АНОТАЦІЯ

на магістерську роботу «Нейромережева система виявлення комп'ютерних атак на основі аналізу мережевого трафіку»,
студента Хезретова Гуванча Нурягдиєвича

Актуальність дослідження полягає в необхідності розробки систем ідентифікації вторгнень та аномальних станів мережевого трафіку для своєчасного виявлення мережевих атак та забезпечення мережевої безпеки.

Мета дослідження – розробка системи ідентифікація аномальних станів комп'ютерних систем на основі паралельної обробки трафіка комп'ютерної мережі з використанням колективу нейромереж.

Задачі дослідження: виконати аналіз методів і технологій виявлення аномалій і вторгнень; описати алгоритм побудови класифікатору на основі нейронних мереж; побудувати віртуальний нейромережевий процесор розв'язку задач виявлення аномальних станів засобами пакету MatLAB Neural Network Toolbox; виконати експериментальні дослідження роботи віртуального нейромережевого процесору.

Об'єкт дослідження – інтелектуальні технології, що базуються на нейромережевих принципах і орієнтовані на розв'язок прикладних задач в області забезпечення мережевої безпеки.

Предмет дослідження – методи та моделі ідентифікації аномальних станів трафіку комп'ютерних мереж колективом нейромереж.

Методи дослідження: нейромережевий аналіз, імітаційне моделювання, методи розпізнавання образів.

Результати, їх новизна, теоретичне та практичне значення: створена модель віртуального процесору, що дає змогу розпізнати (ідентифікувати) аномальні стани в комп'ютерних мережах. Віртуальній процесор дозволяє проаналізувати роботу запропонованої нейромережевої технології для ідентифікації атак. Автоматично генеруються випадкові вхідні набори, які відносяться до одного із класів, на екран одночасно виводиться сгенерований набір і рішення нейромережею завдання класифікації.

Структура магістерської роботи складається з вступу, чотирьох розділів, висновків, переліку посилань на 18 найменуваннь, додатків. Повний обсяг роботи становить 78 сторінки, містить 30 рисунків і 4 таблиці.

КЛЮЧОВІ СЛОВА: комп'ютерна мережа, мережеві атаки, нейронна мережа, аномальний трафік, класифікація.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

CVV(complete cross–validation)	функціонал повного ковзкого контролю
DDoS (distributed DoS)	розподілена атака на відмову в обслуговуванні
HTTP(Hypertext transfer protocol)	протокол передачі гіпертексту
ICMP (Internet Control Message Protocol)	міжмережевий протокол керуючих повідомлень
IDS (Intrusion Detection Systems)	система виявлення вторгнень
IP (Internet Protocol)	протокол мережевого рівня для передачі датаграм між мережами.
LVQ (Learning Vector Quantization)	тип нейронної мережі
SSH (Secure SHell)	мережевий протокол рівня додатків, що дозволяє проводити віддалене управління комп'ютером і тунелювання *TCP*–з'єднань
TCP (Transmision controlr protocol)	протокол керування передачею
БД	база даних
ІС	інформаційна система
КМ	комп’ютерна мережа
КС	комп’ютерна система
МСВВ	мережеві системи виявлення вторгнень
НМ	нейронна мережа
НС	нейронна система
ОС	операційна система
СУБД	система управління базами даних

ВСТУП

Виявлення мережевих атак є в даний момент є однією з найбільш гострих проблем мережевих технологій. Однією з актуальних наукових завдань в даний час є аналіз (і подальше прогнозування) самоподобної структури трафіку в сучасних мультисервісних мережах. Для вирішення цього завдання необхідний збір і подальший аналіз різноманітної статистики в діючих мережах.

Метою та завданням магістерської атестаційної роботи є ідентифікація аномальних станів комп’ютерних систем на основі паралельної обробки трафіка комп’ютерної мережі (КМ) використовуючи колектив нейромереж.

Об'єктом дослідження є інтелектуальні технології, базовані на нейромережевих принципах, які орієнтовані на розв’язок прикладних задач.

Предметом дослідження є методи та моделі ідентифікації аномальних станів трафіка КМ комітетом нейромереж.

Методи дослідження:

- нейромережевий аналіз;
- імітаційне моделювання;
- розпізнавання образів.

Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- виконати аналіз мережевих небезпек та методів і технологій виявлення аномалій і вторгнень;
- виконати аналіз можливості використання нейронних мереж в якості класифікатору аномальних станів трафіку комп’ютерної мережі;
- описати алгоритм побудови класифікатору на основі нейронних мереж, а саме етапів: попередньої обробки даних, підготовки вихідних даних, вибір архітектури мережі;
- побудувати віртуальний нейромережевий процесор розв’язку задач виявлення аномальних станів засобами пакету MatLAB Neural Network Toolbox;
- виконати експериментальні дослідження роботи віртуального нейромережевого процесору.

1 СУЧАСНИЙ СТАН ІДЕНТИФІКАЦІЇ ТА АНАЛІЗУ ТРАФІКУ КОМП'ЮТЕРНИХ СИСТЕМ

Завдання аналізу трафіку магістральних Інтернет-каналів з кожним роком стає все більш необхідним. На даний момент всесвітня мережа Інтернет використовується не тільки для обміну інформацією, а й для надання різних послуг, у тому числі державних. Тим самим гостро постає питання про забезпечення відмово стійкої, безперебійної роботи серверів організацій, що надають такі послуги.

Аналіз трафіку магістральних Інтернет – каналів є складним завданням, що залежить від безлічі параметрів, яка насилу піддається декомпозиції і моделюванню. Однією з причин цього є постійне ускладнення структури глобальної мережі, яка характеризується взаємодією великої кількості пристроїв самих різних типів, які не мають єдиного центру управління.

Трафік є дуже великим і змінюється безперервно, тоді як аномальній трафік є маленьким порівняно з нормальним трафіком і змінами нормального трафіку. Основна мета виявлення аномалії полягає в тому, щоб виявити відносно маленькій трафік аномалії у відносно великому фоновому трафіку. Тому швидке і точне виявлення аномалій трафіку є однією з умов безпечної ефективної роботи мережі.

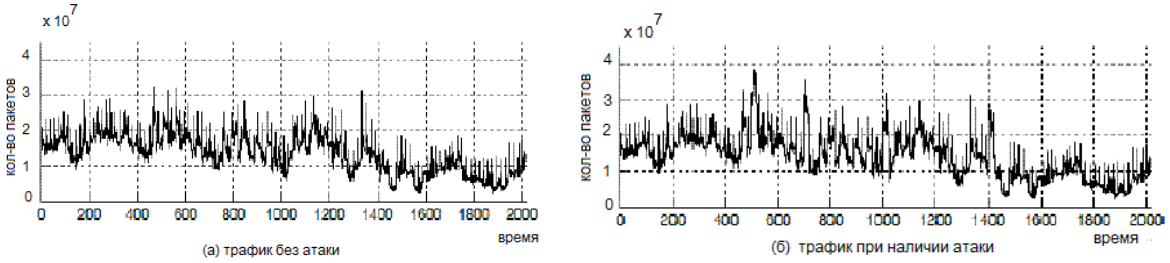
Процес здійснення загроз інформаційній системі отримав назву «атака» або «вторгнення».

Атака – це будь-яка дія порушника, спрямована на порушення заданої функціональності обчислювальної системи або отримання несанкціонованого доступу до інформаційних, обчислювальних або мережевих ресурсів. Атака, як і будь-яка дія, має свій життєвий цикл, що розділяє її на етапи підготовки, вторгнення, атакуючого впливу і розвитку атаки.

Атака на інформаційну систему – подія або сукупність подій, які стосовно кожного окремо взятого об'єкта повинні розглядатися в якості спроб здійснення інформаційного впливу протиправного чи деструктивного характеру. Приклад вигляду нормального і аномального трафіку показано на (рис.1.1).

Виявлення мережевих атак є в даний момент є однією з найбільш гострих проблем мережевих технологій. За даними DARPA(Defense Advanced Research Projects Agency – агентство передових оборонних дослідницьких проектів), незахищений комп'ютер, підключений до мережі Інтернет, буде зламаний не пізніше ніж через 2–3 години. Масштабні епідемії мережевих черв'я-

ків, DDoS атаки з бот-мереж розміром більше 10000 комп'ютерів, автоматизовані засоби пошуку вразливостей в мережах – все це робить забезпечення безпеки локальних мереж дуже трудомісткою справою. Зараз важко знайти мережу, в якій відсутні такі активні засоби попередження атак як антивірус, брандмауер, системи попередження вторгнень рівня хоста і так далі. На жаль, одних активних засобів відбиття атак недостатньо. Тому, на додаток до них застосовують пасивні засоби боротьби з атаками – мережеві системи виявлення вторгнень [1].



а) б)

Рисунок 1.1 – Мережевий трафік: а – при наявності атаки;
б – при її відсутності.

Виявлення атак – це процес оцінки підозрілих дій в мережі, що захищається, який реалізується або за допомогою аналізу журналів реєстрації операційної системи і додатків або мережевого трафіку. Мета виявлення атак – виявити ознаки атак або під час їх, або постфактум. В якості таких ознак можуть виступати:

- повтор певних подій;
- неправильні або невідповідні поточної ситуації команди;
- використання вразливостей;
- невідповідні параметри мережевого трафіку;
- непередбачені атрибути;
- незрозумілі проблеми;
- додаткові знання про порушення.

1.1 Аналіз небезпек мережевої безпеки

Для організації комунікацій в неоднорідному мережевому середовищі застосовується набір протоколів TCP/IP, що забезпечує сумісність між комп'ютерами різних типів. Сумісність – одна з основних переваг TCP/IP, тому

більшість комп'ютерних мереж підтримує ці протоколи. Крім того, протоколи TCP/IP надають доступ до ресурсів глобальної мережі Інтернет. Завдяки своїй популярності TCP/IP став стандартом де-факто для міжмережевої взаємодії. Однак повсюдне поширення стека протоколів TCP/IP оголило і його слабкі сторони. Творці стека TCP/IP не бачили причин особливо турбуватися про захист мереж, що будуються на його основі. Тому в специфікаціях ранніх версій протоколу IP відсутні вимоги безпеки, що призвело до початкової уразливості його реалізації.

Стрімке зростання популярності інтернет технологій супроводжується зростанням серйозних загроз розголошення персональних даних, критично важливих корпоративних ресурсів, державних таємниць і т. д.

Кожен день хакери та інші зловмисники піддають загрозам мережеві інформаційні ресурси, намагаючись отримати до них доступ за допомогою спеціальних атак. Ці атаки стають все більш витонченими по впливу і нескладними у виконанні.

Цьому сприяють два основні чинники. По-перше, це повсюдне проникнення Інтернету. Сьогодні до цієї мережі підключені мільйони комп'ютерів. Багато мільйонів комп'ютерів будуть підключені до Інтернету в найближчому майбутньому, тому ймовірність доступу хакерів до уразливих комп'ютерів і комп'ютерних мережах постійно зростає. Крім того, широке поширення Інтернету дозволяє хакерам обмінюватися інформацією в глобальному масштабі. По-друге, це загальне поширення простих у використанні операційних систем і середовищ розробки. Цей фактор різко знижує вимоги до рівня знань зловмисників. Раніше від хакера були потрібні хороші знання і навички програмування, щоб створювати і поширювати шкідливі програми. Тепер, для того щоб отримати доступ до хакерських засобів, потрібно просто знати IP-адресу потрібного сайту, а для проведення атаки досить клацнути мишкою.

Проблеми забезпечення інформаційної безпеки в корпоративних комп'ютерних мережах обумовлені погрозами безпеки для локальних робочих станцій, локальних мереж і атаками на корпоративні мережі, що мають вихід в загальнодоступні мережі передачі даних.

Мережеві атаки настільки ж різноманітні, як і системи, проти яких вони спрямовані. Деякі атаки відрізняються більшою складністю. Інші здатний здійснити звичайний оператор, який навіть не передбачає, які наслідки може мати його діяльність.

Порушник, здійснюючи атаку, зазвичай ставить перед собою наступні цілі:

- Порушення конфіденційності переданої інформації;
- Порушення цілісності та достовірності переданої інформації;
- Порушення працездатності системи в цілому або окремих її частин.

З точки зору безпеки, розподілені системи характеризуються насамперед наявністю віддалених атак, оскільки компоненти розподілених систем зазвичай використовують відкриті канали передачі даних і порушник може проводити не лише пасивне прослуховування переданої інформації, але і модифікувати переданий трафік (активний вплив). І якщо активний вплив на трафік може бути зафіксовано, то пасивний вплив практично не піддається виявленню. Але оскільки в ході функціонування розподілених систем обмін службовою інформацією між компонентами системи здійснюється теж по відкритих каналах передачі даних, то службова інформація стає таким же об'єктом атаки, як і дані користувача.

Труднощі виявлення факту проведення віддаленої атаки виводить цей вид неправомірних дій на перше місце за ступенем небезпеки, оскільки невиявність перешкоджає своєчасному реагуванню на здійснену загрозу, в результаті чого у порушника збільшуються шанси успішної реалізації атаки. Безпека локальної мережі, в порівнянні з безпекою міжмережевої взаємодії відрізняється тим, що в цьому випадку на перше за значимістю місце виходять порушення зареєстрованих користувач, оскільки в основному канали передачі да–них локальної мережі знаходяться на контрольованій території і захист від несанкціонованого підключення до них реалізується адміністративними методами.

