається функцією, яка створює значення в інтервалі [0, 1]. Ці функції називаються функціями належності.

Нечіткі змінні поділяються на дві групи: попередні змінні, яким присвоюються вхідні дані нечіткої експертної системи, і консеквентні змінні, яким присвоюються результати, обчислені системою.

Нечіткі правила визначають зв'язок між антецедентом і наступними нечіткими змінними, і часто визначаються з використанням лінгвістичних термінів природної мови. Наприклад, нечітким правилом може бути «якщо температура низька та сильний вітер, тоді одягайте теплий одяг», де «температура» та «вітер» є попередніми нечіткими змінними, «одягайте» є наступною нечіткою змінною, а «низька», «сильний» та «теплий одяг» є нечіткими наборами.

Процес створення нечіткої системи має три етапи. Ці кроки – фазифікація, оцінка правила та дефазифікація. На етапі фазифікації вхідні чіткі значення перетворюються на ступені приналежності до нечітких наборів. Ступінь належності кожного чіткого значення до кожного нечіткого набору визначається підключенням значення до функції належності, пов’язаної з нечітким набором. На

етапі оцінки правила кожному нечіткому правилу призначається значення міцності. Міцність визначається ступенями належності чітких вхідних значень до нечітких наборів попередньої частини нечіткого правила. Етап дефазифікації транспонує нечіткі виходи в чіткі значення [33].

3) Штучна імунна мережа – це динамічний метод некерованого навчання. Модель штучної імунної мережі складається з набору клітин, які називаються антитілами, з'єднаних зв'язками певної сили. Ці мережні антитіла (ідіотипова мережа) представляють мережні внутрішні образи патогенів (вхідні шаблони), що містяться в середовищі, якому вони піддаються.

4) Метод рою часток (англ. Particle Swarm Optimization, PSO) зараз привертає значний інтерес дослідницького співтовариства, оскільки метод здатна задовольнити зростаючий попит на надійну та інтелектуальну систему виявлення вторгнень (IDS). PSO бере участь у Swarm intelligence, який є технікою обчислювального інтелекту, що передбачає вивчення колективної поведінки в децентралізованій системі. В алгоритмі PSO точка в просторі пошуку (тобто можливе рішення) називається частинкою. Сукупність частинок у певній ітерації називається роєм. Терміни «частинка» і «рій» аналогічні термінам «індивід» і «популяція», які використовуються в еволюційних алгоритмах, таких як GA. На кожній ітерації кожна частинка в рої переміщується на нову позицію в просторі пошуку. PSO ефективний у задачах нелінійної оптимізації, його легко реалізувати, і потрібно налаштувати лише кілька вхідних параметрів. Оскільки процес оновлення в PSO базується на простих рівняннях, його можна ефективно використовувати для великих наборів даних. Завдяки цим перевагам PSO успішно застосовувався в багатьох сферах, таких як оптимізація функцій, навчання штучних нейронних мереж, керування нечіткою системою та інші області, де можна застосувати GA [34].

5) Регресійний аналіз – це набір статистичних процесів для оцінки зв'язків між змінними. Простими словами, ідея полягає в тому, що є кілька незалежних змінних, які, взяті разом, дають у якості результату залежну змінну. Потім регресійна модель використовується для прогнозування результату невідомої залежної змінної за значеннями незалежних. Подібно до класифікації, регресія є контрольованою проблемою, але вона виводить не клас, а число.

Слід зазначити, що при наявності безперечних переваг застосування методів машинного навчання для систем виявлення вторгнень існує і рід недоліків. Одним з них є вразливість алгоритмів машинного навчання перед змагальними атаками.

Таким чином, аналіз механізмів реалізації змагальних атак та методів протидії є актуальною задачею.

# 2 АНАЛІЗ МЕТОДІВ ПРОТИДІЇ ЗМАГАЛЬНИМ АТАКАМ

## 2.1 Класифікація атак на системи машинного навчання

При реалізації змагальної атаки супротивник прагне заплутати модель машинного навчання, змусивши її прийняти неправильне рішення. Супротивник досягає цього шляхом модифікації вхідних даних, які передаються в модель машинного навчання або під час фази навчання (атака отруєння), або під час фази висновку (атака ухилення). Причина появи змагальних прикладів пов’язана з тим фактом, що більшість моделей машинного навчання залишаються відкрито прив’язаними до поверхневої статистики вхідних даних. Ця прив’язаність до вхідних даних робить машинне навчання дуже чутливим до зміни розподілу, що призводить до невідповідності між семантичними змінами та зміною рішення.

У [35] розглядається модель безпеки для використання машинного навчання в мережній безпеці як комбінацію чотирьох компонентів.

1) Поверхня атаки: ідентифікує різні вектори атаки вздовж типового конвеєра обробки даних машинного навчання в програмах, пов’язаних з безпекою мережі.
2) Модель загроз: забезпечує системну абстракцію для профілювання можливостей супротивника та пов’язаних із ним потенційних загроз.
3) Змагальна структура: деталізує підхід до класифікації різних атак і засобів захисту в кожному домені безпеки мережі.
4) Змагальний ризик: забезпечує оцінку ймовірності та серйозності змагальних атак у системі безпеки мережі.

Основним компонентом змагальної атаки є змагальна вибірка. Як показано на рис. 2.1, змагальна вибірка складається з вхідних даних для моделі машинного навчання, які були заплутані.

Для конкретного набору даних із ознаками «x» і міткою «y» відповідна змагальна вибірка представляє собою конкретну точку даних «x’», яка змушує класифікатор «c» передбачати іншу мітку на «x’», відмінну від «y», але «x’» майже не відрізняється від «x». Змагальні зразки створюються за допомогою одного з багатьох методів оптимізації, відомих як методи змагальної атаки. Створення змагальних вибірок передбачає вирішення задачі оптимізації для визначення мінімальної пертурбації, яка максимізує втрати для нейронної мережі.

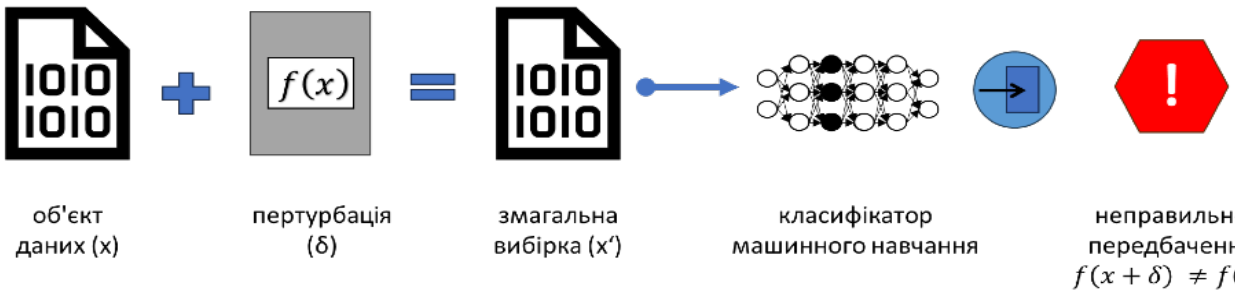



Рисунок 2.1 – Схема змагального машинного навчання [35]

Враховуючи вхідні дані «x» і класифікатор «f», мета оптимізації для супротивника полягає в тому, щоб обчислити таку пертурбацію з малою нормою, виміряною відносно деякої метрики відстані, яка б змінила вихід класифікатора таким чином, що:

$$f(x + \delta) \neq f(x), \tag{2.1}$$

де $\delta$ – це пертурбація.

Якщо $\delta$ застосовується до всіх вхідних даних – це вважається щільною змагальною атакою. Однак, якщо порушуються лише часткові позиції, тоді це називається розрідженою змагальною атакою [36].

Змагальне машинне навчання в мережній безпеці зазвичай є суперечкою між двома агентами. Перший агент – це зловмисник, метою якого є вторгнення в мережу за допомогою шкідливого корисного навантаження. Інший агент – це той, роль якого полягає в захисті мережі від наслідків шкідливого корисного навантаження. Тому для подальшого аналізу змагальних атак, потрібно розглянути різні типи даних, які проходять через мережу в будь-який момент часу.

Таксономія змагальної атаки представлена в таблиці 2.1.

Загрози ворожих атак у сфері мережної безпеки можна розглянути на основі знань зловмисника, простору атаки, стратегії зловмисника, цілі зловмисника та цілі атаки. Такий підхід дозволяє розглядати зловмисні атаки у багатомірному просторовому виміру.

Таблиця 2.1 – Таксономія змагальних атак

| Основа | Тип |
|---|---|
| Знання | Black Box |
| | White Box |
| | Gray Box |
| Простір | Простір ознак (Feature Space) |
| | Простір задач (Problem Space) |
| Стратегія | Ухилення (Evasion) |
| | Отруєння (Poisoning) |
| | Оракул (Oracle) |
| Мета | Доступність (Availability) |
| | Цілісність (Integrity) |
| | Конфіденційність (Confidentiality) |
| Ціль | Фізичний домен |
| | Модель ML |

До цих вимірів відносяться: знання, простір, стратегія, мета, ціль.

1) Знання: компонент знань моделі змагальної загрози описує ступінь, до якого супротивник знає про машинну систему в цілому. Це можна класифікувати як атаки «білого ящика», «сірого ящика» або «чорного ящика» [35]:

– в атаках «білого ящика» передбачається, що зловмисник має повні знання про навчальні дані, алгоритм навчання, навчену модель, а також параметри, які використовувалися під час навчання моделі. Атака «білого ящика» представляє супротивника, який володіє точною інформацією, якою володіє власник або розробник системи машинного навчання, яка піддається атаці. У більшості випадках змагальної атаки в реальних умовах це зазвичай неможливо;

– атаки «сірого ящика» передбачають більш реалістичний підхід і враховують, що супротивник може мати доступ до різної кількості інформації. Зловмисник може мати часткову інформацію про запити моделі або обмежений доступ до навчальних даних. Для атаки «сірого ящика» супротивник не має точних знань, якими володіє розробник моделі, але має

достатньо інформації, щоб атакувати і спричинити збій системи машинного навчання;

– атака «чорного ящика» передбачає, що супротивник абсолютно не знає про систему машинного навчання. При цьому типі атаки супротивник не знає ані про алгоритм навчання, ані про вивчену модель. Можна припустити, що в дійсності атака «чорного ящика» неможлива. Це аргументується тим, що передбачається, що супротивник повинен мати принаймні певну інформацію, наприклад, місце розташування моделі, перш ніж він зможе атакувати модель. Але насправді серйозність атак «чорного ящика» становить більшу загрозу на практиці. Модель для систем реального світу може бути більш обмежувальною, ніж теоретична модель «чорного ящика», де зловмисник може повністю зрозуміти вихідну інформацію нейронної мережі базуючись на вхідних даних, які були вибрані довільно.

2) Простір: у сфері змагального машинного навчання вхідний простір можна визначити як розмірне представлення всіх можливих конфігурацій об’єктів у контексті визначення.

Моделювання простору ознак змагальної вибірки – це метод, у якому використовується алгоритм оптимізації, щоб знайти ідеальне значення з кінцевої кількості довільних змін, внесених до ознак. У змагальній атаці на простір ознак метою зловмисника є залишатися незагрозливим, не створюючи нового екземпляра. Простір ознак визначається як n-вимірний простір, у якому представлені всі змінні у вхідному наборі даних. Атака на простір ознак безпосередньо змінює функції в екземплярі. Таким чином, атака, яка спрямована на простір, лише змінює вектори функцій, але не створює нового шкідливого програмного забезпечення.

Простір задач відноситься до вхідного простору, в якому існують об’єкти, такі як наприклад зображення, файл тощо. Змагальна атака зловмисного програмного забезпечення в просторі задач змінить фактичний екземпляр з джерела, щоб створити новий екземпляр зловмисного програмного забезпечення. Як правило, змагальна атака на простір задач має тенденцію створювати нові об’єкти в таких доменах, як виявлення зловмисного програмного забезпечення, за допомогою чого немає чіткого зворотного відображення простору задач. Типова відмінність між змагальною атакою на просторі задач і змагальною атакою на просторі ознак полягає в тому, що атака на простір ознак не генерує нову вибірку, а лише створює новий вектор ознак. Змагальна атака простору задач змінює сам фактичний екземпляр, створюючи абсолютно новий об’єкт.

3) Стратегія: стратегія зловмисника передбачає фази операції, на яких супротивник починає атаку. Три основні стратегії, які супротивник може використовувати у змагальних атаках: ухилення, отруєння та оракул.

Атаки ухилення, також відомі як дослідницька атака або атака під час прийняття рішення, під час фази тестування або висновку. Зловмисник прагне ввести в оману рішення моделі машинного навчання після того, як воно було вивчено, як показано на рис. 2.2.


Рисунок 2.2 – Атаки ухилення

Атаки ухилення зазвичай передбачають арифметичне обчислення проблеми оптимізації. Метою задачі оптимізації є обчислення крихітної сигми пертурбації, яка призведе до збільшення функції втрат. Тоді зміна функції втрат буде достатньо значною, щоб призвести до неправильного прогнозу моделі машинного навчання. Атаки ухилення класифікуються на: атаки на основі градієнта та атаки без градієнта [35].

Атаки на основі градієнта додатково класифікуються на базі частоти, з якою змагальні зразки оновлюються або оптимізуються: ітераційні або одноразові атаки. Ітераційні атаки забезпечують більш жорсткий контроль пертурбації, щоб створити більш переконливі змагальні зразки. Однак це призводить до вищих обчислювальних витрат. Альтернативою ітераційним атакам є одноразові атаки, які використовують одноетапний підхід без ітерацій. Одноразові атаки – це атаки, у яких змагальні вибірки оптимізуються лише один раз. В той час як ітераційні атаки передбачають багаторазове оновлення змагальних зразків. Завдяки багаторазовому оновленню змагальних зразків вони краще оптимізовані та ефективніші порівняно з одноразовими атаками. Однак ітераційні атаки потребують більше обчислювального часу. Змагальні атаки проти певних методів машинного навчання, які потребують інтенсивних обчислень, таких як навчання з підкріпленням, зазвичай вимагають одноразових атак як єдиного можливого підходу [37].

Безградієнтні атаки [38], на відміну від градієнтних атак, не вимагають знання моделі. Безградієнтні атаки можуть генерувати потужні атаки проти моделі машинного навчання, знаючи лише значення достовірності моделі.

Атаки отруєння, також відомі як причинні атаки, включають змагальне пошкодження навчальних даних або логіки моделі під час фази навчання, щоб спровокувати неправильне передбачення в режимі машинного навчання, як показано на рис. 2.3.

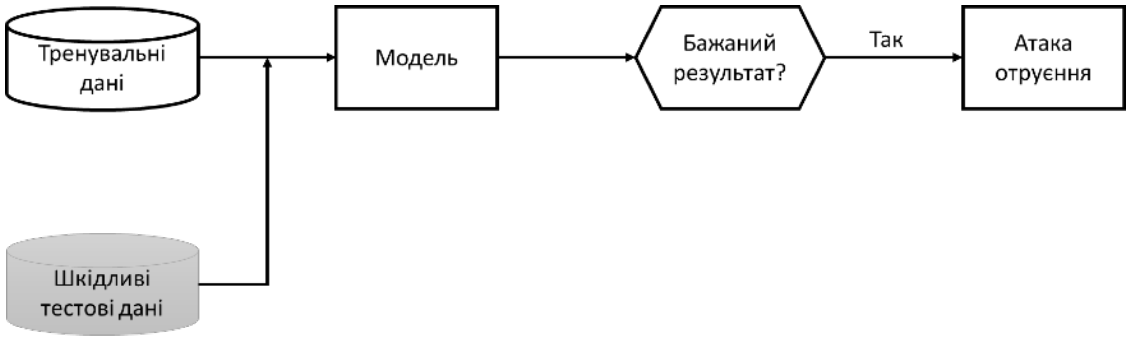



Рисунок 2.3 – Атаки отруєння

Атаки отруєння можуть здійснюватися шляхом ін’єкції даних, маніпуляції даних або порушення логіки [39].

Ін’єкція даних відбувається, коли зловмисник вставляє конкурентні вхідні дані, щоб змінити розподіл даних, зберігаючи початкові функції введення та мітки даних.

Маніпулювання даними стосується ситуації, в якій зловмисник змінює або вхідні функції, або мітки даних початкових навчальних даних.

Порушення логіки – це спроба супротивника змоделювати структуру [35].

Атаки оракул (Oracle) відбуваються, коли зловмисник використовує доступ до прикладного програмного інтерфейсу моделі, щоб створити замінну модель зі зловмисними намірами. Модель-замінник зазвичай зберігає значну частину функціональних можливостей вихідної моделі. У результаті модель заміни може бути використана для інших типів атак, таких як атаки ухилення [39]. Атаки оракул можна далі поділити на атаки вилучення, інверсії та атаки на логічний висновок.

Мета атаки вилучення полягає в тому, щоб вивести архітектурні деталі моделі, такі як параметри та вагові коефіцієнти, зі спостереження прогнозів виходу моделі та ймовірностей класу.

Інверсійні атаки виникають, коли зловмисник намагається реконструювати навчальні дані.

Атаки на логічний висновок дозволяють зловмиснику ідентифікувати конкретні точки даних з розподілом навчального набору даних.

4) Мета: традиційно у сфері комп'ютерного зору атаки суперників розглядаються як цілеспрямовані атаки або атаки на надійність. У цілеспрямованих атаках зловмисник має конкретну мету щодо модельного рішення. Найчастіше зловмисник прагне отримати певний прогноз від моделі машинного навчання. Так атака на надійність виникає, коли зловмисник лише прагне максимізувати помилку передбачення моделі машинного навчання, не обов'язково викликаючи певний результат [35].

Атака конфіденційності означає, що на меті зловмисника - перехопити зв'язок між двома сторонами A та B, щоб отримати доступ до приватної інформації, якою обмінюються. Це відбувається в контексті змагального машинного навчання, за допомогою якого методи машинного навчання використовуються для виконання завдань безпеки мережі.

