Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи

магістра
(назва освітньо–кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній рівень ***магістр***
(назва освітнього рівня)

кваліфікація
(код і назва кваліфікації)

на тему: *Моделі класифікації подій кібербезпеки в розподілених ІС*

Виконавець: студент *2* **курсу, групи** *КБМ–21*

Сахацька Анна Вікторівна
(підпис) (прізвище ім'я по–батькові)

	Прізвище, ініціали	Оцінка	Підпис
Науковий керівник	*Бабенко Т.В.*		
Рецензент			
Нормоконтроль	*Фесенко А.О.*		

Київ
2021

Міністерство освіти і науки України

Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри
кібербезпеки та захисту інформації

_____ Лукова–Чуйко Н.В.

«_____» травня 2021 року

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

студенту *КБМ–21* *Сахацька Анна Вікторівна*
(група) (прізвище ім'я по–батькові)

Тема дипломного роботи *Моделі класифікації подій кібербезпеки в розподілених ІС*

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 2 від 08.10.2020

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень *Процес класифікації подій кібербезпеки в розподілених ІС*

Предмет досліджень *Моделі класифікації подій кібербезпеки в розподілених ІС*

Мета *Розробка моделі класифікації подій кібербезпеки в розподілених ІС*

Вихідні дані для проведення роботи *Класифікації подій кібербезпеки в розподілених ІС*

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна *створення оновленої моделі класифікації подій кібербезпеки в розподілених ІС, за допомогою поєднання існуючих класифікацій*

Практична цінність *полягає у створенні моделі класифікації подій кібербезпеки в розподілених ІС для попередження порушення властивостей інформації*

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок–кінець)
Розробка плану для досягнення мети роботи	01.09.2020 – 16.10.2020
Аналіз літературних джерел	19.10.2020 – 08.01.2021
Розробка моделі класифікації подій кібербезпеки в розподілених ІС	11.01.2021 – 25.03.2021
Оформлення і друк пояснювальної записки	26.03.2021 – 20.05.2021

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект *Зниження збитків через викрадення даних*

Соціальний ефект *Покращення технологій забезпечення захисту інформації в розподілених ІС.*

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ Т. В. Бабенко
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ А. В. Сахацька
(підпис) (прізвище, ініціали)

Дата видачі завдання: _____
Термін подання дипломної роботи до ЕК _____

УДК 004.

РЕФЕРАТ

Пояснювальна записка: 89 сторінок, 24 рисунків, 1 додатків, 45 джерел.

Об’єкт дослідження : процес класифікації подій кібербезпеки.

Мета роботи: розробка моделі класифікації подій кібербезпеки в розподілених ІС.

Методи дослідження: методи аналізу, системний підхід, методи порівняння та імітаційне моделювання.

У роботі досліджено розподільні інформаційні системи, механізми їх захисту. Проведено аналіз подій кібербезпеки в розподілених ІС. Запропонована модель класифікації подій кібербезпеки в розподілених ІС. Розроблено комплексний підхід до вирішення проблеми класифікації подій кібербезпеки в розподілених ІС.

Практичне значення роботи полягає у створенні моделі класифікації подій кібербезпеки в розподілених ІС для попередження порушення властивостей інформації. Результати здійснених у дипломній роботі досліджень можуть бути використані на підприємствах, головною системою яких є розподільна ІС та потребують захисту від інцидентів кібербезпеки.

Наукова новизна дослідження полягає у створенні оновленої моделі класифікації подій кібербезпеки в розподілених ІС (АС 3 класу).

Напрямки подальших досліджень в автоматизації процесу класифікації подій кібербезпеки в розподілених ІС.

Ключові слова: розподілені ІС, події кібербезпеки, інциденти кібербезпеки.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – прикладний програмний інтерфейс.

ІБ – Інформаційна безпека.

ІС – Інформаційна система.

ОС – операційна система.

ПЗ – програмне забезпечення.

ПК – персональний комп’ютер.

ВСТУП

На сьогоднішній день при розвитку технологій, суспільство стикається із цифровим світом, в якому інформаційні технології розвиваються швидкими темпами. Все більше сучасні потужні обчислювальні пристрої знаходять застосування в людському повсякденному житті. Без глобальних мереж вже неможливо уявити сучасне життя суспільства, а без мобільного зв'язку ми вже не відчуваємо себе впевнено. Для нас все більше стаємо залежні від інформаційних систем, які охоплюють нові сфери використання.

На даний момент існує багато різноманітних інформаційних систем які виконують функцію накопичення даних. При цьому існує проблема, що при зловмисних діях над такою системою не завжди є можливість виявити чи відбувається атака, або взагалі класифікувати її. Тому надійність та функціональність ІС стає особливо необхідною, а можливість класифікувати події в системі потребують дослідження, що і проводяться в цій роботі

Відповідно до статистичних даних, таких що протягом 2020 року аналітиками у сфері інформаційної безпеки спостерігалося постійне збільшення число атак на інформаційні системи. Через повсюдне впровадження віддаленої роботи з'явилися нові ризики інформаційної безпеки; соціальна інженерія стала часто застосовуватися для проникнення в мережі організацій.

Глобальний новинний привід – епідемію – почали використовувати всі типи зловмисних угруповань. З темою COVID–19 були пов'язані як масові, так і APT–атаки. При цьому, як і прогнозувалося експертами в 2019 році, число APT–атак в 2020 році продовжило рости.

Як підсумок надійність та функціональність ІС стає особливо необхідною, з причин їх розвитку та ускладнення. Пропорційною до кількості, ростуть і висунуті до них вимоги.

Метою даної дипломної роботи є розробка моделі класифікації подій кібербезпеки в розподілених ІС.

Об'єктом є процес класифікації подій кібербезпеки.

Предметом моделі класифікації подій кібербезпеки в розподілених ІС.

Методом дослідження методи аналізу, системний підхід, методи порівняння та імітаційне моделювання.

Практичне значення отриманих результатів полягає у створенні моделі класифікації подій кібербезпеки в розподілених ІС для попередження порушення властивостей інформації.

Завдання дослідження полягає в аналізі розподілених систем та класифікації подій кібербезпеки в них.

Наукова новизна полягає в створенні оновленої моделі класифікації подій кібербезпеки в розподілених ІС (АС 3 класу), за допомогою поєднання існуючих класифікацій в одну. Практичне значення полягає у покращенні класифікації подій кібербезпеки в розподілених ІС.

Апробація робіт: Бабенко Т. В., Сахацька А. В., Дослідження методів кібербезпеки в розподілених системах. // IV Міжнародна науково–практична конференція "Проблеми кібербезпеки інформаційно–телекомунікаційних систем" (PCSITS)". – 2021.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Аналіз кіберзагроз та способи їх попередження

Цифровий підхід в сучасному суспільстві характеризується високим рівнем автоматизації і зростаючим зв'язком із зовнішніми мережами, що робить об'єкти уразливими для кіберзагроз. Кібератака, крім економічного і репутаційного збитку, може потенційно викликати серйозні події (наприклад, викиди небезпечних матеріалів, пожежі, вибухи) з серйозними наслідками для робітників, населення і навколишнього середовища.

Відповідно до Cybersecurity Ventures Official Annual Cybercrime Report від 26 жовтня 2020 року головний редактор Стів Морган прогнозував що: «Кіберзлочинна діяльність є однією із найбільших викликів, з якими зіткнеться людство протягом наступних двох десятиліть»[1].

Кіберзлочинність є найбільшою загрозою для будь–якого бізнесу у світі та однією з великих проблем, з якими стикається людство. Вплив на суспільство відображається на цифрах.

У серпні 2016 року компанія Cybersecurity Ventures передбачила, що до 2021 року кіберзлочинність буде обходитися світу в 6 трильйонів доларів щорічно, в порівнянні з 3 трильйонами доларів в 2015 році. Це являє собою найбільшу передачу економічного багатства в історії, ставить під загрозу стимули для інновацій і інвестицій та буде приносити більше прибутку, ніж світова торгівля всіма основними нелегальними наркотиками разом узятими.

Прогноз даного видання кіберзлочинності залишається в силі, і за останні чотири роки його підтвердили сотні великих ЗМІ, наукові кола, високопоставлені урядовці, асоціації, галузеві експерти, найбільші технологічні компанії та компанії з кібербезпеки, а також кібербійци по всьому світу.

Оцінка вартості збитку заснована на історичних показниках кіберзлочинності, включаючи нещодавнє зростання по роках, різке збільшення числа ворожих атак дій, спонсорований державами і організованими злочинними угрупованнями, і поверхня кібератак, яка в 2021 році буде на порядок більше, ніж п'ять років тому [2].

Витрати на кіберзлочинність включають пошкодження та знищення даних, викрадені гроші, втрату продуктивності, крадіжку інтелектуальної власності, крадіжку персональних та фінансових даних, розкрадання, шахрайство, зрив звичного бізнесу, судово–медичне розслідування, відновлення та видалення зламаних дані та системи та шкоду репутації.

Надійні дослідження витримують випробування часом. У 2016 році було опубліковано «Hackerpocalypse report», в якому містився перший в історії прогноз вартості збитку від кіберзлочинів. У 2017 прогнозувалося про 3,5 мільйони незаповнених вакансій в сфері кібербезпеки до 2021 року, в порівнянні з одним мільйоном вакансій в 2014 році. У відповідь на це з’явився новий погляд на те, хто є кіберзахисником: «Кожна посада в сфері IT тепер також є посадою в сфері кібербезпеки» [3].

Кожен IT–працівник, кожен технологічний працівник повинен займатися захистом і обороною додатків, даних, пристроїв, інфраструктури і людей ". У звіті за 2018/2019 рік були представлені статистичні дані про поверхню кібератак, витратах на кібербезпеку, вимаганні та кризі робочої сили в даній області, щоб представити оцінку вартості збитку від кіберзлочинів в перспективі. Висновок полягав в тому, що подібно вуличній злочинності, яка історично зростала в залежності від зростання населення, можна спостерігати аналогічну еволюцію кіберзлочинності. Справа не тільки в більш досконалій зброї, але і в зростаючій кількості людських і цифрових мішеней.

COVID–19: Кіберзагроза віддаленим працівникам.

В результаті спалаху коронавірусу (COVID–19) велика кількість працівників було відправлена додому на віддалену роботу по всьому світу.

2020 рік був важким для всіх: компаній, регуляторних органів, приватних осіб. Через обмеження, накладені епідеміологічною ситуацією, певні категорії користувачів та компанії все частіше стають мішенню кіберзлочинців. Оскільки суспільство пристосувалось до віддаленої роботи та інших нових умов, це зробили і шахраї. Як результат, 2020 рік став надзвичайно напруженим роком з точки зору цифрових загроз, особливо для фінансових установ [4].

Тим часом деякі добре відомі групи "Постійні стійкі загрози" (APT – Advanced persistent threats), які зазвичай не націлені на фінансові установи, спробували свої сили. Група Лазарів знаходиться на певному перетині між АПТ та фінансовою злочинністю і вже була однією з найактивніших на фінансовій арені. У 2020 році група спробувала велику гру з вимогами з використанням сімейства викупників VHD. Пізніше пішли інші групи, такі як MuddyWater.

Крім того, у 2020 році було помітно, як регіональні учасники пішли глобальним шляхом. Кілька бразильських сімей зловмисного програмного забезпечення розширили свою діяльність на інші континенти, орієнтуючись на жертв у Європі та Азії. Було названо перші чотири сім'ї (Гільдма, Джавалі, Мелкос, Грандорейро) "Тетраде". Пізніше автори Guildma також створили нову банківську шкідливу програму під назвою Ghimob, націлену на користувачів у Бразилії, Парагваї, Перу, Португалії, Німеччини, Анголи та Мозамбіку [5].

Експерти з питань кібербезпеки закликали віддалених працівників підвищити обізнаність та знання про фішинг–шахрайство– найдинамічніший із кіберзлочинів, багато з яких зараз грають на страху перед коронавірусів. Після спалаху пандемії COVID–19 кількість скарг, отриманих Центром скарг на злочини в Інтернеті ФБР, зросла втричі.

Шахраї відомі тим, що полюють на вразливих людей під час національних катастроф і трагічних подій, коли люди відволікаються і втрачають пильність. Особливо виділяється, що COVID–19 гірше і підстьобує зростання кіберзлочинності,

тому що він носить глобальний характер. До березня (2020 р.) було отримано близько 1000 скарг на місяць, на даний момент їх налічує майже 3000.

Співробітники організацій будь–якого розміру та типу тепер мають мінімальні ресурси кібербезпеки, якщо вони є, порівняно з тими, що вони зазвичай мають у своєму розпорядженні. Якщо віддалені працівники не відразу починають навчання в питаннях безпеки, і компанії не відразу починають пропонувати своїм працівникам навчання з підвищення рівня безпеки, яке зосереджується на загрозі домашнього офісу, глобальна вартість збитків від кіберзлочинності може подвоїтися до кінця цього року [6, 7].

Організації мають короткий проміжок часу для можливостей навчання своїх віддалених працівників способам виявлення та реагування на фішинг–шахрайства та інші типи кібератак. Якщо вони діють оперативно та ретельно, вартість збитків, спричинених кіберзлочинністю, може бути стримана та утримана на рівні, або поруч із поточним рівнем.

Протягом багатьох років фішинг–листи ініціювали більшість кібератак на людей. Вони майже завжди хочуть, щоб користувач щось натиснув, наприклад, щоб оновити свої платіжні реквізити або отримати доступ до останньої інформації про COVID–19.

Оцінка Cybersecurity Ventures, згідно з якою вартість кіберзлочинності може потенційно подвоїтися під час спалаху коронавірусу, впливає не тільки на шахрайство з фішингом, але й на атаки з вимогами, небезпечний віддалений доступ до корпоративних мереж, віддалених працівників, облікові дані та конфіденційні дані для членів сім'ї та відвідувачів. інші загрози.

Корпоративне шпигунство, зриви бізнесу чи фінансова вигода. Незалежно від мотивації, загрози кібербезпеки стали повсюдними і продовжують впливати на всі аспекти цифрової сфери.

Cybersecurity Ventures прогнозує, що глобальна вартість збитків, що вимагаються, досягне 20 млрд. доларів до 2021 року – в 57 разів більше, ніж у 2015 році.

Відповідно до звіту про розслідування порушень даних Verizon (DBIR) за 2020 рік, 86% порушень кібербезпеки були фінансово мотивованими, а 10% – шпигунством [2].

Так, в 2020 році відбувся сплеск використання Emotet, названого Інтерполом "найнебезпечнішим шкідливим ПЗ в світі".

На початку 2021 року правоохоронні органи по всьому світу об'єднали свої зусилля, щоб зруйнувати інфраструктуру ботнету. На думку експертів "Касперського", ця операція зірве діяльність Emotet як мінімум на кілька місяців. Тим часом, принаймні, деякі клієнти Emotet перейшли на Trickbot [8].

Незважаючи на те, що в 2020 році можна було стати свідками все більш витончених кібератак, загальна статистика виглядає обнадійливою: число користувачів, які постраждали від шкідливих програм для комп'ютерів і мобільних пристроїв, знижується, як і число фінансових фішинг.

Однак це не означає, що кіберсвіту став безпечнішим – це означає, що цілі і тактика кіберзлочинців зазнали ряд змін. Незважаючи на зниження загальної статистики, стає помітно, що атаки стали більш цілеспрямованими і орієнтованими на бізнес [9]. У той же час є можливість спостерігати, як кіберзлочинці вміло адаптуються до глобальних змін і отримують вигоду від вразливостей телероботи і зростаючої популярності онлайн–покупок.

Окрім серйозної фінансової шкоди, кібератаки можуть призвести до штрафів, судових процесів, збитку репутації та порушення безперервності бізнесу. У сучасному кібер світі жодна компанія чи IT–організація не захищені [10, 11]. Оскільки кіберзлочинці дедалі більше покладаються на передові технології, організації часто почуваються безнадійно, оскільки їхні конфіденційні дані та важливі активи стають жертвами зловмисних атак.

Крім того, швидке впровадження нових технологій, зокрема штучного інтелекту, Інтернету речей (IoT) та хмарних обчислень, додало нових кіберзагроз для бізнесу та ускладнило існуючі ризики.

Основні важливі моменти за 2020 рік можна переглянути на рисунку 1.1

Рисунок 1.1 – Підсумки за 2020 рік

За 2020 рік «The Verizon Data Breach Investigations Report» (DBIR) створили звіт в якому було проаналізовано 157 525 інцидентів, 30 002 відповідали їх стандартам якості; обсяг даних за цей рік був дуже великим [12].

Під час дослідження особливою інформацією є визначення інциденту та порушення, що спостерігаються в даному виданні:

- Інцидент: Подія безпеки, яка ставить під загрозу цілісність, конфіденційність або доступність інформаційного активу.
- Порушення: Інцидент, який призводить до підтверджене розкриття – або не тільки потенційне розкриття даних неавторизованій стороні.

За роки досліджень з 2014 року компанія почала використовувати шаблони, що представляють собою своєрідні кластери схожих інцидентів.

В результаті було виділено 9 головних кластерів класифікації інцидентів. Це дозволяє нам абстрагуватися і обговорити тенденції в закономірності, а не тенденції в кожній різній комбінацій: дії, активи, виконавці та атрибути.

Якщо проаналізувати 409 000 інцидентів безпеки і майже 22 000 якісних витоків даних з моменту створення даного звіту, то цифри показують, що 94% інцидентів безпеки і 88% витоків даних потрапляють в одну з дев'яти первинних схем. Однак, якщо сфокусувати погляд тільки на даних цього року, то процентне співвідношення знижується до 85% інцидентів безпеки і 78% порушень даних [13].

Ніщо не демонструє це краще, ніж категорія «Все інше», яка фактично була створена як ящик для запасних USB–кабелів, в якому зберігаються дані про порушення, що піднялася на одне з перших місць завдяки зростанню фішингу, в той час як деякі інші моделі різко скоротилися з моменту їх появи [14].

Схоже, що час не чекає ніяких шаблонів, і єдине постійне порушення – це порушення, що змінюються з часом.

Шаблони представляють із себе (розташовані в порядку спадання, відповідно до кількості інцидентів відповідно до них за 2020 рік):

- відмова в обслуговуванні;
- шкідливе ПО;
- «все інше»;
- веб–додатки;
- втрачені і вкрадені активи;
- різні помилки;
- зловживання привілеями;
- кібершпіонаж;
- точка продажу;
- використання платіжних карт.

В 2020 році було зібрано 157 525 інцидентів і 108 069 порушень. Це може здатися вражаючим, поки не починаєш розуміти, що 100 000 і більше з цих порушень були облікові дані окремих користувачів, що були взяті з метою злому банківських рахунків, хмарних сервісів і т.п.Різну кількість інцидентів було зафіксовано в різних сферах промисловості, на наступних таблицях буде представлено декілька з них із найбільшим ураженням від інциденті [13].

Таблиця 1.1 – Аналіз інцидентів освіти зо 2020 рік

У цій галузі фішингові атаки спостерігалися в 28% випадків, а злом за допомогою вкрадених облікових даних – в 23% випадків. За даними інцидентів, на частку програм–вимагачів припадає близько 80% заражень шкідливим ПО в цій вертикалі. Освітні послуги показали низькі результати в плані інформування про фішингових атаках, що призвело до втрати критично важливого часу реагування для постраждалих організацій.	
Частота	798 інцидентів, 521 з підтвердженим розкриттям даних.
Основні типи порушень	«Все інше», різні помилки і веб–додатки складають 81% порушень.
Агенти загроз	Зовнішні (67%), внутрішні (33%), партнерські (1%), множинні (1%).
Мотиви	Фінансові (92%), розваги (5%), зручність (3%), шпигунство (3%), Другорядні (2%).
Скомпрометовані дані	Особисті (75%), облікові дані (30%), інші (23%), внутрішні(13%).
Кращі засоби контролю	Реалізувати програму підвищення обізнаності та навчання з питань безпеки, захисту кордонів, безпечної конфігурації

Таблиця 1.2 – Аналіз інцидентів фінансів та страхування за 2020 рік

Атаки в цьому секторі відбуваються зовнішніми суб'єктами, фінансово мотивованими на отримання легко монетизованими даних (63%), внутрішніми фінансово мотивованими суб'єктами (18%) і внутрішніми суб'єктами, які здійснюють помилки (9%). Атаки на веб–додатки, що використовують вкрадені облікові дані, також продовжують впливати на цю галузь. Порушення, викликані внутрішніми акторами, змістилися від зловмисних дій до доброякісних помилок, хоча і ті, і інші, як і раніше завдають шкоди.

Частота	1509 інцидентів, 448 з підтвердженим розкриттям даних.
Основні типи порушень	Веб–додатки, різні помилки і «все інше» складають 81% порушень.