На практиці IP-мережі уразливі для ряду способів несанкціонованого вторгнення в процес обміну даними. По мірі розвитку комп'ютерних та мережевих технологій (наприклад, з появою мобільних Java-додатків і елементів ActiveX) список можливих типів мережевих атак на IP-мережі постійно розширюється.

Підслуховування (sniffing). Здебільшого дані по комп'ютерних мережах передаються в незахищеному форматі (відкритим текстом), що дозволяє зловмисникові, який отримав доступ до ліній передачі даних у вашій мережі, підслуховувати або зчитувати трафік. Для підслуховування в комп'ютерних мережах використовують сніффер. Сніффер пакетів являє собою прикладну програму, яка перехоплює всі мережеві пакети, передані через певний домен. В даний час сніффери працюють в мережах на цілком законній підставі. Во-

ни використовуються для діагностики несправностей і аналізу трафіку. Однак, з огляду на те, що деякі мережеві додатки передають дані в текстовому форматі (Telnet, FTP, POP3 тощо.), за допомогою сніфферу можна дізнатися корисну інформацію, а іноді і конфіденційну інформацію (наприклад, імена користувачів і паролі) [2].

Перехоплення пароля (password sniffing), переданого по мережі в незашифрованому вигляді, шляхом «підслуховування» каналу є різновидом атаки підслуховування. Якщо програма працює в режимі клієнт-сервер, а аутентифікаційні дані передаються по мережі в читаємому текстовому форматі, цю інформацію з великою ймовірністю можна використовувати для доступу до інших корпоративним або зовнішніх ресурсів. У найгіршому випадку хакер отримує доступ до призначеного для користувача ресурсу на системному рівні і з його допомогою створює атрибути нового користувача, які можна в будь–який момент використовувати для доступу в мережу і до її ресурсів. Запобігти загрозу сніффінга пакетів можна за допомогою таких заходів і засобів [2]:

- застосування для аутентифікації одноразових паролів;
- установка апаратних чи програмних засобів, які розпізнають сніфери;
- застосування криптографічного захисту каналів зв'язку.

Зміна даних. Зловмисник, який отримав можливість прочитати ваші дані, зможе зробити і наступний крок – змінити їх. Дані в пакеті можуть бути змінені, навіть якщо зловмисник нічого не знає ні про відправника, ні про одержувача.

Аналіз мережевого трафіку. Метою атак подібного типу є прослуховування каналів зв'язку і аналіз переданих даних і службової інформації з метою вивчення топології та архітектури побудови системи, отримання критичної інформації користувача (наприклад, паролів користувачів або номерів кредитних карт, переданих у відкритому вигляді). Атакам даного типу схильні такі протоколи, як FTP або Telnet, особливістю яких є те, що ім'я і пароль користувача передаються в рамках цих протоколів у відкритому вигляді. Підміна довіреного суб'єкта. Велика частина мереж і операційних систем використовує IP-адреса комп'ютера для того, щоб визначати, чи той це адресат, який потрібен. У деяких випадках можливе некоректне присвоєння IP-адреси (підміна IP-адреси відправника іншою адресою) – такий спосіб атаки називають фальсифікацією адреси (IP-spoofing). IP-спуфинг має місце, коли зловмисник, що знаходиться усередині корпорації або поза нею, видає себе за

законного користувача. Зловмисник може скористатися IP-адресою, яка перебуває в межах діапазону санкціонованих IP-адрес, або авторизованою зовнішньою адресою, якій дозволяється доступ до певних мережевих ресурсів.

Атаки IP-спуфінга часто є відправною точкою для інших атак. Класичним прикладом є атака типу «відмова в обслуговуванні» (DoS), яка починається з чужої адреси, що приховує справжню особистість хакера. Зазвичай IP-спуфинг обмежується вставкою помилкової інформації або шкідливих команд у звичайний потік даних, переданих між клієнтським і серверним додатком або по каналу зв'язку між однорангових пристроями. Загрозу спуфінгу можна послабити (але не усунути) за допомогою таких заходів: правильне налаштування управління доступом із зовнішньої мережі; припинення спроб спуфинга чужих мереж користувачами своєї мережі.

Слід мати на увазі, що IP-спуфинг може бути здійснений за умови проведення аутентифікації користувачів на базі IP-адрес, тому введення додаткових методів аутентифікації користувачів (на основі одноразових паролів або інших методів криптографії) дозволяє запобігти атаки IP-спуфинга.

Перехоплення сеансу (Session hijacking). Після закінчення початкової процедури аутентифікації з'єднання, встановлене законним користувачем, наприклад, з поштовим сервером, перемикається зловмисником на новий хост, а вихідного сервера видається команда розірвати з'єднання. У результаті «співрозмовник» законного користувача виявляється непомітно підміненим.

Після отримання доступу до мережі у атакуючого зловмисника з'являються великі можливості:

− Він може посилати некоректні дані додаткам і мережевим службам, що призводить до їх аварійного завершення або неправильного функціонування;

− Він може також наводнити комп'ютер або всю мережу трафіком, поки не відбудеться зупинка системи у зв'язку з перевантаженням;

− Атакуючий може блокувати трафік, що призведе до втрати доступу авторизованих користувачів до мережевих ресурсів.

Відмова в обслуговуванні (Denial of Service, DoS). Ця атака відрізняється від атак інших типів. Вона не націлена на отримання доступу до вашої мережі або на витяг з цієї мережі якоїсь інформації. Атака DoS робить мережу організації недоступною для звичайного використання деяких серверних додатків (таких як Web-сервер або FTP-сервер) атаки DoS можуть полягати в тому, щоб зайняти всі з'єднання, доступні для цих додатків, і тримати їх в за-

йнятому стані, не допускаючи обслуговування звичайних користувачів. У ході атак DoS можуть використовуватися звичайні інтернет-протоколи, такі як TCP або ICMP (Internet Control Message Protocol) [2].

Атаки DoS важко запобігти, так як для цього потрібна координація дій з провайдером. Якщо трафік, призначений для переповнення вашої мережі, не зупинити у провайдера, то на вході в мережу це зробити вже неможливо, тому що вся смуга пропускання буде зайнята. Якщо атака цього типу проводиться одночасно через безліч пристроїв, ми говоримо про розподілені атаки на відмову в обслуговуванні DDoS (distributed DoS) [2].

Простота реалізації атак DoS і величезна шкода, заподіяна ними організаціям і користувачам, залучають до цих атак пильну увагу адміністраторів мережевої безпеки. Мережеві та інформаційні технології змінюються настільки швидко, що статичні захисні механізми, до яких відносяться і системи розмежування доступу, і між мережеві екрани, і системи аутентифікації, сильно обмежені і в багатьох випадках не можуть забезпечити ефективного захисту. Тому потрібні динамічні методи, що дозволяють оперативно виявляти і запобігати порушенням безпеки. Однією з технологій, що дозволяє виявляти порушення, які не можуть бути ідентифіковані за допомогою традиційних моделей контролю доступу, є технологія виявлення атак.

По суті, процес виявлення атак є процесом оцінки підозрілих дій, які відбуваються в корпоративній мережі. Інакше кажучи, виявлення атак (instrusion detection) – це процес ідентифікації та реагування на підозрілу діяльність, спрямовану на обчислювальні або мережеві ресурси.

1.2 Методи аналізу мережевої інформації

Ефективність системи виявлення атак великою мірою залежить від застосовуваних методів аналізу отриманої інформації. У перших системах виявлення атак, розроблених на початку 80–х років, використовувалися статистичні методи виявлення атак. В даний час до статистичного аналізу додався ряд нових методик, починаючи з експертної системи, нечіткої логіки і закінчуючи використанням нейронних мереж.

Статистичний метод. Основні переваги статистичного підходу – це використання вже розробленого і зарекомендованого себе апарату математичної статистики і адаптації до поведінки суб'єкта. Спочатку для всіх суб'єктів аналізованої системи визначаються профілі. Будь–яке відхилення використовуваного профілю від еталонного вважається несанкціонованою діяльністю.

Статистичні методи універсальні, оскільки для проведення аналізу не потрібні знання про можливі атаки і вразливості, використовуванні ними. Однак при використанні цих методик виникає і кілька проблем [3]:

1) «Статистичні» системи не чутливі до порядку проходження подій; в деяких випадках одні й ті ж події в залежності від порядку їх слідування можуть характеризувати аномальну або нормальну діяльність.

2) Дуже важко задати граничні (порогові) значення, що відслідковуються, системою виявлення атак характеристик, щоб адекватно ідентифікувати аномальну діяльність.

3) «Статистичні» системи можуть бути з плином часу «навчені» порушниками так, щоб атакуючі дії розглядалися як нормальні.

Слід також враховувати, що статистичні методи незастосовні в тих випадках, коли для користувача відсутній шаблон типової поведінки або коли для користувача типові несанкціоновані дії.

Експертні системи. Експертна система складається з набору правил, які охоплюють знання людини – експерта. Використання експертних систем являє собою поширений метод виявлення атак, при якому інформація про атаки формулюється у вигляді правил. Ці правила можуть бути записані, наприклад, у вигляді послідовності дій або сигнатури. При виконанні будь–якого з цих правил приймається рішення про наявність несанкціонованої діяльності. Важливою перевагою такого підходу є практично повна відсутність помилкових тривог.

База даних експертної системи повинна містити сценарії більшості відомих на сьогоднішній день атак. Для того щоб залишатися завжди актуальними, експертні системи вимагають постійного оновлення бази даних. Хоча експертні системи пропонують гарну можливість для перегляду даних в журналах реєстрації, необхідні оновлення можуть або ігноруватись, або виконаються адміністратором вручну. Як мінімум, це призведе до експертної системи з ослабленими можливостями. У гіршому випадку відсутність належного супроводу знизить ступінь захищеності всієї мережі, вводячи її користувачів в оману щодо дійсного рівня захищеності. З недоліків, основним є неможливість відображення невідомих атак. При цьому навіть невелика зміна вже відомої атаки може стати серйозною перешкодою для функціонування системи виявлення атак [3,4].

1.3. Використання нейронних мереж

Більшість сучасних методів виявлення атак використовують деяку форму аналізу контрольованого простору на основі правил або статистичного підходу. В якості контрольованого простору можуть виступати журнали реєстрації або мережевий трафік. Цей аналіз спирається на набір заздалегідь визначених правил, які створюються адміністратором або самою системою виявлення атак.

Будь-який поділ атаки або в часі, або серед кількох зловмисників є важким для виявлення за допомогою експертних систем. За великої різноманітності атак і хакерів навіть спеціальні постійні оновлення бази даних експертної системи ніколи не дадуть гарантій точної ідентифікації всього діапазону атак. Використання нейронних мереж є одним із способів подолання зазначених проблем експертних систем. На відміну від експертних систем, які можуть дати користувачеві певну відповідь, відповідають чи ні аналізовані характеристики закладеним в базу даних правилам, нейронна мережа проводить аналіз інформації та надає можливість оцінити, чи узгоджуються дані з характеристиками, які вона навчена розпізнавати. У той час як ступінь відповідності нейромережевого подання може досягати 100 %, достовірність вибору повністю залежить від якості системи в аналізі прикладів наданої задачі.

Спочатку нейромережу навчають правильної ідентифікації та попередньо підібраною вибіркою прикладів предметної області. Реакція нейромережі аналізується, і система налаштовується таким чином, щоб досягти задовільних результатів. На додаток до початкового періоду навчання нейромережа набирається також досвіду з протягом часу, у міру того як вона проводить аналіз даних, пов'язаних з предметною областю. Важливою перевагою нейронних мереж при виявленні зловживань є їх здатність «вивчати » характеристики умисних атак та ідентифікувати елементи, які не схожі на ті, що спостерігалися в мережі раніше. Кожен з описаних методів має ряд переваг і недоліків, тому зараз практично важко зустріти систему, що реалізовує тільки один з описаних методів. Як правило, ці методи використовуються в сукупності [5,6].

1.4. Класифікація систем виявлення атак

Механізми виявлення атак, що застосовуються в сучасних системах виявлення атак IDS (Instrusion Detection System), засновані на декількох загаль-

них методах. Слід зазначити, що ці методи не є взаємовиключними. У багатьох системах використовується комбінація декількох методів. Класифікація систем виявлення атак може бути виконана за кількома ознаками [2]:

- за способом реагування;
- за способом виявлення атаки,
- за способом збору інформації про атаку.

За способом реагування розрізняють пасивні та активні IDS. Пасивні IDS просто фіксують факт атаки, записують дані у файл журналу і видають попередження. Активні IDS намагаються протидіяти атаці, наприклад, шляхом реконфігурації брандмауера або генерації списків доступу маршрутизатора. За способом виявлення атаки системи IDS прийнято ділити на дві категорії:

- виявлення аномальної поведінки;
- виявлення зловживань.

Технологія виявлення атак шляхом ідентифікації аномальної поведінки заснована на наступній гіпотезі. Аномальна поведінка користувача (тобто атака або яке–небудь ворожу дію) часто проявляється як відхилення від нормальної поведінки. Прикладом аномальної поведінки може служити велике число з'єднань за короткий проміжок часу, високе завантаження центрального процесора тощо.