Атака на цілісність має на меті спричинити неправильну класифікацію, відмінну від фактичного класу виходу, якому навчена передбачати модель машинного навчання. Атака на цілісність може призвести до цілеспрямованої неправильної класифікації або атаки на надійність. Цільова неправильна класифікація намагається змусити модель машинного навчання створювати конкретні неправильні прогнози. Атака на надійність призводить або до зниження впевненості, або до неправильної класифікації до будь-якого довільного класу, крім правильного класу.

Атака на доступність призводить до ситуації відмови в обслуговуванні для моделі машинного навчання. У результаті модель машинного навчання стає або повністю недоступною для користувача, або якість значно погіршується до такої міри, що система машинного навчання стає непридатною для кінцевих користувачів.

5) Ціль: ціллю може бути конкретна IDS. В загальному випадку це може бути датчики, камери чи інше обладнання.

## 2.2 Змагальні атаки на системи виявлення вторгнень на основі машинного навчання

В загальному випадку для IDS зловмисне корисне навантаження в мережі розподіляється за трьома широкими типами: шкідливі файли (зловмисне програмне забезпечення), шкідливий текст (спам) і шкідливі URL-посилання (фішинг), зазначаючи, що зловмисники можуть використовувати комбінацію всіх трьох корисних навантажень у більшості кібератак. Наприклад, електронний лист зі спамом може також містити посилання на шкідливу URL-адресу або містити шкідливий вкладений файл.

Метод класифікації змагальних атак у мережній безпеці представлений на основі завдання мережної безпеки, враховуючи об’єкт даних, яким маніпулює зловмисник. Сфера застосування ознак змагальної атаки відповідає об’єкту даних, як показано на рис 2.4, а самі змагальні атаки розглядаються на основі фактичного корисного навантаження, яке атакується. Наприклад, коли надсилається електронний лист, корисне навантаження складається з тіла повідомлення, вкладень і URL-посилань. Заголовки та метадані, які допомагають полегшити доставку корисного навантаження, не розглядаються як частина корисного навантаження в контексті дослідження. Таким чином, накладні витрати протоколу не розглядаються як частина фактичних даних [35].

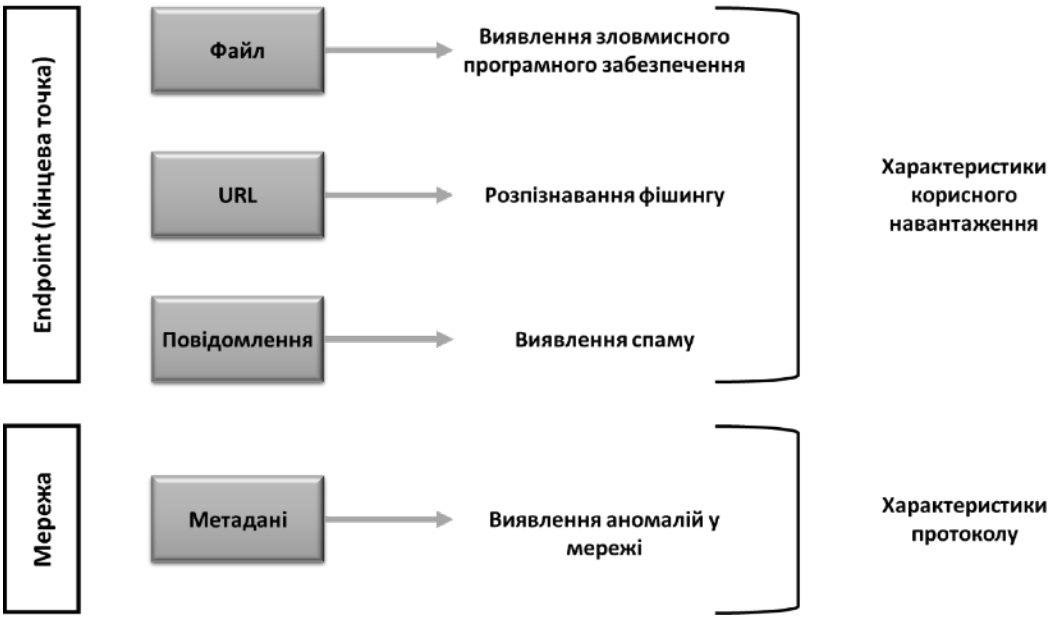



Рисунок 2.4 – Класифікація змагальної атаки

Змагальні атаки проти програм виявлення зловмисного програмного забезпечення, виявлення фішингу та виявлення спаму намагаються порушити такі

функції корисного навантаження, як бінарний файл, URL чи повідомлення електронної пошти. Ці атаки класифікуються як змагальні атаки на системи захисту кінцевих точок. І навпаки, також є змагальні атаки проти програм виявлення мережних аномалій, які намагатимуться порушити такі функції протоколу, як метадані мережі або заголовки протоколу, і класифікуються як агресивні атаки на системи захисту мережі.

Домен мережної безпеки, який використовує методи машинного навчання, поділяється на чотири широкі категорії, а саме виявлення зловмисного програмного забезпечення, виявлення фішингу, виявлення спаму та виявлення мережних аномалій.

Перші три вважаються захистом на основі кінцевої точки. Програми машинного навчання в цій категорії захисту на основі кінцевої точки зазвичай ініціюються функціями корисного навантаження.

Захист мережі насамперед передбачає виявлення мережних аномалій, і програми машинного навчання в цій категорії зазвичай ініціюються за допомогою функцій протоколу.

Слід зазначити, що на відміну від змагальних атак у сфері обробки зображень або комп'ютерного зору, навчання змагальності мережної безпеки є більш складним. Це відбувається тому, що навіть дуже незначні зміни URL, спаму, пакетів або байтів зловмисного програмного забезпечення бінарних файлів можуть суттєво змінити функціональність даних. У комп'ютерному зорі додавання дрібних збурень до зразка зображення не змінює людське сприйняття зображення, як і в обробці мови. Методи обробки тексту та фільтрації мережної безпеки схожі в цьому відношенні, оскільки дуже незначна зміна вхідних даних, як слово чи байт, змінить значення тексту або функціональність даних. Отже, підходи до генерації змагальних зразків у сфері систем фільтрації мережної безпеки на основі машинного навчання мають відбуватися таким чином, щоб шкідливі функції не спотворювалися. Було досліджено кілька підходів для досягнення цих змагальних атак, які розглянуті нижче.

До змагальні атак на системи виявлення вторгнень на основі машинного навчання відносяться наступні.

1) IDSGAN був запропонований у [40] для створення змагальних атак, спрямованих на IDS. IDSGAN базується на системі Wasserstein GAN, яка використовує генератор, дискримінатор і чорний ящик. Дискримінатор використовується для імітації системи виявлення вторгнень чорного ящика

та водночас для надання зразків шкідливого трафіку. IDSGAN може знизити рівень виявлення деяких моделей IDS приблизно до нуля відсотків.

2) Методи обфускації TCP були запропоновані у [41]. Основою цих методів є модифікація різних властивостей мережних з'єднань для обфускації зв'язку TCP. Таке перетворення дозволяє успішно ухиляється від широкого спектру класифікаторів виявлення вторгнень.

3) Поглиблене змагальне навчання у виявленні вторгнень це розширений фреймворк доповнення даних. У [41] запропоновано структуру, яка включає глибоке змагальне навчання зі статистичним навчанням таким чином, щоб використовувати доповнення даних на основі навчання. Була досліджена спільна імовірнісна генеративна модель Пуассона-Гамма використовується для синтезу змагальних вибірок.

4) Генеративні змагальні мережі для запуску та запобігання змагальним атакам на системи виявлення мережних вторгнень. У [42] запропоновано метод атаки на базі GAN, який був першою спробою застосувати змагальні атаки на основі GAN проти IDS із збереженням функціональної поведінки мережного трафіку. У деяких випадках така атака знизила точність моделі виявлення з 84,3 до 43,4 %.

5) Змагальне глибоке навчання для надійного виявлення зловмисного програмного забезпечення з бінарним кодуванням. У [43], представлені чотири змагальні методи атаки для створення змагального прикладу бінарного файлу зловмисного програмного забезпечення, який зберігає свою функціональність (rFGSM, dFGSM, BCA та BGA). Було розроблено структуру для навчання надійних моделей виявлення зловмисного програмного забезпечення, використовуючи формулювання сідлової точки, яка складається з проблем внутрішньої та зовнішньої максимізації. Підхід внутрішньої максимізації використовується для створення потужних змагальних прикладів, які максимізують втрати, а потім вони вводять їх у час навчання. У деяких умовах рівень ухилення від атаки перевищував 99 %.

6) Змагальні атак проти систем виявлення мережних вторгнень у програмно-конфігурованій мережі (англ. Software-defined Networking, SDN). У [44] розглядається приклад DDoS-атаки SYN Flood, в якому було продемонстровано здатність зменшити точність виявлення NIDS зі 100 % до 0 % на кількох класифікаторах за допомогою атак ухилення. Це була одна з найуспішніших спроб змагальних атак проти систем виявлення мережних вторгнень. Експериментальна платформа, яка була запропонована,

базувалася на базі машинного навчання NIDS для програмно визначених мереж під назвою Neptune. Було продемонстровано, що з пертурбацією кількох функцій точність виявлення конкретної атаки SYN Flood Distributed Denial of Service (DDoS) від Neptune знижується зі 100% до 0% за низкою класифікаторів.

Крім того, було запропоновано набір змагальних тестів під назвою Hydra для оцінки впливу змагальних класифікаторів ухилення від NIDS-Neptune на основі аномалій. Інструмент змагальної оцінки Hydra був розроблений, щоб надати користувачеві інтерфейс і платформу для перевірки стійкості свого ML-NIDS до агресивних атак. Ця система здійснює мережні атаки в середовищі SDN, застосовуючи до атак різні змагальні методи, щоб підірвати класифікацію атак. Hydra запускає власний емульований SDN (за допомогою Mininet), у якому виконує атаки на запущену NIDS, забезпечуючи класифікацію поточного трафіку (у даному прикладі, NIDS – це Neptune). Тестовий фреймворк показаний на рис. 2.5 [44].

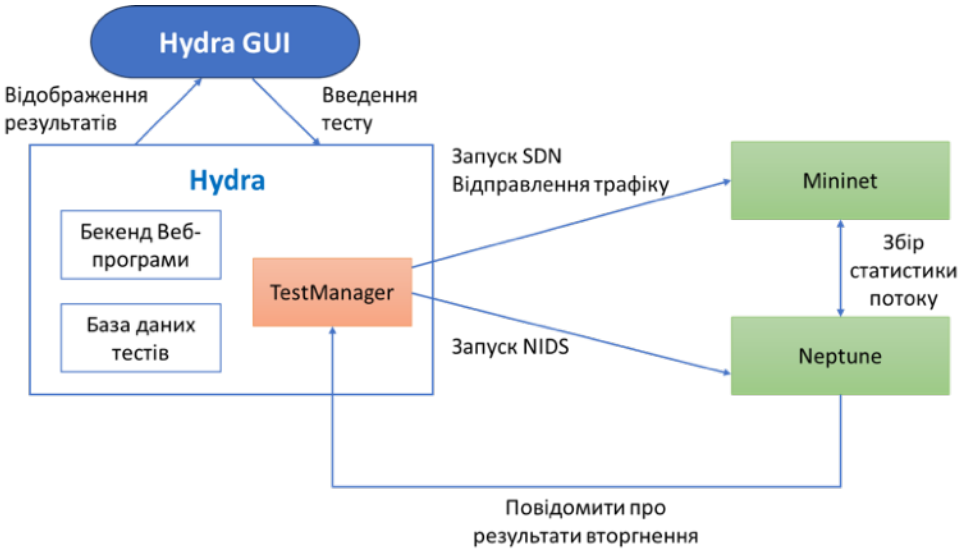



Рисунок 2.5 – Процес повної взаємодії між Hydra та Neptune для тестування змагальної атаки [44]

У дослідженні [44] розглянуто кілька класифікаторів і алгоритмів машинного навчання, які доводять, що алгоритми кластеризації були більш стійкими до змагальних вибірок порівняно з іншими типами машинного навчання. Зокрема, KNN виявився найнадійнішим класифікатором для протиборчих атак, проведених у рамках цього дослідження: лише з однією комбінацією порушень функції вдвічі знижує точність виявлення зі 100% до 50%. І навпаки, випадковий

ліс, LR і опорні вектори були загалом вразливі до тих самих пертурбацій, що призводило до зниження точності виявлення.

7) Боротьба зі змагальними атаками на системи безпеки на основі машинного навчання: у [45] було запропоновано метод атаки та захисту від декількох типів алгоритмів машинного навчання для систем виявлення мережних вторгнень. Було виконана оцінка як отруєння, так і атаки проти трьох контрольованих алгоритмів машинного навчання. Три алгоритми, а саме: Random forest, K-найближчий сусід і штучна нейронна мережа (багатошаровий перцептрон) MLP (англ. multi-layer perceptron), були використані для розробки системи виявлення мережних вторгнень. У них тяжкість отруєння та ухилення становила в середньому 70,1 та 66,4 % відповідно. Вони також продемонстрували, що змагальність була ефективною для підвищення надійності мережних систем виявлення вторгнень на основі глибокого навчання.

8) Змагальні атаки проти глибокого навчання для виявлення вторгнень у мережах IoT було розглянуто в [46]. В цій роботі досліджували ефективність змагальних зразків проти IDS на основі глибокого навчання в контексті мережі IoT. Надано комплексне порівняння між двома різними моделями глибокого навчання: загорткова нейронна мережа (CNN) і нейронна мережа прямого поширення (FNN). Було використано та досліджено нормалізації вхідних функцій у IDS на основі глибокого навчання в змагальному середовищі. Це підвищує надійність моделі глибокого навчання проти різних змагальних атак.

9) У [47] вивчали вплив атаки отруєння навчальними даними на виявлення онлайн центроїдної аномалії (IDS) із кінцевим розсувним вікном. Була проаналізована атака отруєння з обмеженим і повним контролем навчального набору даних, використовуючи реальний трафік HTTP з веб-сервера Інституту Фраунгофера FIRST.

Було проаналізовано поведінку «онлайнових систем виявлення центроїдних аномалій, як атаку з отруєнням даних. Щоб виявити аномалію для тестового прикладу x і даного набору даних X, було позначено x як сторонній приклад, якщо він лежить в області низької щільності порівняно з функцією щільності ймовірності простору вибірки X. Автори використали кінцеве розсувне вікно навчальних даних, де, у міру надходження кожної нової точки даних центр маси змінюється:

$$c' = c + \frac{1}{n} \bullet \left(x - x_i\right) \ . \qquad (2.2)$$

Як показано на рис. 2.6, маючи попередні знання про алгоритм і навчальний набір даних, зловмисник намагатиметься змусити алгоритм виявлення аномалії прийняти точку атаки A, яка лежить за межами суцільного кола, тобто $\left\|A - c\right\| > r$ [48].

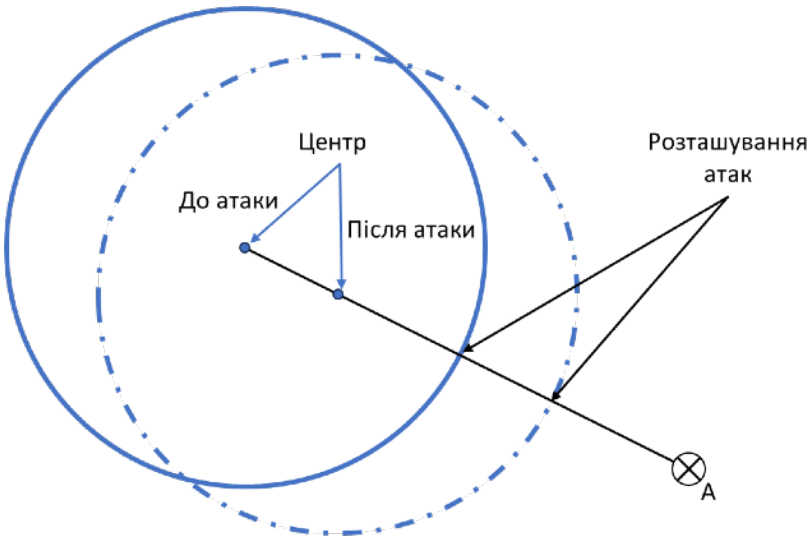



Рисунок 2.6 – Ілюстрація нападу отруєння

Це дослідження показує, що у разі, коли зловмисник має повний контроль над даними, то атакувати легко, тоді як у разі застосування додаткових обмежень для обмеження контролю над навчальними даними, припускаючи, що зловмисник може ввести невелику частину навчального набору даних, атака зазнає невдачі. Таким чином, додавання цих обмежень додає підходи до захисту від атак. У цей самий час, результати показують, що метод не можна вважати безпечним, якщо зловмисник має повний контроль над набором даних.

10) Уникнення моделей виявлення ботнетів машинного навчання за допомогою глибокого навчання з підкріпленням розглянуто у [49]. Була представлена загальна атака «чорного ящика» на моделі машинного навчання виявлення ботнетів. Було використане глибоке навчання з підкріпленням (DRL) для генерації змагальних потоків трафіку, щоб ввести в оману моделі виявлення. Агент навчання з посиленням оновлює змагальні зразки, щоб змінити часові та просторові характеристики потоків трафіку, не змінюючи вихідну функціональність і можливість виконання. Рівень ухилення від атаки коливався від 69,3 до 80,4 %.

11) У [50] запропоновано атаку-GAN для генерації шкідливих суперницьких необроблених пакетів, які можуть ввести в оману поточні системи виявлення вторгнень мереж машинного навчання в Інтернеті речей. Кожен байт у пакеті представлено за допомогою вбудовування слова. Зворотній зв'язок від жертви NIDS необхідний цій атаці чорного ящика для оновлення параметрів генератора. Рівень успіху атаки залежить від багатьох факторів, таких як модель машини та режими вбудовування байтів, але в найкращому випадку він досягав 98,42 %.