Продовження таблиці 1.2

Агенти загроз	Зовнішні (64%), внутрішні (35%), партнерські (2%), множинні (1%).
Мотиви	Фінансові (91%), шпигунство (3%), злість (3%).
Скомпрометовані дані	Особисті (77%), інші (35%), облікові дані (35%), банківські (32%).
Кращі засоби контролю	Реалізувати програму підвищення обізнаності та навчання з питань безпеки, захисту кордонів, безпечної конфігурації

Таблиця 1.3 – Аналіз інцидентів охорони здоров’я за 2020 рік

Фінансово мотивовані злочинні групи продовжують атакувати цю галузь за допомогою програм–вимагачів. Втрачені і вкрадені активи також залишаються проблемою в наборі даних про інциденти. Базові людські помилки жива і здорова в цій вертикалі. Неправильна доставка посіла перше місце серед типів дій, пов'язаних з помилками, в той час як внутрішнє Неправильне використання знизилося.	
Частота	798 інцидентів, 521 з підтвердженим розкриттям даних.
Основні типи порушень	Веб–додатки, різні помилки і «все інше» складають 72% порушень.
Агенти загроз	Зовнішні (51%), внутрішні (48%), партнерські (2%), множинні (1%).
Мотиви	Фінансові (88%), розваги(4%), шпигунство (4%).
Скомпрометовані дані	Особисті (77%), медичні(67%), інші (18%), облікові дані (18%).
Кращі засоби контролю	Реалізувати програму підвищення обізнаності та навчання з питань безпеки, захисту кордонів, безпечної конфігурації

Таблиця 1.4 – Аналіз інцидентів в інформаційному секторі за 2020 рік

У цій галузі широко поширені атаки на веб–додатки через уразливості і використання вкрадених облікових даних. Помилки продовжують залишатися значним фактором і в основному полягають у наступному. Неправильна конфігурація

хмарних баз даних. Зростання числа атак типу «відмова в обслуговуванні» також залишається проблемою для інформаційного сектора.	
Частота	5741 інцидентів, 360 з підтвердженим розкриттям даних.
Основні типи порушень	Веб–додатки, різні помилки і «все інше» складають 88% порушень.

Продовження таблиці 1.4

Агенти загроз	Зовнішні (67%), внутрішні (34%), множинні (2%) партнерські (1%).
Мотиви	Фінансові (69%), шпигунство (41%), розваги(2%), образа (2%), інше(1%)
Скомпрометовані дані	Особисті (69%), облікові дані (41%), інші (34%), внутрішні (16%).
Кращі засоби контролю	Безпечні конфігурації, постійне управління уразливими, впровадження програми, підвищення обізнаності та навчання з питань безпеки

Таблиця 1.5 – Аналіз інцидентів у виробництві за 2020 рік

Виробництво піддається нападу зовнішніх суб'єктів, які використовують шкідливе ПО для скидання паролів і вкрадені облікові дані для злому системи і крадіжки даних. Хоча більшість атак мають фінансову мотивацію, було визначено ряд атак, мотивованих кібершпіонажем. в цій галузі. Внутрішні співробітники, які зловживають своїм доступом з метою розкрадання даних, також залишається проблемою для цього сектору.	
Частота	922 інцидентів, 381 з підтвердженим розкриттям даних.
Основні типи порушень	Зловмисне програмне забезпечення, веб–програми та зловживання привілеями складають 64% порушень.
Агенти загроз	Зовнішні (75%), внутрішні (25%), партнерські (1%).
Мотиви	Фінансові (73%), шпигунство (27%).
Скомпрометовані дані	Облікові дані (55%), особисті (49%), інші (25%), оплати (20%).
Кращі засоби контролю	Захист кордонів , впровадження програми підвищення обізнаності та навчання з питань безпеки та захисту даних

Таблиця 1.6 – Аналіз інцидентів в професійних, наукових та технічних сервісах за 2020 рік

Фінансово мотивовані зловмисники продовжують красти облікові дані і використовувати їх в інфраструктурі веб–додатків. Соціальна інженерія у вигляді фішингу та pretexting атак є поширеною тактика, використовувана для отримання	

Продовження таблиці 1.6

доступу. Ця галузь також регулярно страждає від атак типу «відмова в обслуговуванні».	
Частота	7463 інцидентів, 326 з підтвердженим розкриттям даних.
Основні типи порушень	Веб–додатки, різні помилки і «все інше» складають 79% порушень.
Агенти загроз	Зовнішні (75%), внутрішні (22%), партнерські (3%), множинні (1%).
Мотиви	Фінансові (93%), шпигунство (8%), ідеологія (1%).
Скомпрометовані дані	Особисті (75%), облікові дані (45%), інші (32%), інтернет (27%).
Кращі засоби контролю	Безпечна конфігурація, впровадження програми підвищення обізнаності та навчання з питань безпеки, захист кордонів

Таблиця 1.7 – Аналіз інцидентів в сфері державного управління за 2020 рік

Програми – вимагачі є великою проблемою для цього сектора, оскільки фінансово мотивовані зловмисники використовують його для атак на широкий спектр урядових організацій. Неправильна доставка та помилки неправильної конфігурації також присутні в цьому секторі.	
Частота	8843 інцидентів, 346 з підтвердженим розкриттям даних.
Основні типи порушень	Веб–додатки, різні помилки і «все інше» складають 73% порушень.
Агенти загроз	Зовнішні (59%), внутрішні (43%), множинні (2%), партнерські (1%).
Мотиви	Фінансові (75%), шпигунство (19%), розвага (3%).
Скомпрометовані дані	Особисті (51%), інші (34%), облікові дані (33%), внутрішні (14%).
Кращі засоби контролю	Впровадити систему безпеки, реалізувати програму підвищення обізнаності та навчання, захист кордонів, безпечна заміна

Основні види кібератак представлять із себе такі величини:

1) Шкідливе ПО.

Атаки з використанням шкідливого ПЗ є найбільш поширеним видом кібератак. Шкідливе ПО визначається як шкідливе програмне забезпечення, включаючи шпигунські програми, програми–викупи, віруси і черв'яки, яке встановлюється в систему, коли користувач переходить по небезпечній посиланням або електронній пошті. Потрапляючи в систему, шкідливе ПЗ може блокувати доступ до критичних компонентів мережі, пошкодити систему, збирають конфіденційну інформацію і т.п. [17].

За даними Accenture, середня вартість атаки шкідливого ПЗ становить 2,6 мільйона доларів США.

2) Фішинг.

Кіберзлочинці розсилають шкідливі електронні листи, які здаються що виходять від легітимних ресурсів. Потім користувача обманом змушують перейти по шкідливої посиланням в листі, що призводить до встановлення шкідливого ПО або розкриття конфіденційної інформації, такої як дані кредитної картки і облікові дані для входу в систему.

На частку фішингових атак припадає понад 80% зареєстрованих кібер–інцидентів.

3) Спір–фішинг.

Спір–фішинг – це більш складна форма фішинговою атаки, при якій кіберзлочинці атакують тільки привілейованих користувачів, таких як системні адміністратори і керівники вищої ланки.

Більш 71% цільових атак пов'язані з використанням спір–фішинг.

4) Атака "людина посередині».

Атака "людина посередині" (MitM) відбувається, коли кіберзлочинці поміщають себе між двома сторонами. Як тільки зловмисник інтерпретує комунікацію, він може фільтрувати і красти конфіденційні дані і повертати користувачеві різні відповіді [18].

За даними Netcraft, 95% серверів HTTPS уразливі до MitM.

5) Атака на відмову в обслуговуванні.

Атаки типу "відмова в обслуговуванні" спрямовані на повінь систем, мереж або серверів масивним трафіком, в результаті чого система стає нездатною виконувати законні запити. Атаки також можуть використовувати кілька заражених пристроїв для атаки на цільову систему. Це відомо як розподілена атака типу «відмова в обслуговуванні" (DDoS).

У 2019 було зафіксовано 8,4 мільйона DDoS–атак.

6) SQL–ін'єкція.

Атака з використанням структурованої мови запитів (SQL) відбувається, коли зловмисники намагаються отримати доступ до бази даних шляхом завантаження шкідливих SQL–скриптів. Після успіху зловмисник може переглядати, змінювати або видаляти дані, що зберігаються в базі даних SQL [19].

На частку SQL–ін'єкцій припадає майже 65,1% всіх атак на веб–додатки.

7) Експлойт нульового дня.

Атака "нульового дня" відбувається, коли оголошується про уразливість програмного забезпечення або обладнання, і зловмисники використовують цю вразливість до появи виправлення або рішення.

Згідно з прогнозами, до 2021 року число атак "нульового дня" зросте до однієї в день.

8) Просунуті постійні погрози (APT).

Передова постійна загроза виникає, коли зловмисник отримує несанкціонований доступ до системи або мережі і залишається непоміченим протягом тривалого часу [20].

45% організацій вважають, що вони можуть стати метою APT.

9) Програми–вимагачі.

Програми–вимагачі – це тип атаки шкідливого ПО, при якому зловмисник блокує або шифрує дані жертви і погрожує опублікувати їх або блокує доступ до них, якщо не буде виплачений викуп.

За оцінками, дана атака обійдеться світовим організаціям в 20 млрд доларів США до 2021 року.

10) DNS–атака.

DNS–атака – це кібератака, в ході якої зловмисники використовують уразливості в системі доменних імен (DNS). Зловмисники використовують уразливості DNS для перенаправлення відвідувачів сайтів на шкідливі сторінки (DNS Hijacking) і витоку даних з зламаних систем (DNS Tunneling).

У 2020 році середня вартість DNS–атаки склав 924 000 доларів США.

Варто також зауважити, що у минулому році на кібербезпеку було витрачено або втрачено в результаті кібератаки більше 1 трильйона доларів, що становить приблизно один відсоток світового ВВП. Це випливає зі звіту компанії Atlas VPN, в якому говориться, що велика частина ($ 945 млрд) була втрачена в результаті інцидентів кібербезпеки, а решту $ 145 млрд були витрачені на забезпечення захисту [21].

Витрати на кібербезпеку виросли більш ніж наполовину в порівнянні з 2018 роком, протягом якого компанії витратили трохи більше 600 мільярдів доларів на забезпечення безпеки і реагування на інциденти, в той час як втрати зросли на 81 відсоток за той же період часу.

З огляду на картину, намальовану цими цифрами, Atlas VPN з подивом виявила, що багато організацій до цих пір не усвідомлюють небезпеку, яку представляють кіберзагрози. П'ята частина організацій по всьому світу не має плану щодо захисту від загроз, відзначає компанія [22].

З іншого боку, багато компаній (44 відсотки) відносно добре підготовлені, причому лідирують в цьому відношенні фірми, розташовані в Канаді. Більше половини

(55%) канадських організацій мають план захисту від інцидентів, пов'язаних з кібербезпекою, і управління ними.

Як висновок, жодна організація не може бути повністю захищена від кібератак, а їх наслідки можуть бути руйнівними. Тому як превентивні, так і реактивні стратегії кібербезпеки необхідні, якщо компанія хоче знизити ризики кіберзлочинності. Наявність плану дій на випадок злому будь–якої організації не менш важливо, ніж захист від таких загроз.

Розглянемо наступний важливий елемент захисту ІС, допоможе для класифікації інцидентів кібербезпеки, а саме система управління інформацією та подіями в області безпеки (SIEM – Security Information and Event Management), що має такі функції [23]:

- отримувати журнали з найрізноманітніших засобів захисту: підтримка великої кількості системних типів, власні протоколи, вимоги API;
- нормалізація отриманих даних: Перетворення даних в єдиний формат, придатний для подальшого використання;
- таксономія нормалізованих даних: класифікація нормалізованих повідомлень для формування, а потім аналіз послідовності типів подій із певним значенням та конкретним часом виникнення;
- кореляція класифікованих подій: Кореляція між подіями, які відповідають певним умовам (правила кореляції);
- створення інциденту, надання інструментів розслідування;
- зберігання інформації про події та інциденти протягом тривалого періоду (від 6 місяців) із механізмами стиснення та оптимізації;
- швидкий пошук у даних, що зберігаються в SIEM.

Характерною особливістю систем SIEM є інтеграція зі сторонніми рішеннями для більш всебічного аналізу подій та інцидентів у сфері інформаційної безпеки. Одним із типів такого рішення є Cyber Threat Intelligence, який подає дані в систему

SIEM, яка називається джерелами інформації Threat Intelligence (TI Feeds або TI Feeds, Cyber Intelligence Data Source) [27].

Потоки TI (TI Feeds) представляють з себе низки показників компромісу (IoC), використання яких пов'язане з підходом "Припускається порушення". Складне шкідливе програмне забезпечення можна знайти в інфраструктурі атакованої компанії за десять /ста днів до його виявлення. У свою чергу індикатор компромісу (IoC) – це об’єкт / артефакт в IT–інфраструктурі компанії, наявність якого може свідчити про неминучу, триваючу або вже здійснену комп’ютерну атаку з високим ступенем ймовірності.

Для точного виявлення атаки статичні показники IoC менш придатні (оскільки їх може легко замінити / оновити зловмисник), а найнадійнішими методами є методи, що використовують аналіз TTP зловмисника – також визначається серія кроків, що використовуються зловмисниками за динамічними компромісними показниками, напр В. послідовність запуску певних утиліт та програм, доступ до пам'яті за допомогою системних процесів, доступ до сервісних мережевих ресурсів тощо [25].

Розглянемо таку важливу річ як ситуаційні центри інформаційної безпеки – SOC (Security Operations Center). Завданнями SOC є моніторинг роботи системи захисту інформації та реагування на інциденти у сфері інформаційної безпеки. SOC – це група спеціалістів з інформаційної безпеки, яка постійно відстежує новини технології для швидкого вирішення загроз інформаційної безпеки.

Центри SOC поділяються на :

- внутрішній (виділений структурний підрозділ компанії);
- зовнішній (комерційний / аутсорсинг).

Ситуаційні центри утворюються за допомогою такого рівняння: SOC = технологія + процеси + люди.

Технології центрів SOC представлені за рахунок таких систем :

- SIEM – Інформація про безпеку та управління подіями, Системи управління інформацією про безпеку та Події інформаційної безпеки;

- IRP – Платформа реагування на інциденти кібербезпеки;
- SOAR – Оркестрація безпеки, автоматизація та реагування, системи управління, автоматизація та реагування на аварії;
- SGRC – системи управління ризиками та дотриманням вимог.

Процеси центру SOC представлені у вигляді сценарії реагування (посібники), згідно з якими група реагування вживає певних заходів залежно від деталей інциденту. Розробка сценаріїв, тестування, навчання працівників. Налаштуйте правила кореляції в SIEM. Конфігурація, оптимізація систем SIEM/IRP/SOAR/SGRC. Необхідно взаємодіяти з клієнтами, щоб зменшити помилкові спрацьовування.

Збір і запит подій з різних систем або додатків не завжди вимагає всебічної функціональності SIEM. Часто найкращим вибором є система збору журналів (LM), яку простіше встановити та використовувати. Системи LM включають функції накопичення, зберігання та звітності SIEM, але мають значно вдосконалені інтерфейси, спрощений аналіз та нижчу кореляцію [27].

Багато постачальників стверджують, що їх системи LM мають можливості SIEM. Як результат, часом важко зрозуміти, до якого класу продуктів належить рішення – SIEM або LM.

Описана вище архітектура SIEM також застосовується до більшості рішень LM. Можна виділити наступні відмінності:

- системи LM зазвичай не мають надійного механізму кореляції, вдосконаленого довгострокового аналізу, повної обробки та підтримки ескалації;
- системи LM зазвичай менше обробляють та нормалізують канали даних і зосереджуються на дуже швидких спеціальних функціях пошуку. Для надійної кореляції зазвичай потрібна система SIEM, щоб зрозуміти значення різних елементів даних у кожній події, і більшість систем LM не повністю аналізують свої дані. Натомість вони зосереджуються на швидкому повнотекстовому пошуку або вилученні лише певних функцій, так що надійне співвідношення або неможливе, або вимагає набагато більше роботи з боку автора вмісту;

- оскільки системи LM не роблять стільки попередньої обробки чи подальшої обробки даних, вони можуть обробляти їх набагато швидше за допомогою меншої бази вихідного коду. Це свідомий компроміс, який, як бачимо, має переваги та недоліки.

Ці два продукти багато в чому доповнюють один одного. Більшість SOC, які впровадили SIEM, розпочали роботу з базовим LM та розширили можливості передачі даних та запитів у міру розширення бізнесу.

Багато невеликих SOC без мільйонів бюджетів SIEM можуть додати колектор журналів на свої консолі IDS і скористатися деякими перевагами SIEM, але за низькою вартістю. Крім того, IT–відділи або НОК, як правило, самі забезпечують ЛМ–системи без мети захисту мереж. Нарешті, охоронні організації можуть надавати ЛМ для власних цілей моніторингу. У будь–якому випадку корисно поєднувати ресурси SOC з цими групами для уніфікації архітектури збору даних [28].

Одним із способів зрозуміти різницю між повноцінними системами SIEM та LM є те, що SIEM може служити основою для робочого процесу мережевої безпеки, тоді як LM не може. У багатьох організаціях клієнтів захист мережі здійснюється спеціально (наприклад, командою безпеки). У таких організаціях ресурси обмежені, і не так багато людей присвячується цій роботі. Тому їхні вимоги добре відповідають пристрою LM. Повний SIEM вимагає інвестицій, які можуть запропонувати лише середні та великі SOC.

1.2 Підходи до аналізу подій кібербезпеки

Усі події, що мають можливість відбутися в кіберпросторі можуть бути розкритими за рахунок аналітичних засобів, наприклад таких, що досліджують середовище.

Оскільки дані є рушійною силою сучасного світу, практично кожен стикався з такими поняттями, як наука про дані (**Data science**), машинне навчання (**Machine**

learning), штучний інтелект(**Artificial intelligence**), глибоке навчання та видобуток даних [29]. Але що саме означають ці терміни? Які відмінності та взаємозв'язки між ними?

Наука про дані – це широке наукове дослідження, орієнтоване на розуміння даних. Наприклад, подумайте про системи рекомендацій, які дозволяють клієнтам робити персоналізовані пропозиції на основі їх історії пошуку. Наприклад, якщо один покупець шукає вудку та приманку, а інший – волосінь серед інших товарів, є велика ймовірність, що перший покупець буде зацікавлений також у придбанні волосіні. Наука даних – це широка галузь, яка охоплює всі види діяльності та технології, що сприяють побудові таких систем, особливо ті, про які буде розглянуто нижче [30].

Машинне навчання має на меті навчити машини на історичних даних, щоб вони могли обробляти нові дані на основі вивчених зразків без явного програмування, тобто без ручних вказівок для системи для здійснення певних дій. Без машинного навчання вищезазначені системи рекомендацій не були б доступними, оскільки одній людині важко обробити мільйони пошукових запитів, оцінок "подобається" та оглядів, щоб з'ясувати, які клієнти зазвичай купують вудки та які лінії [31].

Штучний інтелект – складна тема. Для простоти припустимо, що будь–який реальний продукт даних можна описати як штучний інтелект. Давайте зупинимося на прикладі, яким надихає риболовля. Ви хочете придбати конкретну модель вудки, але маєте лише її фотографію і не знаєте назви торгової марки. Система штучного інтелекту – це програмний продукт, який може вивчити вашу фотографію та запропонувати назву продукту та зберегти, де ви можете його придбати. Для створення продукту ШІ потрібно використовувати методи аналізу даних, машинного навчання, а іноді і глибокого навчання [32].

Незважаючи на їх зв'язок, перелічені терміни не можуть використовуватися як взаємозамінні, тому є можливість ефективно аналізувати події кібербезпеки з трьох різних точок зору: джерела даних, методи машинного навчання та кінцеві результати.

Основними джерелами даних про стан інформаційних систем, їх безпеку та діяльність є: журнали подій, реєстрація послуг, аналіз трафіку тощо. Їх використання є важливою частиною забезпечення кібербезпеки мережі загалом. На наступному рисунок 1.2 показано основні джерела цих подій в галузі кібербезпеки [33].