Якщо можна було б однозначно описати профіль нормальної поведінки користувача, то будь-яке відхилення від нього можна ідентифікувати, як аномальна поведінка. Однак аномальна поведінка не завжди є атакою. Наприклад. одночасну посилку великого числа запитів від адміністратора мережі система виявлення атак може ідентифікувати як атаку типу «відмова в обслуговуванні» (denial of service).

При використанні системи з такою технологією можливі два крайніх випадки :

- виявлення аномальної поведінки, яка не є атакою, і віднесення її до класу атак;
- пропуск атаки, яка не підпадає під визначення аномальної поведінки.

1.5 Технологія виявлення аномалій

Цей випадок більш небезпечний, ніж помилкове віднесення аномальної поведінки до класу атак. При налаштуванні і експлуатації систем цієї категорії адміністратори стикаються з наступними проблемами:

− побудова профілю користувача є важким в формалізації і трудомістким завданням, що вимагає від адміністратора великої попередньої роботи;

− визначення граничних значень характеристик поведінки користувача для зниження ймовірності появи одного з двох вищеназваних крайніх випадків.

Технологія виявлення аномалій орієнтована на виявлення нових типів атак. Однак недолік її в необхідності постійного навчання. Поки технологія виявлення аномалій не отримала широкого розповсюдження, і ні в одній комерційно поширюваній системі вона не використовується. Пов'язано це з тим, що дану технологію важко реалізувати на практиці. Однак зараз намітився певний інтерес до неї. Суть іншого підходу до виявлення атак, виявлення зловживань, полягає в описі атаки у вигляді сигнатури (signature) і пошуку даної сигнатури в контрольованому просторі (мережевому трафіку або журналі реєстрації). В якості сигнатури атаки може виступати шаблон дій або рядок символів, що характеризують аномальну діяльність. Ці сигнатури зберігаються в базі даних, аналогічної тій, яка використовується в антивірусних системах. Слід зауважити, що антивірусні резидентні монітори є окремим випадком системи виявлення атак, але оскільки ці напрямки спочатку розвивалися паралельно, то прийнято розділяти їх. Тому дана технологія виявлення атак дуже схожа на технологію виявлення вірусів при цьому система може виявити всі відомі атаки. Однак системи даного типу можуть виявляти нові, ще невідомі види атак.

Підхід, реалізований у таких системах, досить простий, і саме на ньому базуються практично всі пропоновані сьогодні на ринку системи виявлення атак. Однак при експлуатації цих систем адміністратори стикаються з проблемами. Перша проблема полягає у створенні механізму опису сигнатур, тобто мови опису атак. Друга проблема, пов'язана з першою, полягає в тому, як описати атаку, щоб зафіксувати всі можливі її модифікації. Слід зазначити, що перша проблема вже частково вирішена в деяких продуктах. Наприклад, компанією lnternet Security Systems. Inc. реалізована система опису мережевих атак AdvancedPackets Exchange, і з її допомогою розроблена система аналізу захищеності Internet Scanner. Найбільш популярна класифікація за способом збору інформації про атаку [4]:

− виявлення атак на рівні мережі (network-based);

− виявлення атаки на рівні хоста (host-based);

− виявлення атак на рівні додатку (applicacion-based).

Система першого типу (network-based) працює по типу сніфферу, «прослуховуючи » трафік в мережі і визначаючи можливі дії зловмисників. Пошук атаки йде за принципом «від хоста до хоста». Системи, що входять в перший клас, аналізують мережевий трафік, використовуючи, як правило, сигнатури атак і аналіз «на льоту». Метод аналізу «нальоту» полягає в моніторингу мережевого трафіку в реальному або близькому до реального часі і використанні відповідних алгоритмів виявлення. Часто використовується механізм пошуку в трафіку певних рядків які можуть характеризувати несанкціоновану діяльність.

Системи другого типу (host-based) призначені для моніторингу, детектування та реагування на дії зловмисників на певному хості. Система, розташовуючись на хості що захищається, перевіряє і виявляє спрямовані проти неї дії. Ці системи аналізують реєстраційні журнали операційної системи або програми. Аналіз журналів реєстрації є одним з найперших реалізованих методів виявлення атак. Він полягає в аналізі журналів реєстрації (log, audit trail), створюваних операційною системою, прикладним програмним забезпеченням, маршрутизаторами тощо. Записи журналу реєстрації аналізуються й інтерпретуються системою виявлення атак. До переваг цього методу відноситься простота його реалізації. Однак за цією простотою ховається ряд недоліків:

− для достовірного виявлення тієї чи іншої підозрілої діяльності необхідна реєстрація в журналах великого обсягу даних, що негативно позначається на швидкості роботи контрольованої системи;

− при аналізі журналів реєстрації дуже важко обійтися без допомоги фахівців, що істотно знижує коло поширення цього методу;

− до теперішнього моменту немає уніфікованого формату зберігання журналів;

− аналіз записів у журналах реєстрації здійснюється не в реальному режимі часу, тому цей метод не може бути застосований для раннього виявлення атак в процесі їх розвитку [7,8].

1.6 Мережеві системи виявлення вторгнень

Мережеві системи виявлення вторгнень (МСВВ) переглядають весь мережевий трафік (або трафік певної ділянки мережі) і при виявленні будь–яких відхилень у ньому сигналізують про це. Формальні МСВВ працюють за принципом антивірусної програми–пакети, що потрапляють на сенсори, по-

рівнюються з БД сигнатур і, у разі виявлення збіги, оголошується тривога. На жаль, навіть формальних МСВВ стає недостатньо для надійного захисту мережі. За даними CERT, кількість відомих нових методів вторгнення тільки за 2010 рік перевищило 25 000. Це означає, що в середньому, щодня з'являється близько 70 нових атак. Фізично неможливо оновлювати БД сигнатур формальних МСВВ за такі проміжки часу. Крім того, збільшення обсягу сигнатур негативно позначається на продуктивності систем. Вирішенням цієї проблеми є застосування систем виявлення вторгнень на основі виявлення аномальної активності або евристичних МСВВ.

Проблеми класичних методів:

1) Теоретично існує нескінченна кількість методів і варіантів атак, і для їх виявлення знадобиться БД нескінченного розміру. Таким чином, є потенційна можливість, що якась атака, не включена в базу даних, може бути успішно здійснена.

2) Сучасні методи виявлення аномалій викликають велике число помилкових тривог. Таким чином, можуть бути скомпрометовані легальні мережеві події.

3) Сучасні методи виявлення зловживань мають досить високу ймовірність пропуску атаки.

Бурхливий розвиток комп'ютерних мереж та інформаційних технологій породжує безліч проблем, пов'язаних з безпекою інформаційних ресурсів. У зв'язку з недосконалістю існуючих методів захисту комп'ютерних систем від мережевих атак розробка нових методів захисту інформації, що дозволяють підвищити рівень захищеності комп'ютерних систем від несанкціонованого впливу, є актуальною і затребуваною.

Існує три основні підходи, які використовуються при виявленні та класифікації мережевих атак:

- статистичний аналіз;
- експертні системи;
- нейронні мережі.

Крім того, розвиваються підходи, засновані на і генетичних алгоритмах і імуноклітинних методах. Статистичний аналіз знаходить застосування, як правило, при виявленні аномального поведінки. Відхилення від середнього значення(тобто дисперсія) профілю нормальної поведінки дає сигнал адміністратору про те, що зафіксована атака. Середні частоти і величини змінних обчислюються для кожного типу нормального поведінки (наприклад, кількість входів в систему, кількість відмов у доступі, час доби тощо). Про мож-

ливі атаки повідомляється, коли спостерігаються значення випадають з нормального діапазону, тобто перевищують заданий поріг.

Експертна система – це система, яка в контексті виявлення атак приймає рішення про приналежність того чи іншої події до класу атак на підставі наявних правил. Ці правила засновані на досвіді фахівців і зберігаються в спеціальному сховищі. У більшості випадків правила експертної системи спираються на так звані сигнатури, які й шукаються в контрольованому просторі. Одним з найбільш ефективних засобів масового розпаралелювання та прискорення процесів обробки і передачі потоків даних в задачах виявлення закономірностей, розпізнавання образів і класифікації даних є штучні нейронні мережі (НМ). Природним прототипом штучних НМ є біологічний мозок і центральна нервова система людини і тварин. Можливості штучних та біологічних НМ можуть значно розширитися при колективном (Мультиагентний) вирішенні складних інтелектуальних завдань (data mining, knowledge discovery тощо). Висока складність і розмірність багатьох задач виявлення закономірностей, розпізнавання образів і класифікації даних, а також часто виникає необхідність їх вирішення в реальному часі вимагають масового паралелізму і самоорганізації розподілених обчислень на базі НМ [7].

2 АНАЛІЗ АНОМАЛЬНИХ СТАНІВ ТРАФІКА КОМП'ЮТЕРНОЇ МЕРЕЖІ ЯК РОЗВ'ЯЗОК ЗАДАЧІ КЛАСИФІКАЦІЇ

2.1 Застосування нейронних мереж для задач класифікації

Рішення завдання класифікації є одним з найуспішніших застосувань нейронних мереж. Завдання класифікації розуміється як завдання віднесення зразка до одного з декількох попарно непересічних множин. Найчастіше розглядається двійкова класифікація. Прикладом таких завдань може бути, наприклад, завдання визначення кредитоспроможності клієнта банку, медичні завдання, в яких необхідно визначити, наприклад, результат захворювання, рішення задач управління портфелем цінних паперів (продати купити або "притримати" акції залежно від ситуації на ринку), виявлення аномалій мережевого трафіку, завдання визначення життєздатних і схильних до банкрутства фірм.

Дві головні області застосування мереж з прямим зв'язком: задачі класифікації та моделювання часових рядів. Відмінність між завданнями цих двох типів полягає в наявності (тимчасової) впорядкованості прикладів.

Розглянемо, як нейронні мережі з прямим зв'язком (або багатошарові персептрони – MLP, Multilayer Peiceptron) використовуються в задачах класифікації. У чому, власне, полягає ця задача? По-перше, в будь-якій задачі класифікації потрібно віднести наявні статичні зразки (рукописні літери, звукові сигнали, характеристики фінансового становища, стан трафіку) до певних класів. Різноманітність прикладів, що виникають у реальному світі, практично нескінченна. Ефективність класифікації залежить від способу подання цих форм. У числі інших тут є наступні способи: розпізнавання образів, структурне уявлення і статистичне уявлення. У структурному розпізнаванні образів зразки описуються тим, як вони складені зі своїх компонент, тобто структурою, подібно до того, як це робиться в граматиці мови, розпізнавання в цьому випадку ґрунтується на застосуванні певних синтаксичних правил. При статистичному підході необхідно віднести наявні статичні зразки (характеристики ситуації на ринку, дані медогляду, аномалій трафіку, інформація про клієнта) до певних класів. Різноманітні ступені складності в представленні класів представленні на рис.2.1. Можливо кілька способів подання даних. Найбільш поширеним є спосіб, при якому зразок представляється вектором. Компоненти цього вектору являють собою різні характеристики зразка, які впливають на прийняття рішення про те, до якого класу мо-

жна віднести даний зразок. Наприклад, для медичних завдань в якості компонентів цього вектору можуть бути дані з медичної карти хворого. Таким чином, на підставі деякої інформації про приклад, необхідно визначити, до якого класу його можна віднести. Класифікатор таким чином відносить об'єкт до одного з класів відповідно до певного розбиттям N–мірного простору, який називається простором входів, і розмірність цього простору є кількістю компонентів вектору.

а) б) в)

Рисунок 2.1 – Ступені складності в представленні класів:
а – лінійно роздільні, б – нелінійно роздільні,
в – нероздільні.

При вирішенні задачі розпізнавання статистичними методами найважливіше значення має правильний вибір способу статистичного представлення об'єкта. Тим самим, потрібно проробити попередню обробку даних. Для того щоб вибрати характерні відмінні ознаки об'єктів, потрібно, як правило, серйозне вивчення вихідної проблеми. Наприклад, про моделі банкрутства банків важливе значення мають такі показники, як досвід в управлінні фондами та відповідність вимогам адекватності капіталу. Різні набори ознак призводять до різних розподілів. При цьому в різних варіантах дисперсія і властивості опуклості кластерів у вхідному просторі можуть сильно відрізнятися, відповідно, при їх поділі потрібно проводити кордони різного ступеня складності – від лінійних до сильно нелінійних. Чим краще була зроблена попередня обробка, тим легше буде вирішена задача класифікації.

Перш за все потрібно визначитися з вибором рівня складності. У реальних ситуаціях часто буває так, що є лише відносно невелике число зразків, а структура даних дозволяє виділити наступні три рівні складності. Перший (найпростіший) – коли класи можна розділити прямими лініями (або гіперплощинами, якщо простір має розмірність більше двох). Цей випадок називається лінійною відокремлюваністю.

Функції, які не реалізуються одношаровою нейронною мережею, називаються лінійно нероздільними. Наявність таких функцій обмежує одноша-

рові мережі завданнями класифікації, в яких безлічі точок (відповідних вхідним значенням) можуть бути розділені геометрично. У двовимірному випадку для цього використовується пряма лінія, в тривимірному – площина, а при більшій розмірності – гіперплощина.