12) Обман системи виявлення вторгнень за допомогою змагального автокодувальника. У [51] представив структуру автокодувальника проти виявлення вторгнень (англ. Anti-Intrusion Detection AutoEncoder, AIDAE). AIDAE може створювати функції, що відповідають нормальному розподілу ознак, а також зберігає кореляцію між згенерованими безперервними та дискретними функціями. Було використано коефіцієнт збільшення ухилення (англ. Evasion Increase Rate, EIR), щоб оцінити свою атаку. EIR відображає силу ухилення шляхом порівняння коефіцієнта змагального виявлення з початковим (швидкість конкурентного виявлення поділена на початковий коефіцієнт виявлення). EIR був вищим за 0,9 у всіх проведених експериментах.

13) У [52] було представлено переформатування змагального мережного трафіку на основі часу (англ. Timing-Based Adversarial Network Traffic Reshaping, TANTRA), який обманює NIDS, змінюючи мережний трафік атаки за допомогою атрибута timestamp. Згідно з оцінкою авторів, TANTRA мала надзвичайно високий рівень успіху (99,99%). Однак, коли TANTRA було протестовано після навчання NIDS як з доброякісним, так і зі зміненим трафіком, то рівень успіху знизився.

На основі аналізу класифікації та прикладів атак можна зробити висновок, що у найбільш реалістичних атаках зловмисник маніпулює реальним мережним трафіком (де пертурбація виникає у проблемному просторі, а не в просторі ознак), маючи лише часткове знання функцій, без зворотного зв'язку від NIDS і без доступу до навчання даних. Ці данні не вимагають від зловмисника попереднього зламу системи або отримання інформації про певну конфіденційну та, можливо, добре захищену інформацію про NIDS та цільову організацію. Отже, подібні атаки можуть розгортати навіть зловмисники із середньою кваліфікацією та з невеликими витратами [53].

Змагальні атаки, що покладаються на потужність оракула, можуть бути реалізовані лише якщо цільова організація використовує комерційну NIDS, яку зловмисник також може купити, щоб вільно експериментувати з нею. Однак навіть у цій ситуації здійсненність може бути обмежена нестандартними конфігураціями, застосованими в цільовій організації, про які зловмисник, швидше за все, не знатиме та не відтворить у своєму середовищі тестування [53].

Атаки, які вимагають використання тренувальних даних або моделі виявлення NIDS, створеної та підтримуваної всередині цільової організації, або повного знання набору функцій, є нереальними, оскільки ці дані можуть бути отримані лише коли зловмисник вже зміг скомпрометувати системи, що належать до цільової мережі, яка зберігає інформацію. Ці ж висновки також стосуються атак, які вимагають використання системи виявлення як оракула. В принципі, подібного результату може досягти зловмисник, який надсилає зразки зловмисного трафіку системі запобігання вторгненню в мережу (тобто системі виявлення, яка також налаштована на блокування зловмисних мережних комунікацій). Однак це дозволяє отримати лише часткове знання, збільшує вартість і тривалість кампанії атаки, а також збільшує ймовірність того, що ці допоміжні дії атаки можуть викликати інше попередження.

Найбільш малоймовірними є атаки, які вимагають повного доступу до тренувальних даних або моделі виявлення комерційної IDS, оскільки зловмисник повинен мати можливість скомпрометувати постачальника комерційного захисного рішення. Навіть атаки, які вимагають даних моделі виявлення комерційної NIDS, є досить нереалістичними, оскільки вони вимагають від зловмисника довільної модифікації детектора на основі машинного навчання, який виконує комерційний пристрій NIDS, придбаний цільовою організацією. Нарешті, атаки, які можна здійснити лише в просторі ознак, потрапляють до цієї категорії, оскільки зловмиснику потрібно буде скомпрометувати компонент детектора, який витягує функції з необробленого трафіку [53].

## 2.3 Класифікація та характеристика методів протидії атакам на системи виявлення вторгнень

На основі аналізу джерел можна зробити висновок, що розробка методів протидії змагальним атакам на системи виявлення вторгнень базується на методах протидії змагальним атакам на системи розпізнавання зображень. Це пов’язано з тим, що саме системи розпізнавання зображень на основі машинного навчання

насамперед зазнали впливу таких атак. Тому при аналізі методів протидії деякі будуть розглянуто на прикладах систем розпізнавання зображень.

У статті [54] автори вперше запропонували три широкі підходи до захисту алгоритмів машинного навчання від агресивних атак: регулярізація, рандомізація та приховування інформації. Вони визначають відповідні властивості для аналізу атак на системи машинного навчання, які впливають на вибір методу захисту.

1) Вплив:

– причинний – атаки змінюють процес навчання через вплив на навчальні дані;

– дослідницький – атаки не змінюють процес навчання, але використовують інші методи, такі як зондування учня або офлайн-аналіз для виявлення інформації.

2) Специфіка:

– цілеспрямована – специфікою цієї атаки є безперервний спектр. У точці призначення фокус атаки зосереджений на конкретній точці або невеликій групі точок;

– невибіркова – у невибірковій точці призначення супротивник має більш гнучку мету, яка включає дуже загальний клас балів, наприклад «будь-який хибний негативний результат».

3) Порушення безпеки:

– цілісність – атака на цілісність призводить до того, що точки вторгнення класифікуються як нормальні (хибний негативний результат);

– доступність – атака на доступність є ширшим класом атак, ніж атака на цілісність. Атака на доступність призводить до такої кількості помилок класифікації, як помилково негативних, так і помилково позитивних результатів, що система стає фактично непридатною для використання.

Ці три групи визначають простір атак наведені в таблиці 2.2.

У причинних атаках супротивник певною мірою контролює навчання учня. Атака, яка змушує учня неправильно класифікувати точки вторгнення, наприклад атака, яка змушує IDS не позначити відомий експлойт як вторгнення, є причинною атакою на цілісність. Різниця між цілеспрямованими та невибірковими причинними атаками на цілісність полягає в різниці між вибором одного конкретного експлойта чи просто пошуком будь-якого експлойта. Причинна атака на доступність призводить до погіршення продуктивності учня. Наприклад, зловмисник може змусити IDS відхилити багато законних HTTP-з’єднань. Причинна атака доступності може бути використана, щоб змусити

системного адміністратора вимкнути IDS. Цілеспрямована атака зосереджена на певній службі, тоді як невибіркова атака має ширший масштаб. Дослідницькі атаки не намагаються вплинути на навчання, натомість вони намагаються знайти інформацію про стан учня. Дослідницькі атаки на цілісність прагнуть знайти вторгнення, які не розпізнає учень.

Таблиця 2.2 – Простір атак

| | | Цілісність | Доступність |
|---|---|---|---|
| Причинний вплив | Цілеспря-мована специфіка | Дозволити конкретне вторгнення | Створити достатню кількість помилок, щоб зробити систему непридатною для використання однією особою або службою |
| | Невибіркова специфіка | Дозволити принаймні одне вторгнення | Створити достатню кількість помилок, щоб зробити учня непридатним |
| Дослідницький вплив | Цілеспря-мована специфіка | Знайти дозволене вторгнення з невеликого набору можливостей | Знайти набір точок, неправильно класифікованих учнем |
| | Невибіркова специфіка | Знайти дозволене вторгнення | |

Після визначення можливих атак, у [54] визначаються потенційні методи захисту від атак (таблиця 2.3) у відповідності із простір атак, який було розглянуто.

Регуляризація використовується в статистиці для обмеження або зміщення вибору гіпотези, коли проблема страждає від браку даних або шумових даних. Це також можна інтерпретувати як кодування попереднього розподілу на параметри, штрафуючи вибір параметрів, який є менш ймовірним апріорі. Регуляризацію та попередні розподіли можна розглядати як штрафні функції.

Рандомізація пропонується як потенційний інструмент проти цілеспрямованих причинних атак. Під час такої атаки супротивник повинен

виконати певний обсяг роботи, щоб перемістити межу прийняття рішення за межі цільової точки. Якщо є певна випадковість у розташуванні межі, а супротивник має недосконалий зворотний зв'язок від учня, тоді потрібна додаткова робота.

У статті [54] автори класифікують методи захисту за двома стратегіями, проактивні та реактивні:

– проактивна стратегія робить глибокі нейронні мережі більш надійними до того, як зловмисники генеруватимуть приклади змагальності;

– реактивна стратегія має на меті виявляти суперечливі приклади після побудови глибоких нейронних мереж.

У таблиці 2.4 наведені приклади стратегій.

Таблиця 2.3 – Методи захисту від атак

| | | Цілісність | Доступність |
|---|---|---|---|
| Причинний вплив | Цілеспрямована специфіка | Регулярізація; Рандомізація | Регулярізація; Рандомізація |
| | Невибіркова специфіка | Регулярізація | Регулярізація |
| Дослідницький вплив | Цілеспрямована специфіка | Приховування інформації; Рандомізація | Приховування інформації |
| | Невибіркова специфіка | Приховування інформації | |

Таблиця 2.4 – Стратегії методів захисту

| | Оборонні стратегії |
|---|---|
| Реактивні | Виявлення змагальності |
| | Реконструкція вхідних даних |
| | Перевірка мережі |
| Проактивні | Мережна дистиляція |
| | (Пере)підготовка змагальності |
| | Посилення класифікатора |

Оскільки змагальні приклади представляють найгірший сценарій зміни розподілу, завдання створення змагальної вибірки є проблемою, яку можна розв'язати лише приблизно. У загальному випадку, методи змагальних атак

представляють собою здебільшого алгоритми оптимізації в пошуку нижнього граничного збурення, яке відповідає змагальній вибірці.

Найпоширеніші методи захисту, які використовуються сьогодні та класифіковані на основі стратегії та підходу представлені на рис.2.7.


Рисунок 2.7 – Методи захисту від атак

До них відносяться наступні.

1) Градієнтне маскування. Оскільки більшість методів змагальних атак базуються на використанні градієнта, метод градієнтного маскування змінює модель машинного навчання, намагаючись приховати її градієнт від зловмисника. У статті [55] автори продемонстрували ефект маскування градієнта шляхом насичення сигмоподібної мережі, що призводить до зникнення ефекту градієнта в атаках на основі градієнта. Нейронні мережі змушують працювати в нелінійній насичувальній системі. Використовуючи регулярізацію Якобіана для кожного мережного рівня, включаючи вихідний рівень, модель стає нечутливою до пертурбацій, які генеруються за допомогою методу знаків швидкого градієнта (FGSM) і ітераційних змагальних атак. Доведено, що проста, біологічно натхненна стратегія для пошуку дуже нелінійних мереж, що працюють у насиченому режимі, надає цікаві механізми для захисту DNN від ворожих прикладів без жодного їх обчислення. У результаті отримано покращену продуктивність порівняно з мережами, навченими змагальністю, на прикладах змагальності, згенерованих методом знакового швидкого градієнта. Крім цього, мережі із насиченням є відносно надійними проти ітераційних цільових методів, включаючи противників другого порядку.

Однак слід зазначити, що градієнтне маскування реагує як надмірне пристосування в деяких експериментах. Досі вважалося, що градієнтне маскування відбувається, коли DNN навчаються захищатися. Це явище

відбувається за нормальних умов навчання без спеціальних регуляризацій або стратегій захисту від змагальних градієнтів. Потенційні захисні механізми спонукають NN зближуватися до певних локальних рішень, і мінімум не керується лише методологіями захисту.

2) Захисна дистиляція. Техніка дистиляції спочатку була запропонована у [56] для передачі знань від великих нейронних мереж до менших. Щоб реалізувати дистиляційний підхід, було побудовано 10 моделей DNN з однаковою архітектурою та методом навчання та використано м'які цілі, щоб уникнути переобладнання, яке виникає під час використання жорстких мішеней. Було доведено, що ансамблева модель здатна передавати знання дистильованій моделі краще, ніж індивідуальні моделі. Однак ансамбль вимагає великих обчислювальних моделей, які мають великі мережі та великі набори даних. Тому в цьому випадку використовували спеціальні моделі навчання, кожна з яких використовує підмножину класів набору даних, щоб зменшити обсяг обчислень. Представлений метод був адаптований для захисту від змагального впливу, використовуючи вихідні дані оригінальної нейронної мережі для навчання меншої мережі (рис 2.8).

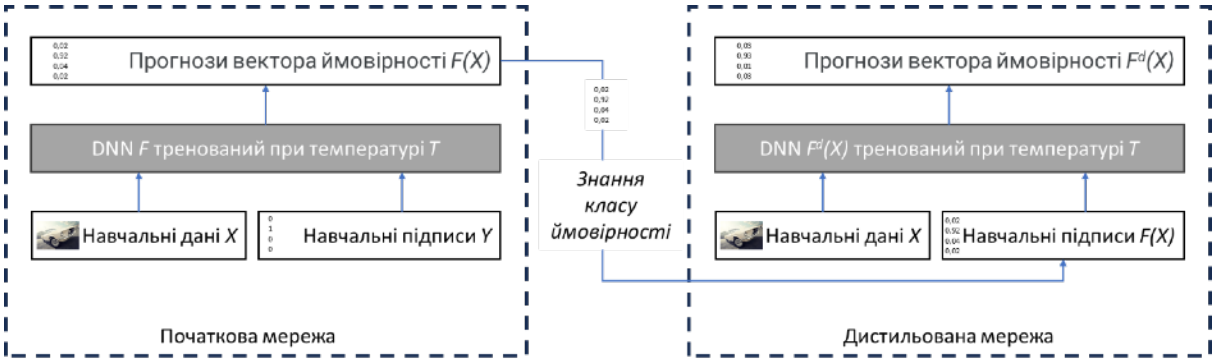



Рисунок 2.8 – Мережна дистиляція глибоких нейронних мереж [56]

Захисну дистиляцію спочатку перевіряли проти агресивних атак у комп'ютерному зорі, але необхідні подальші дослідження, щоб визначити її ефективність, наприклад у IDS.

3) Тренування змагальності. Тренування змагальності – це метод, спрямований на підвищення стійкості моделі машинного навчання до змагальних зразків шляхом мінімізації втрат L на парах даних/міток $\{X_i, y_i\}$ при максимальному збільшенні відповідної функції втрат.

В статті [57] автори запропонували метод з трьох етапів, відомий як тренування змагальності для захисту від ворожих атак:

– навчання класифікатора на вихідному наборі даних;

– створення змагальних зразків;

– ітерація додаткових епох навчання з використанням змагальних зразків.

Загалом тренування змагальності базується на мінімально-максимальному формулюванні, яке вирішує дві проблеми: напади – як внутрішню проблему максимізації та захист – як зовнішню проблему мінімізації для досягнення оптимізації. Внутрішня максимізація спрямована на генерацію змагальної версії зразків, що призводить до максимізації втрати моделі. Зовнішня мінімізація має на меті мінімізувати втрати шляхом знаходження параметрів моделі, які створюють більш надійну модель з меншими змагальними втратами. За результатами випробування методу змагального навчання встановлено її ефективність для мережної безпеки. Було встановлено, що це покращує ефективність класифікації моделі машинного навчання та робить її більш стійкою до змагального впливу. Однак змагальне навчання має певні обмеження, особливо в контексті змагального машинного навчання в галузі мережної безпеки. По-перше, зловмисник може застосувати інший метод атаки, відмінний від того, який використовувався під час навчання системи. По-друге, зловмисник може розробити змагальні пертурбації для моделі глибокого навчання, яка вже була навчена за допомогою тренування змагальності, і створити нові змагальні пертурбації, які зроблять попереднє тренування змагальності неефективним. Було також показано, що тренування змагальності може знизити ефективність моделей глибокого навчання на чистих вхідних даних.

4) Градієнтна регуляризація – це техніка, яка штрафує великі зміни у виході деякого рівня нейронної мережі, щоб налаштувати моделі машинного навчання, мінімізувати функцію втрат, підвищити надійність моделі та запобігти надмірному або недостатньому оснащенню. Цей підхід застосовується як захист від агресивних атак. Було виявлено [58], що навчання DNN з градієнтною регуляризацією покращує стійкість до змагальних збурень настільки ж або більше, ніж тренування змагальності. Крім цього, поєднання обох підходів (регуляризація градієнта та тренування змагальності) досягає більшої надійності. Основний недолік цього методу полягає в тому, що вона подвоює час навчання.

5) Виявлення змагальних зразків. Є декілька підходів для виявлення наявності змагальних зразків на етапі навчання моделі машинного навчання.

Один з таких підходів, запропонований в роботі [59], ґрунтується на припущенні, що вибірки противника мають більшу невизначеність, ніж чисті дані,

і використовує нейронну мережу Байєса, яка знаходиться у шарах відсіву нейронних мереж, для оцінки ступеня невизначеності вхідних даних з метою виявлення змагальних зразків. Інші підходи включають використання розбіжностей ймовірностей, які були використані для реалізації засобу захисту зразків для нейронних мереж від змагальних атак. Цей засіб обробляє ненадійний вхід двома методами. Він виявляє змагальні зразки з великими збуреннями за допомогою детекторних мереж і підштовхує приклади з невеликими пертурбаціями до різноманіття нормальних прикладів. Ці два методи працюють спільно для підвищення точності класифікації. Крім того, використовуючи автокодувальник як мережу детектора, засіб вчиться виявляти змагальні приклади, не вимагаючи змагальних прикладів або знання процесу їх генерації, що веде до кращого узагальнення. Експерименти показують, що засіб ефективно захищався від сучасних атак. Для випадку, якщо зловмисник знає навчальні приклади засобу, була описана нова модель загроз «сірого ящика» та використана різноманітність для ефективного захисту від цієї атаки.

Використання допоміжної мережі вихідної мережі було запропоновано у [60]. Було емпірично показано, що змагальні зразки можна якісно виявити за допомогою підмережі детектора, приєднаної до основної мережі класифікації. Хоча це не дозволяє прямо класифікувати змагальні зразки правильно, це дозволяє пом’якшити змагальні атаки на системи машинного навчання, вдаючись до резервних рішень, тобто додаткового підтвердження. На жаль, таке рішення може бути важко реалізувати для IDS.