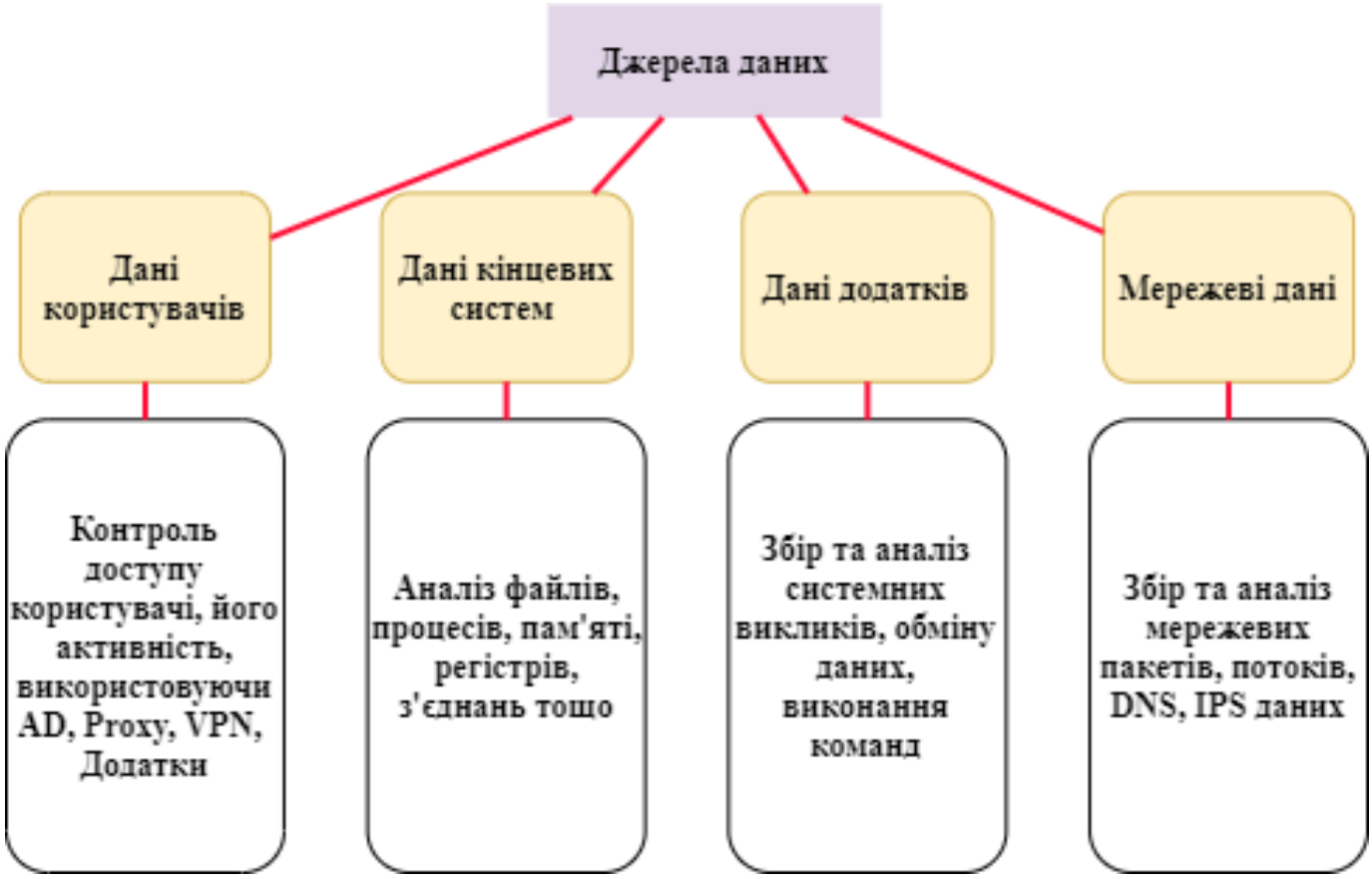

Рисунок 1.2 –Джерела даних

В даний час існує багато продуктів, реалізованих для збору та аналізу даних різного типу: аналізатори мережевих пакетів, антивірусні програми, брандмауери, системи IDS / IPS та т.п..

У цій частині дипломної роботи детально розглядається збір та аналіз даних про мережеві інтерфейси та мережеві служби.

Накопичення та рутинний аналіз даних із діагностичних систем не завжди призводить до бажаного результату. Бувають ситуації, коли обсяг інформації настільки великий, що знайти відповідну інформацію в потрібному контексті стає неможливою

задачею, і відповідно постійне виникнення нових інцидентів кібербезпеки, які можуть дуже відрізнятися від минулих, не дозволяє керівнику зосередитися на події для реагування в кіберсистемі. Методи машинного навчання часто використовуються для підвищення швидкості та точності рішень щодо управління кібернетичною системою. З кожними нещодавно набутим знанням якість та швидкість майбутніх аналізів постійно зростає. Методи машинного навчання включають наступне (рисунок 1.3) [34].

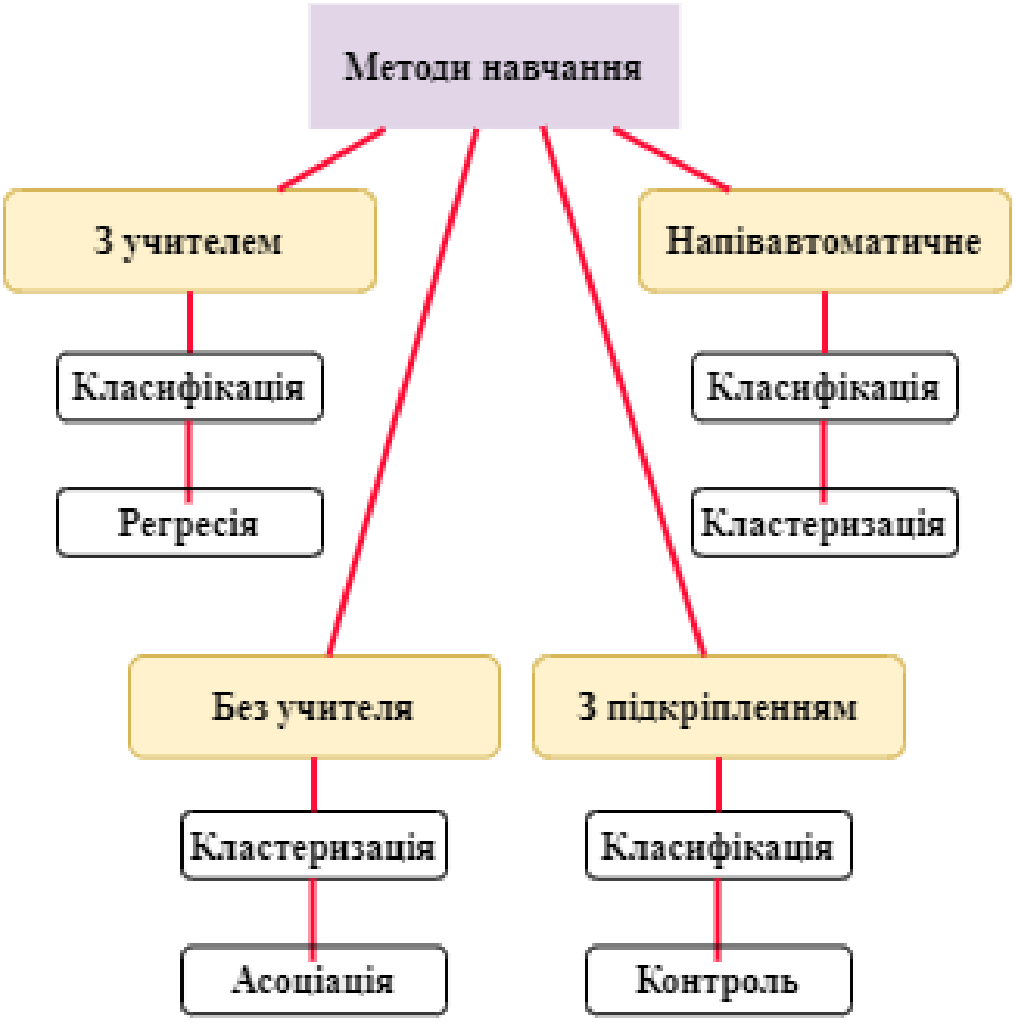

Рисунок 1.3 –Методи машинного навчання

Контрольоване навчання з учителем. Метод машинного навчання, при якому тестова система змушена вчитися на існуючих наборах прикладів "стимул–реакція", щоб визначити "відповідь" на "подразники", що не мають зв’язку з існуючим набором

прикладів, включають . Машина використовує попередні дані, які вже були позначені як хороші чи погані (реальні інциденти або несправні системи безпеки, шахрайські або звичайні дії). Навчання з викладачем включає класифікацію, регресію та глибоке навчання [35].

Навчання без учителя. Один із способів представлення роботи машинного навчання, коли система не має інформації про пройдені події. Тестова система вчиться спонтанно виконувати завдання без втручання. Зазвичай це підходить лише для проблем, коли опис групи об'єктів відомий (навчальний посібник), і необхідно виявити внутрішні взаємозв'язки, залежності та закономірності, що можуть існувати між об'єктами. Навчання без викладачів включає кластеризацію, правила призначення та узгодження зразків.

Навчання з підкріпленням. Цей метод машинного навчання, який готує модель, яка не має інформації про систему, але має можливість виконувати над нею дії. Дії приводять систему в інший стан, і модель отримує деяку відповідь від системи.

Напівавтоматичне навчання. Метод машинного навчання, тип підготовки з вчителем, який також використовує для навчання немічені дані – зазвичай незначну кількість маркованих даних та значну кількість протилежних даних. Напівавтоматичне навчання собою займає проміжне положення між навчанням без викладача (без позначених даних для навчання) та навчанням з викладачем (лише з позначеними даними). Багато дослідників в сфері машинного навчання визначили, що немічені дані в поєднанні з незначною кількістю маркованих даних мають змогу помітно покращити точність у навчанні [37].

Розглянута в дипломній роботі модель нейронної мережі класифікації подій кібербезпеки спирається на підготовку з вчителем. Роль викладача змоделювали за допомогою особливо розробленого ПЗ, що містить алгоритм обробки необроблених даних та переведення їх в необхідний формат, що буде визначений в практичній частині диплому.

Для кожної моделі навчання є необхідність базуватися на якомусь алгоритмі.

На наступному рисунку представлена повна інформація про кожен із існуючих та використаних алгоритмів моделювання машинного навчання та про те, як вони працюють (рисунок 1.4).

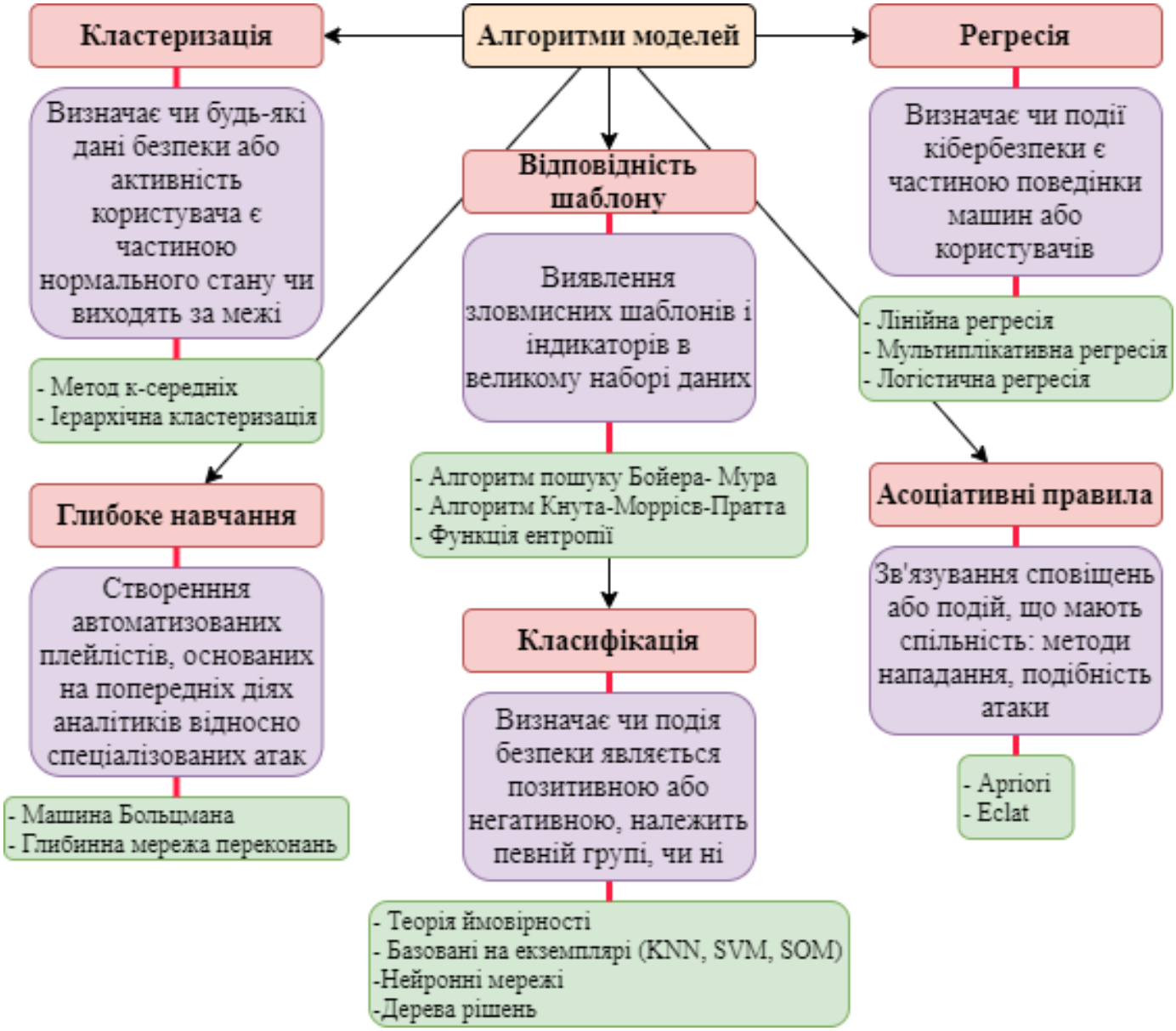

Рисунок 1.4 –Алгоритми моделей машинного навчання

Ось кілька прикладів того, як ці моделі можна використовувати: Для створення спам–фільтрів для визначення фішингу використовуйте метод Байєса (теорія ймовірності), який класифікує звичайні повідомлення та спам–повідомлення. Цей метод заснований на теорії ймовірностей та статистичному аналізі.

Нейронні мережі та системи піддержки прийняття рішень щодо шахрайства використовуються для виявлення інтернет–шахрайства.

Методи кластеризації використовуються для виявлення внутрішніх загроз, таких як порушення доступу користувачів або витоку даних.

Ботів можна виявити за допомогою особливих функцій ентропії, які застосовуються до результатів взаємодії машина–машина.

Асоціативний аналіз дозволяє ідентифікувати групи зловмисників, використовуючи звичайні (відомі) методи атаки в мережі.

Для ведення успішного бізнесу компанія зобов'язана вміти аналізувати можливі інциденти безпеки, успішно протидіяти атакам та розуміти всі ризики. Це найвищий крок у побудові інтелектуальної системи. Для цього потрібно використовувати такі аналітичні методи (рисунок 1.5).

Рисунок 1.5 –Аналіз подій кібербезпеки

Описова аналітика. Перегляд минулого корисний, оскільки маємо можливість повчитися з минулих моделей поведінки та зрозуміти, як вони можуть вплинути на майбутні результати.

Інтелектуальна аналітика. Зрозуміти майбутнє – дає оцінку ймовірності майбутніх результатів. Необхідно пам'ятати, що ніякий статистичний алгоритм не має можливості «передбачити» майбутнє із 100% вірогідністю. Компанії користуються цією статистикою для передбачування того, що буде в майбутньому.

Прескриптивна аналітика. Поради про можливі результати – дозволяє користувачам "прописувати" низку різних можливих дій та направляти їх до рішень. Аналітика, що виписує рецепт, прагне кількісно оцінити вплив майбутніх рішень, щоб надати поради щодо можливих результатів до того, як рішення буде прийнято насправді. У найкращому випадку, рецептурний аналіз не тільки передбачає, що станеться, але і чому це відбудеться.

Діагностична аналітика – це форма розширеного аналізу, що аналізує інформацію або вміст, щоб отримати відповідь на запитання "Чому це сталося?".

Детективна аналітика – заснована на аналізі та виявленні об'єктів, які досі невідомі на практиці та можуть становити загрозу.

Багато сучасних засобів виявлення та аналізу, такі як EDR та криміналістика мережі, є хорошими зразком діагностичного та детективного аналізу. IBM Watson є прикладом аналітичної політики, оскільки вона збирає інформацію із глобальних джерел та представляє свою аналітику під час обробки інцидентів. Інструменти для аналізу поводження користувачів можуть забезпечувати прогнозований аналіз на базі минулих даних.

В кібербезпеці не існує єдиного методу або захисного механізму, який би міг захищати від усіх можливих загроз. В теорії інформаційного аналізу ситуація однакова: наприклад, неможливо визначити усі можливі типи атак одним єдиним методом.

Таблиця 1.8 – Важливість аналітичної системи щодо головних джерел небезпеки

ДЖЕРЕЛО НЕБЕЗПЕКИ	АНАЛІЗ МЕРЕЖІ	АНАЛІЗ ПОВЕДІНКИ КОРИСТУВАЧА	АНАЛІЗ КІНЦЕВИХ СИСТЕМ	АНАЛІЗ ДОПОВНЕНЬ
Продвинуте зловмисне ПЗ	+		+	–
Соціальна інженерія	+	+	+	–
«Боковий рух мережі»	+	+	+	–
Інсайдер	–	+	–	–
Транзакції	–	–	–	+
Отримання облікового запису другого користувача	–	+	–	–
Видалення даних	+	+	–	+
Виконання експлоїтів	–	–	–	+
Шифровані інциденти	–	–	+	+

Ефективна система аналізу має складатися з деякої кількості спеціалізованих систем аналізу та об’єднувати результат обробки будь–кого з них.

В універсальному випадку повноцінна модель приймання рішень, що стосується певної події в ІС, повинні охоплювати всі три вищезазначені поняття, тобто модкль повинна характеризуватися такими аспектами:

– повинні бути постійні джерела інформації про події (журнали подій, інформація про мережевий трафік і т.п.);

– на базі постійного надбання та зберігання даних з метою опублікування результатів їх обробки з безупиним вдосконаленням цих результатів;

– повинен виконуватися остаточний аналіз та надаватися поради аналітикам.

Основною моделлю побудови інтелектуальної системи виявлення та класифікації подій кібербезпеки є наступне:

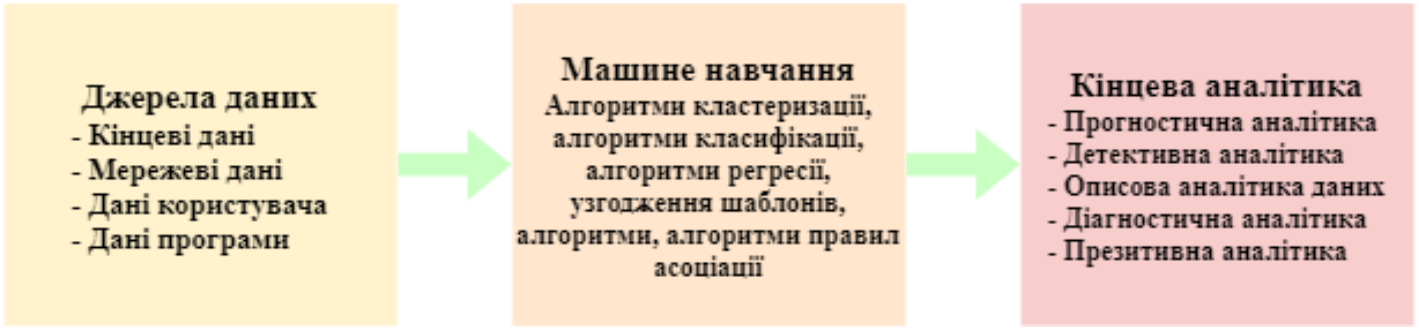

Рисунок 1.6 – Модель побудови інтелектуальної системи класифікації подій кібербезпеки

1.3 Постановка задачі

Об’єктом дослідження даної роботи є процес класифікації подій кібербезпеки. Для виконання поставленої задачі необхідно розбити його на менші завдання, тому надалі для виконання мети роботи, будуть вирішені наступні завдання:

– аналіз існуючих атак та дані про них, такі як журнал подій, накопичення даних про трафік;

– зведення даних до загального вигляду, тобто скласти в потоки даних;

– алгоритм підготовки даних, його розробка;

– модель нейронної мережі, її навчання – загальна розробка;

– тестування моделей, їх перевірка на адекватність.

1.4 Висновки за розділом 1

В першому розділі дипломної роботи було проведено аналіз кіберзагроз, що мали особливий вплив у 2020 році з урахуванням пандемії. Досліджено вплив атак на різні сфери життєдіяльності людини а також було проаналізовані підходи для аналізу подій кібербезпеки.

РОЗДІЛ 2
РОЗРОБКА МОДЕЛІ КЛАСИФІКАЦІЇ ПОДІЙ КІБЕРБЕЗПЕКИ

2.1Основи нейронної мережі та її архітектури

Нейронні мережі відображають поведінку людського мозку і дозволяють комп'ютерним програмам розпізнавати закономірності та вирішувати загальні проблеми в ШІ, машинному навчанні та глибокому навчанні.