Так як лінійна роздільність обмежує можливості персептронного подання, то важливо знати, чи є дана функція роздільною. Тому одношарові персептрони застосовуються для вирішення відносно простих завдань.

У другому випадку однієї гіперплощини для розділення недостатньо (нелінійна роздільність), а в третьому випадку класи перетинаються, і тому розділити їх можна тільки в імовірнісному сенсі.

В ідеальному варіанті попередня обробка повинна дати такий набір ознак, щоб завдання виявилося лінійно роздільною, класифікація після цього суттєво спрощується. На жаль, при вирішенні реальних завдань ми маємо обмежену кількість зразків, на підставі яких і проводиться побудова класифікатора. При цьому ми не можемо провести таку попередню обробку даних, при якій буде досягнута лінійна роздільність зразків [9].

2.2 Використання нейронних мереж в якості класифікатора

Мережі з прямим зв'язком є універсальним засобом апроксимації функцій, що дозволяє їх використовувати у вирішенні задач класифікації. Як правило, нейронні мережі виявляються найбільш ефективним способом класифікації, тому що генерують фактично велике число регресійних моделей (які використовуються у вирішенні задач класифікації статистичними методами).

Багаті можливості відображення особливо важливі в тих випадках, коли на основі кількох оцінок будується Високорівнева процедура прийняття рішень. Відомо багато додатків нейронних мереж з прямим зв'язком до завдань класифікації. Як правило, вони виявляються ефективніше інших методів, тому що нейронна мережа генерує нескінченне число нелінійних регресійних моделей.

На жаль, у застосуванні нейронних мереж у практичних завданнях виникає ряд проблем. По-перше, заздалегідь не відомо, якої складності (розміру) може знадобитися мережа для досить точної реалізації відображення. Ця складність може виявитися надмірно високою, що потребує складної архітектури мереж. Так Мінський у своїй роботі "Персептрони" довів, що найпростіші одношарові нейронні мережі здатні вирішувати тільки лінійно роздільні

завдання. Це обмеження можна подолати при використанні багатошарових нейронних мереж. У загальному вигляді можна сказати, що в мережі з одним прихованим шаром, вектор, відповідний вхідному зразку, перетворюється в прихований шар в деякий новий простір, який може мати іншу розмірність, а потім гіперплощини, відповідні нейронам вихідного шару, поділяють його на класи. Таким чином мережа розпізнає не тільки характеристики вихідних даних, але і характеристики характеристик, сформовані прихованим шаром [10].

Все це підкреслює важливість етапу попередньої обробки даних. Чим більш компактно представлені характеристики зразків, тим менше залежність від настроюваних параметрів мережі (0 або 1).

2.3 Попередня обробка даних

Підвищення якості навчання нейронної мережі можливе при використанні ефективних методів попередньої обробки даних. Для обробки даних перед навчанням нейронної мережі пропонується використовувати метод, заснований на понятті профілю компактності і комбінаторних формулах для ефективного обчислення функціонала ковзкого контролю. Метод застосовується для підготовки даних в задачах класифікації [2]. Щоб звести своє завдання попередньої обробки даних у моделюванні до задачі попередньої обробки даних при класифікації даних (яка має рішення, що використовує профіль компактності), пропонується провести кластерний аналіз на вихідних параметрах нейронної мережі. Таким чином, будемо мати задачу класифікації, для якої відоме рішення попередньої обробки даних.

Метод, починаючи з повної вибірки, послідовно виключає об'єкти. На кожному кроці вибирається той об'єкт, виключення якого мінімізує функціонал. Виявляється, що процес відсіву об'єктів розбивається на дві стадії. Спочатку виключаються шумові, потім виключаються неінформативні периферійні об'єкти. Процес зупиняється, коли залишаються об'єкти, виключення яких помітно збільшує функціонал, тоді в масиві даних залишаються опорні об'єкти.

Основним результатом застосування комбінаторної формули для оцінки функціоналу повного ковзкого контролю є те, що вона однаково добре підходить як для виключення шумових об'єктів, так і для скорочення множини прецедентів, будучи при цьому ефективно обчислюваним, точним значенням функціоналу.

Метод спирається на припущення, яке називається гіпотезою компактності: схожі об'єкти набагато частіше лежать в одному класі, ніж в різних. У цьому випадку межа між класами має досить просту форму, а класи утворюють компактно локалізовані області в просторі об'єктів (у математичному аналізі компактними називаються обмежені замкнуті множини, гіпотеза компактності не має нічого спільного з цим поняттям).

Як правило, об'єкти навчання не є рівноцінними. Серед них можуть знаходитися типові представники класів – еталони. Якщо класифікується об'єкт близький до ідеалу, то, швидше за все, він належить тому ж класу. Ще одна категорія об'єктів – неінформативні, або периферійні. Вони щільно оточені іншими об'єктами того ж класу. Якщо їх видалити з вибірки, це практично не позначиться на якості навчання. Нарешті, у вибірку може потрапити деяка кількість шумових викидів – об'єктів, що знаходяться в чужому класі. Зазвичай їх видалення тільки покращує якість класифікації.

Виключення з вибірки шумових і неінформативних об'єктів дає кілька переваг одночасно: підвищується якість класифікації, скорочується обсяг збережених даних і зменшується час класифікації, що витрачається на пошук найближчих еталонів [5].

Перейдемо до розгляду функціоналу вибірки, що мінімізується. Нехай є множина об'єктів X і множина імен класів Y. Задана навчальна вибірка пар «об'єкт–відповідь»:

$$X^{im}=\{(x_1,y_1),\ldots,(x_m,y_m)\}\in X\times Y \qquad (2.1)$$

Нехай на множині об'єктів задана функція відстані $\rho(x, x')$. Ця функція повинна бути досить адекватною моделлю подібності об'єктів. Чим менше значення цієї функції, тим більше схожі об'єкти x, x′.

Для довільного об'єкта u розташуємо об'єкти навчальної вибірки xi в порядку зростання відстаней до u:

$$\rho(u,x_{1u})\le\rho(u,x_{2u})\le\cdots\le\rho(u,x_{mu}), \qquad (2.2)$$

де через x_{iu} позначається елемент навчальної вибірки, який є *i*–м сусідом об'єкта *u*. Аналогічне позначення введемо і для відповіді на *i*–му сусіді – y_{iu}.

Кожен об'єкт $u \in X$ породжує свою перенумерацію вибірки.

Розглядається метод найближчого сусіда, який відносить об'єкт u, що класифікується, до того класу, якому належить найближчий до u об'єкт навчальної вибірки: $a(u, X^m) = y_{1u}$.

Профіль компактності вибірки X^m є функція:

$$R(j, X^m) = \frac{1}{m}\sum_{i=1}^{m}[y_i \neq y_{ix}] \quad (2.3)$$

Іншими словами, профіль компактності R (j) – це частка об'єктів вибірки, для яких j–й сусід лежить в іншому класі.

Профіль компактності є формальним виразом гіпотези компактності – припущення про те, що схожі об'єкти набагато частіше лежать в одному класі, ніж в різних. Вибірка X^L розбивається всілякими $N = C_L^k$ способами на дві непересічні підвибірки: $X^L = X_n^m \cup X_n^k$, де X_n^m – навчальна підвибірка довжини m; X_n^k – контрольна підвибірка довжини; k = L – m, n = 1,..., N – номер розбиття.

Для кожного розбиття n будується алгоритм $a_n(u, X)_n^m$. Функціонал повного ковзкого контролю (complete cross–validation, CCV) визначається як середня (по всіх розбиттям) помилка на контролі:

$$CCV(X^L) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{k}\sum_{x_2 \in x_n^2}[a_n(x_i, X_n^m) \neq y_i] \quad (2.4)$$

Функціонал повного ковзкого контролю характеризує узагальнюючу здатність методу найближчого сусіда.

Справедлива формула для ефективного обчислення CCV через профіль компактності:

$$CCV(X^L) = \sum_{j=1}^{k} R(j, X^L)\Gamma(j), \quad (2.5)$$

де $\Gamma(j) = \frac{C_{L-1-j}^{m-1}}{C_{L-1}^{m}}$.

Комбінаторний множник Г(j) швидко убуває із зростанням j. Для мінімізації функціоналу CCV достатньо, щоб при малих j профіль R (j, X^L) брав значення, близькі до нуля. Це означає, що близькі об'єкти повинні лежати переважно в одному класі. Таким чином, профіль дійсно є формальним виразом гіпотези компактності [5].

Пропонується використовувати кластерний аналіз для розділення значень виходів мережі на групи, щоб звести задачу попередньої обробки даних при моделюванні за допомогою нейронної мережі до задачі попередньої обробки даних при класифікації, в якій використовується теорія профілю компактності.

В якості характеристики близькості вихідних значень нейронної мережі взято евклідову відстань між точками. Для довільного вектору v з числом елементів n евклідова норма знаходиться наступним чином:

$$\|v\| = \sqrt{\sum_{i=1}^{m} |v_i|2} \tag{2.6}$$

Евклідова відстань є найпопулярнішою метрикою в кластерному аналізі: вона відповідає інтуїтивним уявленням про близькість і, крім того, дуже вдало вписується своєю квадратичною формою у традиційно статистичні конструкції. Геометрично вона найкраще об'єднує об'єкти в кулястих скупченнях, які дуже типові для слабо корельованих сукупностей.

На першому кроці кластерного аналізу кожен об'єкт вважається окремим кластером. На наступному кроці об'єднуються два найближчих об'єкта, які утворюють новий клас, визначаються відстані від цього класу до всіх інших об'єктів, і розмірність матриці відстаней скорочується на одиницю. Процедура повторюється на поточній матриці відстаней, поки не буде досягнуто деяке число кластерів.

Таким чином, пропонований метод попередньої обробки даних дає більш якісне навчання нейронної мережі. Попередня обробка полягає у видаленні з маси–ву суперечливих прикладів. Пошук таких прикладів заснований на теорії профілю компактності в задачі класифікації. Щоб використовувати відомі рішення (теорію профілю компактності) в задачі моделювання, необхідно за допомогою кластерного аналізу виділити групи над значеннями вихідних параметрів нейронної мережі [9].

2.4. Підготовка вихідних даних

Для побудови класифікатора необхідно визначити, які параметри впливають на прийняття рішення про те, до якого класу належить зразок. При цьому можуть виникнути дві проблеми. По-перше, якщо кількість параметрів мала, то може виникнути ситуація, при якій один і той же набір вихідних даних відповідає прикладам, що знаходяться в різних класах. Тоді неможливо навчити нейронну мережу, і система не буде вірно працювати (неможливо знайти мінімум, який відповідає такому набору вихідних даних). Вихідні дані обов'язково повинні бути несуперечливі. Для вирішення цієї проблеми необхідно збільшити розмірність простору ознак (кількість компонентів вхідного вектору, відповідного зразку). Але при збільшенні розмірності простору ознак може виникнути ситуація, коли число прикладів може стати недостатнім для навчання мережі, і вона замість узагальнення просто запам'ятає приклади з навчальної вибірки і не зможе вірно функціонувати. Таким чином, при визначенні ознак необхідно знайти компроміс з їх кількістю.

Далі необхідно визначити спосіб представлення вхідних даних для нейронної мережі, тобто визначити спосіб нормування. Нормування необхідне, оскільки нейронні мережі працюють з даними, представленими числами в діапазоні 0.. 1, а вихідні дані можуть мати довільний діапазон або взагалі бути нечисловими даними. При цьому можливі різні способи, починаючи від простого лінійного перетворення в необхідний діапазон і закінчуючи багатовимірним аналізом параметрів і нелінійним нормуваннями залежно від впливу параметрів один на одного.

Завдання класифікації при наявності двох класів може бути вирішена на мережі з одним нейроном у вихідному шарі, який може приймати одне з двох значень 0 або 1, залежно від того, до якого класу належить зразок. За наявності декількох класів виникає проблема, пов'язана з поданням цих даних для виходу мережі. Найбільш простим способом представлення вихідних даних у такому випадку є вектор, компоненти якого відповідають різним номерам класів. При цьому і-та компонента вектору відповідає і-му класу. Всі інші компоненти при цьому встановлюються в 0. При інтерпретації результату зазвичай вважається, що номер класу визначається номером виходу мережі, на якому з'явилося максимальне значення. Наприклад, якщо в мережі з трьома виходами ми маємо вектор вихідних значень (0.2,0.6,0.4), то ми бачимо, що максимальне значення має друга компонента вектору, значить клас, до якого належить цей приклад, –2. При такому способі кодування іноді вво-

диться також поняття впевненості мережі в тому, що приклад відноситься до цього класу. Найбільш простий спосіб визначення впевненості полягає у визначенні різниці між максимальним значенням виходу і значенням іншого виходу, яке є найближчим до максимального. Наприклад, для розглянутого вище прикладу впевненість мережі в тому, що приклад відноситься до другого класу, визначиться як різниця між другою і третьою компонентою вектору і дорівнює 0,6–0,4 = 0,2. Відповідно чим вище впевненість, тим більше вірогідність того, що мережа дала правильну відповідь. Цей метод кодування є найпростішим, але не завжди найоптимальнішим способом представлення даних.