Крім того, можливість виявлення змагальних збурень може в майбутньому дозволити краще виявляти змагальні зразки шляхом застосування мережної інтроспекції до мережі детектора. Також градієнт, що поширюється назад через детектор, може бути використаний як джерело регуляризації класифікатора проти змагальних прикладів.

В статті [61] автори також запропонували методи виявлення змагальної атаки та розпізнавання змагальних зразків за допомогою техніки причинного висновку для створення причинно-наслідкової моделі для опису створення та ефективності змагальних зразків, які атакують DNN. Було використано методологію причинно-наслідкового висновку, щоб створити причинно-наслідкову модель для опису формування та ефективності змагальних вибірок. Крім того, ця причинно-наслідкова модель дозволила віднести вихід моделі розпізнавання до підобластей вхідного зображення та інтерпретувати робочий механізм змагальних зразків. Таким чином вдалося виявити багато повчальних

феноменів змагальних атак і змагальних зразків. На основі запропонованої каузальної моделі було розроблено метод виявлення змагальної атаки та метод розпізнавання змагальної вибірки. Крім того, на основі потужних трансформерів стало можливим виявляти та розпізнавати змагальні зразки без додаткових моделей чи навчання. Результати експериментів демонструють перевагу таких методів, особливо здатність до узагальнення.

6) Зменшення ознак. Просте зменшення ознак було оцінено в роботі [62], але було визнано недостатнім для захисту від змагальних атак. Крім цього, це може знижати загальну ефективність самої IDS.

7) Рандомізація вхідних даних. Слід зазначити, що цей метод, насамперед, використовується для систем розпізнавання зображень. Для таких систем був запропонований метод на основі рандомізації, який додає випадковий шар зміни розміру та випадковий шар заповнення на початку класифікаційних мереж. Немає необхідності в перенавчанні або точному налаштуванні, що робить запропонований метод дуже простим у застосуванні [63].

Перший рівень рандомізації – це шар випадкової зміни розміру, який змінює розмір вихідного зображення $X_n$ із розміром W×H×3 на нове зображення $X'_n$ із випадковим розміром W′×H′×3. Рекомендовано, що |W′ - W| і |H - H′| мають бути в розумно малому діапазоні, інакше продуктивність мережі на неконкурентних зображеннях значно впаде. Беручи за приклад мережу Inception-ResNet, вихідний розмір вхідних даних становить 299×299×3. Емпірично було виявлено, що продуктивність мережі майже не падає, якщо контролювати висоту та ширину зміненого зображення $X'_n$ в діапазоні [299, 331].

Другий рівень рандомізації – це шар випадкового заповнення, який випадковим чином додає нулі навколо зміненого розміру зображення. Зокрема, додавши змінене зображення $X'_n$ до нового зображення $X''_n$ n із розміром W″×H″×3, ми можемо заповнити ω нульових пікселів ліворуч, W″-W′-ω нульових пікселів праворуч, h нульових пікселів згори та H″-H′-h нульових пікселів внизу. Це призводить до загальної кількості (W″-W′+1) (H ″-H′+1) різних можливих шаблонів заповнення.

Під час реалізації вихідне зображення спочатку проходить через два рівні рандомізації, а потім трансформоване зображення передається оригінальному CNN для класифікації, як показано на рис 2.9. Вхідне зображення $X_n$ спочатку проходить через рівень випадкової зміни розміру із застосуванням випадкового масштабу. Потім випадковий шар заповнення доповнює змінене зображення $X'_n$

випадковим чином. Отримане доповнене зображення $X'_n$ використовується для класифікації [63].

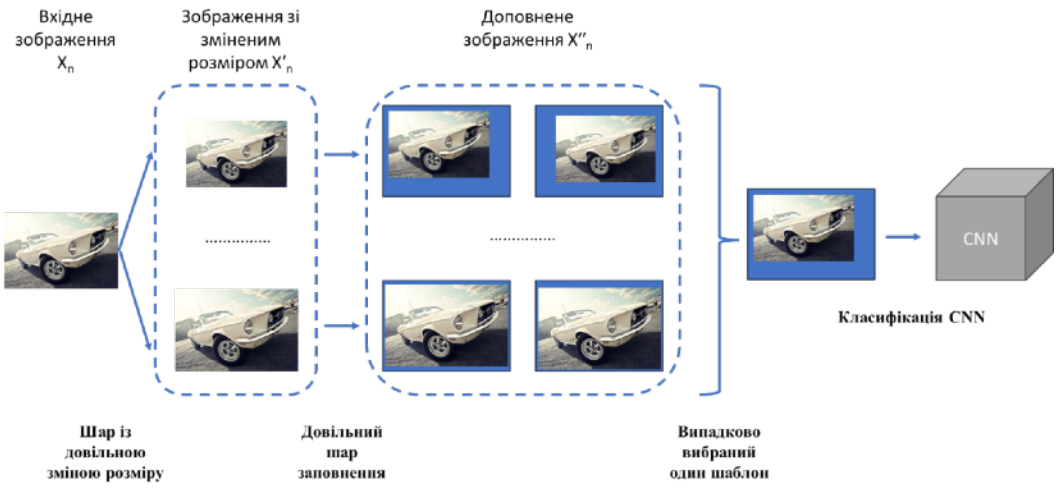



Рисунок 2.9 – Механізм захисту на основі рандомізації [63]

Автори у [64] також спробували подібний метод шляхом введення випадкового гаусового шуму. Такий метод має численні переваги, такі як простота, низька обчислювальна складність і усунення необхідності додаткового навчання. Основним недоліком є те, що використання цього методу захисту в домені безпеки мережі може змінити функціональність вхідних даних (виконуваних файлів, пакетів тощо). Цей метод потребує оцінки можливості використання в області мережної безпеки.

8) Ансамблеві захисти. Подібно до ідеї ансамблевого навчання, яке поєднує одну або декілька технік машинного навчання, цей метод пропонує використовувати кілька захисних стратегій як техніку захисту від змагальних зразків. PixelDefend був запропонований в [65] для поєднання методів виявлення змагальності з одним або декількома іншими методами для створення більш надійного захисту від атак противника.

Основна ідея PixelDefend полягає в тому, щоб очистити вхідні данні, вносячи в них невеликі зміни, щоб повернути їх до тренувального розподілу, тобто перемістити зображення в область високої ймовірності. Потім виконується класифікація очищеного зображення за допомогою будь-якого існуючого класифікатора. Очищені зображення зазвичай можна правильно класифікувати. Формально є розподіл тренувального зображення p(X) і вхідне зображення X роздільної здатності I×J з X[i; j; k] піксель у місці (i; j) і канал $k \in \{1,\cdots,C\}$. Головна мета – знайти зображення X*, яке максимізує p(X) з урахуванням обмеження, що X* знаходиться всередині $\epsilon_{defend}$-кулі X:

$$\max_{X^*} p(X^*), \text{ такий, що } \left\| X^* - X \right\|_\infty \leq \epsilon_{defend} \ . \qquad (2.3)$$

Тут $\epsilon_{defend}$ відображає компроміс, оскільки великий $\epsilon_{defend}$ може змінити значення X, тоді як малий $\epsilon_{defend}$ може бути недостатнім для повернення X до правильного розподілу. Цей метод також потребує оцінки можливості використання в області мережної безпеки.

Як вже було зазначено, деякі представлені методи протидії змагальним атакам було розроблено для систем розпізнавання зображень. Ефективність їх застосування для IDS потребує додаткових досліджень. Крім цього, є нагальна необхідність в розробці нових методів, адаптованих для конкретних типів ISD та області застосування. Окремою задачею є розвиток методик випробувань існуючих та нових методів протидії змагальним атакам на IDS. На жаль, станом на сьогодні відсутній загальноприйнятий підхід для порівняння різних методик протидії. Таким чином є актуальною задача розробки методики кількісного порівняльного аналізу методів протидії змагальним атакам на IDS.

# 3 КІЛЬКІСНИЙ ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПРОТИДІЇ ЗМАГАЛЬНИМ АТАКАМ НА СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ

## 3.1 Огляд властивостей IDS

На основі аналізу [66-71] можна сформулювати загальний перелік властивостей IDS.

1) Висока керованість IDS (B1):проста інсталяція та ефективне керування режимами системи.
2) Висока точність виявлення (B2): точність ідентифікації вторгнень. Це означає, що вторгнення ідентифікується саме як вторгнення, яке потребує реагування. Тоді як всі інші події не ідентифікуються як вторгнення.
3) Повнота (B3): виявлення широкого спектру вторгнень. Для NIDS це множина мережних атак різного виду.
4) Своєчасність (B4): виявлення та реагування на вторгнення в режимі реального часу. Негайне виявлення та реагування при наявності вторгнення. Тобто мінімальна тривалість часу між виникненням вторгнення та його виявленням і реагуванням.
5) Мінімізація споживання ресурсів (B5): обсяг обчислювальних ресурсів (тобто споживання енергії, використання мережі, процесора, ПЗП та ОЗП). Це означає, що споживання ресурсів оптимізується з метою вивільнення обчислювальних ресурсів, які можуть бути використані іншими компонентами системи. Ця властивість є особливо важливою, наприклад, для IoT.
6) Мінімальний вплив на інформаційно-комунікаційну систему (ІКС) (B6): прямий або непрямий вплив IDS не має негативного впливу на коректну роботу ІКС в цілому. Елементи IDS не створюють додаткове навантаження на інші компоненти ІКС. Це особливо важливо, оскільки навантаження на мережу в сучасних системах є дуже високим. Це також означає, що програмні компоненти промислової IDS, встановлені не на спеціалізованому обладнанні IDS, а на інших пристроях ІКС, не впливають негативно на роботу інших компонентів системи.
7) Висока продуктивність обробки (B7): обчислювальні операції, що виконуються компонентами IDS виконуються за найкоротший проміжок часу, забезпечуючи при цьому надійні та якісні результати.

8) Відмовостійкість (B8): IDS здатна забезпечувати певний рівень захисту або функціональності навіть за наявності несправностей у її компонентах.

9) Надійність (B9): IDS здатна підтримувати робочий стан навіть за наявності неочікуваних або несприятливих впливів, які порушують її функціонування.

10) Стійкість (B10): IDS здатна відновлюватися після неочікуваних або несприятливих впливів, які могли спричинити збої в її роботі.

11) Достовірність даних (B11): дані, що обробляються IDS, зберігають цілісність. Ці дані повинні бути також автентифіковані та захищені від несанкціонованих модифікацій. Насамперед це стосується структури та даних, що використовуються для навчання.

## 3.2 Аналіз показників IDS

В наукових публікаціях активно розглядається питання таксономії показників IDS. Значення показників можуть бути визначені наступними способами: аналіз прямого спостереження в лабораторних умовах або аналіз вихідного коду, а також аналіз документації (специфікації, технічні документи, надані постачальником або користувачами). Кожний показник призначений для вимірювання одним або обома цими способами. За видом представлення показники можна розділити на наступні [72].

1) Чітко визначені: показники які є спостережуваними, відтворюваними, кількісно вимірюваними та характерними (показник повинен чітко диференціювати схожі системи).

2) Дискретне оцінювання: спрощує процес присвоєння значень показників для даної системи.

3) Гнучке зважування: дозволяє використовувати будь-яку послідовну числову систему ваг, дискретну або неперервну з верхньою або нижньою межею, визначеною оцінювачем.

В рамках цієї роботи пропонується виконати аналіз показників IDS та виявити рівень впливу на них застосування методів протидії змагальним атакам.

На основі аналізу [71, 72] можна класифікувати показники IDS за наступними ознаками.

1) Ефективність архітектури (П1). Це суб'єктивні показники, які потребують оцінки користувача. До таких показників відносяться:

– розподілене управління (П1.1). Визначає можливості розподілу керування між різними аналізаторами. Використовується для визначення того, наскільки IDS підтримує розподілене управління;

– складність конфігурації (П1.2). Рівень складності встановлення та налаштування IDS користувачем. Від того, наскільки добре користувач розуміє процес розгортання IDS, залежить правильність розгортання IDS;

– простота управління політиками та ліцензіями (П1.3). Рівень простоти налаштування політик безпеки та виявлення вторгнень, а також складність отримання, оновлення та продовження ліцензій;

– доступність оновлень (П1.4). Доступність і вартість оновлень продукту;

– регульована чутливість (П1.5). Легкість зміни чутливості IDS в різний час і для різних середовищ з метою досягнення балансу між помилковими спрацьовуваннями і помилковими відмовами;

– масштабоване балансування навантаження (П1.6) Продуктивність балансувальників навантаження та їх вплив. Вимірює здатність IDS розділяти трафік на незалежні, збалансовані навантаження датчиків, а також здатність підпроцесу балансування навантаження масштабуватися вгору і вниз;

– відстеження стану (П1.7) Це ще один важливий суб'єктивний показник, оскільки вона ефективно зменшує кількість хибних спрацьовувань. Мережний IDS, який виконує відстеження стану, знає, які сеанси бачить ціль, і не піднімає сповіщення про трафік, який ціль відкине як недійсний. Це корисно для захисту NIDS від штормів випадкового трафіку.

2) Показники інтерактивності (П2):

– взаємодія з брандмауером (П2.1). Здатність взаємодіяти з брандмауером;

– взаємодія з маршрутизатором (П2.2). Ступінь, до якого IDS взаємодіє з маршрутизатором (наприклад, перенаправляє трафік зловмисника на Honeypot);

– зручність для користувача (П2.3). Простота встановлення та налаштування IDS в середовищі користувача.

3) Показники виявлення (П3) розраховуються на основі значень, отриманих в результаті ідентифікації або помилкової ідентифікації

вторгнень, виконаних IDS, і часто отримуються з тестів IDS, оскільки необхідно знати загальну кількість вторгнень і ненавмисних подій, які надходять до IDS. Ці події можна представити матрицею плутанини, яка включає чотири виміри (таблиця 3.1). До цих значень відносяться:

- істинно позитивний (англ. True Positive,TP): кількість подій вторгнення, ідентифікованих як вторгнення;
- хибно негативний (англ. False Negative, FN): кількість подій вторгнення, не ідентифікованих як вторгнення;
- істинно негативний (англ. True Negative, TN): кількість подій, які не є вторгненнями та не були ідентифіковані як вторгнення;
- хибно позитивні (англ. False Positive, FP): кількість подій, які не є вторгненнями, але були ідентифіковані як вторгнення.

Таблиця 3.1 – Матриця плутанини

| Передбачив<br><br>Насправді | позитивна | негативна |
|---|---|---|
| позитивна | TP<br>(True Positive) | FN<br>(True Positive) |
| негативна | FP<br>(False Positive) | TN<br>(True Negative) |

З цих значень можна отримати набір оціночних показників. Характерною особливістю цих показників є те, що вони використовуються в машинному навчанні для оцінки алгоритмів, які застосовуються для задач класифікації. Це відбувається тому, що виявлення вторгнень за своєю суттю є задачею класифікації. Таким чином до показників виявлення можна віднести:

- частота істинних позитивних результатів (англ. True Positive Rate, TPR) (П3.1): Відношення реальних виявлених вторгнень до загальної кількості наявних вторгнень. Він також відомий як чутливість (англ. Sensitivity), відгук (англ. Recall) або частота виявлення (англ. Detection Rate, DR). Обчислюється за виразом (3.1):

$$\mathrm{TPR} = \mathrm{Sensitivity} = \mathrm{Recall} = \mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} . \qquad (3.1)$$

– істинно негативний показник (англ. True Negative Rate, TNR) (П3.2): також відомий як специфічність (англ. Specificity). Відношення кількісті істинно негативних результатів, до загальної кількості негативних результатів. Обчислюється за виразом (3.2):

$$\mathrm{TNR} = \mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \quad . \tag{3.2}$$

– коефіцієнт хибно позитивних спрацьовувань (англ. False Positive Rate, FPR) або частота помилок першого роду (П3.3): Також відомий як частота хибних тривог (англ. False Alarm Rate, FAR) або випадіння (англ. Fallout). Відношення кількості хибно позитивних подій, які були помилково ідентифіковані як вторгнення, до загальної кількості негативних результатів. Обчислюється за виразом (3.3):

$$\mathrm{FPR} = \mathrm{FAR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \quad . \tag{3.3}$$

– коефіцієнт хибних спрацьовувань (англ. False Negative Rate, FNR) або частота помилок другого роду (П3.4). Відношення невиявлених вторгнень до загальної кількості реальних вторгнень. Обчислюється за виразом (3.4):

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \quad . \tag{3.4}$$

– прецизійність (англ. Precision, Pr)) або позитивна прогностична цінність (англ. Positive Predictive Value, PPV) (П3.5) [73]. Оцінює здатність класифікатора уникати помилкової класифікації позитивних зразків. Відношення правильно ідентифікованих вторгнень до всіх виявлених вторгнень. Обчислюється за виразом (3.5):

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad . \tag{3.5}$$

– точність (англ. Accuracy, ACC) (П3.6). Відношення правильно класифікованих вторгнень до загальної кількості прогнозів. Обчислюється за виразом (3.6):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad . \tag{3.6}$$

– слід зазначити, що показник точності рекомендовано використовувати для випадків, коли кількість зразків кожного класу приблизно однакові. Для інших випадків рекомендується показник F-міра (англ. F-score) (П3.7). Обчислюється за виразом (3.7):

$$Fscore = \frac{Pr \bullet DR}{Pr + DR} = \frac{2 \bullet TP}{2 \bullet TP + FP + FN} = \frac{2}{\frac{1}{Pr} + \frac{1}{Recall}} \quad . \tag{3.7}$$

– G-mean: Середнє геометричне значення прецизійності та відгуку (тобто, DR або TPR) (П3.8). Високе G-середнє вказує на високу точність виявлення вторгнень. Обчислюється за виразом (3.8):

$$G - mean = \sqrt{Pr \bullet DR} \quad . \tag{3.8}$$

– коефіцієнт кореляції Метьюса (англ. Matthews Correlation Coefficient, MCC) (П3.9). Коефіцієнт кореляції між спостережуваними та виявленими вторгненнями, який використовується, коли різниця між кількістю вибірок двох класів занадто велика. Його значення знаходиться в діапазоні від -1 (повна помилкова ідентифікація вторгнення) до 1 (ідеальне виявлення). Значення 0 означає випадкову ідентифікацію [74]. Обчислюється за виразом (3.9):

$$MCC = \frac{TP \bullet TN + FP \bullet FN}{\sqrt{(TP + FP) \bullet (TP + FN) \bullet (TN + FP) \bullet (TN + FN)}} \quad . \tag{3.9}$$

– графік робочої характеристики приймача (Receiver Operating Characteristic, ROC) (П3.10) дозволяє проаналізувати ефективність

швидкості виявлення. Зазвичай на графік наносяться значення TPR проти FPR (рис. 3.1). Чисельно розглядається площа під кривою ROC [75].