Історія нейронних мереж довша, ніж думає більшість людей. Хоча ідею "машини, яка думає" можна простежити ще у стародавніх греків, зупинимось на ключових подіях, що призвели до розвитку мислення щодо нейронних мереж, які з часом стають все більш популярними:

1943: Уоррен С. Мак–Каллок і Уолтер Піттс публікують «Логічне обчислення ідей, іманентних нервовій діяльності». Це дослідження мало на меті зрозуміти, як людський мозок може створювати складні зразки, використовуючи підключені клітини мозку або нейрони. Однією з головних ідей, що виникла в цій роботі, було порівняння бінарних порогових нейронів з булевою логікою (тобто 0/1 або твердженнями true / false).

1958: Френку Розенблату приписують розробку перцептрону, як це задокументовано в його дослідженні Perceptron: імовірнісна модель для зберігання та організації інформації в мозку. Він робить один крок далі з Мак–Каллохом та Піттом, вводячи вагові коефіцієнти в рівняння. За допомогою IBM 704 Розенблатт зміг дозволити комп'ютеру навчитися розрізняти карти, позначені ліворуч, і карти, позначені праворуч.

1974: Хоча багато дослідників сприяли ідеї зворотного розповсюдження, Пол Вербос першим у США встановив її використання в нейронних мережах у своїй дисертації.

1989: Ян ЛеКун публікує статтю, яка ілюструє використання обмежень зворотного розповсюдження та їх інтеграцію в архітектуру нейронної мережі для навчання алгоритмів. У цьому дослідженні нейронна мережа була успішно використана для розпізнавання рукописних цифр поштового індексу, наданих Поштовою службою США.

Нейронні мережі, також відомі як штучні нейронні мережі (ANN) або модельовані нейронні мережі (SNN), є підмножиною машинного навчання і становлять серце алгоритмів глибокого навчання. Їх назва та структура натхненні людським мозку та імітують спосіб передачі біологічних нейронів сигналів один одному.

Штучні нейронні мережі (ANN) складаються з шарів вузлів, які містять вхідний шар, один або кілька прихованих шарів та вихідний шар. Кожен вузол або штучний нейрон підключається до іншого і має відповідну вагу та поріг. Коли вихід окремого вузла перевищує заданий поріг, цей вузол переходить в активний стан та передає інформацію на наступний шар мережі. В іншому випадку дані не передаватимуться на наступний рівень мережі.

Нейронні мережі покладаються на навчальні дані для навчання та підвищення їх точності з часом. Однак, як тільки алгоритми навчання точно налаштовані, вони стають потужними інструментами в галузі інформатики та штучного інтелекту, які дозволяють класифікувати та групувати дані з високою швидкістю (рисунок 2.1).

Завдання розпізнавання голосу або зображення можуть зайняти хвилини в порівнянні з годинами порівняно з ідентифікацією людей, проведеною вручну. Однією з найвідоміших нейронних мереж є алгоритм пошуку Google.

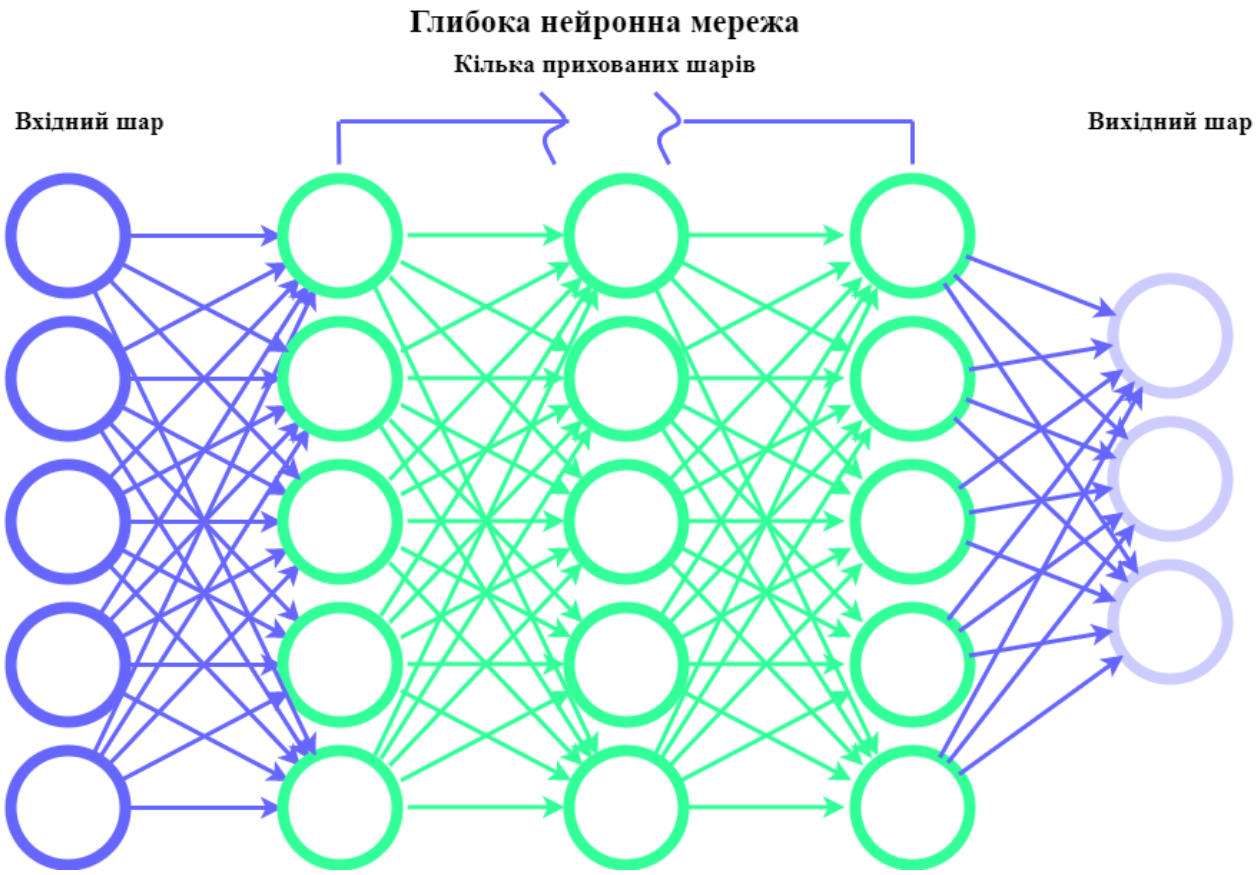

Рисунок 2.1 – Модель глибокої нейронної мережі

Але головне питання полягає у тому як працюють нейронні мережі?

Проаналізуємо кожний окремий вузол як про власну модель лінійної регресії, що складається з входів, ваг, спотворень (або порогових значень) та виходів. Формула виглядає приблизно так рисунок 2.2 –2.3:

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Рисунок 2.2 – Математична формула для визначення підсумовування

$$\text{output} = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases}$$

Рисунок 2.3 – Математична формула для визначення результату

Після визначення вхідної площини призначаються ваги. Ці ваги допомагають визначити важливість змінної, причому більші ваги вносять більший внесок у результат, ніж інші вхідні дані. Потім усі вхідні дані помножуються на їх відповідні ваги та додаються. Потім вихід виходить через функцію активації, яка визначає вихід. Коли вихід перевищує заздалегідь заданий поріг, вузол «спрацьовує» (або активується) і передає дані на наступний рівень мережі.

В результаті вихід одного вузла стає входом наступного вузла. Цей процес передачі даних з одного рівня на наступний визначає цю нейронну мережу як мережу прямого зв'язку.

Необхідно також розглянути, як може виглядати один вузол із двійковими значеннями. Маємо можливість застосувати це поняття до більш конкретного прикладу, наприклад: « Чи варто зайнятися серфінгом? (так: 1, ні: 0)».Рішення їхати чи не їхати – це прогнозований результат. Наприклад, припустимо, що є три фактори, які приймають рішення:

- хвилі хороші? (Так: 1, Ні: 0)
- берегова лінія порожня? (Так: 1, Ні: 0)
- чи був нещодавно напад акули? (Так: 0, Ні: 1)

Тоді припустимо наступне, що дає нам такий вхід:

X1 = 1, оскільки хвилі накочуються.

X2 = 0, бо немає натовпу.

X3 = 1, оскільки недавно не було нападу акул.

Тепер нам потрібно призначити деякі ваги, щоб визначити важливість. Вищі ваги означають, що певні змінні важливіші для рішення або результату.

W1 = 5, оскільки припливи рідкісні.

W2 = 2, як ви звикли до натовпу.

W3 = 4, бо ви боїтеся акул.

Нарешті, припустимо також, що поріг дорівнює 3, що означає значення зміщення –3. З усіх різних входів маємо можливість почати додавати значення у формулу, щоб отримати бажаний результат.

Результат = (1 * 5) + (0 * 2) + (1 * 4) – 3 = 6

Використовуючи функцію включення на початку цього розділу, маємо можливість визначити, що вихід цього вузла дорівнює 1, оскільки 6 більше ніж 0. У цьому випадку ви займаєтесь серфінгом.

Однак, якщо відрегулюємо ваги або поріг, маємо можливість отримати різні результати з моделі. Коли спостерігаємо рішення, як у наведеному вище прикладі, м маємо можливість побачити, як нейронна мережа може приймати дедалі складніші рішення в залежності від результатів попередніх рішень або рівнів.

У наведеному вище прикладі використовували перцептрони для ілюстрації деяких математичних принципів, але нейронні мережі використовують сигмоподібні нейрони, які відрізняються тим, що мають значення від 0 до 1. Оскільки нейронні мережі діють як дерева рішень, дані каскадуються від одного вузла до іншого.

Значення x від 0 до 1 зменшують вплив зміни змінної на вихід будь–якого вузла, а потім на вихід нейронної мережі.

Коли думаємо про більш практичні випадки використання нейронних мереж, такі як розпізнавання зображень або класифікація, використовуємо контрольоване навчання або позначені набори даних для навчання алгоритму. Навчаючи модель, хочеться оцінити її точність, використовуючи функцію витрат (або збитків). Його також зазвичай називають середньоквадратичною помилкою (MSE). У наступному рівнянні є:

- i – індекс вибірки,
- y–hat – прогнозований результат,
- y – фактичне значення i,
- m – кількість зразків.

$$Cost\ Function = MSE = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y)^2$$

Рисунок 2.4 – Математична формула для визначення функції витрат

Зрештою, мета – мінімізувати функцію витрат, щоб забезпечити правильну відповідність даному спостереженню. Оскільки модель регулює свої ваги та спотворення, вона використовує функцію витрат і вчиться досягати точки збіжності або місцевого мінімуму. Процес, за допомогою якого алгоритм регулює свої ваги, здійснюється за допомогою градієнтного спуску, що дозволяє моделі визначати напрямок руху для зменшення помилок (або мінімізації функції витрат).

З кожним прикладом навчання параметри моделі коригуються і поступово зближуються до мінімуму.

Більшість глибинних нейронних мереж є прямолінійними, тобто вони рухаються лише в одному напрямку – від входу до виходу. Однак ви також можете навчити свою модель, використовуючи зворотне розповсюдження, що означає рух для зворотного напрямку від виходу до входу. Зворотне розповсюдження дозволяє нам розрахувати і призначити помилку, пов'язану з кожним нейроном, що дозволяє нам точно налаштувати та налаштувати параметри моделей відповідно (рисунок 2.5).

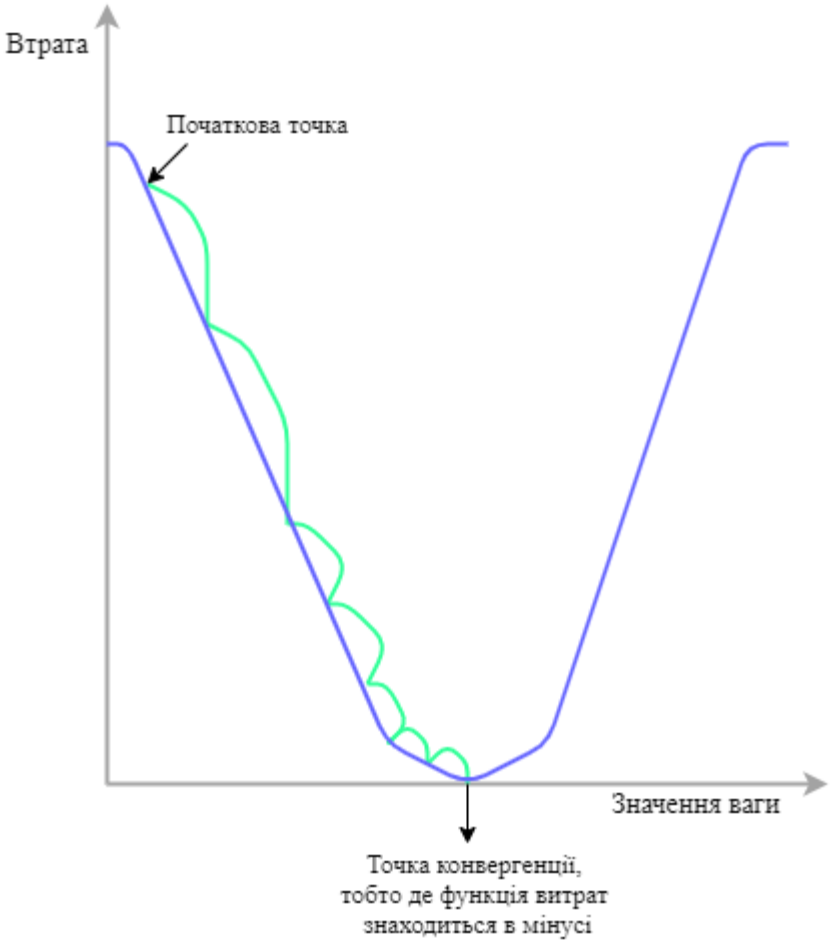

Рисунок 2.5 – Математична формула для визначення функції витрат

Нейронні мережі можна класифікувати на різні типи, які використовуються для різних цілей. Незважаючи на те, що це не вичерпний перелік типів, наведені нижче найпоширеніші типи нейронних мереж, з якими можна зіткнутися під час використання:

Перцептрон – найстаріша нейронна мережа, створена в 1958 році Френком Розенблаттом. Він має нейрон і є найпростішою формою нейронної мережі (рисунок 2.6):

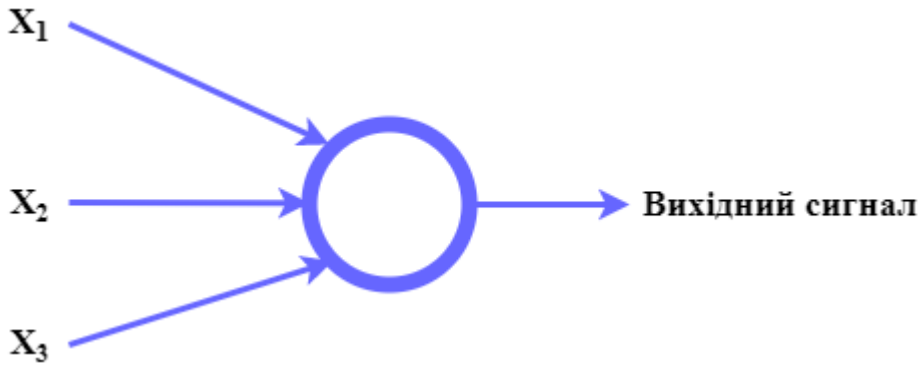

Рисунок 2.6 – Представлення схеми Перцептрона

Живі нейронні мережі або багатошарові перцептрони (MLP) – це те, на чому в основному зосередилися в цій статті. Вони складаються з вхідного рівня, прихованого рівня або рівнів та вихідного рівня. Хоча ці нейронні мережі також часто називають MLP, важливо зазначити, що вони насправді складаються з сигмоподібних нейронів, а не з перцептронів, оскільки більшість проблем реального світу є нелінійними. Дані зазвичай подаються в ці моделі для навчання і становлять основу для комп'ютерного зору, обробки природної мови та інших нейронних мереж.

Світові нейронні мережі (CNN) подібні до прямих мереж, але зазвичай використовуються для розпізнавання осіб, розпізнавання образів та / або комп'ютерного зору. Ці мережі використовують принципи лінійної алгебри, зокрема множення матриць, для виявлення закономірностей на зображенні.

Повторювані нейронні мережі (RNN) визначаються наявністю зворотного зв'язку. Ці алгоритми навчання в основному використовуються при використанні даних часових рядів для прогнозування майбутніх результатів, наприклад Б. прогноз фондового ринку або прогноз продажів.

Загалом, можна зробити висновок, що кожна архітектура нейронної мережі має свої особливі переваги завдяки теоретичним методам проектування та практичній значущості. Існують інші типи нейронних мереж, які можна використовувати для вирішення різних проблем у житті людини. Основною метою розробленої моделі є класифікація подій. Отже, мережа повинна бути класифікатором для деякого набору

даних. Найпростішим завданням класифікації нейронних мереж є перцептрон, тому в цій роботі використано одне з його понять.

Синтез моделі нейронної мережі базується на багатошаровому перцептроні Румельхарта (MLP) (особливий випадок перцептрона Розенблатта), який характерний тим, що вага нейрона регулюється алгоритмом зворотного поширення помилок. Насправді існування більше одного шару визначається як його характеристика.[24].

MLP продемонстрував здатність знаходити приблизні рішення надзвичайно складних проблем. Зокрема, вони є апроксиматорами загального призначення функцій, і тому їх успішно використовують для побудови регресійних моделей. Оскільки класифікацію можна розглядати як особливий випадок регресії, коли вихідною змінною є класифікація, класифікатор може бути побудований на основі MLP.

Пік популярності MLP у машинному навчанні з'явився у 1980–х роках, в таких областях як розпізнавання мови та зображень, системи машинного перекладу. Однак пізніше вони зіткнулися з конкуренцією з боку інших технік машинного навчання (таких як підтримка векторних машин). Завдяки досягненню глибокого навчання, інтерес до багатошарових перцептронів знову з'явився.

Ф. Розенлатт вперше запропонував багатошаровий персептрон. Однак багатошаровий персептрон був розроблений Д. Румельхартом у формі, що використовується зараз.

Перцептрон Руммелхарта відрізняється від перцептрона Розенблата наступними способами:

- використовує нелінійну функцію активації;
- кількість прихованих шарів більше одного (як правило, не більше трьох);
- вхідний сигнал не є двійковим, а кодується десятковими числами, нормованими на інтервал [0,1];
- визначення помилки вихідної мережі – це не кількість помилково ідентифікованих прикладів, а певне значення залишку;

- навчання полягає не в мінімізації помилок, а в стабілізації ваги мережі, щоб уникнути переобладнання.

Коли у відповідь на стимул виробляється більш ефективна реакція (оскільки будь–який тип реакції в датчику можна отримати заздалегідь), багатошаровий перцептрон має функціональну перевагу перед перцептроном Розенблатта. Це покращує узагальнюючу здатність, тобто здатність правильно реагувати на подразники, для яких перцептрон раніше не навчався.

Однак поки що такої загальної теореми в науковій літературі немає, насправді є лише дослідження різних стандартних тестів, що використовують для зіставлення різних типів архітектури.

Для виконання поставлених завдань буде використовуватися саме такий тип мережі, який матиме лише три шари, що включають вхідний, внутрішній та вихідний.

Нейронні мережі – це складні структури, що складаються зі штучних нейронів, які можуть отримувати кілька входів для отримання одного виходу. Це основна робота нейронної мережі – перетворення вхідних даних у значущі результати.

У нейронній мережі всі нейрони впливають один на одного і тому всі зв’язані між собою. Мережа може бачити та спостерігати кожен аспект набору даних та те, як різні дані можуть бути пов’язані між собою, а можуть і не. Через це нейронні мережі можуть знаходити надзвичайно складні закономірності у великих обсягах даних.

У нейронної мережі потік інформації відбувається двома способами:

– Мережі з прямою передачею: У цій моделі сигнали йдуть тільки в одному напрямку – до вихідного прошарку. Мережі з прямою передачею мають вхідний шар і один вихідний шар з нульовим або декількома прихованими шарами. Вони широко використовуються в розпізнаванні образів.

– Мережі зі зворотним зв'язком: У цій моделі рекурентні або інтерактивні мережі використовують свій внутрішній стан (пам'ять) для обробки послідовності вхідних сигналів. У них сигнали можуть проходити в обох напрямках через контури

(прихований шар / и) мережі. Вони зазвичай використовуються в задачах, пов'язаних з тимчасовими рядами і послідовністю.