Відомі й інші способи. Наприклад, вихідний вектор являє собою номер кластера, записаний в двійковій формі. Тоді при наявності 8 класів нам буде потрібно вектор з 3 елементів, і, скажімо, 3 класу буде відповідати вектор 011. Але при цьому у разі отримання невірного значення на одному з виходів ми можемо отримати невірну класифікацію (невірний номер кластера), тому має сенс збільшити відстань між двома кластерами за рахунок використання кодування виходу за кодом Хеммінга, який підвищить надійність класифікації [11].

Інший підхід полягає в розбитті завдання з до класами на k * (k–1) / 2 підзадач з двома класами (2 на 2 кодування) кожна. Під підзадачею в даному випадку розуміється те, що мережа визначає наявність однієї з компонент вектору [11].

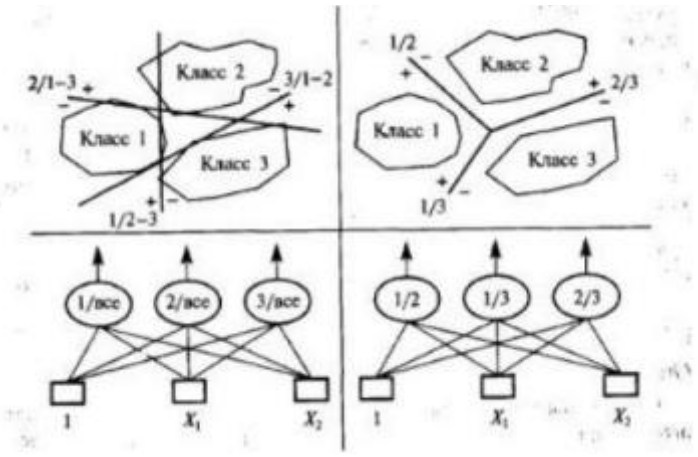


Рисунок 2.2 – Кодування виходу на прикладі двомірної задачі з трьома класами

Тобто вихідний вектор розбивається на групи по два компоненти в кожній таким чином, щоб у них увійшли всі можливі комбінації компонент ви-

хідного вектору. Число цих груп мо–жна визначити як кількість невпорядкованих вибірок по два з вихідних компонент. з комбінаторики відповідно до формули:

$$A_k^n = \frac{k!}{n!\,(k-n)!} = \frac{k!}{2!\,(k-2)!} = \frac{k(k-1)}{2} \qquad (2.7)$$

Тоді, наприклад, для завдання з чотирма класами ми маємо 6 виходів (підзадач) розподілених відповідно до даних поданих в таблиці.2.1.

Таблиця 2.1 – Розподілення підзадач

N підзадача (виходу)	Компоненти виходу
1	1–2
2	1–3
3	1–4
4	1–5
5	1–6
6	1–7

Де 1 на виході говорить про наявність однієї з компонент. Тоді ми можемо перейти до номера класу по результату розрахунку мережею наступним чином: визначаємо, які комбінації отримали одиничне (точніше близьке до одиниці) значення виходу (тобто які підзадачі у нас активувалися), і вважаємо, що номер класу буде той, який увійшов в найбільшу кількість активованих підзадач (табл. 2.2).

Таблиця 2.2 – Відношення активованих виходів

N класу	Акт. виходи
1	1,2,3
2	1,4,5
3	2,4,6
4	3,5,6

Це кодування в багатьох задачах дає кращий результат, ніж класичний спосіб кодування.

2.5 Вибір архітектури мережі

При виборі архітектури мережі звичайно випробовується кілька конфігурацій з різною кількістю елементів. При цьому основним показником є обсяг навчальної множини і узагальнююча здатність мережі.

Здатність аналітичної моделі, побудованої на основі навчання (нейронної мережі, дерева рішень, карти Кохонена та ін.) видавати правильні результати не тільки для прикладів, які брали участь у процесі навчання, а й для будь-яких нових, які не брали участь в ньому. Узагальнююча здатність є найважливішою властивістю аналітичної моделі, що здобувається в процесі навчання.

Якщо з якоїсь причини модель не набула спроможність до узагальнення, її практичне використання безглузде, оскільки на будь-який приклад з навчальної множини вона завжди буде видавати правильний результат, а на будь-який новий приклад – довільне значення. Здатність до узагальнення модель може придбати тільки за рахунок великої кількості різноманітних комбінацій вхідних і цільових значень в прикладах навчальної множини. При цьому число навчальних прикладів повинно в кілька разів перевищувати інформаційну ємність моделі.

Для перевірки узагальнюючої здатності моделі використовується тестова множина, сформована із прикладів навчальної вибірки, що не використовувалися в процесі навчання. Якщо модель забезпечує низьку вихідну помилку як на навчальній, так і на тестовій множині, то з високою часткою впевненості можна стверджувати, що вона придбала узагальнюючу здатність. Якщо низька помилка має місце тільки на навчальній множині, а на тестовій вона висока, то, швидше за все, здатність до навчання була досягнуто. Для того щоб поліпшити здатність до узагальнення і усунути небезпеку перенавчання, застосовуються також зменшення ваг і їх виключення (проріджування дерева). При цьому змінюється архітектура мережі : видаляються деякі зв'язки і вивчається, який вплив вони чинили на ефективність.

Після того, як вибір моделі (тобто архітектури мережі) зроблений і проведена її перевірка, її можна використовувати для передбачення, пояснення та діагностики. З її допомогою можна визначати, до якого з класів належить пред'явлений зразок, або вивчати можливі зв'язки між різними характеристиками об'єктів і прийнятим рішенням, або виявляти причини, які потягли за собою неправильну класифікацію.

Хоча на нейронні мережі часто дивляться як на «чорну скриню», є деякі можливості з'ясувати вплив кожного фактору на рішення, прийняте в задачі класифікації. На даний час формального методу, що дозволяє витягувати з навченої мережі інформацію про завдання або про правила класифікації, не існує. Як правило, аналіз мереж проводиться евристично.

По завершенні всіх зазначених процедур мережу можна використовувати в складних комплексах прийняття рішень у поєднанні з традиційними підходами, а також з іншими мережами, навченими незалежно і налаштованими на інші характеристики об'єктів.

2.6 Алгоритм побудови класифікатора на основі нейронних мереж

1) Робота з даними

– скласти базу даних із прикладів, характерних для даної задачі

– розбити всю сукупність даних на дві множини : навчальна і тестова (можливо розбивка на 3 множини : навчальну, тестову і підтверджуючу).

2) Попередня обробка

– вибрати систему ознак, характерних для даного завдання, і перетворити дані відповідним чином для подачі на вхід мережі (нормування, стандартизація і тощо). В результаті бажано отримати лінійно відокремлюваний простір множини зразків;

– вибрати систему кодування вихідних значень (класичне кодування, 2 на 2 кодування і тощо).

3) Конструювання, навчання та оцінка якості мережі

– вибрати топологію мережі : кількість шарів, число нейронів у шарах і тощо;

– вибрати функцію активації нейронів (наприклад "сигмоїда");

– вибрати алгоритм навчання мережі;

– оцінити якість роботи мережі на основі підтверджуючої множини або іншому критерію, оптимізувати архітектуру (зменшення ваг, проріджування простору ознак);

– зупиниться на варіанті мережі, який забезпечує найкращу здатність до узагальнення та оцінити якість роботи по тестовій множині.

4) Використання та діагностика

– з'ясувати ступінь впливу різних факторів на прийняте рішення (евристичний підхід);

– переконатися, що мережа дає необхідну точність класифікації (число неправильно розпізнаних прикладів мало);

– при необхідності повернуться на етап 2, змінивши спосіб представлення зразків або змінивши базу даних;

– практично використовувати мережу для вирішення завдання.

Для того, щоб побудувати якісний класифікатор, необхідно мати якісні дані. Жоден з методів побудови класифікаторів, заснований на нейронних мережах або статистичний, ніколи не дасть класифікатор потрібної якості, якщо наявний набір прикладів не буде достатньо повним і представницьким для того завдання, з якою доведеться працювати системі.

Методи виявлення комп'ютерних атак на основі нейронних мереж застосовують для попередньої класифікації аномалій в інформаційній системі. Вони базуються на ідентифікації нормальної поведінки системи по функції розподілу отримання пакетів даних (виконання заданих команд оператора), навчанні нейронної мережі та порівняльного аналізу подій за навчальною вибіркою. Аномальне відхилення в інформаційній системі виявляється тоді, коли ступінь довіри нейромережі своєму рішенню лежить нижче заданого порогу. Передбачається, що застосуванню моделі нейронних мереж для реалізації механізмів захисту інформації інформаційної системи від комп'ютерних атак передує навчання цих мереж заданих алгоритмах нормального функціонування. Недоліками методів виявлення комп'ютерних атак з використанням нейронної мережі є складний математичний апарат, який недостатньо ефективно працює в системах квазіреального масштабу часу, і складність навчання мережі для виявлення невідомих атак [11].

2.7. Аналіз вторгнень за допомогою файлів системних журналів

Покажемо як керування ризиками, пов'язаними з мережними атаками реалізується на практиці. Насамперед, необхідно встановити й налаштувати систему моніторингу мережевого трафіка. Після цього можна приступати до аналізу підозрілого трафіку, подій і різного роду мережних атак, оцінювати ризики й управляти ними.

До засобів моніторингу мережних атак відносяться такі програмні продукти, як SNORT (IDS), для запобігання атак використовуються різні системи типу Firewall. Приклад підозрілого трафіка було взято із загальнодоступної бази KDD99 [12], обрана інформація заслуговує уваги експерта, розглянемо наступний фрагмент журналу реєстрації подій програми Tcpdump (фра-

гмент журнального файлу ZoneAlarm різновид FireWall) представлений на рис.2.3.

```
        FWIN,2005/08/19,14:25:04+4:00
GMT,61.235.154.103:44666,194.85.70.31:1027,UDP
        FWIN,2005/08/19,14:39:36+4:00
GMT,220.168.156.70:37740,194.85.70.31:1026,UDP
        FWIN,2005/08/19,14:39:36+4:00
GMT,220.168.156.70:37740,194.85.70.31:1027,UDP
        FWIN,2005/08/19,14:44:34+4:00
GMT,222.241.95.69:32875,194.85.70.31:1027,UDP
```

Рисунок 2.3 – фрагмент журналу реєстрації подій програми Tcpdump

Ця роздруківка демонструє спроби промацування ЕОМ з IP–адресою 194.85.70.31 на предмет відгуків з боку портів 1026 і 1027 (протоколи cap і exosee). Зондування проводиться за декількох різних адрес (61.235.154.103, 220.168.156.70 і 222.241.95.69). Об'єктом атаки в даному випадку є робоча станція, яка не підтримує ці протоколи.

Існує досить багато стандартних діагностичних засобів, зокрема в ОС UNIX. Серед цих засобів, програми ведення журнальних файлів ОС і деяких додатків, на–приклад, Apache (файли access_log, error_log і ssl_access_log), Samba, Squid та ін.

Для відстеження роботи ОС і додатків зазвичай передбачається система журнальних файлів, яка фіксує всі події (прихід запитів, відповідність запитів певним критеріям і тощо).

Розглянемо використання журнальних файлів на прикладі аналізу успішної атаки вторгнення через додаток SSH.

Якщо виникла підозра щодо можливого вторгнення, треба починати з перегляду файлів secure і messages (каталог / var / log / ОС LINUX). У нашому випадку атака почалася в п'ятницю ввечері (8-го січня 2017 року). На рис.2.4 представлені фрагменти журнальних файлів, що ілюструють характер атаки.

Із записів видно, що машина була атакована з 7 точок. Чотири розташовані в США (IP=166.70.74.35; 207.232.63.45; 129.79.240.86 і 129.237.101.171), по одній в Італії, Румунії та Угорщини (IP = 80.98.194.185). Проводиться підбір параметрів доступу ім'я – пароль. Підбір тривав близько двох діб. Успішний варіант був знайдений машиною з Італії (IP-адреса = 80.18.87.243 Венеція). Практично відразу атака з боку всіх ЕОМ була перервана і хакер увійшов на ЕОМ, що атакувалася (імя_ЕВМ = fender) з машини з IP = 81.181.128.181 (Румунія).

```
Jan 8 18:23:18 fender sshd[15017]:
Illegal user anonymous from 207.232.63.45
Jan 8 18:23:20 fender sshd[15019]:
Illegal user bruce from 207.232.63.45 (Нью-Йорк, США)
Jan 8 18:23:22 fender sshd[15021]:
Illegal user chuck from 207.232.63.45
Jan 8 18:23:23 fender sshd[15023]:
Illegal user darkman from 207.232.63.45
…
Jan 9 13:15:13 fender sshd[16764]:
Illegal user bruce from 129.237.101.171
Jan 9 13:15:14 fender sshd[16766]:
Illegal user chuck from 129.237.101.171
Jan 9 13:15:16 fender sshd[16768]:
Illegal user darkman from 129.237.101.171
Jan 9 13:15:17 fender sshd[16770]:
Illegal user hostmaster from 129.237.101.171
….
Jan 10 15:25:34 fender sshd[28450]:
Did not receive identification string from 80.18.87.243\par
Jan 10 16:56:16 fender sshd[28457]:
Illegal user lynx from 80.18.87.243\par
Jan 10 16:56:17 fender sshd[28459]:
Illegal user monkey from 80.18.87.243\par
Jan 10 16:56:18 fender sshd[28461]:
Illegal user lion from 80.18.87.243\par
….
Jan 10 02:42:02 fender sshd[18064]:
Did not receive identification string from 166.70.74.35
Jan 10 03:09:13 fender sshd[18067]:
Illegal user admin from 166.70.74.35 (Солт Лейк Сити, США)
Jan 10 03:09:14 fender sshd[18069]:
Illegal user admin from 166.70.74.35
Jan 10 03:09:16 fender sshd[18071]:
Illegal user admin from 166.70.74.35
….
Jan 10 16:56:16 fender sshd[28457]:
Illegal user lynx from 80.18.87.243 (Венеция, Италия)
Jan 10 16:56:17 fender sshd[28459]:
Illegal user monkey from 80.18.87.243
Jan 10 16:56:18 fender sshd[28461]:
Illegal user lion from 80.18.87.243
….
Jan 10 16:56:40 fender sshd[28509]:
Failed password for root from 80.18.87.243 port 45208 ssh2
Jan 10 16:56:41 fender sshd[28511]:
Accepted password for root from 80.18.87.243 port 45298 ssh2
….
Jan 10 19:13:49 fender sshd[31152]:
Accepted password for root from 81.181.128.181 port 4943 ssh2
```

Рисунок 2.4 – Фрагменти журнальних файлів secure і messages

Для подальшого аналізу подій нами були використані дані з файлу. Bash_history, куди записуються всі команди виконувані користувачем в тер-

мінальному режимі. Записи цього файлу і результати роботи демона syslogd говорять про те, що через 3 хвилини після успішного вторгнення хакер заблокував роботу syslog.