4) Часові показники (П4) розраховуються на основі значень, отриманих в результаті вимірювання часової тривалості подій. Ці події включають події, пов'язані з IDS, вторгненнями або іншими компонентами, які можуть взаємодіяти з IDS або існувати в тому ж середовищі, що й IDS:

– час навчання (англ. Training Time, TT) (П4.1): час, необхідний для навчання IDS на основі аномалій очікуваної нормальної поведінки;

– тривалість вторгнення (англ. Intrusion Duration) (П4.2): час, що пройшов між початком і закінченням вторгнення;

– час виявлення (англ. Detection Time, DT) (П4.3): час, що минув від початку вторгнення до його виявлення IDS;

– час реакції (англ. Response Time, RsT) (П4.4): час, що минув з моменту виявлення вторгнення до моменту виконання дії реагування;

– час затримки (англ. Delay Time) (П4.5): різниця між очікуваним часом події та пізнішим часом, коли вона насправді відбувається. Прикладом цього є мережна затримка. Мережна затримка – це затримка доставки пакетів у мережі. Ця затримка може бути спричинена IDS, вторгненням або іншими компонентами мережі;

– час обробки події (англ. Processing Time per Event) (П4.6): час, що минув з моменту, коли подія почала оброблятися IDS, до моменту прийняття остаточного рішення щодо цієї події (тобто, чи є вона вторгненням, чи ні).

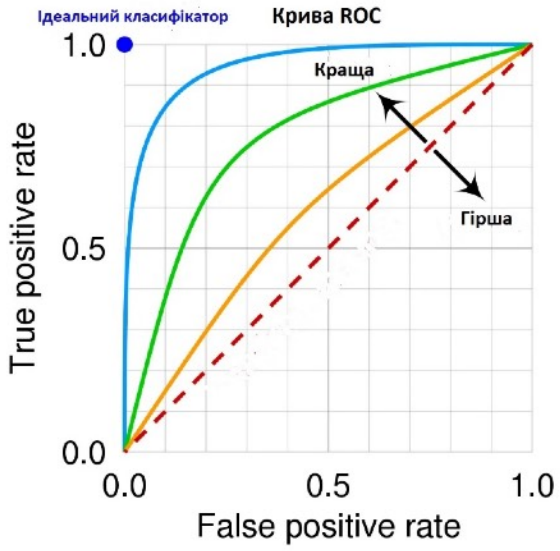



Рисунок 3.1 – Графік робочої характеристики приймача

5) Показники обчислювальних ресурсів (П5) оцінюють обчислювальні ресурси певних компонентів системи. Ці компоненти можуть бути частиною IDS або існувати в одному середовищі з нею.

Відсоток використання ресурсів (англ. Percentage of Resource Utilization) (П5.1), використаних за певний період часу. Ресурси, які можна виміряти – це використання мережі, центрального процесора та пам'яті (оперативної та постійної пам'яті).

Пропускна здатність мережі (англ. Network Bandwidth) (П5.2) – це кількість даних (в одиницях або відсотках), переданих через мережу.

6) Показники спроможності IDS (П6) вимірюють специфічні можливості IDS, які не пов'язані з точністю виявлення.

Пропускна здатність IDS (англ. IDS Throughput) (П6.1) – це кількість подій, які може обробити IDS за певний проміжок часу. У NIDS – це мережний трафік.

Середній час до компрометації (англ. Mean Time-to-Compromise, MTTC) (П6.2): Показник безпеки, який оцінює час, необхідний зловмиснику для успішного впливу на інформаційно-телекомунікаційну систему.

Зазначені показники впливають на властивості IDS. Слід зазначити, що цей вплив має різний ступінь. В таблиці 3.2 зазначені наступні рівні впливу: «-» – відсутній вплив, 1 – середній вплив, 2 – високий вплив.

Показники виявлення вторгнень дають пряму оцінку високої точності виявлення (В2) IDS. Їх можна отримати з тесту на виявлення вторгнень. Крім цього вони також впливають на властивість повноти (В3). Це підтверджується, коли для тестування використовується безліч різних вторгнень.

Часові показники, отримані під час тесту на виявлення вторгнень, дають оцінку своєчасності (В4). Час виявлення (DT), час реагування (RsT) і час обробки події дають чітку оцінку можливостей IDS щодо часу реагування на події. Час навчання (TT) дає уявлення про зручність використання часових можливостей IDS щодо готовності до роботи.

Показники виявлення вторгнень і часу можуть використовуватися як міра якості для високої продуктивності обробки (В7), відмовостійкості (В8), надійності (В9) і стійкості (В10), оскільки вони дають уявлення про продуктивність IDS. Однак, самі по собі ці показники недостатні для оцінки цих властивостей.

Показники ефективності архітектури насамперед впливають на властивість керованості IDS та частково на відмовостійкість (В8) та надійність (В9).

Більш точну оцінку високої продуктивності обробки (B7) може бути отримана, якщо врахувати час навчання (TT), час виявлення (DT), час реагування (RsT), час обробки на подію та пропускну здатність IDS.

Відмовостійка, надійна та стійка IDS підтримує високу точність та можливості роботи в режимі реального часу за наявності збоїв або інших непередбачуваних обставин.

Таблиця 3.2 – Вплив показників IDS на властивості

| Властивості / Показники | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| П1.1 | 2 | - | 1 | 1 | - | - | 1 | - | - | - | - |
| П1.2 | 2 | - | - | - | - | - | - | - | - | - | - |
| П1.3 | 2 | - | 1 | 1 | - | - | - | - | - | - | - |
| П1.4 | 2 | - | - | - | - | - | - | - | 1 | 1 | - |
| П1.5 | 2 | 2 | 1 | - | - | - | - | - | - | - | - |
| П1.6 | 2 | - | - | - | - | - | 1 | - | - | - | - |
| П1.7 | 1 | - | - | 1 | - | - | - | - | 1 | - | - |
| П2.1 | 1 | - | - | 1 | - | - | - | - | - | - | - |
| П2.2 | 1 | - | - | 1 | - | - | - | - | - | - | - |
| П2.3 | 2 | - | - | 1 | - | - | - | - | - | - | - |
| П3.1 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.2 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.3 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.4 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.5 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.6 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.7 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.8 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.9 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П3.10 | - | 2 | 1 | - | - | - | 1 | 1 | 1 | 1 | - |
| П4.1 | - | - | - | 1 | - | - | 2 | - | - | - | - |
| П4.2 | - | - | - | - | - | - | - | 1 | 1 | 1 | - |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| П4.3 | - | - | - | 2 | - | 1 | 2 | 1 | 1 | 1 | - |

Продовження таблиці 3.2

| Властивості / Показники | В1 | В2 | В3 | В4 | В5 | В6 | В7 | В8 | В9 | В10 | В11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| П4.4 | - | - | - | 2 | - | 1 | 2 | 1 | 1 | 1 | - |
| П4.5 | - | - | - | - | - | 2 | - | 1 | 1 | 1 | - |
| П4.6 | - | - | - | 2 | - | 1 | 2 | 1 | 1 | 1 | - |
| П5.1 | - | - | - | - | 2 | 2 | - | - | - | - | - |
| П5.2 | - | - | - | - | 2 | 2 | - | - | - | - | - |
| П6.1 | - | - | - | 1 | - | 1 | 2 | 1 | 1 |  | - |
| П6.2 | - | - | - | - | - | - | - | 2 | 2 | 1 | 2 |

Вона також має поведінку, максимально наближену до оптимальної. Крім цього, вона має високий MTTC, що означає, що зловмиснику потрібно більше часу для того, щоб скомпрометувати систему. Оцінка MTTC дозволяє визначити механізми для посилення або захисту IDS. Це забезпечує покращення властивості достовірності даних (В11).

## 3.3 Аналіз впливу змагальних атак та методів протидії на показники IDS

Слід зазначити, що змагальні атаки та методи протидії мають різний рівень впливу на показники IDS. На основі аналізу показників, які були розглянути, було зроблено висновки щодо рівнів впливу на них методів протидії змагальним атакам IDS.

Крім рівня впливу треба розрізняти характер впливу:

– постійний, який практично не залежить від часу або самого зовнішнього впливу на IDS;

– змінний, який залежить від часу або самого зовнішнього впливу на IDS.

Результат аналізу представлений в таблиці 3.3. В таблиці вказано наступні рівні впливу: 1 – практично відсутній, 2 – мінімальний, 3 – середній, 4 – значний.

Таблиця 3.3 – Рівні впливу методів протидії на показники IDS

| Характер впливу<br>Показники | Постійний | Змінний |
|---|---|---|
| 1 | 2 | 3 |
| 1. Ефективність архітектури | | |
| Розподілене управління | 1 | |
| Складність конфігурації | 2 | |
| Простота управління політиками та ліцензіями | 2 | |
| Доступність оновлень | 1 | |
| Регульована чутливість | 2 | |
| Масштабоване балансування навантаження | 1 | |
| Відстеження стану | 2 | |
| 2. Показники інтерактивності | | |
| Взаємодія з брандмауером | 1 | |
| Взаємодія з маршрутизатором | 1 | |
| Зручність для користувача | 2 | |
| 3. Показники виявлення | | |
| Частота істинних позитивних результатів | | 4 |
| Істинно негативний показник | | 4 |
| Коефіцієнт хибно позитивних спрацьовувань | | 4 |
| Коефіцієнт хибних спрацьовувань | | 4 |
| Прецизійність | | 3 |
| Точність | | 3 |
| F-міра | | 3 |
| G-mean | | 3 |
| Коефіцієнт кореляції Метьюса | | 3 |
| Графік робочої характеристики приймача | | 3 |
| 4. Часові показники | | |
| Час навчання | 3 | |
| Тривалість вторгнення | 1 | |

| | | |
|---|---|---|
| Час виявлення | 3 | |
| Час реакції | 3 | |
| Час затримки | 3 | |
| Час обробки події | 4 | |
| 5. Показники обчислювальних ресурсів | | |
| Відсоток використання ресурсів | 3 | |
| Пропускна здатність мережі | 3 | |
| 6. Показники спроможності IDS | | |
| Пропускна здатність IDS | 3 | |
| Середній час до компрометації | 3 | |

Як видно з таблицею 3.3, впровадження методів протидії змагальним атакам на IDS мають значний вплив на показники виявлення. Цей вплив змінюється з часом. Як було зазначено, ці показники недостатні для оцінки стійкості IDS до змагальних атак. Для оцінки стійкості та здатності систем до відновлення систем в [76] запропоновано використовувати поняття резильєнтність – здатність передбачати, протистояти, відновлюватись та пристосовуватися до несприятливих умов, зовнішніх впливів, атак чи порушення нормального функціонування системи.

Існує різноманіття підходів до визначення показників резільєнтності, які відносяться до різнорідних систем. Однак більша частина досліджень спрямована на аналіз кривих резільєнтності, які побудовані у відношенні часу та показників продуктивності системи. Ці криві описують реакцію резільєнтної системи на деструктивний збурюючий вплив. Для IDS в якості продуктивності можна розглянути інтегральну оцінку показників виявлення У роботах [77, 78] було запропоновано наступні основні показники резільєнтності системи.

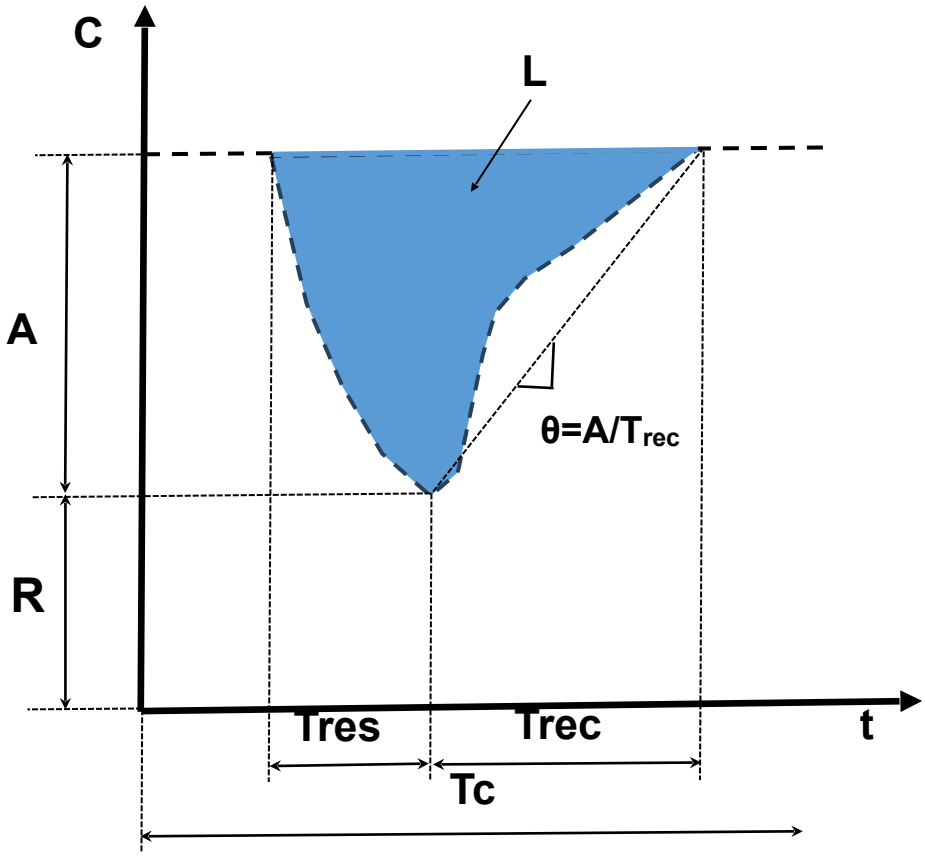
1) Час відгуку (Tres) відображає швидкість реагування IDS на деструктивний вплив. IDS з малим часом відгуку ефективніше зменшують вплив, що призводить до менших втрат продуктивності.

2) Час відновлення (Trec) визначає період, необхідний для відновлення функціональності IDS до очікуваного рівня, при якому система може працювати так само або навіть краще, ніж перед впливом.

3) Просідання продуктивності (A) визначає максимальне зменшення продуктивності системи внаслідок збурюючого впливу, тоді як показник втрати продуктивності (L) описує загальні втрати продуктивності під час

етапів реагування та відновлення. Втрата продуктивності представлена темною областю на рис. 3.2.

Рисунок 3.2 – Крива резільєнтності та її основні показники




4) Робастність (R) визначає здатність системи витримувати певний рівень стресу і утримувати свою функціональність, не суттєво погіршуючи або втрачаючи продуктивність. Ця характеристика дозволяє системі поглинати та протистояти деструктивним впливам. Система, яка володіє високим рівнем робастності, зможе зберегти основну частину своїх функціональних характеристик під час впливу деструктивних факторів на достатньому рівні. Робастність можна визначити як залишкову функціональність після екстремального деструктивного впливу. Обчислюється за виразом (3.10).

$$R = 1 - \tilde{A}\left(m_A, \sigma_A\right), \tag{3.10}$$

де $\tilde{A}$ – випадкова змінна, яка є функцією від $m_A$ та $\sigma_A$;

$m_A$ – середнє значення A;

$\sigma_A$ – середньоквадратичне відхилення A.

Швидкість відновлення (θ) представляє собою здатність IDS ефективно та оперативно відновлювати свою функціональність, обмежуючи втрати і запобігаючи можливим майбутнім збоям. У математичному визначенні швидкість відновлення визначається як нахил кривої продуктивності протягом періоду відновлення (див. рис. 3.2). Обчислюється за виразом (3.11).

$$\theta = \frac{d\,C(t)}{dt}, \tag{3.11}$$

де d/dt – оператор диференціювання;

C(t) – функція залежності продуктивності від часу.

Альтернативою може бути усереднена оцінка швидкості відновлення. Обчислюється за виразом (3.12).

$$\theta = \frac{A}{Trec}, \tag{3.12}$$

де A – просідання продуктивності;

Trec – час відновлення.

Для одночасного врахування змінних часу і продуктивності під час оцінювання резільєнтності IDS в [79] пропонуються різноманітні варіанти інтегральних показників. Ці показники відображають різницю або відношення між номінальною продуктивністю та втратою продуктивності в часі внаслідок збурюючих впливів. Такий інтегральний показник можна представити наступним чином:

$$R = \frac{\frac{1}{E} \bullet \sum_E \int_{t=0}^{Tc} C(t)dt}{\int_{t=0}^{Tc} C^{норм}(t)dt}, \tag{3.13}$$

де C(t) – залежність поточного значення продуктивності або функціональності системи від часу;

$C^{норм}(t)$ – значення продуктивності при нормальному режимі роботи IDS;

$T_c$ – тривалість періоду контролю, який визначається на основі результатів попереднього оцінювання середнього інтервалу між випадками деструктивних впливів;

E – кількість деструктивних впливів за період контролю.

Таким чином, до основних показників резільєнтності IDS можна віднести:

– робастність;

– швидкість відновлення;

– інтегральна міра резільєнтності.