У нейронної мережі процес навчання (або тренування) починається з поділу даних на три різних набору:

– навчальний набір даних – цей набір даних дозволяє нейронної мережі зрозуміти ваги між вузлами;

– перевірки набір даних – цей набір даних використовується для точного налаштування продуктивності нейронної мережі;

– тестовий набір даних – цей набір даних використовується для визначення точності і меж похибки нейронної мережі;

Після того як дані розділені на ці три частини, до них застосовуються алгоритми нейронної мережі для навчання нейронної мережі. Процедура, яка використовується для полегшення процесу навчання нейронної мережі, відома як оптимізація, а використовуваний алгоритм називається оптимізатором. Існують різні типи алгоритмів оптимізації, кожен з яких має свої унікальні характеристики і аспекти, такі як вимоги до пам'яті, точність обчислень і швидкість обробки.

Навчання нейронних мереж складне, оскільки ваги цих проміжних шарів тісно пов'язані. Отже, якщо ви злегка потягнете одне з з'єднань, ефект вплине не тільки на нейрон, що перетягується, але також поширюється на всі нейрони в наступних шарах і, отже, вплине на всі виходи. Через це неможливо досягти найкращого набору ваг, оптимізуючи по одній вазі за раз. Однак необхідно одночасно дослідити весь простір потенційних груп ваг. Звідси випливає концепція застосування випадкових ваг до відношень та багаторазової оцінки записів для досягнення майже оптимального результату.

Але наскільки ефективним є цей випадковий вибір ваг, адже насправді частка досить хорошого набору ваг величезна. Якщо хочемо застосувати випадкову грубу силу, нам слід уточнити оцінку, щоб отримати чудовий набір ваг.

Давайте розглянемо приклад, якщо мережа має 750 вхідних нейронів, 20 нейронів у прихованому шарі та 12 нейронів у вихідному шарі. Тоді кількість ваг можна розрахувати як 750 × 20 + 20 × 12 = 15 240 ваг. Звідси випливає, що потрібно враховувати 15 240 параметрів, якщо є помилки, то додавання цих параметрів до цього числа означає, що повинні зробити 1015 240 здогадок, що є величезною кількістю.

Цей принцип, виявлений у цьому прикладі, відомий як "прокляття виміру". Незалежно від того, наскільки малу розмірність додаємо до простору пошуку, це призводить до експоненціального збільшення кількості вибірок. Тому потрібно було розробити методи ефективного обчислення таких проблем.

Одним із способів буде лінійна регресія. Лінійна регресія – це робота з побудови "лінії, що найкраще підходить" у наборі точок даних, але існує також набагато більш надійний алгоритм, який надає нейромережам набагато більшу гнучкість, ніж модельні функції.

Метод градієнтного спуску є одним із інструментів парадигми для ітеративної оптимізації складних функцій, що містяться в задачі. Його можна отримати, визначивши поточний стан градієнта, а потім зменшивши градієнт, щоб мінімізувати функції втрат. Величина зміни параметрів вибирається для мінімізації помилки, і цей процес повторюється, поки не буде знайдено задовільний бал.

Кожен алгоритм має унікальні переваги та недоліки. Це лише декілька алгоритмів, що використовуються для тренування нейронних мереж, і їх функції демонструють лише вершину айсберга – у міру того, як фреймворки глибокого навчання прогресують, і функції цих алгоритмів будуть зростати.

Зазвичай, зворотні розподільні мережі складаються з двох етапів, таких як навчання та тестування. На етапі навчання для мережі є "доречними" вибіркове введення та правильна класифікація. Наприклад, на вході може бути шифроване зображення особи, а на виході – код, який співвідноситься імені особи.

Як і більшість алгоритмів навчання, нейронна мережа повинна закодовувати входи і виходи у відповідності до будь–якої заданої користувачем схемою. Шаблон

визначає архітектуру в мережі, тому, як тільки мережа пройде навчання, її не можна буде змінити без складання абсолютно іншої мережі.

При проходженні моделлю навчання та тестування для визначення її спроможності до синтезування, потрібно скласти алгоритм дій, що буде необхідний для виконання побудови моделі (рисунок 2.7).

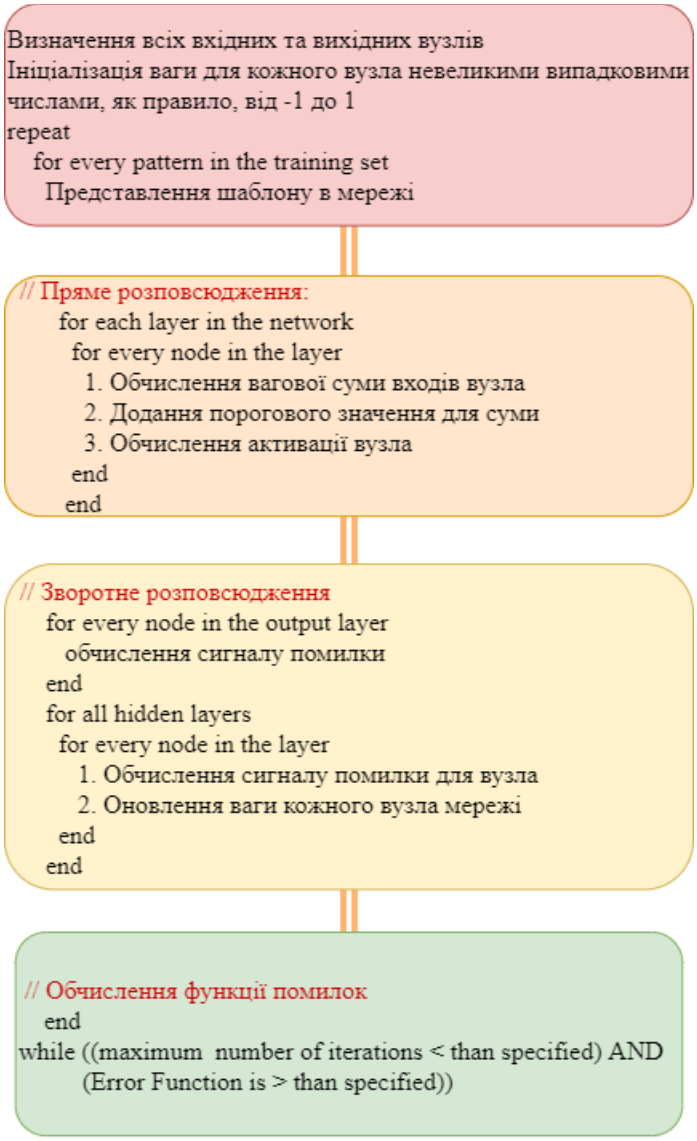

Рисунок 2.7 – Представлення алгоритму в схематичному вигляді

2.2Дані для моделі: синтез, підготовка, навчання

Необхідно провести попередній збір зразків даних для навчання, контролю та тестування, аби надалі синтезувати та проаналізувати модель класифікації інцидентів кібербезпеки.

В загалом модель процесу упорядкування даних показано на рисунку 2.8.

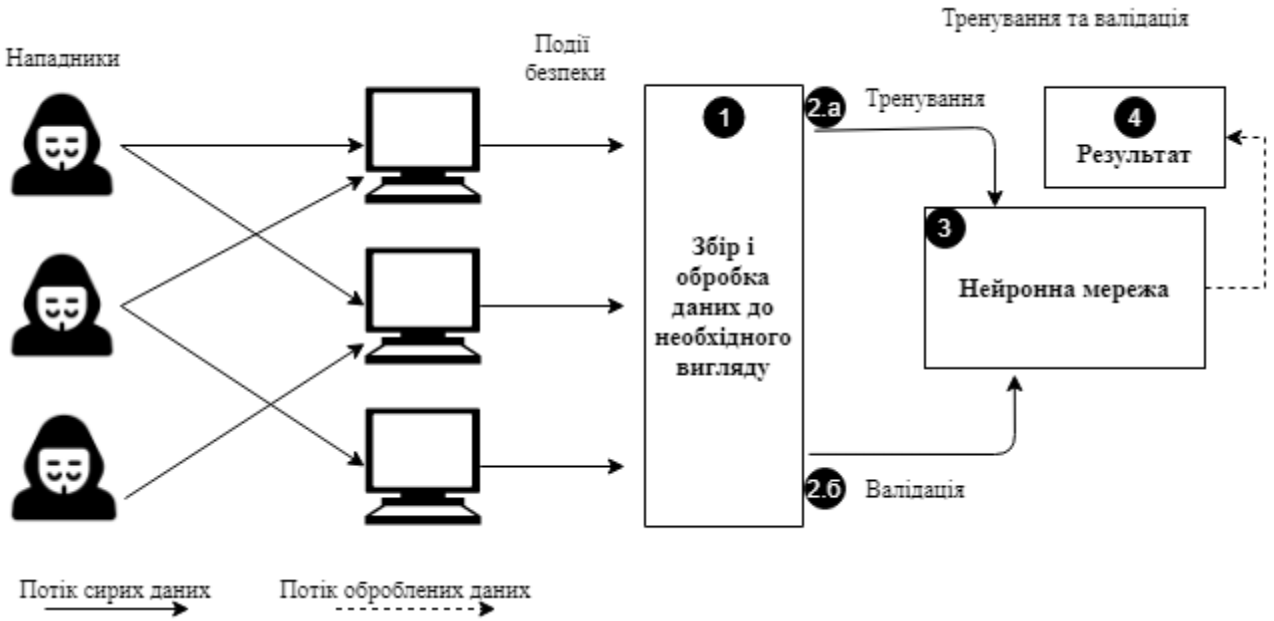

Рисунок 2.8 –Процес упорядкування даних

Процес оброблювання даних про стан системи складається з 4 головних етапів:

1) Збирає дані про мережеву активність, системні журнали та журнали подій.

2) Обробка минулих даних та їх перетворення в дані, що необхідні для подальшої обробки в логічній формі.

3) Навчання нейронної мережі та її тестування.

4) Аналіз отриманих результатів.

Ці кроки описують загальні підходи для розробки моделі класифікації подій, які будуть використовуватися в майбутньому. Щойно створена модель має власний погляд на кожному етапі і може суттєво відрізнятися при виявленні різних типів інцидентів. У

цьому розділі представлена модель, яка визначає різні типи мережевих атак шляхом аналізування реакції трафіку мережеві. На базі отриманих даних оцінюється потенційна можливість вдосконалення моделі для виявлення окремих інцидентів.

Щоб навчити та протестувати інтелектуальну модель, потрібно багато даних. Найкращим способом збирання даних про події кібербезпеки є використання єдиної направленої системи або скомпрометованої мережі та мережі зловмисника. Кінцевими результатами такого збирання є створення спеціального набору даних, що містить інформацію щодо мережевого трафіку, журналів подій, системних журналів і т.п.

В даному розділі в процесі моделювання використовувався набір даних, який містить шість сценарії атак. Мережевий сектор зловмисника налічує 40 станцій. Організація жертв має 6 відділів і складається з 410 робочих станцій користувачів та 25 серверів. Набір даних містить дані про перехвачений мережевий трафік та системні журнали для кожного комп'ютера. Також в ньому міститься приблизно 590 ГБ логічної інформації, яка доступна для подальшої обробки.

У наборі використовується концепція файлів конфігурації для систематичного генерування набору даних, який містить детальні описи вторгнень та абстрактні моделі розповсюдження для додатків нижчого рівня, протоколів або мережевих об'єктів. Через абстрактну природу сформованих файлів конфігурації є можливість застосовувати їх до різних мережевих протоколів з різною топологією. Це можете використовувати профілі разом для створення наборів даних, які відповідають конкретним потребам. Ми створимо два різні класи профілю.

Файл конфігурації B – профіля. Є можливість використовувати різні методи машинного навчання й статистичного аналізу (наприклад, K–Means, Random Forest, SVM та J48) для інкапсуляції поведінки користувачів. Особливою характеристикою інкапсуляції є розподіл розміру пакета протоколу, кількість пакетів на потік, конкретний шаблон корисного навантаження, розмір корисного навантаження та розподіл часу запиту протоколу.

У тестовому середовищі будуть змодельовані такі протоколи: HTTPS, HTTP, IMAP, SMTP, SSH, POP3 та FTP. За попередніми спостереженнями, більша частина трафіку надходить від HTTP та HTTPS.

Файл конфігурації M – профіля. При спробі чітко описати сценарій атаки. У простому випадку одна людина може інтерпретувати ці конфігураційні файли та виконати їх пізніше. В ідеалі для інтерпретації та виконання цих сценаріїв слід використовувати окремий агент разом із компілятором. Для було розглянуто шість сценаріїв атак (таблиця 2.1):

Проникнення зсередини мережі. В такому випадку надсилається шкідливі файли жертвам електронною поштою та використовуються вразливості додатків. Після успішного використання на комп'ютері жертви буде виконуватися операція, яка виконується в тилу, а потім буде використовуватися його комп'ютер для пошуку інших вразливих вікон у внутрішній мережі та використання їх, коли це можливо.

HTTP: відмова в обслуговуванні. В такому випадку використовується Slowloris та LOIC як головні інструменти. Доведено, що ці інструменти унеможливлюють доступ одного веб–сервера до іншого. Slowloris спочатку встановлює повне TCP–з'єднання з віддаленим сервером. Інструмент тримає з'єднання відкритим, регулярно надсилаючи на сервер діючі неповні HTTP–запити, щоб запобігти закриттю сокета. Оскільки кожен веб–сервер здатний обробляти підключення лише обмежено, це лише питання часу, коли всі сокети будуть використані, а інші зв'язки встановити не вдасться. Крім того, HOIC – це є ще однією відомою програмою, яка може запускати DoS–атаки на веб–сайтах.

Атаки на веб–додаток. В такому випадку використовується «Проклята й вразлива веб–програму (DVWA)», яка призначена для того, щоб допомогти професіоналам безпеки перевірити свої вміння. Це веб–додаток для жертв. Першим кроком є сканування веб–сайту за допомогою сканера вразливостей веб–додатків, а потім здійснюється різні види веб–атак на вразливі веб–сайти, включаючи введення SQL–ін'єкції, введення команд та необмежену кількість завантажень файлів.

Атаки грубої сили. Атаки грубої сили дуже поширені в мережах, оскільки вони, як правило, використовують слабкі комбінації імені користувача та пароля для проникнення в облікові записи. Метою остаточного рішення є отримання облікових записів SSH та MySQL шляхом виконання грубої сили на головному сервері.

Атака останніх оновлень. Це деякі атаки, засновані на відомих уразливостях, які мають можливість бути виконані протягом певного проміжку часу (ці аномальні вразливості іноді зачіпають мільйони серверів та жертв і, зазвичай, тривають місяці. Heartbleed є одним із найвідоміших комп'ютерів за останні роки для виправлення всіх вразливих місць (серед усіх комп'ютерів у всьому світі).

Таблиця 2.1 – Список атак і їх тривалість

Атака	Інструменти	Тривалість	Нападник	Жертва
Атаки грубої сили	FTP та SSH – patator.	1 день	Kali linux	Ubuntu 16.4 (Web Server)
DoS атака	GoldenEye, Hulk, Slowloris Slowhttptest.	1 день	Kali linux	Ubuntu 16.4 (Apache)
DoS атака	Heartleech	1 день	Kali linux	Ubuntu 12.04 (Open SSL)
Веб–атака	«Проклята й вразлива веб–програма (DVWA)». Власний фреймворк selenium (XSS та груба сила).	2 дні	Kali linux	Ubuntu 16.4 (Web Server)
Атака з проникненням	Перший рівень: завантаження Dropbox на машині windows. Другий рівень: Nmap та сканування портів.	2 дні	Kali linux	Windows Vista and Macintosh
Ботнет–атака	Ares (розроблений Python): видалення оболонки, завантаження/вивантаження, перехват. Скріншоти та реєстрація натискання клавіш.	1 день	Kali linux	Windows Vista, 7, 8.1, 10 (32–bit) and 10 (64–bit)

Продовження таблиці 2.1

DDoS та сканування портів	Низькоорбітальний іонний канал (LOIC) для запитів UDP, TCP або HTTP.	2 дні	Kali linux	Windows Vista, 7, 8.1, 10 (32–bit) and 10 (64–bit)

Підготовка даних для навчання розпочинається з аналізу мережевих атак, які можна розглянути з різних сторін:

– аналіз діяльності в мережі;

– аналіз пакетів даних та їх вмісту.

Другий аналіз є складнішим процесом і його розгляд буде присутній в подальшій роботі.

Активність мережі аналізувалася за допомогою ПЗ CICFlowMeter.

CICFlowMeter – це генератор мережевого трафіку, написаний мовою Java, що забезпечує більшу гнучкість у виборі обчислюваних об'єктів, додавання нових функцій та кращий контроль тривалості часу очікування потоку. Він створює двонаправлений потік (Biflow), де першим пакетом визначається прямий (від джерела до пункту призначення) та зворотній (від пункту призначення до джерела) напрямки, тому існує 83 статистичні характеристики, такі як тривалість, кількість байтів, кількість пакетів, довжина пакета і т.п. Також обчислюються у прямому та зворотному напрямках відповідно.

Вихідні дані у форматі файлу CSV, і кожен потік має шість стовпців тегів, а саме FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort та протоколи з більш ніж 80 характеристиками мережевого трафіку. Зазвичай потік TCP припиняється, коли з'єднання від'єднується (через пакет FIN), а потік UDP припиняється через час очікування потоку. Значення часу очікування потоку можна довільно призначити за окремою схемою, наприклад, 600 секунд на TCP та UDP. CICFlowMeter–V3 може витягувати більше 80 функцій, ці функції перелічені в наступній таблиці (для зручності

призначимо для максимальних значень код –MAX,мінімальний – MIN та середній AVG; відповідно до їх англійського аналогу).

Таблиця 2.2 – Список вилучених функцій в мережевому трафіку

Назва характеристики	Опис
FL_DUR	Довжина потоку
TOT_FW_PK	Кількість переадресованих пакетів
TOT_BW_PK	Кількість пакетів для зворотного напрямку
TOT_L_FW_PKT	Повний розмір переадресованого пакету
FW_PKT_L_MAX	MAX розмір пакета в прямому напрямку
FW_PKT_L_MIN	MIN розмір пакета в прямому напрямку
FW_PKT_L_AVG	AVG розмір пакета в прямому напрямку
FW_PKT_L_STD	Звичайна похибка розміру пакета в прямому напрямку
BW_PKT_L_MAX	MAX розмір пакета для зворотного напрямку
BW_PKT_L_MIN	MIN розмір пакета для зворотного напрямку
BW_PKT_L_AVG	AVG розмір пакета для зворотного напрямку
BW_PKT_L_STD	Звичайна похибка розміру пакета
FL_BYT_S	Швидкість потоку байтів (в секунду)
FL_PKT_S	Швидкість потоку пакетів (в секунду)
FL_IAT_AVG	AVG час між двома потоками
FL_IAT_STD	Звичайний час похибки між двома потоками
FL_IAT_MAX	MAX час між двома потоками

Продовження таблиці 2.2

FL_IAT_MIN	MIN час між двома потоками
FW_IAT_TOT	Повний час між двома відправленими пакетами
FW_IAT_AVG	AVG час між двома відправленими пакетами
FW_IAT_STD	Звичайна похибка між двома відправленими пакетами
FW_IAT_MAX	MAX час між відправленням двох пакетів у прямому напрямку
FW_IAT_MIN	MIN час між відправленням двох пакетів у прямому напрямку
BW_IAT_TOT	Повний час між двома пакетами, відправленими для зворотного напрямку
BW_IAT_AVG	AVG час між двома пакетами, відправленими для зворотного напрямку
BW_IAT_STD	Звичайна похибка між двома пакетами, відправленими для зворотного напрямку
BW_IAT_MAX	MAX час між відправленням двох пакетів для зворотного напрямку
BW_IAT_MIN	MIN час між відправленням двох пакетів для зворотного напрямку
FW_PSH_FLAG	Кількість встановлення прапора PSH пакетів у прямому напрямку (для UDP – 0)
BW_PSH_FLAG	Кількість встановлення прапора PSH для зворотного напрямку (UDP дорівнює 0)
FW_URG_FLAG	Кількість встановлення прапора URG у прямому напрямку (UDP дорівнює 0)
BW_URG_FLAG	Кількість встановлення прапора URG для зворотного напрямку (UDP дорівнює 0)
FW_HDR_LEN	Кількість байтів, використаних для заголовка у прямому напрямку
BW_HDR_LEN	Кількість байтів, використаних для заголовка для зворотного напрямку
FW_PKT_S	Швидкість пересилання пакетів у прямому напрямку на секунду
BW_PKT_S	Швидкість пересилання пакетів для зворотного напрямку на секунду
PKT_LEN_MIN	MIN величина потоку

Продовження таблиці 2.2

PKT_LEN_MAX	MAX величина потоку
PKT_LEN_AVG	AVG величина потоку
PKT_LEN_STD	Звичайна похибка потоку
PKT_LEN_VA	MIN час між надходженням пакету
FIN_CNT	Кількість пакетів FIN
SYN_CNT	Кількість пакетів SYN
RST_CNT	Кількість RST–пакетів
PST_CNT	Кількість пакетів Push
ACK_CNT	Кількість пакетів ACK
URG_CNT	Кількість пакетів URG
CWE_CNT	Кількість пакетів CWE
ECE_CNT	Кількість пакетів ЄЕК
DOWN_UP_RATIO	Коефіцієнт завантаження та розвантаження
PKT_SIZE_AVG	AVG розмір пакету
FW_SEG_AVG	AVG розмір пакету у прямому напрямку
BW_SEG_AVG	AVG розмір пакету для зворотного напрямку
FW_BYT_BLK_AVG	AVG кількість байтів у прямому напрямку
FW_PKT_BLK_AVG	AVG показник якості переадресації у прямому напрямку
FW_BLK_RATE_AVG	AVG кількість об’ємної норми пакету в прямому напрямку
BW_BYT_BLK_AVG	AVG кількість байтів для зворотного напрямку

Продовження таблиці 2.2

BW_PKT_BLK_AVG	AVG кількість пакетів зворотної швидкості потоку
BW_BLK_RATE_AVG	AVG кількість об'ємної норми пакету для зворотного напрямку
SUBFL_FW_PK	AVG кількість пакетів для кожного в прямому напрямку
SUBFL_FW_BYT	AVG кількість байтів для кожного в прямому напрямку
SUBFL_BW_PKT	AVG кількість пакетів на субпотоці для зворотного напрямку
SUBFL_BW_BYT	AVG кількість байтів у протилежному напрямку в субпотоці
FW_WIN_BYT	Кількість байтів, відправлених в прямому напрямку
BW_WIN_BYT	# Байти, відправлені для зворотного напрямку
FW_ACT_PKT	# Пакет містить принаймні 1 байт корисного навантаження даних TCP (вперед)
FW_SEG_MIN	MIN розмір сегмента в прямому напрямку
ATV_AVG	AVG час активного потоку, перш ніж стане вільним
ATV_STD	Звичайна похибка часу активного потоку, перш ніж стане вільним
ATV_MAX	MAX час активного потоку, перш ніж стане вільним
ATV_MIN	MIN час активного потоку, перш ніж стане вільним
IDL_AVG	AVG час простою, перш ніж потік стане активним
IDL_STD	Звичайна похибка часу простою, перш ніж потік стане активним
IDL_MAX	MAX час простою, перш ніж потік стане активним
IDL_MIN	MIN час простою, перш ніж потік стане активним

Після створення набору кожен потік був позначений відповідно до інциденту, який стався під час потоку, або позитивно, якщо атака не відбулася (звичайне користування служб).