Далі хакер заблокував доступ до системи інших користувачів, завантажив туди файл pass_file (обсяг 696 057 байт), що містить комбінації ім'я – пароль (невеликі фрагменти вмісту файлу представлені на рис. 2.5).

```
lynx lynx
monkey monkey
lion lion
heart heart
michel michel
alibaba alibaba
…..
root 123456
root 1234567
….
root 1234567890
root rootroot
root rootrootroot
root 123root123
root 987654321
….
root 4321
root 321
root root!
root root!@
root root!@#
….
```

Рисунок 2.5 – фрагмент вмісту файлу pass_file

Хакер розраховує на те, що користувач ЕОМ ледачий, і вибирає простий пароль (легше запам'ятати – легше підібрати). Крім того, хакер скопіював на зламану ЕОМ кілька скриптів і файл зі списком адрес – кандидатів на злом. Після цього машина включилася в роботу з підбору паролів на інших ЕОМ.

На жаль, факт атаки був встановлений лише вранці в понеділок. Спочатку було проведене часткове блокування. Хакер відчув недобре і видав команди last і ps, намагаючись зрозуміти, що відбувається, подальша його робота була повністю блокована.

Які висновки з цієї історії можна зробити? На атакованої робочої станції була встановлена SSH застарілої версії (що мала вразливість) і використаний досить простий пароль. З цієї причини потрібно своєчасно оновлювати ОС і версії додатків. Особливо небезпечними з точки зору атак є ніч і вихідні

дні. Якщо немає нагальної необхідності, краще на цей час блокувати доступ до ЕОМ або навіть виключати її.

Крім журнальних файлів ОС треба переглядати і відповідні файли додатків, наприклад, Firewall (BlackIce Defender, ZoneAlarm і тощо), Apache, баз даних тощо Якщо навіть у вашій зоні відповідальності тільки один комп'ютер, перегляд всіх важливих файлів досить трудомісткий. З цієї причини слід розглянути можливість використання спеціалізованих скриптів, які візьмуть цю роботу на себе, інформуючи вас в разі виявлення тривожних подій. Результати роботи скриптів повинні накопичуватися в базі даних. Ці дані можуть використовуватися для отримання даних про атакерів і формування ACL (списків управління доступом).

Хакер може спробувати знищити сліди свого перебування, стерши або очистивши певні журнальні файли. Цілком можливо, в нашому випадку хакер так би і поступив, відновивши перед відходом і доступ по SSH. З цієї причини слід заздалегідь потурбуватися про періодичне копіюванні журнальних файлів на недоступний для хакера пристрій або збереження їх у зашифрованому вигляді. Але хакер може діяти й жорстокіше, наприклад, зробивши розмітку системного диску. Хороша схема захисту повинна запобігати такого роду дії або дозволяти хоча б швидко відновлювати зруйновану конфігурацію системи.

Слід враховувати, що самі журнальні файли можуть стати об'єктом атаки типу DoS. Великий потік запитів, що надходять з декількох ЕОМ, і звернених до одного або декількох ресурсів машини, можуть привести до швидкого зростання журнальних файлів, переповнити дисковий запам'ятовуючий пристрій і блокувати роботу.

3 ВІРТУАЛЬНИЙ НЕЙРОМЕРЕЖЕВИЙ ПРОЦЕСОР РОЗВ'ЯЗКУ ЗАДАЧ ВИЯВЛЕННЯ АНОМАЛЬНИХ СТАНІВ ЗАСОБАМИ ПАКЕТУ MATLAB

На даний час існують такі пакети для проектування нейронних мереж :

SNNS(Stuttgart Neural Network Simulator) – потужна бібліотека, розроблена в Штуттгардському університеті. Велика частина коду написана ще на початку 90-х на чистому C, без використання об'єктно- орієнтованого підходу. Це сильно ускладнює подальший розвиток. Joone – сучасніший пакет, написаний на Java, що кілька відбивається на швидкості роботи. Перевагою є повністю об'єктна модель. Але сама по собі бібліотека малорозвинених, основний упор поставлений на візуалізацію, мабуть, тому розробники мало часу приділили розвитку ядра.

Matlab Neural Network Toolbox – це пакет розширення MATLAB, що містить засоби для проектування, моделювання, розробки та візуалізації нейронних мереж [13].

Основний недолік – дуже низька швидкість роботи. Загальним недоліком для всіх існуючих пакетів, є однопоточні обробки, а також представлення нейронної мережі виключно шарами. Шарувату уявлення спрощує розробку програми, але відсікає можливість проектування нейронних мереж з довільною топологією. Перевага пакету MATLAB полягає в тому, що при його використанні користувач не обмежений моделями нейронних мереж та їх параметрами.

3.1 Огляд можливостей системи MATLAB з точки зору створення віртуальних процесорів на базі НМ

Систему MATLAB (матрична лабораторія), розроблену програмістом Молером (C.B. Moler) як середовище програмування високого рівня для технічних обчислень, з кінця 70-х років широко використовували на великих ЕОМ. На початку 80-х років Дж. Літл (John Little) з фірми Math Works, Inc. розробив першу версію системи PC MATLAB для комп'ютерів класу IBM PC та Macintosh, з якої і почалася еволюція версій системи для персональних комп'ютерів.

Архітектурно система MATLAB складається з базової програми і декількох десятків так званих пакетів розширення, які у своїй сукупності забезпечують винятково широкий діапазон розв'язуваних задач. Інтеграція всіх

цих засобів у єдиному робочому середовищі забезпечує необхідну гнучкість використання сотень вбудованих функцій, які реалізують різноманітні математичні процедури та обчислювальні алгоритми.

Зараз можливості системи значно перевершують можливості первісної версії матричної лабораторії Matrix Laboratory. Нинішній MATLAB – це високоефективна мова інженерних і наукових обчислень. Він підтримує математичні обчислення, візуалізацію наукової графіки та програмування з використанням операційного оточення, що легко освоюється, коли завдання і їх рішення можуть бути представлені в нотації, близької до математичної. Найбільш відомі області застосування системи MATLAB [13]:

– математика і обчислення;
– розробка алгоритмів;
– обчислювальний експеримент, імітаційне моделювання, макетування;
– аналіз даних, дослідження та візуалізація результатів;
– наукова та інженерна графіка;
– розробка додатків, включаючи графічний інтерфейс користувача.

Система MATLAB – це одночасно і операційне середовище і мова програмування. Одна з найбільш сильних сторін системи полягає в тому, що мовою MATLAB можуть бути написані програми для багаторазового використання. Користувач може сам написати спеціалізовані функції і програми, які оформляються у вигляді М-файлів. У міру збільшення кількості створених програм виникають проблеми їх класифікації і тоді можна спробувати зібрати родинні функції в спеціальні папки. Це призводить до концепції пакетів прикладних програм (ППП), які представляють собою колекції М-файлів для вирішення певної задачі або проблеми.

MATLAB широко використовується в таких областях, як:

– обробка сигналів та зв'язок,
– обробка зображень і відео,
– системи управління,
– автоматизація тестування і вимірювань,
– фінансовий інжиніринг,
– обчислювальна біологія і т.п.

MATLAB являє собою основу всього сімейства продуктів MathWorks і є головним інструментом для вирішення широкого спектра наукових і прикладних задач, в таких областях як: моделювання об'єктів та розробка систем

управління, проектування комунікаційних систем, обробка сигналів та зображень, вимірювання сигналів і тестування, фінансове моделювання, обчислювальна біологія тощо.

Доступні наступні операції:

- інтерполяція і регресія;
- диференціювання та інтегрування;
- системи лінійних рівнянь;
- фур'є аналіз;
- власні значення і сингулярні числа матриць;
- звичайні диференціальні рівняння;
- розріджені матриці.

Розширення MATLAB надають спеціалізований функціонал в таких областях як статистика, оптимізація, обробка сигналів, машинне навчання.

Високу ефективність дослідження моделей у середовищі Matlab дозволяють забезпечувати спеціалізовані професійні тулбокси (набори інструментальних засобів). Зокрема, за допомогою тулбоксу Simulink конструюються моделі аналогових, дискретних і гібридних динамічних схем та розв'язуються задачі їх аналізу. Введення моделей в Simulink та виконання їх аналізу здійснюється з використанням наступних бібліотек: Sources (джерел сигналів), Sinks (виводу результатів), Continuous (неперервних систем), Discontinuous (розривних систем), Discrete (дискретних систем), Linear (лінійних блоків), Nonlinear (нелінійних блоків), Connections (з'єднань), Functions & Tables (таблично заданих трансцендентних функцій, виразів та функцій), Math (математичних, логічних операцій та операцій відношення), Signals & Systems (систем і підсистем).

3.2 Можливості пакету Simulink для створення віртуального процесора

Simulink – це графічне середовище імітаційного моделювання, що дозволяє за допомогою блок-діаграм у вигляді направлених графів, будувати динамічні моделі, включаючи дискретні, безперервні і гібридні, нелінійні і розривні системи.

Програма Simulink є додатком до пакету MATLAB. При моделюванні з використанням Simulink реалізується принцип візуального програмування, відповідно до якого, користувач на екрані з бібліотеки стандартних блоків створює модель пристрою і здійснює розрахунки. При цьому, на відміну від класичних способів моделювання, користувачеві не потрібно досконально

вивчати мову програмування і чисельні методи математики, а досить загальних знань потрібних при роботі на комп'ютері і, природно, знань тієї предметної області в якій він працює.

Simulink є досить самостійним інструментом MATLAB і при роботі з ним зовсім не потрібно знати сам MATLAB і інші його додатки. З іншого боку доступ до функцій MATLAB і інших його інструментів залишається відкритим і їх можна використовувати в Simulink. Частина входять до складу пакетів має інструменти, що вбудовуються в Simulink. Є також додаткові бібліотеки блоків для різних областей застосування (наприклад, Power System Blockset – моделювання електротехнічних пристроїв, Digital Signal Processing Blockset – набір блоків для розробки цифрових пристроїв і т.д).

Інтерактивне середовище Simulink, дозволяє використовувати вже готові бібліотеки блоків для моделювання електросилових, механічних і гідравлічних систем, а також застосовувати розвинений модельно-орієнтований підхід при розробці систем управління, засобів цифрового зв'язку і пристроїв реального часу. Додаткові пакети розширення Simulink дозволяють вирішувати весь спектр завдань від розробки концепції моделі до тестування, перевірки, генерації коду і апаратної реалізації. Simulink інтегрований в середовище MATLAB, що дозволять використовувати вбудовані математичні алгоритми, потужні засоби обробки даних. Можна виконувати симуляцію динамічних властивостей системи і переглядати результати, як тільки симуляція почалася.

Щоб гарантувати задану швидкість симуляції і точність, Simulink надає ODE вирішувачі з фіксованим і змінним кроком, графічний відладчик і підпрограму оцінки часу виконання окремих функцій моделі. Архітектура нейронної мережі представляється S-моделлю, відтворюється за допомогою системи Simulink, бібліотекою Neural Network Toolbox Block Library, що містить блоки, необхідні для формування S-моделі нейронної мережі.

3.3 Neural Network Toolbox для розробки та візуалізації нейронних мереж

Neural Network Toolbox – це пакет розширення MATLAB, що містить засоби для проектування, моделювання, розробки та візуалізації нейронних мереж. Neural Network Toolbox надає функції і додатки для моделювання складних нелінійних систем, які складно описуються рівняннями. Neural Network Toolbox підтримує навчання з учителем і прямим поширенням, з ра-

діальними базисними функціями і динамічними мережами. Також є підтримка навчання без вчителя з самоорганізуючими картами і конкурентними шарами. З даним інструментом можливо створювати, навчати, візуалізувати і моделювати нейронні мережі. Neural Network Toolbox можна використовувати для таких завдань, як апроксимація даних, розпізнавання образів, кластеризація, прогноз часових рядів, моделювання динамічних систем та їх управління [14].