Для використання цих показників необхідно визначити єдиний показник ефективності виявлення для IDS. Згідно з літературними джерелами, більшість робіт оцінювали продуктивність ISD за допомогою DR, FPR та площі під ROC (AUC). Однак, підкреслюється, що DR, FPR та AUC не здатні розрізнити продуктивність IDS в деяких особливих випадках. Для подолання цього обмеження рекомендується єдиний об'єктивний показник, який представлений в [80]: здатність виявлення вторгнень (англ. Intrusion Detection Capability, $C_{ID}$).

Слід зазначити, що $C_{ID}$ має наступні властивості:

– він природно враховує всі важливі аспекти здатності виявлення, тобто FPR, FNR, позитивне прогностичне значення (PPV), негативне прогностичне значення (NPV) і базову швидкість (ймовірність вторгнень);

– він об'єктивно формує важливу міру здатності виявлення вторгнень;

– він дуже чутливий до параметрів роботи IDS, таких як базова швидкість, FPR і FNR.

$$C_{ID} = -B \bullet (1-\beta) \bullet \log(PPV) - B \bullet (1-\beta) \bullet \log(1-NPV) - \\ -(1-B) \bullet (1-\alpha) \bullet \log(NPV) - (1-B) \bullet \alpha \bullet \log(1-PPV), \qquad (3.14)$$

де B – базова ставка: ймовірність того, що є вторгнення в спостережуваних даних аудиту;

α – частота хибно позитивних спрацювань: ймовірність того, що є тривога, за відсутності вторгнення;

(1 – β) – частота істинно позитивний спрацювань або частота виявлення: ймовірність того, що є тривога, коли є вторгнення;

(1 – α) – істинно негативний показник, ймовірність відсутності тривоги, коли немає вторгнення;

PPV – позитивне прогностичне значення: ймовірність того, що вторгнення присутнє, коли IDS видає тривогу;

NPV – негативне прогностичне значення: ймовірність того, що вторгнення відсутнє, коли IDS не видає тривогу.

На основі аналізу впливу методів протидії змагальним атакам на показники IDS та з врахуванням характеру впливу можна виділити декілька груп показників для порівняльного аналізу методів протидії:

1) Часові показники: збільшення часу навчання, збільшення часу виявлення, збільшення часу реакції, збільшення часу затримки, збільшення часу обробки події.

2) Показники обчислювальних ресурсів: збільшення відсотку використання ресурсів, зменшення пропускної здатності мережі.

3) Показники спроможності IDS: зменшення пропускної здатності IDS, зміна середнього часу до компрометації.

4) Показники резільєнтності IDS: робастність, швидкість відновлення, інтегральна міра резільєнтності.

## 3.4 Модель кількісного порівняльного аналізу методів протидії змагальним атакам

Для реалізації кількісного порівняльного аналізу методів протидії змагальним атакам на IDS в роботі розглянути наступні методи:

– на основі теорії багатокритеріальної корисності (англ. Multi-attribute Utility Theory, MCUT);

– обробки аналітичних ієрархій (англ. Analytical Hierarchy Process, AHP);

– визначення вагових коефіцієнтів на основі функції втрати ефективності систем.

Основні характеристики зазначених методів наведені в таблиці 3.4.

На основі аналізу основних характеристик вказаних методів та враховуючи позитивний досвід використання для вирішення аналогічної задачі з порівняння алгоритмів геш-функцій [81], для реалізації кількісного порівняльного аналізу методів протидії змагальним атакам на IDS було обрано метод визначення вагових коефіцієнтів на основі функції втрати ефективності систем.

Перелік показників, їх умовні позначення та можливі одиниці виміру наведено в таблиці 3.5. За результатами огляду джерел з результатами випробування методів протидії змагальним атакам, було встановлено, що ці результати мають різний перелік параметрів. За параметрами з цих переліків неможливо виконати підготовку та розрахунок параметрів, які було визначено для порівняльного аналізу методів протидії. Тому в подальшому в цій роботі для

перевірки моделі кількісного порівняльного аналізу методів протидії змагальним атакам на IDS буде використано тестовий набір параметрів для трьох різних методів ($M^{(1)}$, $M^{(2)}$, $M^{(3)}$) (таблиця 3.6.).

Вимоги щодо необхідного переліку параметрів треба враховувати при поведені в подальшому випробувань систем протидії змагальним атакам.

Опис метода визначення вагових коефіцієнтів на основі функції втрати ефективності систем виконаний на основі [81].

Згідно даних, які використовуються для тестування, представлено три метода протидії змагальним атакам $M^{(1)}$, $M^{(2)}$, $M^{(3,}$, які необхідно порівняти між собою.

Кожний з цих методів охарактеризований певним набором параметрів:

$$A = \left\{\alpha_{1.1},\ \alpha_{2.1},\ \alpha_{3.1},\ \alpha_{4.1}, \alpha_{1.2}, \alpha_{2.2}, \alpha_{1.3}, \alpha_{2.3}, \alpha_{1.4}, \alpha_{2.4}, \alpha_{3.4}\right\} . \qquad (3.15)$$

Тобто, для кожного метода можна записати наступний вираз:

$$M^{(k)} \rightarrow A^{(k)} = \left\{ \alpha_{1.1}^{(k)}, \alpha_{2.1}^{(k)}, \alpha_{3.1}^{(k)}, \alpha_{4.1}^{(k)}, \alpha_{1.2}^{(k)}, \alpha_{2.2}^{(k)}, \alpha_{1.3}^{(k)}, \alpha_{2.3}^{(k)}, \alpha_{1.4}^{(k)}, \alpha_{2.4}^{(k)}, \alpha_{3.4}^{(k)} \right\}, \qquad (3.16)$$

де k=1,2,3 – номер метода, який поданий для порівняння.

Таблиця 3.4 – Характеристика методів визначення ваг показників

| № | Показник | Метод на основі теорії багатокритеріальної | Метод обробки аналітичних ієрархій | Метод визначення вагових коефіцієнтів на основі функції втрати |
|---|---|---|---|---|
| 1 | Область застосування | Застосовується до багатокритеріальних завдань, де важливо визначити | Використовується для прийняття рішень в умовах ієрархічних | Застосовується до визначення ваг критеріїв у випадках втрати ефективності |
| 2 | Основна ідея | Використовує експертні оцінки або інші методи для визначення | Використовує парні порівняння для визначення вагових коефіцієнтів | Оцінює важливість параметрів на основі їх впливу на ефективність системи при збуренні |
| 3 | Ступінь залучення | Залучає експертів для надання оцінок та визначення | Залучає експертів для проведення парних порівнянь та | Може використовувати експертні оцінки для визначення ваг |
| 4 | Точність | Залежить від об'єктивності та експертної точності. | Залежить від об'єктивності та експертної точності | Точність визначається якістю функцій втрати ефективності. |
| 5 | Особливості | Враховує багато параметрів при прийнятті рішень. | Вимагає чіткої специфікації ієрархії та проведення парних порівнянь, | Забезпечує можливість кількісного визначення ваг параметрів. |

Таблиця 3.5 – Перелік показників методів протидії змагальним атакам на IDS

| № | Назва показника (параметра) | Умовне позначення | Одиниці виміру | Прим. |
|---|---|---|---|---|
| 1 | Часові показники | | | |
| 1.1 | збільшення часу навчання | $\alpha_{1.1}$ | с | |
| 1.2 | збільшення часу виявлення | $\alpha_{2.1}$ | мкс | |
| 1.3 | збільшення часу реакції | $\alpha_{3.1}$ | мкс | |
| 1.4 | збільшення часу затримки | $\alpha_{4.1}$ | мкс | |
| 1.5 | збільшення часу обробки події | $\alpha_{5.1}$ | мкс | |
| 2 | Показники обчислювальних ресурсів | | | |
| 2.1 | збільшення відсотку використання ресурсів | $\alpha_{1.2}$ | % | |
| 2.2 | зменшення пропускної здатності мережі | $\alpha_{2.2}$ | Мбіт/с; % | |
| 3 | Показники спроможності IDS | | | |
| 3.1 | зменшення пропускної здатності IDS | $\alpha_{1.3}$ | кількість подій/с | |
| 3.2 | зміна середнього часу до компрометації | $\alpha_{2.3}$ | с | |
| 4 | Показники резільєнтності IDS | | | |
| 4.1 | робастність | $\alpha_{1.4}$ | - | |
| 4.2 | швидкість відновлення | $\alpha_{2.4}$ | $с^{-1}$ | |
| 4.3 | інтегральна міра резільєнтності | $\alpha_{3.4}$ | - | |

Порівняння методів виконується шляхом обчислення та подальшого порівняння окремих та загальних показників ефективності. Часткові показники характеризують ефективність системи з точки зору деякої функціональної задачі.

Частковий показник обчислюється за виразом:

$$\gamma_j = \sum_{i=1}^{l_j} \rho_{ij} \bullet \eta_{ij}^{(k)}, \qquad (3.17)$$

де $\rho_{ij}$ – ваговий коефіцієнт параметра в групі параметрів;

j – номер групи параметрів (функціональна задача);

$l_j$ – кількість параметрів у групі j;

$\eta_{ij}^{(k)}$ – формування значення параметра k-го метода;

k=1,2,3 – номер метода, який поданий для порівняння.

Таблиця 3.6 – Тестовий набір параметрів

| № групи | Параметр | Метод | | |
|---|---|---|---|---|
| | | $M^{(1)}$ | $M^{(2)}$ | $M^{(3)}$ |
| 1 | $\alpha_{1.1}$ | 1200 | 700 | 2000 |
| | $\alpha_{2.1}$ | 3 | 5 | 3 |
| | $\alpha_{3.1}$ | 6 | 11 | 5 |
| | $\alpha_{4.1}$ | 12 | 18 | 8 |
| | $\alpha_{5.1}$ | 13 | 22 | 25 |
| 2 | $\alpha_{1.2}$ | 5 | 8 | 3 |
| | $\alpha_{2.2}$ | 25 | 12 | 30 |
| 3 | $\alpha_{1.3}$ | 1020 | 530 | 880 |
| | $\alpha_{2.3}$ | 57 | 33 | 26 |
| 4 | $\alpha_{1.4}$ | 0,8 | 0,5 | 0,7 |
| | $\alpha_{2.4}$ | 350 | 200 | 120 |
| | $\alpha_{3.4}$ | 0,94 | 0,95 | 0,91 |

Узагальнена оцінка метода визначається за виразом:

$$\Gamma^{(k)} = \sum_{j=1}^{J} \beta_j \bullet \gamma_j^{(k)}, \qquad (3.18)$$

де $\beta_j$ – ваговий коефіцієнт часткового показника (функціональної задачі);

$\gamma_j^{(k)}$ – значення часткового показника для k-го метода;

J=4 – кількість груп параметрів;

k=1,2,3 – номер метода, який поданий для порівняння.

Розглянутий метод дозволяє визначити:

– нормовані значення параметрів системи $\eta_{ij}$;

– вагові коефіцієнти параметрів $\rho_{ij}$ ;

– вагові коефіцієнти груп показників (функціональних задач) $\beta_j$ без залучення експертів. На основі чого обчислюється узагальнений системний показник.

Для цього необхідно виконати наступні обчислення.

1) Визначити в кожній групі параметрів такі параметри:

– що підвищують, підвищення його значення призводить до підвищення ефективності системи у цілому;

– що понижують, підвищення його значення призводить до погіршення ефективності системи у цілому.

Для кожного параметра, що підвищує, обчислюється $\alpha_{ij\ max}$, а для кожного параметра, що понижує, обчислюється $\alpha_{ij\ min}$ за формулами:

$$\alpha_{ij\ max} = \max_{k=1,2,3} \alpha_{ij}^{(k)}, \tag{3.19}$$

$$\alpha_{ij\ min} = \min_{k=1,2,3} \alpha_{ij}^{(k)}, \tag{3.20}$$

де $\alpha_{ij}^{(k)}$ – значення і -го параметру з j -ї функціональної групи для системи k.

Результат представлений в таблиці 3.7.

2) Частковий показник γj величина безрозмірна. Саме по цій величині проводиться нормування параметрів. Нормування параметрів виконується у відповідності з виразом (3.21). Результат представлений в таблиці 3.6.

3) Визначається середнє значення кожного з нормованих параметрів згідно з формулою (3.22). Результат представлений в таблиці 3.7.

4) Розраховується середнє значення розбіжності кожного нормованого параметра, яке характеризує відхилення параметрів систем від середнього значення за формулою (3.23). Результат представлений в таблиці 3.7.

$$\eta_{ij}^{(k)} = \begin{cases} \dfrac{\alpha_{ij}^{(k)}}{\alpha_{ij\ max}} \text{ для показників, що підвищують} \\ \dfrac{\alpha_{ij\ min}}{\alpha_{ij}^{(k)}} \text{ для показників, що понижують} \end{cases}. \tag{3.21}$$

$$\vec{\eta}_{ij} = \frac{1}{3} \bullet \sum_{k=1}^{3} \eta_{ij}^{(k)} \ . \qquad (3.22)$$

$$\Delta\vec{\eta}_{ij} = \frac{1}{3} \bullet \sum_{k=1}^{3} \left| \eta_{ij}^{(k)} - \vec{\eta}_{ij} \right| \ . \qquad (3.23)$$

5) Розраховується нормоване значення розбіжності за формулою (3.24). Результат представлений в таблиці 3.7.

$$d_{ij} = \frac{\Delta\vec{\eta}_{ij}}{\eta_{ij}} \ . \qquad (3.24)$$

6) Розраховується нормоване значення вагових коефіцієнтів по кожній групі параметрів за формулою (3.25). Результат представлений в таблиці 3.7.

$$\rho_{ij} = \frac{d_{ij}}{\sum_{i=1}^{l_j} d_{ij}} , \qquad (3.25)$$

де $l_j$ – кількість параметрів у групі j.

Значення цього коефіцієнта вказує на залежність, на скільки є різними відповідні параметри у різних методах. Чим більше ця різниця, тим більше влив цих параметрів.

7) Розраховується значення часткових показників ефективності по кожній групі параметрів за формулою (3.26). Результат представлений в таблиці 3.8.

$$\gamma_j^{(k)} = \sum_{i=1}^{l_j} \rho_{ij} \bullet \eta_{ij}^{(k)} , \qquad (3.26)$$

де $l_j$ – кількість параметрів у групі j.

Таблиця 3.7 – Проміжні дані з обчислення коефіцієнтів значущості параметрів у межах функціональних груп

| № групи | Пара-метр | $\alpha_{ij\ max}$ | $\alpha_{ij\ min}$ | $\eta_{ij}^{(1)}$ | $\eta_{ij}^{(2)}$ | $\eta_{ij}^{(3)}$ | $\vec{\eta}_{ij}$ | $\Delta\vec{\eta}_{ij}$ | $d_{ij}$ | $\rho_{ij}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $\alpha_{1.1}$ | - | 700 | 0,58 | 1,00 | 0,35 | 0,64 | 0,24 | 0,37 | 0,26 |
| | $\alpha_{2.1}$ | - | 5 | 1,00 | 0,60 | 1,00 | 0,87 | 0,18 | 0,21 | 0,15 |
| | $\alpha_{3.1}$ | - | 11 | 0,83 | 0,45 | 1,00 | 0,76 | 0,21 | 0,27 | 0,19 |
| | $\alpha_{4.1}$ | - | 18 | 0,67 | 0,44 | 1,00 | 0,70 | 0,20 | 0,28 | 0,20 |
| | $\alpha_{5.1}$ | - | 22 | 1,00 | 0,59 | 0,52 | 0,70 | 0,20 | 0,28 | 0,20 |
| 2 | $\alpha_{1.2}$ | - | 8 | 0,60 | 0,38 | 1,00 | 0,66 | 0,23 | 0,35 | 0,47 |
| | $\alpha_{2.2}$ | - | 12 | 0,48 | 1,00 | 0,40 | 0,63 | 0,25 | 0,40 | 0,53 |
| 3 | $\alpha_{1.3}$ | - | 530 | 0,52 | 1,00 | 0,60 | 0,71 | 0,20 | 0,28 | 0,47 |
| | $\alpha_{2.3}$ | 57 | - | 1,00 | 0,58 | 0,46 | 0,68 | 0,21 | 0,32 | 0,53 |
| 4 | $\alpha_{1.4}$ | 0,8 | - | 1,00 | 0,63 | 0,88 | 0,83 | 0,14 | 0,17 | 0,28 |
| | $\alpha_{2.4}$ | 350 | - | 1,00 | 0,57 | 0,34 | 0,64 | 0,24 | 0,38 | 0,64 |
| | $\alpha_{3.4}$ | 0,94 | - | 0,99 | 1,00 | 0,96 | 0,98 | 0,02 | 0,02 | 0,03 |

Таблиця 3.8 – Часткові показники порівнюваних систем

| № групи | Метод | | |
|---|---|---|---|
| | $M^{(1)}$ | $M^{(2)}$ | $M^{(3)}$ |
| 1 | 0,79 | 0,64 | 0,73 |
| 2 | 0,54 | 0,71 | 0,68 |
| 3 | 0,78 | 0,78 | 0,52 |
| 4 | 0,95 | 0,57 | 0,49 |

Для розрахунку загальносистемного показника необхідно розрахувати вагові коефіцієнти $\beta_j$ кожної групи показників. Слід зазначити, що для кількісної оцінки умов порівняння методик можна використати функцію втрат ефективності k-тої методики, яка характеризує ступінь наближення ефективності системи даних $\beta$ до максимально можливої при любих значеннях $\beta$ (3.27).

$$\Theta^{(k)} = 1 - \frac{\Gamma^{(k)}}{\max_{\beta} \Gamma^{(k)}} \quad . \tag{3.27}$$

Шляхом обчислення розкиду значень функції втрат ефективності можна проводити дослідження припустимих областей значень вагових коефіцієнтів на підставі функції (3.27). Для цього для кожної фіксованої сукупності значень вагових коефіцієнтів необхідно знайти максимальні і мінімальні значення функції втрат ефективності і побудувати функцію ρ(B), яка характеризує величину максимального розкиду, виду:

$$\rho(B) = \max_k \Theta^{(k)} - \min_k \Theta^{(k)} \quad . \tag{3.28}$$

З виразу (3.28) можна визначити діапазон вагових коефіцієнтів, при якому розкид показників не перевищить деякої величини або прийме максимальне значення. Це можна зробити наступним чином:

Розраховується вагові коефіцієнти для першої групи параметрів. Для цього змінюється $\beta_1$ в діапазоні 0,1 – 0,9 з кроком 0,1. Для інших груп β розраховується із врахуванням умови нормування:

$$\sum_{i=1}^{4} \beta_i = 1 \ . \tag{3.29}$$

Для кожного набору β визначимо значення узагальненої оцінки метода $\Gamma^{(k)}$ за формулою (3.18).