До останнього запису додано атрибут Label, який вказує, що потік відповідає певній атаці або плюсовому трафіку. Дані будуть позначені відповідно до значень, що із себе представляє атака: FTP–BruteForce, Benign, DoS–GoldenEye, SQL Injection, DDoS–LOIC–UDP, SSH–Bruteforce, DoS–Slowloris, DoS–Hulk, DDoS–Attacks–LOIC–HTTP, DoS–SlowHTTPTest, DDOS – HOIC, Brute Force, Brute Force – XSS, Infiltration, Bot.

Підготовка моделі даних навчання розпочинається із синтезу моделі нейронної мережі, яка проводився на основі багатошарового персептрона Румельхарта.

Багатошаровий перцептрон (MLP) – це глибока штучна нейронна мережа. Вона складається з більш ніж одного перцептрону. Вони складаються з вхідного шару для прийому сигналу, вихідного шару, який приймає рішення або робить прогноз щодо вхідного сигналу, і між цими двома шарами – довільну кількість прихованих шарів, які є справжнім обчислювальним двигуном MLP. MLP з одним прихованим шаром здатні апроксимувати будь–яку безперервну функцію.

Багатошарові перцептрони часто застосовуються для вирішення завдань контрольованого навчання: вони навчаються на наборі пар вхід–вихід і вчяться моделювати кореляцію (або залежність) між цими входами і виходами. Навчання включає в себе настройку параметрів, або терезів і зсувів, моделі з метою мінімізації помилки. Для коригування ваг і зміщень щодо помилки використовується метод зворотного поширення, а сама помилка може бути виміряна різними способами, в тому числі середньоквадратичної помилкою (RMSE).

Нейронні мережі з прямим зв’язком, такі як MLP, схожі на теніс або пінг–понг. Вони в основному беруть участь в двох рухах – постійному русі вперед–назад. Можна уявити цей пінг–понг з припущень і відповідей як свого роду прискорену науку, оскільки кожна думка – це перевірка того, що, як нам здається, ми знаємо, а кожен

відповідь – це зворотний зв'язок, що дозволяє нам зрозуміти, наскільки ми помиляємося. Приклад даного алгоритму представлений у спеціальному ПЗ Weka.

Weka – це додаток з відкритим вихідним кодом, що використовується для аналізу даних, випущений під загальною публічною ліцензією GNU, що містить набір алгоритмів машинного навчання для завдань з інтелектуального аналізу даних розроблене в Університеті Вайкато в Гамільтоні, Нова Зеландія. Він включає інструменти для таких функцій як класифікації, підготовки даних, кластеризації, відображення правил вилучення, регресії, та візуалізації. Це зручний інструмент, який можна використовувати для розробки моделей, здатних відповісти на вищевказані питання при правильному застосуванні.

Машинне навчання – це підмножина ШІ, в якому комп'ютери навчаються приймати рішення на основі наданих наборів даних при обмеженому втручанні людини. Ідея полягає в тому, щоб комп'ютери постійно вдосконалювали цей процес в автономному режимі. Чим більше даних надходить в модель машинного навчання, поряд з ітераційним поліпшеннями на основі результатів тестування, тим точніше виходять висновки. Вхідні дані можуть надходити в модель з декількох різних типів джерел, таких як файли CSV, аудіо, зображення, реляційні бази даних, відео та текстові формати. Точний алгоритм машинного навчання може відрізняти собак від кішок, розпізнавати перешкоди для самохідних автомобілів, визначати потенційні нові продукти для клієнтів на основі їх історії покупок, рекомендувати фільми на платформах потокового відео або вирішувати, чи стосується слово суддя до члену суду, який приймає ключові юридичні рішення, або до акту винесення оцінок (суддівства).

Дисципліну машинного навчання можна розділити на дві категорії: контрольоване і неконтрольоване [14].

На етапі навчання в контрольованому навчанні зростаюча модель споживає велику кількість мічених даних. Керована модель може виконувати такі оцінки, як ідентифікація букв алфавіту, написаних від руки (розрізнення скоропису і друкованого

шрифту), або визначення того, чи є настрій, що лежить в основі письмового тексту, позитивним або негативним.

Непідконтрольне навчання, з іншого боку, не залежить від мічених даних, а шукає схожості всередині даних, а потім розділяє їх на окремі категорії. Наприклад, Google може використовувати неконтрольоване навчання для угруповання результатів пошуку на основі подібності між окремими сторінками, а додатки для оренди нерухомості, такі як Airbnb, можуть використовувати неконтрольоване навчання для класифікації об'єктів оренди, розміщених на їх сервісі, за ціною, рейтингу або доступності.

Weka містить API (інтерфейс прикладного програмування), написаний на Java, який реалізує вже існуючі види алгоритмів для навчання нейромережевих моделей. Залишається охарактеризувати і сформувати сам процес моделювання і підготувати необхідні дані для навчання. Для опису навчання і тестування моделі було створено спеціальне програмне забезпечення. Логіку моделі є можливість представити в схематичному вигляді (таблиця 2.3) [15].

Алгоритм для початку навчання нейронної мережі представляє із себе виконання деяких функцій які будуть представлені далі, адже вони є незамінимою частиною для роботи з Weka та й для запуску роботи загалом, тому необхідно розібрати кожний крод для побудови.

Початковий момент представляє із себе ініціалізацію класової моделі, за рахунок створення базового класу генерації моделі (ModelGenerator), що включає функції для завантаження інформації до оперативної пам’яті для подальшої обробки, ініціалізації класів нейромережі, тестування та зберігання моделей для майбутнього використання.

Наступним відбувається завантаження представленого вище набору даних до оперативної пам’яті за допомогою функції (loadDataset), що формує інтерфейс для завантаження початкових даних з раніше підготовленого файлу до формату, що використовує Weka –.arff. Після ініціалізується клас описаний раніше як – Instances.

На порядку далі йде визначення значень, а саме переведення номінальних в числові, так як нейронні мережі являють значення з певними залежностями, і ці значення регулюють ваги значень. При нормалізації даних взаємозв'язку між деякими атрибутами можуть бути невірно витлумачені і спотворені процесом навчання. Для особливо критичних атрибутів числового виду, наприклад, такі як визначення портів від 1 до 65565, необхідно змінити вид представлення. Щоб вирішити цю проблему, необхідно використовувати формат номінального значення.

При нормалізації даних в нейронних мережах відбувається процес, а саме оптимізація значень наборів даних (від великого діапазону до діапазону значень між 0 і 1) для числових видів при збереженні пропорційності. Нормовані значення можуть показово підвищити швидкість процесу навчання моделі без порушення коректності навчання.

Розподіл навчальних та тестових зразків відбувається за рахунок поділу вибірки, що необхідно при первинній перевірці адекватності моделі. У даній роботі навчальний набір становить 90% від загальної множини переданих навчальних даних .

Наступний крок являється процесом навчання нейромережі за рахунок спеціального класу MultilayerPerceptron. Даний клас є одним із класів Weka API, який відображає математичну архітектуру нейромережі на основі багатошарового перцептрона. Також для такого випадку необхідна функція buildClassifier, яка ініціалізує приклад класу нейрмережі із необхідними властивостями і виконує його навчання.

Далі відбувається етап важливий для моделі, а саме первинне тестування через функцію evaluateModel, що класифікує дані за допомогою тестового набору, а потім порівнює отримані значення з шаблоном. На основі вдалих зрівнянь формуються значення помилок навчання [16].

Нижче наведено рисунок 2.9, що представляє собою реалізований код моделі для класифікації подій кібербезпеки на основі аналізу мережевих властивостей, але саме та частина, що відповідає за навчального модуля моделі (Java).

```
import weka.core.Debug;
import weka.core.Instances;
import weka.filters.Filter;
import weka.filters.unsupervised.attribute.Normalize;
import weka.filters.unsupervised.attribute.NumericToNominal;
public class Test {
    public static final String DATASETPATH = "./data/test.arff";
    public static final String MODElPATH = "./data/model.bin";
    public static void main(String[] args) throws Exception {
        ModelGenerator mg = new ModelGenerator();
        Instances dataset = mg.loadDataset(DATASETPATH);
        // set Numeric Port and Protocol to Nominal value
       Filter numToNominalFilter = new NumericToNominal();
        String[] options = new String[2];
        options[0] = "-R";
        options[1] = "2";
        numToNominalFilter.setOptions(options);
        numToNominalFilter.setInputFormat(dataset);
        dataset = Filter.useFilter(dataset, numToNominalFilter);
        Filter filter = new Normalize();
       // розділення 80% даних для тренування and 20% для тестування
        int trainSize = (int) Math.round(dataset.numInstances() * 0.8);
        int testSize = dataset.numInstances() - trainSize;
        dataset.randomize(new Debug.Random(1)); // перемішуємо дані для отримання більш
якісного співвідношення тренування / тестування
        // нормалізація числових даних в наборі
        filter.setInputFormat(dataset);
        Instances datasetnor = Filter.useFilter(dataset, filter);
        Instances traindataset = new Instances(datasetnor, 0, trainSize);
        Instances testdataset = new Instances(datasetnor, trainSize, testSize);
        // створюємо класифікатор на основі набору даних
     MultilayerPerceptron ann = (MultilayerPerceptron) mg.buildClassifier(traindataset);
        // Оцінюємо класифікатор набором тестів
        String evalsummary = mg.evaluateModel(ann, traindataset, testdataset);
        System.out.println("Evaluation: " + evalsummary);
        // Save model
        mg.saveModel(ann, MODElPATH);
    }
}
```

Рисунок 2.9 – Уривок модуля навчальної моделі

Етап навчання моделі починається за причин великого недоліка цього алгоритму, а саме є неможливість постійного навчання [17]. Для того, щоб ідентифікувати подію з певною точністю, відносна похибка не повинна перевищувати 4%.

Через великий обсяг даних, необхідних для навчання, останніх булл розділено на блоки залежно від типу інциденту, і було вирішено синтезувати окрему нейромережеву модель з ціллю класифікувати кожний тип атаки [18].

Початковий етап дослідження характерний проведенням синтезу моделі класифікації атак грубої сили FTP та SSH на базі спеціального набору із даних, що представлений на рисунку 2.10.

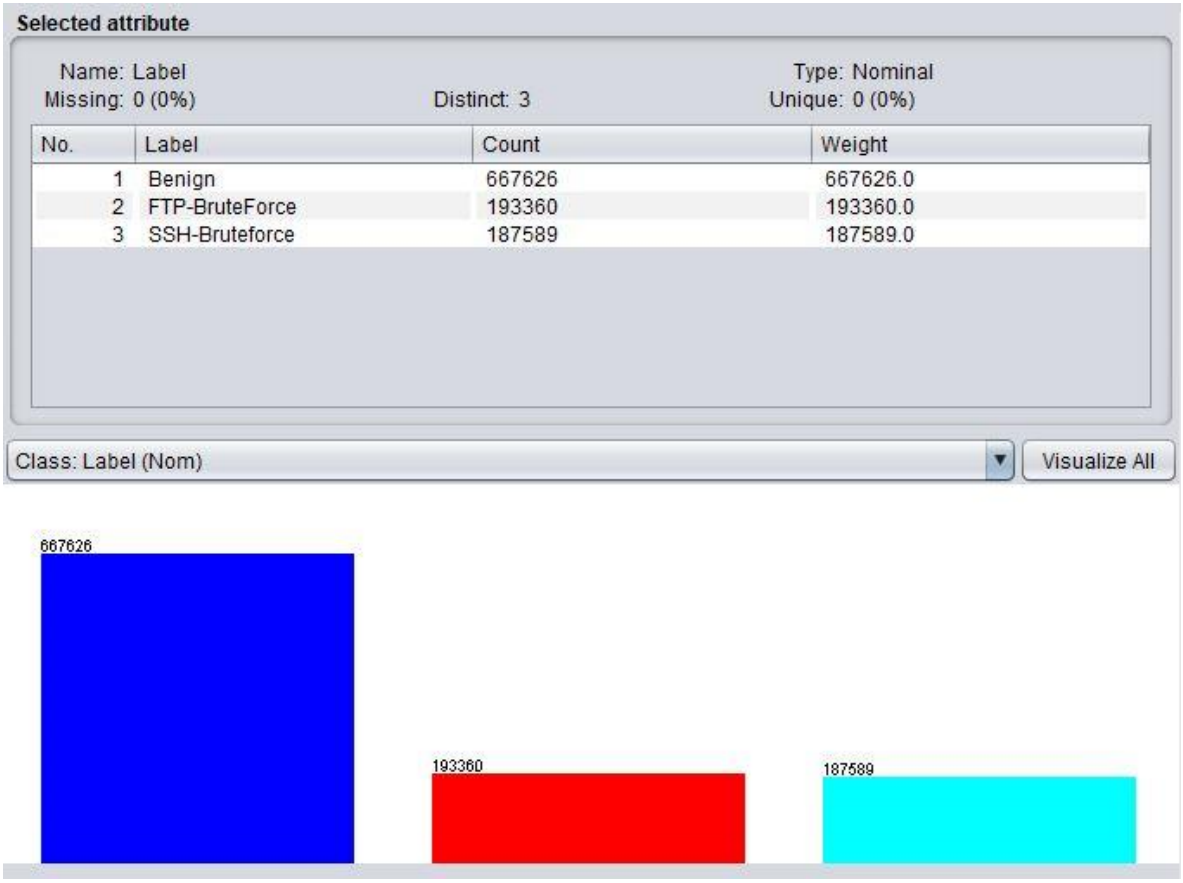

Рисунок 2.10 – Набір даних для навчання

Ініціалізація алгоритму нейронної мережі з наявністю спеціальних параметрів представлена на рисунку 2.11.

```
public Classifier buildClassifier(Instances traindataset) {
        MultilayerPerceptron m = new MultilayerPerceptron();
        // m.setGUI(true);
        // m.setValidationSetSize(0);
        // m.setBatchSize("100");
        // m.setLearningRate(0.3);
        // m.setSeed(0);
        // m.setMomentum(0.2);
        m.setTrainingTime(50);// тривалість навчання (кількість епох)
        // m.setNormalizeAttributes(true);
        //m.setHiddenLayers("2,3,3") три приховані шари з 2 вузлами в першому шарі, 3 вузла в другому і 3 вузли в третьому.
        try {
            m.buildClassifier(traindataset);
        } catch (Exception ex) {
           Logger.getLogger(ModelGenerator.class.getName()).log(Level.SEVERE, null, ex);
        }
        return m;
```

Рисунок 2.11 – Функція ініціалізації алгоритму

Виходячи з даних зображення необхідно виділити особливо важливі моменти, які необхідні для кращого розуміння поставлених значень, які будуть надалі представлені у вигляді висновків [19]:

- швидкість навчання повинна бути встановлена в діапазоні від 0 до 1, за бай дефолт це 0,3;
- коефіцієнт моменту повинен бути встановлений в діапазоні від 0 до 1, бай дефолт це 0,2;
- кількість навчених періодів бай дефолт це 500. 50 використовується по причині надмірності даних;
- відсотковий розмір перевірочного набору, використовуваного для виходу, повинен бути від 0 до 100, бай дефолт це 0;
- значення, що використовується для створення генератором випадкових чисел, має бути більше або дорівнює 0 і менше long;

- число прихованих шарів має являти собою список натуральних чисел або літер "a" = (атрибут + клас) / 2 через кому, "i" = атрибут, "o" = клас, і "t" = атрибут + клас. Значення бай дефолт це "a";
- передбачення за замовчуванням вимагає розмір пакета = 100;
- для впевненості в точності розрахунків моделі вихідний набір ділиться на менші набори: 100000, 500000 та 1000000 відповідно.

Результати навчання моделі для набору даних з 100000 записами показані на малюнку 2.12.

```
Evaluation: Summary
Correctly Classified Instances       19893               99.465  %
Incorrectly Classified Instances       107                0.535  %
Kappa statistic                          0.9889
K&B Relative Info Score            1927505.1563 %
K&B Information Score                23721.0376 bits      1.1861 bits/instance
Class complexity | order 0           24377.6345 bits      1.2189 bits/instance
Class complexity | scheme             1035.0185 bits      0.0518 bits/instance
Complexity improvement     (Sf)      23342.616  bits      1.1671 bits/instance
Mean absolute error                      0.0118
Root mean squared error                  0.0606
Relative absolute error                  3.6326 %
Root relative squared error             15.0765 %
Total Number of Instances            20000

Detailed Accuracy By Class
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,003    0,999      1,000    0,999      0,998    0,998     0,996     SSH-Bruteforce
                 0,971    0,000    1,000      0,971    0,985      0,982    0,994     0,987     Benign
                 1,000    0,005    0,969      1,000    0,984      0,982    0,996     0,957     FTP-BruteForce
Weighted Avg.    0,995    0,002    0,995      0,995    0,995      0,993    0,997     0,989

Confusion Matrix
     a     b     c   <-- classified as
 13625     0     5 |     a = SSH-Bruteforce
    16  3425    86 |     b = Benign
     0     0  2843 |     c = FTP-BruteForce
```

Рисунок 0.12. Представлення процесу навчання моделі

Завдяки програмі було створено файл, що має розширення .bin. Середнім значенням проміжку часу генерації моделі на відповідний набір даних – 6,5 хвилин. Цей файл є необхідним для використання його для класифікації вистежених нових даних.

Для виконання класифікації інцидентів із допомогою створення синтезованої моделі був згенерований метод створений спеціальний метод в роботу якого закладено особливий алгоритм дій.

Важливість алгоритму та його роботи представляє із себе особливе використання нейромережі при цьому повторення виконаних дій, таких як ініціалізація моделі, завантаження даних, визначення номіналів та у відзначенні похибки, що можливі при роботі моделі [20].

Для представлення роботи новосформованої моделі були обрані окремо 15 особливих записів: 5–Benign (стандартний трафік), 5–SSH Brute Force потоків та 5–FTP Brute Forcei відповідно.