Нейромережеві технології дозволяють вирішувати такі завдання, вирішення яких класичними формальними методами утруднене або неможливо. Пакет забезпечує всебічну підтримку типових нейромережевих парадигм і має відкриту модульну архітектуру. Пакет містить функції командного рядка і графічний інтерфейс користувача для швидкого покрокового створення нейромереж. Крім цього Neural Network Toolbox забезпечує підтримку, що дозволяє моделювати нейромережі і створювати блоки на основі розроблених нейромережевих структур. Основні характеристики [14]:

– навчання мереж з учителем: багатошарові, з радіальними базисними функціями, з часовою затримкою, нелінійні авторегресійні, а також рекурентні мережі;

– навчання мереж без вчителя, включаючи самоорганізуючі карти і мережі з конкурентними шарами;

– програми для апроксимації даних, розпізнавання образів і кластеризації;

– паралельні обчислення та підтримка графічних процесорів для прискорення;

– навчання (з використанням Parallel Computing Toolbox);

– попередня обробка і післяобробка даних для підвищення ефективності навчання мережі та оцінки якості мережі;

– модульне уявлення мереж для управління і візуалізації мережі заданого розміру;

– блоки Simulink для побудови та оцінки нейронних мереж і додатків систем управління;

– графічний інтерфейс користувача для покрокового створення, навчання та імітаційного моделювання нейронних мереж;

– підтримка найбільш поширених керованих і некерованих мережевих структур;

– повний перелік навчальних і тестових функцій;

– динамічні алгоритми навчання мереж, що включають тимчасову затримку, нелінійну авторегресії (NARX), ланцюгові і настроюються динамічні структури;

– блоки Simulink для створення нейронних мереж і розвинених блоків для систем контролю;

– автоматична генерація блоків Simulink з об'єктів нейронної мережі;

– модульне подання мережі, що дозволяє створювати необмежену кількість вхідних шарів і об'єднаних мереж, а також графічне представлення архітектури мережі;

– функції попередньої і пост обробки та блоки Simulink для поліпшення процесу навчання та оцінки продуктивності мережі;

– візуалізація топології і процесу навчання нейронної мережі.

Нейронна мережа, як і її прообраз в біологічній нервовій системі, може вчитися і бути навчена для пошуку рішення, розпізнавання образів, класифікації даних і прогнозу майбутніх подій. Поведінка нейронної мережі визначається тим, як пов'язані її окремі обчислювальні елементи, а також силою цих зв'язків або вагами. Ваги автоматично налаштовуються шляхом навчання мережі у відповідності з конкретними навчальними правилами доти, поки мережа не почне виконувати коректно поставлене завдання.

Neural Network Toolbox включає в себе функції командного рядка і програми для створення, навчання та моделювання нейронних мереж. Додатки дозволяють легко розробляти нейронні мережі для таких завдань, як апроксимація даних (у тому числі даних часових рядів), розпізнавання образів і кластеризація. Після створення мереж на базі цих інструментів можна автоматично генерувати код MATLAB для перетворення отриманого рішення в програмний код і автоматизації вирішення завдань. Neural Network Toolbox підтримує різні архітектури контрольованих і неконтрольованих мереж. З модульним підходом створення нейромереж можливо розробляти власні мережеві архітектури для конкретної задачі. Є можливість переглянути мережеву архітектуру, включаючи всі входи, шари, виходи і взаємозв'язки.

Контрольовані нейронні мережі навчаються для отримання заданих результатів у відповідь на вхідну вибірку даних, що робить їх особливо придатними до моделювання та управління динамічних систем, класифікації зашумленних даних і передбачення майбутніх подій.

Neural Network Toolbox включає в себе чотири типи контрольованих мереж: з прямим розповсюдженням, радіально-базисні, динамічні та LVQ.

Мережі прямого поширення мають односторонні з'єднання від вхідних до вихідних шарів. Вони найчастіше використовуються для прогнозування, розпізнавання образів та апроксимації нелінійних функцій. Підтримувані мережі прямого поширення включають алгоритми зворотного поширення помилки, каскадного розповсюдження, прямого зв'язку із вхідною затримкою зустрічного поширення, лінійні мережі та мережі типу персептрон.

Радіально базисні мережі забезпечують альтернативний, швидкий спосіб конструювання нелінійних мереж прямого поширення. Підтримувані варіанти включають узагальнені регресії і імовірнісні нейронні мережі.

Динамічні мережі використовують пам'ять і рекурентні зворотні зв'язки для розпізнавання просторових і часових закономірностей в даних. Вони широко використовуються для прогнозування часових рядів, моделювання нелінійних динамічних систем і в додатках систем управління. Готові динамічні мережі в Neural Network Toolbox включають мережі із сфокусованою і розподіленою затримкою часу, нелінійні авторегресійні мережі (NarX), мережі з рекурентними шарами, мережі Ельмана і Хопфілда. Набір інструментів також підтримує динамічне навчання користувальницької мережі з довільними сполуками.

LVQ-мережі використовують метод класифікації моделей, які не є лінійно роздільними. LVQ дозволяє визначити межі класу і ступінь деталізації класифікації.

Неконтрольовані нейронні мережі навчаються, постійно пристосовуючись до нових вхідних значеннь. Такі мережі знаходять залежності в даних і можуть автоматично визначати схеми класифікації. Neural Network Toolbox включає два типи самоорганізованих неконтрольованих мереж з навчанням без вчителя: конкурентні шари і самоорганізуються карти. Конкурентні шари розпізнають подібні вхідні вектори, автоматично сортуючи їх на категорії. Дані мережі зазвичай використовуються для класифікації та розпізнавання образів.

Самоорганізуючі карти навчаються класифікації вхідних векторів за подібністю. Подібно конкурентним шарам вони використовуються для задач класифікації та розпізнавання образів, проте відрізняються від конкурентних шарів тим, що можуть зберігати топологію вхідних векторів, призначаючи прилеглі входи до прилеглих категорій [15,16].

3.4 Використання нейронних процесорів

Однією з перших можливостями нейронних мереж та їх промисловим застосуванням зацікавилася компанія Intel. З подачі міністерства оборони США було розпочато роботи з проектування та розробки нейропроцесора. У 1989 році вже був представлений перший промисловий зразок i80170NX ETANN (Electrically Trainable Analog Neural Network). Застосування розпаралелених архітектур в нейропроцесорах дозволило добитися продуктивності 2000000000 операцій в секунду. Цей процесор (і його наступник - i80160NC) досить успішно працює в різних системах, в яких необхідне рішення неформалізованих задач. Слідом за Intel підтягнулися й інші провідні світові виробники обчислювальної техніки. Свої нейропроцесори створили такі компанії, як Motorola, Echelon, IBM, Siemens, Fujitsu та інші. Окремо можна відмітити успіхи Росії на цьому терені. У 1998 році на світовий ринок нейрочіпів вийшла фірма - НТЦ "Модуль", представивши нейропроцесори NM6403. Його спроектували і розробили російські інженери, правда, виготовлюються нейропроцесори на потужностях компанії Samsung. Область застосування NM6403 вельми широка. Він застосовується для обробки відеоданих, в радіолокаційних системах і в криптографії. Досить цікавий створений на базі нейропроцесора апаратно-програмний комплекс "Трафік -Монітор", який вимірює в реальному масштабі часу статистичні характеристики транспортного потоку для подальшого прийняття рішення з організації та регулювання дорожнього руху. Він дозволяє виміряти не тільки загальну кількість минулих транспортних засобів, а й класифікувати їх за типами.

Основні перспективні напрями розвитку нейрокомп'ютерних технологій: нейропакет, нейромережеві експертні системи, СУБД з включенням нейромережевих алгоритмів, управління динамічними системами, обробка зображень, обробка сигналів, управління фінансовою діяльністю, оптичні нейрокомп'ютери і віртуальна реальність [17].

3.5 Нейропроцесори NM6403

Нейропроцесори NM6403 володіє наступними характеристиками. Тактова частота - 40 МГц, напруга живлення - 3,0-3,6 В, споживана потужність - 1,3 Вт Основні обчислювальні вузли процесора: керуюче RISC-ядро і векторний співпроцесор.

Продуктивність нейропроцесора становить 120 мільйонів операцій в секунду для 32- бітових операндів. До речі, ці процесори доступні у вільному продажі, та й коштують не так вже й дорого - близько 50 $. Загальна структура процесора NeuroMatrix NM 6403 представлена на рисунку 3.1.

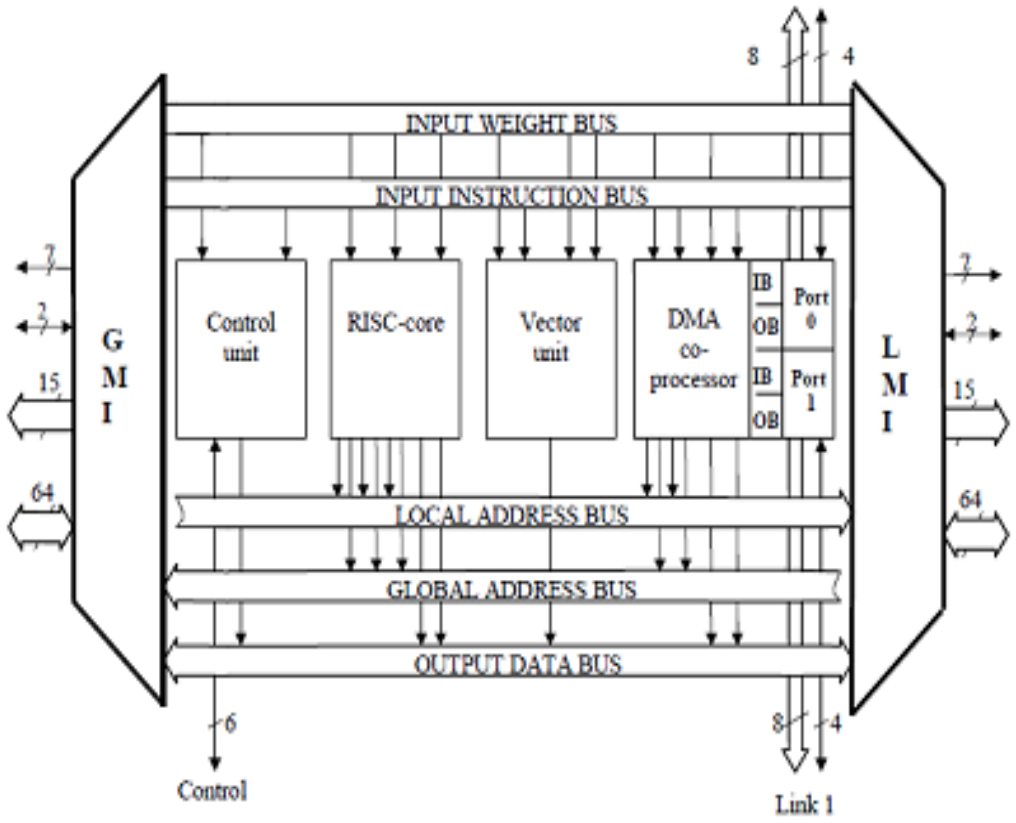


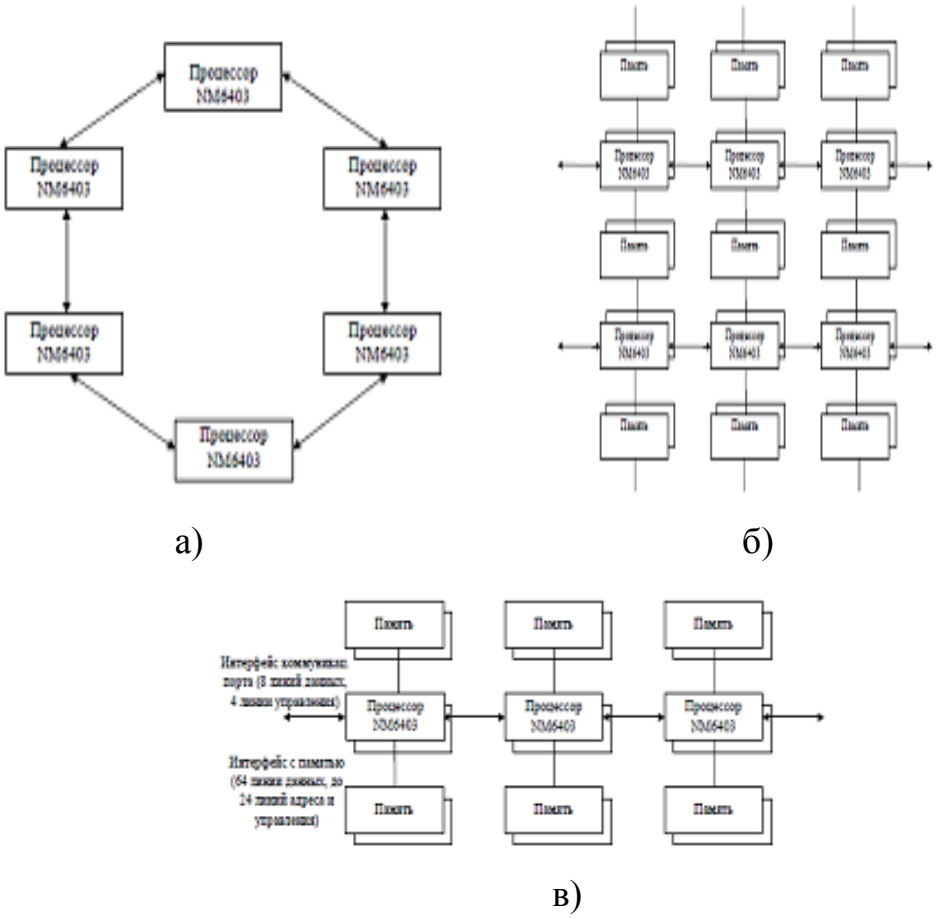
Рисунок 3.1 – Загальна структура процесора NeuroMatrix NM 6403

Хоча NM6403 моделює тільки певний клас нейромереж, число додатків для нього велике. Архітектура кристала дозволяє створювати мультипроцесорні конфігурації, за допомогою яких можна ефективно емулювати складні нейронні мережі для багатьох додатків. Нейрочіп показав себе високопродуктивним процесором широкого застосування для вбудованих систем. Особливо він хороший для обробки відеоданих, радіолокації і криптографії. Зараз на замовлення однієї південнокорейської фірми на основі плати з чотирма чіпами NeuroMatrix NM6403 DSP завершується проектування комплексу контролю дорожнього руху. По суті, це система машинного зору, яка і в жорстких погодних умовах вміє розпізнавати декілька видів транспорту, що рухається зі швидкістю до 200 км / год по шести смугах. Система використовує вхідний сигнал тільки від однієї телевізійної камери, розташованої на висоті 12 метрів, і обчислює число транспортних засобів кожного з заданих класів, що

пройшли за годину, їх середню швидкість, середню дистанцію між ними і завантаженість дороги. Помилка визначених величин не перевищує 5 %. Накопичену інформацію комплекс щогодини передає на пункт управління.