За формулою (3.27) розраховується значення $\theta^{(k)}$.

За формулою (3.28) розраховується значення $\rho(\beta_j)$ та будується відповідний графік (рис. 3.3). Результати розрахунків наведені у додатку А.

Визначається значення $\beta_1$ , при якому функція $\rho(\beta_1)$ приймає мінімальне значення. Саме це значення $\beta_1$ приймається в якості вагового коефіцієнта. Аналогічні дії виконуються для $\beta_2$, $\beta_3$ та $\beta_4$.

Виконується нормалізація отриманих значень згідно формули:

$$\hat{\beta}_j = \frac{\beta_j}{\sum_{r=1}^{4} \beta_r} \quad . \tag{3.30}$$

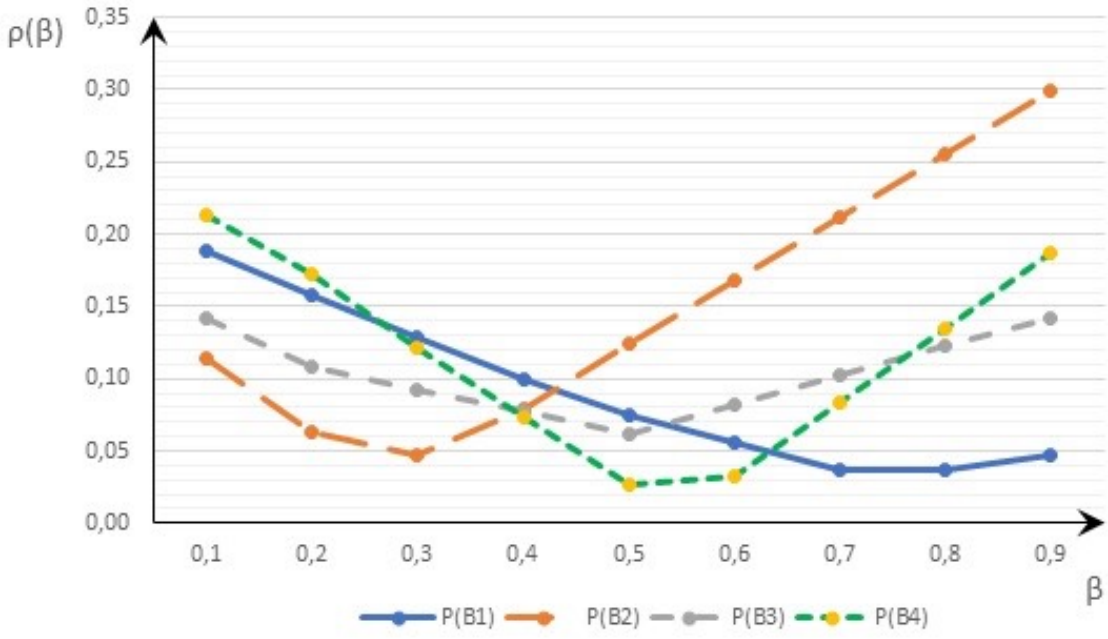



Рисунок 3.3 – Графіки функцій $\rho(\beta_j)$

Узагальнена оцінка методів із врахуванням $\hat{\beta}_j$ визначається за виразом:

$$\hat{\Gamma}^{(k)} = \sum_{j=1}^{1} \hat{\beta}_j \bullet \gamma_j^{(k)} \ . \tag{3.31}$$

За результатами обчислень: $\hat{\Gamma}^{(1)} = 0{,}789$, $\hat{\Gamma}^{(2)} = 0{,}667$, $\hat{\Gamma}^{(3)} = 0{,}619$.

Найкращою вважається методика з максимальним значенням узагальненої оцінки:

$$\Gamma_{\text{опт.}} = \max_{k} \hat{\Gamma}^{(k)} \ . \tag{3.32}$$

Таким чином, для заданих значень параметрів для трьох методів протидії змагальним атакам на системи виявлення вторгнень оптимальним є перший метод.

Слід зазначити, що для застосування запропонованої моделі кількісного порівняльного аналізу методів протидії змагальним атакам тестування рекомендовано виконувати в однакових умовах із застосуванням єдиного набору даних.

# 4 РЕКОМЕНДАЦІЇ ЩОДО ВДОСКОНАЛЕННЯ СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ

На основі аналізу сучасного рівня захисту IDS та запропонованого переліку показників ефективності методів протидії змагальним атакам на IDS можна сформулювати наступні рекомендації щодо вдосконалення та застосування методів протидії змагальним атакам і самих IDS:

1) Використання ансамблів моделей, а саме – інтеграція кількох моделей машинного навчання та поєднання різних стратегій ідентифікації.

Переваги:

– покращення загальної точності за рахунок комбінування прогнозів декількох моделей для виправлення помилок одна одної. Це знижує вразливість до помилок, які можуть бути спричинені змагальними атаками;

– різні моделі можуть вчитися на різних складових даних і робити помилки в різних областях вхідного простору. За рахунок цього IDS може ефективніше справлятися зі спотвореннями, які призначені для обману одиночних моделей;

– підвищення стійкості до шуму та зміненню даних, що є типовими для змагальних атак, завдяки «голосуванню» або усередненню прогнозів, які зменшують вплив викидів або спотворень;

– зниження ризику перенавчання за рахунок незалежності моделей;

– підвищення стійкості до змагальних атак: змагальні атаки часто розробляються так, щоб впливати на конкретні моделі ML. Ансамбль моделей може включати різноманітні обчислювальні підходи, що ускладнює зловмисникам створення ефективних атак, які впливають на всі моделі одночасно;

– використання моделей, що спеціалізуються на різних типах даних (наприклад: мережний трафік, логи серверів), що дозволяє враховувати більше контексту при виявленні атак.

Недоліки:

– складність реалізації порівняно з одиночними моделями, тому що необхідно забезпечити координацію та інтеграцію між різними моделями;

– підвищені вимоги до обчислювальних ресурсів та енергоспоживання;

– збільшення затримки у виявленні та реакції на атаки, що може бути критичним для систем, які вимагають швидкої реакції;

– збільшення трудомісткості налаштування та оновлення, оскільки потрібно оптимізувати параметри для кожної моделі в ансамблі і забезпечити їх правильну взаємодію;

– наявність небезпеки «голосування більшості», коли більшість моделей в ансамблі неправильно класифікують новий тип атаки, тоді ансамбль як ціле також прийме неправильне рішення.

Таким чином, використання ансамблів моделей доцільно використовувати в IDS, які насамперед вимагають покращення робастності та інтегральної міри резільєнтності.

2) Зменшення ознак або редукція розмірності даних дозволяє зменшити обсяг даних і при цьому зберегти важливу інформацію.

Переваги:

– зменшення кількості ознак може значно знизити час навчання та класифікації моделі, що дозволяє IDS швидше реагувати на потенційні загрози;

– усунення незначних ознак може допомогти зменшити вплив шуму на модель IDS, покращуючи її здатність виявляти справжні загрози;

– ускладнення реалізації змагальних атак за рахунок зменшення можливості маніпулювати всіма ознаками. Це може зробити більш важким створення зразків, які ефективно обдурюють систему;

– спрощення моделі тому, тому що моделі з меншою кількістю ознак легше інтерпретувати та перевіряти. Це може допомогти аналітикам краще розуміти, як модель виявляє атаки та реагує на змагальні атаки;

– зменшення вимог до обчислювальних ресурсів та енергоспоживання;

– зменшення кількості ознак може значно знизити час навчання та класифікації моделі, що дозволяє IDS швидше реагувати на потенційні загрози;

– системи з редукцією розмірності можуть бути більш гнучкими та адаптованими для впровадження в різні середовища, що дозволяє масштабувати IDS відповідно до змінюваних потреб безпеки.

Недоліки:

– втрата важливих ознак, які є критичними для виявлення специфічних типів атак. Це може призвести до зниження точності та ефективності IDS;

– змагальні зразки можуть бути ще більш ефективними, оскільки система втратила частину інформації, необхідної для їх виявлення;

– вибір ознак для редукції розмірності вимагає глибокого розуміння даних і потенційних атак, що може бути складним без глибоких знань у сфері кібербезпеки;

– під час тренування моделі з використанням редукованих ознак, існує ризик, що модель навчиться виявляти лише відомі шаблони атак, ігноруючи нові або модифіковані змагальні атаки.

Таким чином, використання зменшення ознак доцільно використовувати в IDS, які насамперед вимагають покращення наступних показників: відсоток використання ресурсів, часові показники, пропускна здатність мережі, наприклад для IoT.

Слід зазначити, що представлені методи по суті є діаметрально протилежні за характеристиками та разом з іншими були розглянуті в розділі 2.

В якості додаткового методу можна запропонувати застосування технік регуляризації, які використовуються в машинному навчанні для запобігання перенавчання моделей, коли модель занадто точно відтворює тренувальний набір даних і втрачає здатність узагальнювати свої прогнози на нові дані. У контексті протидії змагальним атакам IDS, регуляризація може також допомогти зробити моделі менш вразливими за рахунок покращення їх здатності протистояти спробам експлуатації надмірної складності моделі.

Можна застосувати наступні методи регуляризації:

– L1 і L2 регуляризація (також відомі як регуляризація Лассо та регуляризація Ріджа): ці методи включають до функції витрат додатковий член, який штрафує великі ваги в моделі. L1 регуляризація може привести до спарсності ваг, тобто деякі ваги стають рівними нулю, що робить модель простішою. L2 регуляризація штрафує ваги квадратично, зменшуючи ризик перенавчання без встановлення ваг рівними нулю;

– Dropout техніка регуляризації полягає в випадковому видаленні (відключенні) нейронів під час тренування, що змушує модель навчатися робити прогнози, використовуючи лише частину доступних даних. Це допомагає запобігти спиранню моделі на невелику групу особливостей і покращує її узагальнення.

Слід зазначити, що за результатами аналізу змагальних атак найбільш реальною є атака «чорного ящика». Цю атаку можна описати як цілеспрямований вплив змагальної атаки на IDS з метою отримання необхідної реакції. Таким чином, схема цієї атаки схожа на схему системи автоматичного керування, де вплив змагальної атаки можна представити як сигнали керування. Тоді задачу

протидії змагальним атакам можна сформулювати як погіршення керованості такої системи.

Наприклад можна збільшити час реакції системи на вплив змагальної атаки. Це можна зробити шляхом затримки впливу вхідних даних ISD на навчальні данні.

Безумовно, вплив такого підходу на ефективність протидії змагальним атакам на ISD на основі машинного навчання та на ефективність самої ISD потребує додаткових досліджень.

# ВИСНОВКИ

У кваліфікаційній роботі вирішено задачу щодо розробки моделі кількісного порівняльного аналізу методів протидії змагальним атакам на базі методу визначення вагових коефіцієнтів на основі функції втрати ефективності систем.

Для досягнення мети роботи, згідно із завданням, було проаналізовано класифікатори мережних атак, а також виконаний їх аналіз. Було визначено, згідно результатів досліджень Держспецзв'язку та Cisco, що одними із розповсюджених типом атаки є DDoS. Слід зазначити, що саме DoS/DDoS атаки були успішно застосовані при реалізації змагальних атак.

На основі аналізу сучасних систем виявлення вторгнень, можна зробити висновок, що IDS, які використовують методи на основі машинного навчання, мають ряд переваг, але низьку стійкість. Це та інші фактори обумовлюють їх вразливість перед змагальними атаками. В роботі виконано огляд методів машинного навчання для систем виявлення вторгнень.

За результатами класифікації та аналізу змагальних атак на системи виявлення вторгнень можна зробити висновок, що деякі види змагальних атак мають достатньо великий вплив на ефективність IDS. Слід зазначити, що більшість досліджень змагальних атак виконана для систем розпізнавання зображень. Тому для подальших досліджень треба враховувати наступне.

1) У розпізнаванні зображень основною ознакою, що використовується для збурення противника, є пікселі зображення. Однак у мережній безпеці існує велика варіативність типів ознак, які можуть бути використані, і, таким чином, обсяг збурень для ворожих атак значно збільшується.
2) Ворожі атаки в мережній безпеці відрізняються від комп'ютерного зору, оскільки розглядаються об'єкти даних, а не зображення. Як наслідок, збурені функції є більш різноманітними та неоднорідними.

Аналіз методів протидії атакам на системи виявлення вторгнень виявив схожу ситуацію. Серед представлених методів тільки частина була застосована саме для захисту IDS. Інші потребують підтвердження своєї ефективності. Тобто, на сьогодні є певні успіхи в розробці універсальних засобів захисту, але дослідження в області мережної безпеки потребують розвитку.

Окремо слід зазначити відсутність єдиних показників ефективності та єдиної методики кількісного порівняльного аналізу методів протидії змагальним атакам.

На основі аналізу впливу змагальних атак та методів протидії на показники та властивості IDS, в роботі розроблена система показників ефективності методів протидії змагальним атакам, які складають наступні групи:

– часові показники;
– показники обчислювальних ресурсів;
– показники спроможності IDS;
– показники резільєнтності IDS.

На базі показників, які були запропоновано, розроблена модель кількісного порівняльного аналізу методів протидії змагальним атакам на базі методу визначення вагових коефіцієнтів на основі функції втрати ефективності систем. Для перевірки запропонованої моделі були використані тестові значення показників. Розрахунки були виконання в додатку Excel.

В роботі запропоновані рекомендації щодо вдосконалення IDS.

Результати роботи доцільно використовувати при обробці результатів досліджень та випробувань методів протидії атакам на системи виявлення вторгнень.

Подальший розвиток запропонованої методики може полягати у впровадженні додаткових вагових коефіцієнтів, які будуть враховувати важливість параметрів у відповідності до особливостей цільової системи. Ці вагові коефіцієнти можуть бути отримані на основі експертної оцінки.

Окремі результати роботи доповідались на XVIII Міжнародній конференції «Проблеми використання інформаційних технологій у сфері освіти, науки та промисловості» [2, 3, 4].

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Звіт про роботу системи виявлення вразливостей і реагування на кіберінциденти та кібератаки (2023 (Q3)). URL: https://scpc.gov.ua/api/files/22c75b41-d1d8-4da6-bd46-fa5489af9c6e (дата звернення: 10.10.2023).

2. Кручінін О. В., Тимофєєв Д.С. Класифікатори мережевих атак. *Проблеми використання інформаційних технологій в освіті, науці та промисловості* : зб. наукових праць. № 8 XVIII Міжнар. конф. Дніпро : НТУ «ДП», 2023. С. 177 – 181. URL: https://pzks.nmu.org.ua/ua/science/conf2023_01.pdf (дата звернення: 13.01.2024).

3. Святошенко В.О., Кручінін О. В. Методи машинного навчання в системах виявлення вторгнень. *Проблеми використання інформаційних технологій в освіті, науці та промисловості* : зб. наукових праць. № 8 XVIII Міжнар. конф. Дніпро : НТУ «ДП», 2023. С. 182 – 185. URL: https://pzks.nmu.org.ua/ua/science/conf2023_01.pdf (дата звернення: 13.01.2024).

4. Кручінін О. В., Святошенко В.О. Характеристика атак на системи виявлення вторгнень на основі машинного навчання. *Проблеми використання інформаційних технологій в освіті, науці та промисловості* : зб. наукових праць. № 8 XVIII Міжнар. конф. Дніпро : НТУ «ДП», 2023. С. 204 – 207. URL: https://pzks.nmu.org.ua/ua/science/conf2023_01.pdf (дата звернення: 13.01.2024).

5. НД ТЗІ 1.1-003-99 Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу, наказ ДСТСЗІ СБУ від 28.04.99 № 22.

6. Кібервійни, Інтернет-розвідка. URL: https://www.pollawlife.com.ua/2015/05/blog-post_4.html (дата звернення: 10.10.2023).

7. Lindqvist U., Jonsson E. How to systematically classify computer security intrusions. *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097),* м. Oakland, CA, USA. URL: https://doi.org/10.1109/secpri.1997.601330 (дата звернення: 10.10.2023).

8. Bisbey, R., Hollingworth, D. Protection analysis: Final report. URL: https://csrc.nist.rip/publications/history/bisb78.pdf (дата звернення: 10.10.2023).

9. Грайворонський М. В., Новіков О. М. Безпека інформаційно-комунікаційних систем. Київ : Видавнича група BHV, 2009. 608 с.

10. John D. Howard, Thomas A Longstaff. A common language for computer security incidents. Office of Scientific and Technical Information (OSTI), 1998. URL: https://doi.org/10.2172/751004 (дата звернення: 10.10.2023).

11. Hansman S., Hunt R. A taxonomy of network and computer attacks. *Computers & Security*. 2005. Т. 24, № 1. С. 31–43. URL: https://doi.org/10.1016/j.cose.2004.06.011 (дата звернення: 11.10.2023).

12. Lough D. L. A Taxonomy of Computer Attacks with Applications to Wireless Networks : dissertation. 2001. URL: http://hdl.handle.net/10919/27242 (дата звернення: 11.10.2023).

13. Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. AVOIDIT: A cyber attack taxonomy. *In Proc. of 9th Annual Symposium On Information Assurance-ASIA*. Т. 14. URL: https://nsarchive.gwu.edu/sites/default/files/documents/4530310/Chris-Simmons-Charles-Ellis-Sajjan-Shiva.pdf . (дата звернення: 11.10.2023).

14. Бурячок В. Л. Технології забезпечення безпеки мережевої інфраструктури: підручник. Київ : КУБГ, 2019. 218 с.

15. Md Mehedi Hassan Onik, Nasr Al-Zaben, Hung Phan Hoo and Chul-Soo Kim. A Novel Approach for Network Attack Classification Based on Sequential Questions. *Annals of Emerging Technologies in Computing*. 2018. Т. 2, № 2. С. 1–14. URL: https://doi.org/10.33166/aetic.2018.02.001 (дата звернення: 11.10.2023).

16. What is a denial of service attack (DoS)?. URL: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos (дата звернення: 23.09.2023).

17. SQL Injection Attack: Quick View. URL: https://www.researchgate.net/publication/368511340_SQL_Injection_Attack_Quick_View (дата звернення: 27.09.2023).

18. Maulana M., Luthfi A., Wibowo D. K. Network Attacks Classification for Network Forensics Investigation: Literature Reviews. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. 2023. Т. 7, № 5. С. 1132–1139. URL: https://doi.org/10.29207/resti.v7i5.5153 (дата звернення: 15.10.2023).

19. Поліщук С.М. Ландшафт загроз спуфінгу: тенденції, вразливості та напрямки досліджень. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2023. Т. 34 (73), № 4. С. 99–103. URL: https://doi.org/10.32782/2663-5941/2023.4/16 (дата звернення: 23.09.2023).

20. What is Sniffing Attacks?. URL: https://intellipaat.com/blog/tutorial/ethical-hacking-cyber-security-tutorial/sniffing-attacks (дата звернення: 27.09.2023).

21. What Is an Intrusion Detection System? Latest Types and Tools. URL: https://www.dnsstuff.com/intrusion-detection-system (дата звернення: 10.10.2023).

22. Intrusion Detection Systems Explained: 12 Best IDS Software Tools Reviewed. URL: https://www.comparitech.com/net-admin/network-intrusion-detection-tools/ (дата звернення: 24.10.2023).

23. 8 Best HIDS Tools—Host-Based Intrusion Detection Systems. URL: https://www.dnsstuff.com/host-based-intrusion-detection-systems (дата звернення: 27.10.2023).

24. Top 10 BEST Intrusion Detection Systems (IDS) [2023 Rankings]. URL: https://www.softwaretestinghelp.com/intrusion-detection-systems/ (дата звернення: 01.11.2023).

25. Tereykovsky I., Korchenko A., Parashchuk T., Pedchenko Y. Open intrusion detection systems analysis. *Ukrainian Scientific Journal of Information Security.* 2018. Т. 24, № 3. URL: https://doi.org/10.18372/2225-5036.24.13431 (дата звернення: 01.11.2023).

26. Толюпа С., Лукова-Чуйко Н., Шестяк Я. Засоби виявлення кібернетичних атак на інформаційні системи. *Information and communication technologies, electronic engineering*. 2021. Т. 1, № 2. С. 19–31. URL: https://doi.org/10.23939/ictee2021.02.019 (дата звернення: 01.11.2023).

27.Liu H., Lang B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*. 2019. Т. 9, № 20. С. 4396. URL: https://doi.org/10.3390/app9204396 (дата звернення: 01.11.2023).

28. Michie D., Spiegelhalter D.J., Taylor C.C. Machine Learning, Neurall and Statistical Classification. New York : Ellis Horwood, 1994. 290 с. URL: https://www1.maths.leeds.ac.uk/~charles/statlog/whole.pdf (дата звернення: 01.11.2023).

29. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res*. 2010. № 11. С 3371–3408. URL: https://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf (дата звернення: 01.11.2023).

30. Deng, J.; Zhang, Z.; Marchi, E.; Schuller, B. Sparse Autoencoder-Based Feature Transfer Learning for Speech Emotion Recognition. *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*, м. Geneva, Switzerland, 2–5 верес. 2013 р. 2013. URL: https://doi.org/10.1109/acii.2013.90 (дата звернення: 05.11.2023).

31. Hinton G. E. A Practical Guide to Training Restricted Boltzmann Machines. *Lecture Notes in Computer Science*. Berlin, Heidelberg, 2012. С. 599 – 619. URL: https://doi.org/10.1007/978-3-642-35289-8_32 (дата звернення: 11.11.2023).

32. Pillai, M. M., Jan HP Eloff, and H. S. Venter. An approach to implement a network intrusion detection system using genetic algorithms. *Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. 2004. URL: https://dl.acm.org/doi/10.5555/1035053.1035080 (дата звернення: 07.11.2023).

33. Morshedur Hassan M. M. Current Studies On Intrusion Detection System, Genetic Algorithm And Fuzzy Logic. *International Journal of Distributed and Parallel systems*. 2013. Т. 4, № 2. С. 35–47. URL: https://doi.org/10.5121/ijdps.2013.4204 (дата звернення: 11.11.2023).

34. Hansman S., Hunt R. A taxonomy of network and computer attacks. *Computers & Security*. 2005. Т. 24, № 1. С. 31–43. URL: https://doi.org/10.1016/j.cose.2004.06.011 (дата звернення: 11.11.2023).

35. Ibitoye, Olakunle, et al. The Threat of Adversarial Attacks on Machine Learning in Network Security: A Survey. School of Information Technology, Carleton University, Ottawa, Canada. URL: https://arxiv.org/abs/1911.02621 (дата звернення: 15.11.2023).

36. Y. Fan, B. Wu, T. Li, Y. Zhang, M. Li, Z. Li, and Y. Yang. Sparse Adversarial Attack via Perturbation Factorization. / Y. *Computer Vision – ECCV 2020*. Cham, 2020. URL: https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123670035.pdf (дата звернення: 16.11.2023).

37. A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary. Robust deep reinforcement learning with adversarial attacks. Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018. С. 2040 – 2042 URL: https://arxiv.org/abs/1712.03632 (дата звернення: 16.11.2023).

38. N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. URL: https://arxiv.org/abs/1902.06705 (дата звернення: 16.11.2023).

39. E. Tabassi, K. J. Burns, M. Hadjimichael, A. D. Molina-Markham, and J. T. Sexton. A taxonomy and terminology of adversarial machine learning. URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8269-draft.pdf (дата звернення: 16.11.2023).

40. Lin Z., Shi Y., Xue Z. IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection. *Advances in Knowledge Discovery and Data Mining*. Cham, 2022. С. 79 – 91. URL: https://doi.org/10.1007/978-3-031-05981-0_7 (дата звернення: 16.11.2023).

41. I. Homoliak, M. Teknos, M. Ochoa, D. Breitenbacher, S. Hosseini, and P. Hanacek. Improving Network Intrusion Detection Classifiers by Non-payload-Based Exploit-Independent Obfuscations: An Adversarial Approach. *ICST Transactions on Security and Safety*. 2019. Т. 5, № 17. С. 1 – 15. URL: https://arxiv.org/abs/1805.02684 (дата звернення: 18.11.2023).

42. M. Usama, M. Asim, S. Latif, J. Qadir, et al. Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems. *2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC)*, м. Tangier, Morocco, 24 – 28 черв. 2019 р. 2019. С. 78-83. URL: https://doi.org/10.1109/iwcmc.2019.8766353 (дата звернення: 23.11.2023).

43. A. Al-Dujaili et al. Adversarial Deep Learning for Robust Detection of Binary Encoded Malware. *2018 IEEE Security and Privacy Workshops (SPW)*, м. San Francisco, CA, 24 трав. 2018 р. 2018. С. 76 – 82. URL: https://arxiv.org/abs/1801.02950 (дата звернення: 23.11.2023).

44. Aiken J., Scott-Hayward S. Investigating Adversarial Attacks against Network Intrusion Detection Systems in SDNs. *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, м. Dallas, TX, USA, 12 – 14 листоп. 2019 р. 2019. С. 1 – 7. URL: https://doi.org/10.1109/nfv-sdn47374.2019.9040101 (дата звернення: 23.11.2023).

45. G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti. Addressing Adversarial Attacks Against Security Systems Based on Machine Learning. *2019 11th International Conference on Cyber Conflict (CyCon)*, м. Tallinn, Estonia, 28 – 31 трав. 2019 р. 2019. С. 1 – 18. URL: https://doi.org/10.23919/cycon.2019.8756865 (дата звернення: 23.11.2023).

46. Ibitoye O., Shafiq O., Matrawy A. Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks. *GLOBECOM 2019 - 2019 IEEE Global Communications Conference*, м. Waikoloa, HI, USA, 9 – 13 груд. 2019 р. 2019. URL: https://arxiv.org/abs/1905.05137 (дата звернення: 23.11.2023).

47. M. Kloft and P. Laskov. Online anomaly detection under adversarial impact. *Journal of Machine Learning Research*: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 13-15 May 2010 р. 2010. Т. 9. С.

405 – 412. URL: https://proceedings.mlr.press/v9/kloft10a/kloft10a.pdf (дата звернення: 25.11.2023).

48. Chakraborty, Anirban, et al. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*. 2021. Т. 6, № 1. С. 25 – 45. URL: https://doi.org/10.1049/cit2.12028 (дата звернення: 25.11.2023).

49. D. Wu, B. Fang, J. Wang, Q. Liu, and X. Cui. Evading Machine Learning Botnet Detection Models via Deep Reinforcement Learning. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, м. Shanghai, China, 20 – 24 трав. 2019 р. 2019. С. 1 – 6. URL: https://doi.org/10.1109/icc.2019.8761337 (дата звернення: 25.11.2023).

50. Q. Cheng, S. Zhou, Y. Shen, D. Kong, and C. Wu. Packet-level adversarial network traffic crafting using sequence generative adversarial networks. 2021. URL: https://arxiv.org/abs/2103.04794 (дата звернення: 25.11.2023).

51. J. Chen, D. Wu, Y. Zhao, N. Sharma, M. Blumenstein, and S. Yu. Fooling intrusion detection systems using adversarially autoencoder. *Digital Communications and Networks*, 2020. Т. 7, № 3, С. 453 – 460. URL: https://www.sciencedirect.com/science/article/pii/S2352864820302868?via%3Dihub (дата звернення: 25.11.2023).

52. Y. Sharon, D. Berend, Y. Liu, A. Shabtai, and Y. Elovici . TANTRA: Timing-Based Adversarial Network Traffic Reshaping Attack. *IEEE Transactions on Information Forensics and Security*. 2022. С. 1. URL: https://doi.org/10.1109/tifs.2022.3201377 (дата звернення: 25.11.2023).

53. Apruzzese, Giovanni, et al. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digital Threats: Research and Practice*. 2021. URL: https://arxiv.org/abs/2106.09380 (дата звернення: 28.11.2023).

54. Barreno, Marco, et al. Can machine learning be secure?. *The 2006 ACM Symposium*, м. Taipei, Taiwan, 21 – 24 берез. 2006 р. New York, New York, USA, 2006. URL: https://people.eecs.berkeley.edu/~tygar/papers/Machine_Learning_Security/ asiaccs06.pdf (дата звернення: 03.12.2023).

55. Nayebi, Aran, and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. URL: https://arxiv.org/abs/1703.09202 (дата звернення: 03.12.2023).

56. Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. URL: https://arxiv.org/abs/1503.02531 (дата звернення: 03.12.2023).

57. Szegedy, Christian, et al. Intriguing properties of neural networks. URL: https://arxiv.org/abs/1312.6199 . (дата звернення: 03.12.2023).

58. Ross, Andrew, and Finale Doshi-Velez. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018. Т. 32, № 1. URL: https://arxiv.org/abs/1711.09404 (дата звернення: 03.12.2023).

59. Feinman, Reuben, et al. Detecting adversarial samples from artifacts. URL: https://arxiv.org/abs/1703.00410 (дата звернення: 05.12.2023).

60. Metzen, Jan Hendrik, et al. On detecting adversarial perturbations. URL: https://arxiv.org/abs/1702.04267 (дата звернення: 05.12.2023).

61. Ren, Min, Yun-Long Wang, and Zhao-Feng He. Towards Interpretable Defense Against Adversarial Attacks via Causal Inference. *Machine Intelligence Research*. 2022. Т. 19, № 3. С. 209 – 226. URL: https://doi.org/10.1007/s11633-022-1330-7 (дата звернення: 07.12.2023).

62. Grosse, Kathrin, et al. Adversarial perturbations against deep neural networks for malware classification. URL: https://arxiv.org/abs/1606.04435 (дата звернення: 07.12.2023).

63. Xie, Cihang, et al. Mitigating adversarial effects through randomization. URL: https://arxiv.org/abs/1711.01991 (дата звернення: 07.12.2023).

64. Zhang, Yuchen, and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. URL: https://arxiv.org/abs/1903.10586 (дата звернення: 07.12.2023).

65. Song, Yang, et al. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. URL: https://arxiv.org/abs/1710.10766 (дата звернення: 09.12.2023).

66. Martinez, Cyntia Vargas, Birgit Vogel-Heuser. A Taxonomy of Metrics and Tests to Evaluate and Validate Properties of Industrial Intrusion Detection Systems. URL: https://www.scitepress.org/Papers/2019/78339/78339.pdf (дата звернення: 10.12.2023).

67. Oryspayuli, O. D. What intrusion detection approaches work well if only TCP/IP packet header information is available?. URL: https://www.utwente.nl/en/eemcs/dacs/assignments/completed/master/reports/MSc-Thesis-Dossay%20Oryspayev.pdf (дата звернення: 12.12.2023).

68. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., Payne, B. D. Evaluating Computer Intrusion Detection Systems. *ACM Computing Surveys*. 2015. Т. 48, № 1. С. 1 – 41. URL: https://doi.org/10.1145/2808691 (дата звернення: 14.12.2023).

69. Zarpelao, B. B., Miani, R. S., Kawakani, C. T., de Al- ˜varenga, S. C. A survey of intrusion detection in Internet of Things. *Journal of Network and Computer*

*Applications*. 2017. Т. 84. С. 25 – 37. URL: https://doi.org/10.1016/j.jnca.2017.02.009 (дата звернення: 14.12.2023).

70. Lussier, B., Chatila, R., Ingrand, F., Killijian, M. O., Powell, D. On fault tolerance and robustness in autonomous systems. *Third IARP-IEEE/RAS-EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments*. 2004. С. 1 – 7. URL: https://homepages.laas.fr/mkilliji/docs/workshops/IARP04.pdf (дата звернення: 14.12.2023).

71. Zhu, Q., Basar, T. 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*. 2015. Т. 35, № 1. С. 46 – 65. URL: https://doi.org/10.1109/mcs.2014.2364710 (дата звернення: 14.12.2023).

72. Fink G. A., Chappell B. L., Turner T., Odonoghue K. F., N. S. W. Center. A metrics-based approach to intrusion detection system evaluation for distributed real-time systems. *Proceedings 16th International Parallel and Distributed Processing Symposium. IPDPS 2002*, м. Ft. Lauderdale, FL, 15–19 квіт. 2001 р. 2002. URL: https://doi.org/10.1109/ipdps.2002.1016475 (дата звернення: 14.12.2023).

73. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., and Payne, B. D. Evaluating Computer Intrusion Detection Systems. *ACM Computing Surveys*. 2015. Т. 48, № 1. С. 1 – 41. URL: https://doi.org/10.1145/2808691 (дата звернення: 14.12.2023).

74. Al-Jarrah, O. Y., Al-Hammdi, Y., Yoo, P. D., Muhaidat, S., Al-Qutayri, M. Semi-supervised multi-layered clustering model for intrusion detection. *Digital Communications and Networks*. 2018. Т. 4, № 4. С. 277 – 286. URL: https://doi.org/10.1016/j.dcan.2017.09.009 (дата звернення: 14.12.2023).

75. Buczak A. L., Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*. 2016. Т. 18, № 2. С. 1153 – 1176. URL: https://doi.org/10.1109/comst.2015.2494502 (дата звернення: 14.12.2023).

76. Лисенко С. М. Резильєнтність комп'ютерних систем в умовах кіберзагроз: таксономія та онтологія. *Radioelectronic and computer systems*. 2020. № 1. С. 17 – 28. URL: https://doi.org/10.32620/reks.2020.1.02 (дата звернення: 18.12.2023).

77. Yodo N., Wang P. Engineering Resilience Quantification and System Design Implications: A Literature Survey. *Journal of Mechanical Design*. 2016. Т. 138, № 11. URL: https://doi.org/10.1115/1.4034223 (дата звернення: 18.12.2023).

78. Cimellaro G. P., Reinhorn A. M., Bruneau M. Framework for analytical quantification of disaster resilience. *Engineering Structures*. 2010. Т. 32, № 11. С. 3639 – 3649. URL: https://doi.org/10.1016/j.engstruct.2010.08.008 (дата звернення: 18.12.2023).

79. Yodo N., Wang P. Engineering Resilience Quantification and System Design Implications: A Literature Survey. *Journal of Mechanical Design*. 2016. Т. 138, № 11. URL: https://doi.org/10.1115/1.4034223 (дата звернення: 18.12.2023).

80. Gu, G., Fogla, P., Dagon, D., Lee, W., Skori´c, B. Measuring intrusion detection capability. *The 2006 ACM Symposium*, м. Taipei, Taiwan, 21–24 берез. 2006 р. New York, New York, USA, 2006. С. 90 – 101. URL: https://doi.org/10.1145/1128817.1128834 (дата звернення: 18.12.2023).

81. Бойко А. О., Горбенко І. Д., Даценко С. І. Методика і результати порівняння алгоритмів геш-функцій, що приймають участь у 3-му раунді конкурсу NIST SHA-3. *Прикладная радиоэлектроника* : науч.-техн. журн., м. Харків, 2011. Т. 10, № 2. С. 171 – 175. URL: http://openarchive.nure.ua/handle/document/4188 (дата звернення: 18.12.2023).