За замовчуванням всі дані спеціально позначаються як FTP Brute Force attacks, як показано нижче (рисунок 2.13).

```
@data
21,6,1,1,1,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,40,20,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,
21,6,2,1,1,0,0,0,0,0,0,0,0,0,2,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,20,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,
21,6,1,1,1,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,20,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,
21,6,2,1,1,0,0,0,0,0,0,0,0,0,2,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,20,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,
21,6,1,1,1,0,0,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,40,20,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,1,
67,17,685,1,1,300,329,300,300,300,0,329,329,329,0,685,0,685,685,0,0,0,0,0,0,0,0,0,0,0,0,0,8,8,300,329,309
0,0,112640738,3,0,0,0,0,0,0,0,0,0,0,56320369,94.752309,56320436,56320302,112640738,56320369,94.752309,563
0,0,112640743,3,0,0,0,0,0,0,0,0,0,0,56320371.5,13.435029,56320381,56320362,112640743,56320371.5,13.435029
0,0,112640627,3,0,0,0,0,0,0,0,0,0,0,56320313.5,4401.739713,56323426,56317201,112640627,56320313.5,4401.73
0,0,112640733,3,0,0,0,0,0,0,0,0,0,0,56320366.5,51.618795,56320403,56320330,112640733,56320366.5,51.618795
22,6,555650,22,22,1928,2665,640,0,87.636364,137.780552,976,0,121.136364,258.64156,12922.093023,38554.860045
22,6,560585,22,22,1944,2665,640,0,88.363636,137.869001,976,0,121.136364,258.64156,13036.860465,48855.286144
22,6,530547,22,20,1928,2665,640,0,87.636364,137.780552,976,0,133.25,268.771254,12940.170732,45365.27566,271
22,6,533609,22,21,1960,2665,640,0,89.090909,137.953377,976,0,126.904762,263.574639,12704.97619,42668.192189
22,6,565469,22,21,1944,2665,640,0,88.363636,137.869001,976,0,126.904762,263.574639,13463.547619,45697.10508
```

Рисунок 0.13 – Дані, що використовуються для тестування моделі та класифікації атак

При початку роботі класифікатора потоки даних відповідають атакам, як представлено на рисунку 2.14.

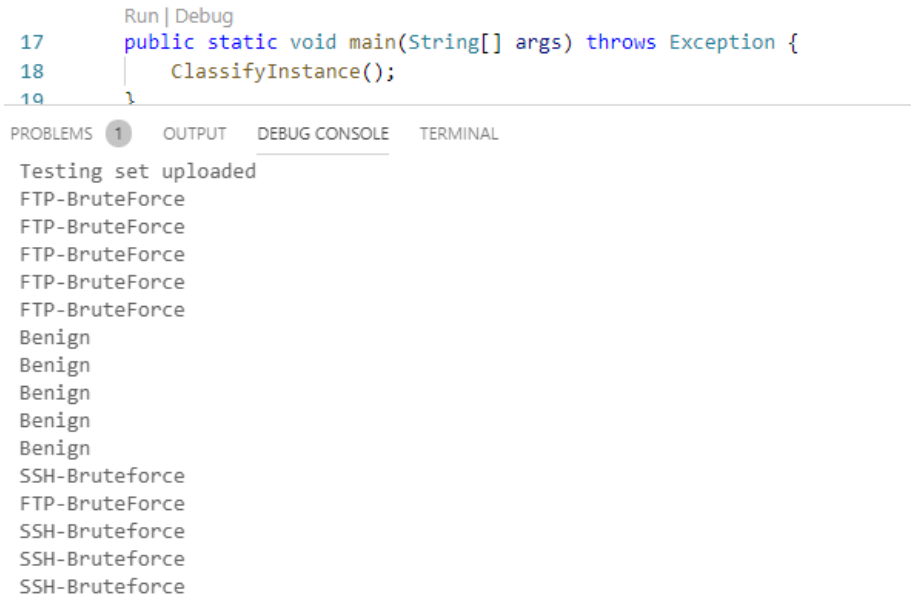

Рисунок 0.14. Екземпляр класифікації атак

Тільки одна атака показала помилки, які не SSH–Bruteforce–FTP Bruteforce. Демонстрація ілюструє роботу трафіку мережі, але не представляє реальну життєздатність. Щоб переконатися, що він працює належним чином. він був протестований на великих наборах додаткових даних і інших видах інцидентів [21].

Для перевірки моделі на адекватність використовуються два набора додаткових даних кожного типу. Після початку роботи класифікатора було отримано наступну інформацію про атаку SSH, FTP–Bruteforce для групи 1 до 100000, як показано на рисунку 2.15.

```
Benign: [90, 0, 0]
FTP-BruteForce: [0, 48340, 0]
SSH-Bruteforce: [0, 0, 46897]
Relative error: 0.0%
```

Рисунок 2.15. Приклад процесу тестування

Конкретна інформація по кожному з типів атак і результатах навчання наведена у таблиці 2.3.

Таблиця 0.3 – Результати навчальної моделі

Назва атаки	Навчальна група	Тестова група 1	Тестова група 2	Відносне відхилення, %
FTP–Bruteforce	96680	48340 / 48340	48200 / 48339	0.15
SSH–Bruteforce	93795	46897 / 46897	46801 / 46896	0.1
Dos–атака GoldenEye	20754	197 / 10378	330 / 10378	96.4
Dos–атака Slowloris	5496	2 / 2749	0 / 2747	9.95
Dos–атака LOIC–UDP	865	0 / 432	0 / 431	100
Dos–атака HOIC	343005	171504 / 171504	171504/171504	0
Dos–атака HULK	230954	115477/115477	115477/115477	0
HTTP благотворний	172012	18224 / 86005	15466 / 86004	80.3
Інфільтрація	45474	0 / 22737	0 / 22737	100
Ботнет	140816	70406 / 70408	70404 / 70407	0.003
Inf–Bot благотворний	469802	234901 / 234901	234901 / 234901	0

2.3Висновки за розділом 2

В другому розділі було описані основні види архітектури нейронної мережі, що почала своє формування з двадцятого століття і продовжує у сьогоднішніх днях. Основна відмінність між кожною із запропонованих прикладів архітектури полягає в концептуальній задачі, що вони повинні виконувати. В роботі було вирішено використовувати саме багатошаровий перцептрон Румельхарта як приклад архітектури

нейронної мережі на його основі, основним завданням якої є включення патернів вхідного мережевого трафіку в якості типу атаки.

Для створення шаблонів наборів даних використовувалося ПЗ CICFlowMeter V3 та інше програмне забезпечення, розроблене на мові програмування Python. Як джерела наборів даних використовувалися опубліковані набори даних про мережеві атаки.

Описана концепція проектування нейронних мереж і відображення результатів її опрацювання. Відповідно до таблиці 2.3 є можливість дійти висновку, що модель найкраще підходить для визначення Bruteforce–атак на FTP і SSH сервіси, HULK, Dos–атак використовуючи HOIC та активності ботнетів.

РОЗДІЛ 3
РОЗРОБКА МОДЕЛІ КАСИФІКАЦІЇ ПОДІЙ КІБЕРБЕЗПЕКИ В РОЗПОДІЛЕНИХ ІС

3.1Опис класифікації подій на веб–сервіс

Сьогодні багато додатків і веб–сторін покладаються на веб–сервіси для безперешкодного обміну інформацією один з одним. Веб–сервіси надають метод передачі різнорідних даних через Інтернет, який може допомогти організаціям/фізичним особам повторно використовувати функціональність різних сервісів. Між користувачами і веб–сервісами існують динамічні відносини, оскільки існує безліч сценаріїв, контрольованих призначеним для користувача введенням. Характер цієї динаміки часто викликає занепокоєння[22].

Протоколи веб–служб включають Simple Object Access Protocol (SOAP) для обміну повідомленнями, XML–RPC і JSON–RPC. Мова опису веб–служб (WSDL) і Universal Directory and Discovery Integration (UDDI) для опису і виявлення веб–служб. Основні фреймворки, використовувані для створення/розробки веб–сервісів, включають Apache Axis, .NET і Zend.Із через різноманітність протоколів і фреймворків, які беруть участь в розробці веб–сервісів, вони стають уразливими і часто піддаються атакам [23].

Більшість атак на веб–сервіси – це XML–ін'єкції, XPath–ін'єкції, SQL–ін'єкції, спуфинг, відмова в обслуговуванні і атаки "людина посередині".

DOS–атаки можуть вплинути на систему і її доступність для правильно запитаних ресурсів. Рекурсивні корисні навантаження використовують розширену вкладеність тегів XML для перевантаження синтаксичного аналізатора, що призводить до XDOS–атакам. Обов'язкова корисне навантаження завантажує в пам'ять довге XML–повідомлення, яке займає багато ресурсів процесора, роблячи сервер недоступним, що призводить до DOS–атак.

Ін'єкційні атаки виникають, коли шкідливий код впроваджується через призначені для користувача введення для отримання доступу до обмежених даними. Ін'єкції SQL, XPath і XML є найбільш частими подіями в веб–сервісах.

Хоча запити із відповідними параметрами, підготовлені операторами та проходять перевірку білих списків часто використовуються в якості контрзаходів, ці звичайні виправлення не можуть повністю усунути атаку. Підробка виконується шляхом маскування під дійсного користувача і відправлення запитів в обхід всіх засобів контролю доступу.

Існує цілий ряд атак на веб–сервіси, починаючи від ін'єкційних атак і закінчуючи атаками типу "відмова в обслуговуванні". Це докладно описано далі [24].

Ін'єкція.

Ін'єкція коду відбувається, коли зловмисник відправляє неприпустимі дані в веб–додаток, щоб змусити його виконати дію, для якого програма не було розроблено / запрограмовано.

Ймовірно, найпоширенішим прикладом усунення цієї уразливості безпеки є SQL–запит, який використовує недовірені дані. Один із прикладів OWASP: String query = "SELECT * FROM account where custID = '" + request.getParameter ("id") +' '.

Цей запит можна використовувати, викликавши веб–сторінку, яка його виконує, використовуючи наступний URL: http://example.com/app/accountView?id= 'або' 1 '=' 1, в результаті чого він буде збережений в таблиці бази даних. В основі вразливості ін'єкції коду лежить відсутність валідації і очищення даних, що використовуються веб–додатком, що означає, що вразливість може виникнути практично в будь–якому типі технології, пов'язаної з веб–сайтом [25].

Будь–який контент, що приймає параметри в якості вхідних даних, може бути уразливим для атаки з впровадженням коду.

Збій аутентифікації

Пошкоджена вразливість аутентифікації може дозволити зловмиснику використовувати ручні та/або автоматизовані методи, щоб спробувати отримати

контроль над будь–обліковим записом в системі або, що ще гірше, отримати повний контроль над системою. Сайти з пошкодженими або уразливою аутентифікацією дуже часто зустрічаються в Інтернеті. Збій аутентифікації зазвичай є логічною проблемою, яка виникає в механізмі аутентифікації додатків, наприклад, погане управління сеансами, схильне до перерахування імен користувачів – коли зловмисник використовує грубу силу, щоб вгадати або підтвердити дійсного користувача в системі [26].

Щоб мінімізувати ризик порушення аутентифікації, необхідно не надавати всім відвідувачам підпорядкованого сайту загальнодоступну сторінку входу адміністратора:

- /administrator на Joomla!;
- /wp–admin/ на WordPress;
- /index.php/admin на Magento;
- /user/login для входу в Drupal.

XML External Entities (XXE).

Згідно до визначення, атака XML External Entity – це тип атаки на програму, яка аналізує вхідні дані XML. Ця атака відбувається, коли вхід XML, що містить посилання на зовнішню сутність, обробляється погано налаштованим аналізатором XML [27]. Більшість аналізаторів XML за замовчуванням вразливі до атак XXE. Як результат, відповідальність за те, щоб програма не демонструвала цієї вразливості, покладається головним чином на розробника.

Згідно з OWASP основними векторами атак для зовнішніх сутностей XML (XXE) є використання:

- вразливі процесори XML, коли зловмисники можуть завантажувати XML або включати ворожий вміст у документ XML;
- вразливий код;
- вразливі залежності;
- вразливі інтеграції.

Таким чином є можливість запобігти атакам на зовнішні сутності XML [28]. Відповідно до OWASP, для запобігання XML–атакам зовнішніх сутностей, серед іншого:

- якщо можливо, необхідно використовувати менш складні формати даних, такі як JSON, і уникати серіалізації конфіденційних даних;
- виправляти або оновлювати будь–які XML–процесори та бібліотеки, що використовуються додатком або базовою операційною системою;
- використовувати засоби перевірки залежностей (оновлювати SOAP до SOAP 1.2 або вище);
- вимикати обробку XML та DTD у всіх аналізаторах XML у додатку відповідно до шпаргалки OWASP "XXE Prevention";
- впроваджувати позитивну перевірку, фільтрацію чи очищення входів на стороні сервера ("дозволити списки"), щоб запобігти ворожим даним у документах XML, заголовках або вузлах;
- впевненість, що функція завантаження файлів XML або XSL перевіряє вхідний XML, використовуючи перевірку XSD або щось подібне;
- засоби SAST можуть допомогти виявити XXE у вихідному коді – хоча ручний огляд коду є найкращою альтернативою у великих, складних додатках з багатьма інтеграціями.

Порушений контроль доступу.

У безпеці веб–сайтів контроль доступу означає обмеження кількості областей або сторінок, до яких відвідувачі можуть зайти за потреби.

Наприклад, при визначенні ролі як власник магазину електронної комерції, при такому розкладі, ймовірно, знадобиться доступ до адміністративної панелі, щоб додати нові товари або налаштувати рекламну акцію на майбутні свята [29, 30]. Однак навряд чи комусь ще це знадобиться. Якщо дозволити іншим відвідувачам веб–сайту перейти на сторінку входу, магазин електронної комерції буде відкритий лише для атаки.

І в цьому проблема майже всіх поширених систем управління вмістом (CMS) сьогодні. За замовчуванням вони надають доступ у всьому світі до сторінки входу адміністратора. Більшість з них також не змушують вас встановлювати двофакторний метод автентифікації (2FA) [29].

Вищесказане змушує багато замислюватися про розробку програмного забезпечення із філософією, що стосується першої лінії безпеки.

Приклади неправильного контролю доступу, далі представлені кілька прикладів того, що вважається "доступом" до веб–сервісу [31]:

- доступ до панелі управління / управління хостингом;
- доступ до сервера через FTP / SFTP / SSH;
- доступ до вікна адміністрування веб–сайту;
- доступ до інших програм на вашому сервері;
- доступ до бази даних.

Зловмисники можуть використовувати помилки авторизації такими способами:

- доступ до несанкціонованих функцій та/або даних;
- перегляд конфіденційних файлів;
- змінити права доступу.

Неправильні налаштування безпеки.

По суті, груба сила випробовує багато можливих комбінацій, але існує багато варіантів цієї атаки, щоб збільшити швидкість її успіху [30, 31]. Ось найпоширеніші:

- невиправлені дефекти;
- стандартні конфігурації;
- невикористані сторінки;
- незахищені файли та каталоги;
- непотрібні послуги.

Однією з найпоширеніших помилок веб–майстрів є дотримання стандартних конфігурацій CMS. Сьогоднішні програми CMS (хоча і прості у використанні) можуть бути складними для кінцевих користувачів з міркувань безпеки. Найпоширеніші атаки

на сьогоднішній день повністю автоматизовані. Багато з цих атак вимагають, щоб користувачі мали лише налаштування за замовчуванням.

Це означає, що велику кількість атак можна заблокувати, змінивши налаштування за замовчуванням під час встановлення CMS. Є налаштування, які ви можете налаштувати для управління коментарями, користувачами та видимістю інформації про користувачів [32]. Файлові дозволи – ще один приклад налаштування за замовчуванням, який можна посилити.

При аналізі контексту і результатів попередніх розділів дипломної роботи, було визнано за необхідне деталізувати генеровану модель для одного типу атак на веб–сервіси (тобто SQL–ін'єкції).

3.2Розробка моделі класифікації SQL–атаки

Дані – один з найважливіших компонентів інформаційних систем. Веб–додатки, що працюють на базі даних, використовуються організацією для отримання даних від клієнтів. SQL – це абревіатура мови структурованих запитів. Він використовується для отримання і маніпулювання даними в базі даних.

SQL–ін'єкція – це метод, який зловмисники застосовують для вставки SQL–запитів в поля введення, які потім обробляються базою даних SQL. Цими слабкими місцями можна зловживати, коли форми входу дозволяють генеруються користувачем SQL–запитах безпосередньо запитувати базу даних.

Для прикладу візьмемо типову форму входу в систему, що складається з поля введення користувача, електронної пошти і поля введення пароля. Після введення інформації для входу вона об'єднується з SQL–запитом на вашому веб–сервері. У PHP команда записується в такий спосіб:

```
<?php
$query = "SELECT * FROM users WHERE username = '" .
$_POST['username'] . "'"; $query .= " AND password = '" .
```

$_POST['password'] . """; ?>

Вона відправляється на сервер, щоб перевірити, чи було дано дійсне ім'я користувача з відповідним паролем [33]. Ім'я користувача "james" з паролем "1111" призведе до такої команді:

SELECT * FROM users WHERE username='james' AND password='1111'

Але якщо вони поставлять щось на кшталт "james '; –", запит буде виглядати наступним чином:

SELECT * FROM users WHERE username='james'; — ' AND password='1111'

У цьому сценарії зловмисник використовує синтаксис коментаря SQL. Щоб залишився код після послідовності подвійних тире (–), що призведе до не виконання запитів. Це означає, що SQL буде виглядати наступним чином:

SELECT * FROM users WHERE username='james';

Потім він поверне дані користувача, які були введені в поле пароля. Цей хід може дозволити обійти екран входу в систему [34]. Зловмисник також може піти далі, додавши ще одну умову Select, "OR 1 = 1", що призведе до наступного запиту :

SELECT * FROM users WHERE username='james' OR 1=1;

Запит повертає непорожній набір даних для будь–якого потенційного входу в систему з усією базою даних таблиці "users". Наведений вище хак показав нам істотний недолік безпеки будь–якого сайту, але це лише невеликий приклад того, що він може зробити. Більш просунуті зломи дозволять зловмисникові запускати довільні оператори.

Це може привести до[35, 36]:

- вилучення приватних даних, таких як кредитні картки, паспорти, лікарняні листи;
- збір даних користувача для аутентифікації, що дозволяє використовувати ці логіни на інших сайтах;
- пошкодження бази даних, виконання команд ОС, видалення або вставка даних і знищення операцій для всього веб–сайту;

- повна компрометація системи.

Процес валідації спрямований на перевірку того, чи дозволений тип введення, представлений користувачем. Валідація введення перевіряє, що він має допустимий тип, довжину, формат і так далі. Тільки значення, що пройшли валідацію, можуть бути оброблені. Це допомагає протистояти будь–яким командам, вставленим в рядок введення. У певному сенсі це схоже на перевірку того, хто стукає, перш ніж відкрити двері.

Валідація повинна застосовуватися не тільки до полів, що дозволяють користувачам вводити дані, тобто необхідно в рівній мірі подбати і про таких ситуаціях[37]:

- необхідно використовувати регулярні вирази як білих списків для структурованих даних (таких як ім'я, вік, дохід, відповідь на опитування, поштовий індекс), щоб забезпечити надійну перевірку введення;
- у разі фіксованого набору значень (наприклад, список, що випадає, радіокнопка) необхідно визначити, яке значення буде повернуто. Дані, що вводяться повинні точно відповідати одному із запропонованих варіантів.

Необхідно також згадати деякі особливо важливі правила для SQL–ін’єкції, такі як уникнення використання додатків до бази даних, використовуючи обліковий запис з доступом root [38]. Це слід робити тільки в разі крайньої необхідності, оскільки зловмисники можуть отримати доступ до всієї системи. Навіть сервер неадміністративних облікових записів може створити ризик для додатка, тим більше, якщо сервер баз даних використовується кількома програмами і базами даних.

З цієї причини для захисту програми від SQL–ін'єкцій краще застосовувати найменші привілеї до бази даних. Переконайтеся, що кожен додаток має свої власні облікові дані бази даних і що ці облікові дані мають мінімальні права, необхідні додатком [39].

Замість того щоб намагатися визначити, які права доступу слід відібрати, зосередьтеся на визначенні того, які права доступу або підвищені дозволу потрібні

вашому додатку. Якщо користувачеві потрібен доступ тільки до деяких частин, можна створити режим, виконує тільки цю функцію.

Одним з кращих методів виявлення атак SQL–ін'єкцій є використання брандмауера веб–додатків (WAF) [40]. WAF, що працює перед веб–серверами, відстежує трафік, який входить і виходить з веб–серверів, і виявляє шаблони, які становлять загрозу. По суті, це бар'єр, встановлений між веб–додатком і Інтернетом.

WAF працює на основі певних параметрів, що правил веб–безпеки. Ці набори політик повідомляють WAF, які слабкі місця і поведінку трафіку він повинен шукати. Грунтуючись на цій інформації, WAF буде стежити за додатками і одержуваними GET і POST запитами, щоб знайти і заблокувати шкідливий трафік [41].

Цінність WAF почасти обумовлена простотою зміни політик. Нові політики можуть бути додані в найкоротші терміни, що забезпечує швидке впровадження правил і швидке реагування на інциденти.

Такі методи запобігання, як під час введення, параметризрвані запити, збережені процедури і екранування, добре працюють з різними векторами атак. Однак через велику різноманітності шаблонів атак SQL–ін'єкцій вони часто не можуть захистити бази даних.

Тому, якщо є необхідність охопити всі бази, вам слід застосовувати вищезгадані стратегії в поєднанні з надійним WAF. Основна перевага WAF полягає в тому, що він забезпечує захист призначених для користувача веб–додатків, які в іншому випадку залишилися б незахищеними.

Для побудови логічної моделі необхідно, необхідне виконання умови, що полягає в можливості співвідносити вхідні HTTP–запити з атаками SQL–ін’єкції [42]. Дана модель повинна складатись із трьох головних частин:

- URL генератора;
- URL класифікатор;
- на основі нейронної мережі модель.

URL генератор – це модуль тестового виду, що необхідний для створення вихідних наборів даних, складається з двох елементів: генерація стандартних URL шляхом збирання даних з відомих сайтів (як приклад, аналізуючи sitemap.xml файли), і генерація зловмисних адрес шляхом додавання SQL типовим способом. Вставка параметрів запиту URL в URL, що був отриманий минулим методом. Другий спосіб наповнити зразок використовуваними зловмисними параметрами – використовувати набір даних з відкритого джерела.

Типовий запит SQL–ін'єкції містить головні слова, написані на SQL мові, що зазвичай використовують для реалізації операцій над таблицями SQL бази даних. Запити можна застосовувати як до всієї бази, включаючи створення та/або видалення таблиць і т.п., так і до виділених таблиць, тобто змінювати записи, оптимізувати пошук і т.п [43].

Аналіз та синтез моделі розпізнавання атаки потребує особливих умов функціонування, тому необхідно провести початкову підготовку тестового, навчального та контрольного набору даних. До навчального набору даних входять параметри об'єкта навчання, причому вибір параметрів здійснюється евристичний на основі проведеного аналізу основних ознак інциденту, які можуть містити URL.

Відомо, що моделі на базі нейронної мережі можуть використовуватися тільки з числовими даними в певному діапазоні значень, по цій причині перший етап дослідження полягав у розробленні класифікатора URL, який являє з себе програмне забезпечення, що перетворює URL у бінарний формат і встановлює логічні модуль. "TRUE" – (1), якщо адреса є атакуючою, і "FALSE" – (0) в інших випадках [44].

Отже, для кожного URL генерується вектор вхідного характеру з присвоєнням йому номера. Враховуючи що параметрів запиту в наявності є дванадцять відповідно до шаблонів параметрів SQL, загальна кількість нейронів, що матиме нейромережа на вхідний шар складатиме дванадцять.

Використовуючи Node JS і сторонній модуль Synaptic буде розроблено відповідне програмне забезпечення для навчання моделі.

Synaptic – це бібліотека нейромереж JavaScript для node.js, яка дозволяє генерувати і навчати практично будь–який вид архітектури нейромережі різних порядків [45].

Відповідно згенерована модель має 3 шари із нейронів, з S–образними функціями активації для нейронів у вхідному і прихованому шарах і лінійним вихідним шаром. Модель має таку архітектуру: вхідний шар із двандцяти нейронів; прихований шар із шести нейронів та вихідний шар один. На вході моделі вводиться вхідний вектор, що компонований з 12 складників X, з допустимою межею значення від 0 до 1. Ініціалізація в моделі здійснюється шляхом розстановки випадкових величин в проміжку від –1 до 1.

Число нейронів у прихованому шарі визначається наступними важливими правилами:

- величина прихованого шару є оптимальною, коли це значення, яке зазвичай знаходиться між кількістю вузлів вхідного шарі і кількістю вузлів вихідного шару;
- число нейронів у даному шарі, представляє із себе середнє число між кількістю нейронів у вихідному шарі і кількістю нейронів у вхідному шарі.

Щоб налаштувати нейромережу, необхідно визначити безліч параметрів, заснованих на отриманні найкращих результатів при найкращій конфігурації (значення відхилення, швидкість навчання і т.п.).

Швидкість навчання – в машинному навчанні і статистиці являє параметр настройки в алгоритмі оптимізації, який прораховує розмір шагу кожної ітерації і рух до мінімального значення функції помилки. Діапазон значень становить від 0 до 1. В роботі це значення "швидкості навчання" – 0,2.

Якщо мінімальна помилка не була досягнута, то для завершення навчання потрібна максимальна кількість ітерацій (періодів). Кількість ітерацій в роботі не перевищує 200. Мінімальна помилка – значення помилки, при якому нейронна мережа припиняє навчання. В роботі це значення становить 0,004.

Вихідний збір ділиться на 60% (навчальний набір), 20% (контрольний) і 20% (тестовий). Для зствореннія вибірки було згенеровано до 30200 повністю різних URL–адрес.

Набір із навчальних даних складався з 20150 записів, з яких 12900 були звичайними URL–адресами, а 7200 – шкідливими URL–адресами.

Набір із контрольованих даних складається з 5020 записів: 3200 нормальних записів і 1 800 шкідливих записів.

Тестова вибірка містить 5015 записів: 3200 звичайних записів і 1800 шкідливих записів. Після навчання інтегрована модель може класифікувати вхідні дані з точністю 96%.

Нижче показані графіки навчання в моделі, що відображають зміни помилки навчання для кожного етапу. Процес навчання відбувається як на малих, так і на великих наборах.

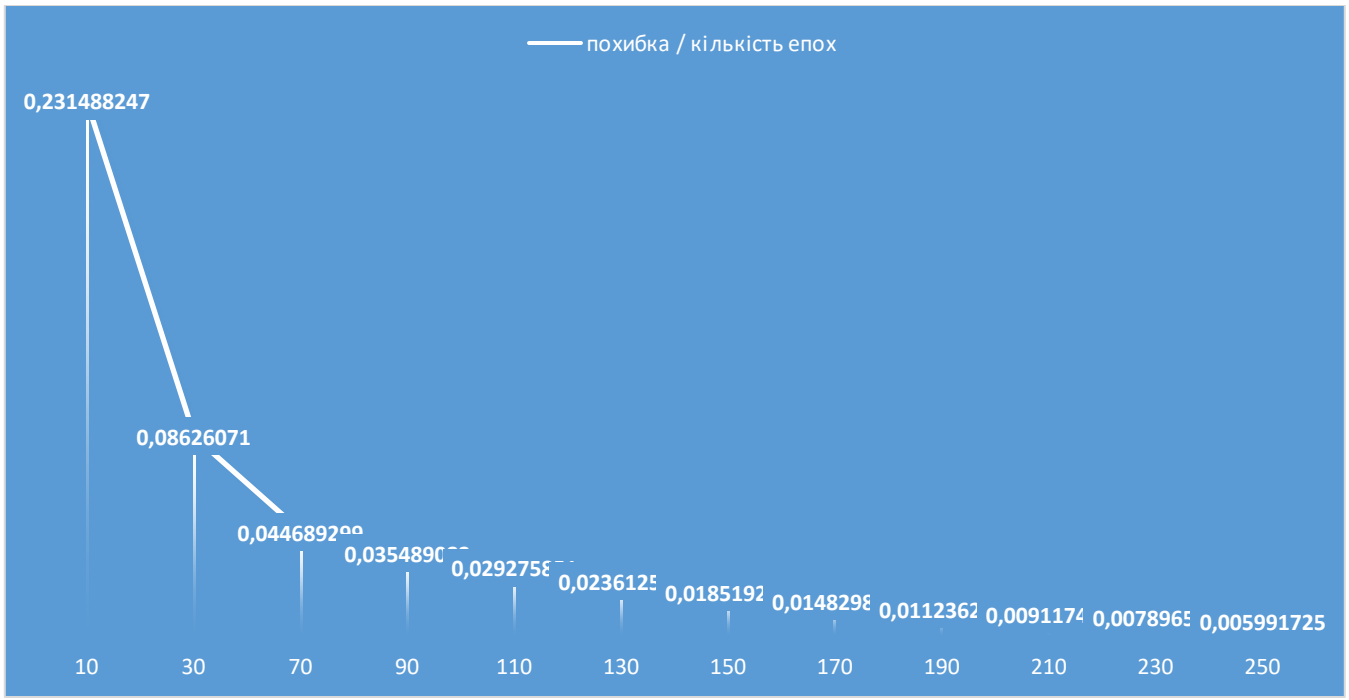

Рисунок 0.1. Графік малої вибірки

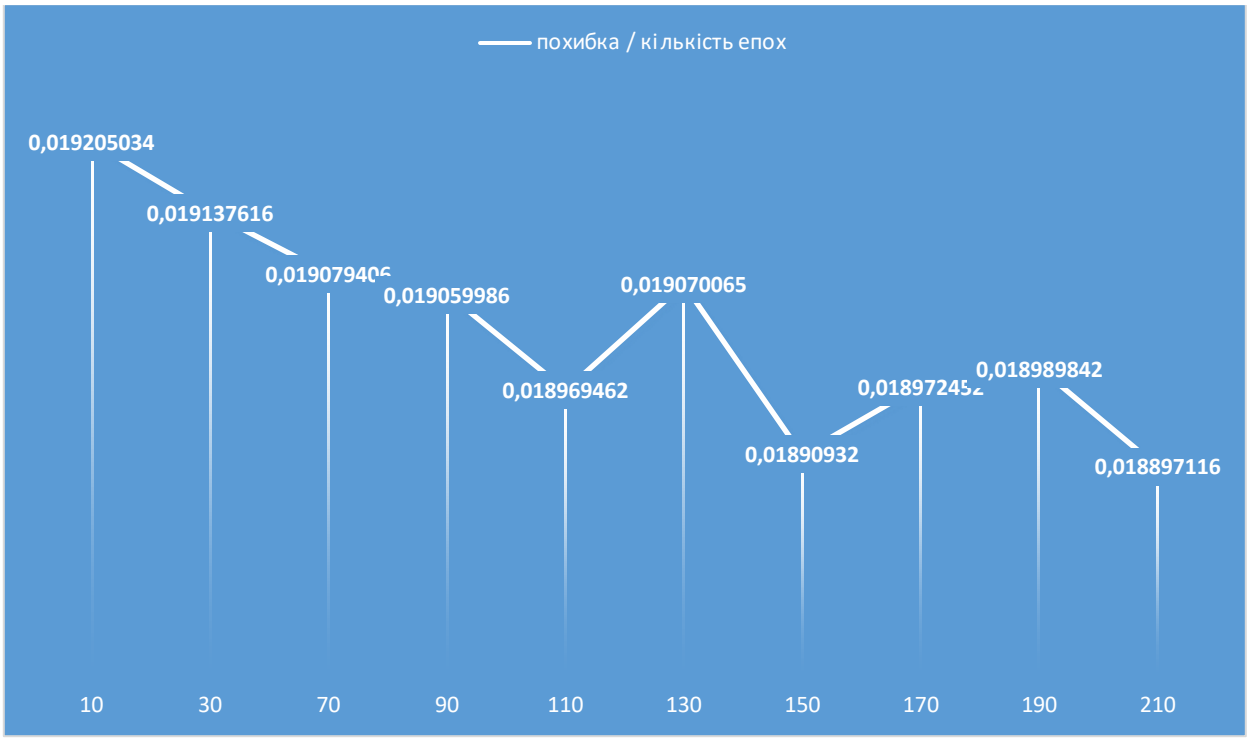

Рисунок 0.2. Графік великої вибірки

Як видно з наведених вище графіків, відносна помилка навчання для великої навчальної вибірки становить 1,87%.

В процесі роботи нейронної мережі може трапитися така ситуація: досліджувані параметри від SQL–ін'єкцій відображаються в URL–адресах в будь–якому порядку і не завдають ніякої шкоди веб–серверам і базам даних відповідно (помилкове спрацьовування), але для системи вони визначаються як атака. Для подальших досліджень з метою вирішення такої проблеми можна використовувати безумовні шаблони SQL–ін'єкцій і враховувати стан HTTP в коді відповіді веб–сервера.

Підсумовуючи, в цілому, відносні помилки у згенерованій моделі в контрольній і тестової вибірках не переходять за 5%. Результати класифікації наведені в таблиці. 3.2.

Таблиця 3.1 – Результати моделі (вдалі/загальна)

Тип трафіку	Навчальний набір	Контрольний набір	Тестовий набір	Відносне відхилення, %
Стандартний	12900	3193 / 3214	3190 / 3217	0,8 %
SQL– ін'єкція	7253	1715 / 18020	1711 / 1807	3,7 %

3.3Висновки за розділом 3

Відповідно до виконаної роботи, при аналізі результатів моделі на тестовому наборі даних є можливість зробити остаточне ствердження, що запропонований вище метод класифікації SQL–ін'єкції може ідентифікувати цей тип атаки із загальною точністю близькій до 96%. Висновком такої ситуації можна вважати, що точність класифікації може бути поліпшена шляхом навчання даної моделі на більшому наборі даних.

ВИСНОВКИ

У своїй дипломній роботі я проаналізувала підходи до досягнення поставленої мети у вигляді моделі класифікації подій кібербезпеки в розподілених ІС. Було проаналізовано стан питання та мною було поставлено задачі для виконання мети роботи, а саме досліджено аналіз подій кібербезпеки на світовому рівні за минулий рік у різних сферах життєдіяльності людини з урахуванням епідеміологічної ситуації у світі та представлені шляхи протидії їм. Проаналізовано нейронну мережу та використання її для досягнення поставленої мети.

Для вирішення проблеми дослідження було зібрано дані та проаналізовано моделі, розібрано алгоритми, синтезовано та навчено моделі нейронних мереж. У результаті дослідження було проведено аналіз на адекватність синтезованої моделі.

Таким чином, мета роботи, яка полягала у розробці моделі класифікації подій кібербезпеки в розподілених ІС, досягнута, всі поставлені завдання виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Global Cybercrime Damages Predicted To Reach $6 Trillion Annually By 2021 URL:https://cybersecurityventures.com/cybercrime–damages–6–trillion–by–2021/.
2. 2020 Data Breach Investigations Report. // Verizon. – 2020.
3. Cybercrime cost the world over $1 trillion in 2020 URL:https://www.itproportal.com/news/cybercrime–cost–the–world–over–dollar1–trillion–in–2020/.
4. CSE–CIC–IDS2018 on AWS URL: https://www.unb.ca/cic/datasets/ids–2018.html.
5. Neural Networks URL: https://www.ibm.com/cloud/learn/neural–networks.
6. ДСТУ 2392–94. Інформація та документація. Базові поняття. Терміни та визначення
7. Wang, H., LO, M. K., and Wang, C: «Consumer Privacy Concerns about Internet Marketing.» Commun. ACM, vol. 41, no. 3, pp. 63–70, Mar. 1998.
8. Mishra, K. S., & Tripathi, A. K. (2014). Some issues, challenges and problems of distributed software system. International Journal of Computer Science and Information Technologies. Varanasi, India, 7(3).
9. Nadiminti, K., De Assunao, M. D., & Buyya, R. (2006). Distributed systems and recent innovations: Challenges and benefits. InfoNet Magazine, 16(3), 1–5.
10. How to prevent SQL injection attacks URL: https://www.ptsecurity.com/ww–en/analytics/knowledge–base/how–to–prevent–sql–injection–attacks/
11. Laprie, J.–C: «Dependability — Its Attributes, Impairments and Means.» In Randell, B., Laprie, J.–C, Kopetz, H., and Littlewood, B. (eds.), Predictably Dependable Computing Systems, pp. 3–24. Berlin: Springer–Verlag, 1995.
12. Pfleeger, C: Security in Computing. Upper Saddle River, NJ: Prentice Hall, 2nded., 1997.
13. Shahabi, Reza Nayebi. ”Security Techniques in Distributed Systems.” system 2: 3.

14. Liu, H., Luo, P., & Wang, D. (2008). A scalable authentication model based on public keys. Journal of Network and Computer Applications, 31(4), 375–386.
15. Wang, F., & Zhang, Y. (2008). A new provably secure authentication and key agreement mechanism for SIP using certificateless public–key cryptography. Computer Communications, 31(10), 2142–2149.
16. Malacaria, P., & Smeraldi, F. (2013). Thermodynamic aspects of confidentiality. Information and Computation, 226, 76–93.
17. Chandra, S., &Khan, R.A.(2010).Confidentialitycheckinganobject–orientedclasshierarchy. Network Security, 2010(3), 16–20.
18. Andress, J. (2014). The basics of information security: understanding the fundamentals of InfoSec in theory and practice. Syngress.
19. Prowell, S., Kraus, R., & Borkin, M. (2010). Seven deadliest network attacks. Elsevier.
20. Cheswick, W. and Bellovin, S.: Firewalls and Internet Security. Reading, MA: Addison–Wesley, 2nd ed., 2000.
21. Zwicky, E., Cooper, S., Chapman, D., and Russell, D.: Building Internet Firewalls. Sebastopol, CA: O'Reilly & Associates, 2nd ed., 2000.
22. Kumar, P. A. R., & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. Computer Communications, 34(11), 1328–1341.
23. NIST Special Publication 800–61, Revision 2, Computer Security Incident Handling Guide
24. Indrajit, R. E. (2011). Pengantar Konsep Keamanan Informasi di Dunia Cyber. Jakarta: APTIKOM..
25. Sarno, Riyanarto., Iffano, Irsyat. (2009). Sistem Manajemen Keamanan Informasi. ITSPress.
26. Deris Stiawan, Intrusion Prevention System (IPS) dan Tantangan dalam pengembanganya, Jurusan Sistem Komputer, FASILKOM, Universitas Sriwijaya.
27. O'Brien, James A. 2005. Introduction to Information System. Mc Graw Hill.

28. Research, G. (2011). How to deploy SIEM technology. Technical report..
29. Mokube,I. & Adams M.,2007.Honeypots:Concepts,Approaches,and Challenges.ACMSE 2007, March 23–24, 2007, Winston–Salem, North Carolina, USA ,pp.321–325..
30. Spitzner, Lance. (2012), Honeypots: tracking hackers. Addison–Wesley Longman Publishing Co., Inc.
31. Global Cybercrime Damages Predicted To Reach $6 Trillion Annually By 2021 URL:https://cybersecurityventures.com/cybercrime–damages–6–trillion–by–2021/.
32. Hastie T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman. – New York: Springer, 2009. – 764 с. – (2).
33. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities / Hopfield. // Proceedings of National Academy of Sciences. – 1982. – №79. – С. 2554–2558.
34. LSTM: A Search Space Odyssey / K.Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink., 2015. – 12 с.
35. Murphy K. P. Machine Learning: A Probabilistic Perspective / Kevin Murphy. – Cambridge: MIT Press, 2012. – 247 с.
36. Newcomer E. Uber Paid Hackers to Delete Stolen Data on 57 Million People URL:https://www.bloomberg.com/news/articles/2017–11–21/uber–concealed–cyberattack–that–exposed–57–million–people–s–data.
37. OWASP. SQL Injection URL:https://owasp.org/www–community/attacks/SQL_Injection.
38. Ponemon L. What’s New in the 2019 Cost of a Data Breach Report URL:https://securityintelligence.com/posts/whats–new–in–the–2019–cost–of–a–data–breach–report/.
39. Rouse M. SOAR (Security Orchestration, Automation and Response) URL:https://searchsecurity.techtarget.com/definition/SOAR.

40. Total number of Websites URL:https://www.internetlivestats.com/total–number–of–websites/.
41. WEKA Manual for Version 3–9–3 / [R. Bouckaert, E. Frank, M. Hall та ін.]. – Hamilton: University of Waikato.
42. What is SQL Injection (SQLi) and How to Prevent It URL:https://www.acunetix.com/websitesecurity/sql–injection/.
43. Williams A. Improve IT Security With Vulnerability Management URL:https://www.gartner.com/en/documents/480703/improve–it–security–with–vulnerability–management.
44. Хайкін С. Нейронные сети: полный курс = Neural Networks: A Comprehensive Foundation / Саймон Хайкін. – Москва: Вільямс, 2006. – 1104 с. – (2).
45. Sumit S. A Comprehensive Guide to Convolutional Neural Networks URL:https://towardsdatascience.com/a–comprehensive–guide–to–convolutional–neural–networks–the–eli5–way–3bd2b1164a53.

ДОДАТКИ

ДОДАТОК А
Передік наукових публікацій

1. Бабенко Т. В., Сахацька А. В., Дослідження методів кібербезпеки в розподілених системах. // IV Міжнародна науково–практична конференція “Проблеми кібербезпеки інформаційно–телекомунікаційних систем” (PCSITS)”. – 2021.