Основними архітектурними особливостями процесора NM6403 для побудови різних багатопроцесорних обчислювальних систем є наявність двох високошвидкісних двонаправлених байтових комунікаційних портів, апаратно сумісних з портами сигнального процесора TMS320C40, та підтримки режиму роботи із загальною пам'яттю.

Шляхом об'єднання процесорів NM6403 різними способами можна домогтися реалізації великого числа високопродуктивних паралельних систем різноманітної конфігурації. На рисунку 3.2 наведені приклади побудови таких обчислювальних мереж.



а) б)

в)

Рисунок 3.2 – Обчислювальна мережа на базі процесора NM6403: а – двонаправлене кільце; б – структура типу двомірної решітки; в – двонаправлений конвеєр

Крім перерахованих вище можливостей можна створювати обчислювальні мережі практично будь-якої конфігурації з використанням сигнального процесора TMS320C40 в якості комутуючого елемента. Використовуваний інтерфейс з пам'яттю дозволяє будувати обчислювальні системи з різною архітектурою на основі процесора NM6403:

- архітектура з загальною пам'яттю;
- архітектура з розподіленою пам'яттю;
- змішана архітектура.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ІДЕНТИФІКАЦІЇ АНОМАЛЬНИХ СТАНІВ РЕАЛЬНИХ КОМП'ЮТЕРНИХ МЕРЕЖ ПАРАЛЕЛЬНОЮ ОБРОБКОЮ КОЛЕКТИВОМ НЕЙРОННИХ МЕРЕЖ

4.1. Створення моделей трафіку, дослідження продуктивності мережі

Інженери використовують середовище MATLAB і Simulink для створення і виконання моделей поведінки мережевого трафіку і прогнозування продуктивності мережі – цей процес критично важливий і дає конкурентну перевагу при проектуванні та розгортанні мереж і послуг зв'язку наступного покоління.

Сучасні мережі передачі даних можуть використовуватися для передачі аудіо та відеоінформації з певним рівнем якості. Мережеві оператори використовують продукти MathWorks для створення моделей і прогнозування факторів, що впливають на продуктивність мережі, таких як перевантаження, конфлікт при поділі ресурсів і затримки процесів. Використання моделей трафіку для прогнозування продуктивності мережевого обладнання.

Виробники мережного обладнання використовують продукти MathWorks для створення точних і володіють можливістю повторного використання моделей комунікаційного устаткування і протоколів для моделювання використання мережі і каналів, а також для вивчення впливу додавання нових послуг в існуючі мережі.

Інженери використовують моделі трафіку для визначення проблем затримки, блокування і браку ресурсів, властивих конфігурації мережі передачі даних. Ці моделі дозволяють виробникам устаткування демонструвати і доводити продуктивність і справність своїх продуктів планувальникам мереж і постачальникам послуг. Планувальники мереж використовують моделі, створені виробниками обладнання, для вивчення і порівняння різних варіантів конфігурації мережі і рекомендують найбільш ефективні рішення постачальникам послуг.

Створення моделей фізичного і мережевого рівнів в єдиному середовищі інженерам необхідно моделювати і перевіряти продуктивність мережі до того, як буде розроблений фізичний рівень, або ж незалежно від поведінки фізичного рівня. Щоб допомогти в проведенні незалежного моделювання продуктивності мережі, середа MATLAB і Simulink підтримує створення як залежать від часу, так і дискретно-подієвих моделей, що враховують черги, сервери і генератори трафіку.

4.2 Нейромережева реалізація технології виявлення атак і її реалізація в системі MATLAB

Завдання ідентифікації атак на КС розглядається як завдання класифікації засобами нейронніх мереж різних типів.

Припустимо, тренувальні набори (x_1,c_1), (x_2,c_2),..., (x_n, c_n) містять множину $X_t = \{x_x,x_2,\ldots,x_{N1}\}$ і N_t тренувальних спостережень в n-мірному просторі ($x_j \in R^n$, $n \geq 2$) і їхній зв'язаний клас-індикатор – вектор c_j =1,2... N_t. Обмежимося розглядом проблеми двох класів («атака на КС», «відсутність атаки»), хоча в реальних умовах це обмеження може бути критичним, тому що залежно від типу даних вибір ваг матриці зв'язків може бути досить складним і число класів прийдеться збільшувати.

Відповідно до прийнятої постановки завдання потрібно, щоб індикатор c_j був двомірним вектором, $c_j = (c_1,c_2)$ T, а x_j належали або класу ω_1, або класу ω_2. Компоненти c_{1j}, c_{2j} визначені як нуль або одиниця відповідно класу приналежності x_j, тобто c_{lj} =1 і c_{2j} =0 або x_j є ω_1; c_{1j} =0 і c_{2j}=1 або x_j є ω_1. Клас-індикатор вектор c_j передбачає декомпозицію множин X_t на підмножини відповідно окремим класам. Позначимо N_n число тренувальних спостережень у класі ω_1. Тренувальні дані x_t, можна нормалізувати. Надалі використаються ненормалізовані дані.

Позначимо через b (x; w) функції відтворення багатошарового персептрона MLP нейронної мережі для класифікації з вектором w регульованих ваг нейромережі. Тренування мережі виконується шляхом мінімізації середньоквадратичної MS погрішності.

Для подальшого зменшення можливо його регуляризацію для зменшення узагальнюючих властивостей нейромережі. Погрішність нейромережі з нелінійною активаційною функцією на схованому шарі моделі є істотно нелінійна функція. Послідовна її мінімізація може бути проведена з використанням ітераційних оптимизаційних алгоритмів, реалізованих у середовищі MATLAB. Головна мета – пошук глобального мінімуму E^{MPL} (w). Найпростіше використання тренувального алгоритму – локальна мінімізація E^{MPL}. Обчислена величина спостережуваного мінімуму може бути чистим локальним мінімумом. Рішення w строго залежить від стартових значень локального оптимізатора. Будемо використати рекурсивні (евристичні) методи для пошуку декількох невеликих (наприклад, для досягнення величини 10^6) локальних мінімумів, з яких вибирається один. Саме на цьому мінімумі фіксу-

ється матриця ваг нейромережі, використовувана надалі в якості класифікатора.

Специфіка завдання ідентифікації атаки полягає в наступному. Задано два вектори, $x^{(1)}=\{x^{(j1)}\}$,$x^{(2)}=\{x^{(j2)}\}$j= 1, 9, компоненти яких характеризують трафік,$x^{(1)} \in \omega_1$, $x^{(2)} \in \omega_2$ де ω_1 й ω_2 – класи ситуації, що відповідають наявності (100 % гарантія) і відсутності атаки. Оскільки наявності двох ідеалізованих векторів недостатньо для реалізації гарантованого результату, сформовані тестові вибірки $x^{(1t)}$ й $x^{(2t)}$ [11].

Загальна структура віртуального нейронного процесора для ідентифікації аномалій мережеві технології стали невід'ємною частиною життєдіяльності сучасного суспільства. При цьому для ефективної роботи мереж велике значення має надійність передачі даних по каналах зв'язку. Однією з головних причин, що впливають на ефективність роботи обчислювальної мережі (ОМ) є аномалії трафіку. Аномалії в трафіку ОМ можуть бути викликані несправністю мережевого обладнання, випадковими або навмисними діями з боку легітимних користувачів, невірної роботою додатків, діями зловмисників і тощо. Таким чином, для надійної передачі даних в ОМ можуть бути вжиті заходи щодо своєчасного виявлення аномалії. Отже, для забезпечення надійної передачі даних в ОМ велике значення набуває розробка нових методів виявлення аномалій і заходи щодо їх усунення.

Нейромережа – це навчальна система. Вона діє не тільки відповідно до заданого алгоритму і формул, а й на підставі минулого досвіду. Застосування нейронних мереж дає багато переваг у виявленні атак. Гнучкість використання нейромережевих технологій є безперечно кращою альтернативою для сьогоденних систем виявлення аномальних станів КС, що зумовлено тим, що нейромережевий базис це, перш за все, інтелектуальна технологія, здатна постійно розвиватися та доповнюватися постійно новими значущими компонентами. Нейромережа здатна аналізувати дані від мережі, навіть якщо ці дані є неповними або спотвореними, володіє можливістю проводити аналіз даних в нелінійному режимі.

Для моделювання аномальних станів КМ та їх виявлення розробленно наступну схему в Simulink яка показана на рис. 4.1.

Загальна схем ідентифікації атак включає наступні блоки:

– Perceptron NN – нейронна мережа типу Perceptron, що включає в себе два шари, що ідентифікує атаку.

– FeedForward NN – нейронна мережа типу FeedForward, що включає в себе два шари, що ідентифікує атаку.

– Recurent NN – нейронна мережа типу Recurent, що включає в себе два шари, що ідентифікує атаку.

– Блок вхідних наборів.

– Identify attacks – ідентифікатор атаки.

– output – залежно від комбінації показує наявність або відсутність атаки.

– Inputs – відображує значення сигналу у вигляді чисел.

– Інші допоміжні інструменти.

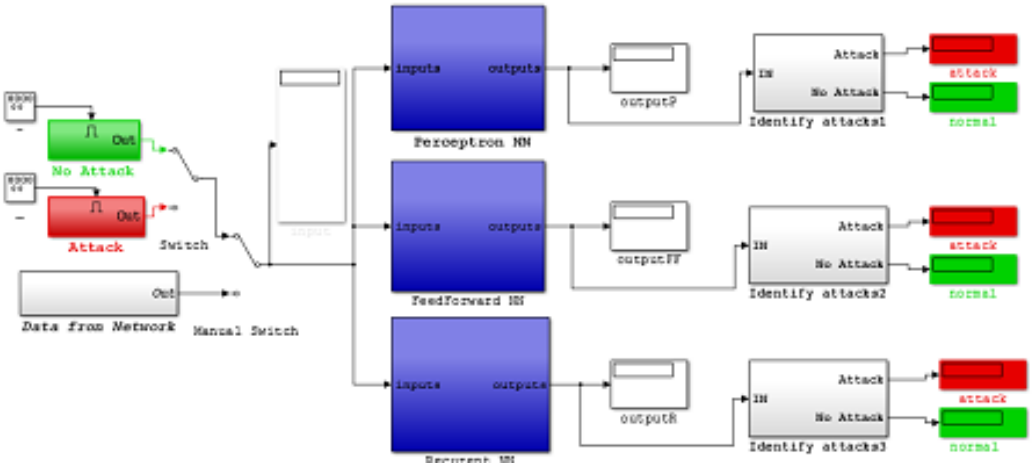


Рисунок 4.1 – Загальна схема моделювання ідентифікації атак у нотації MATLAB/SIMULINK

Загальна схема нейронної мережі типу Perceptron наведена на рис. 4.2 [18].

Нейронна мережа складається із одного шару: LW(1,1) показаного відповідно на рис. 4.3, та LW(2,1) показаного на рис. 4.4.

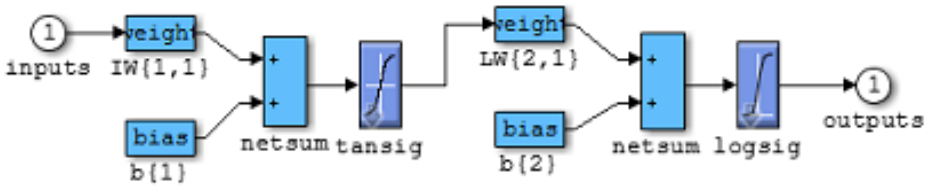


Рисунок 4.2 – Загальна схема нейронної мережі

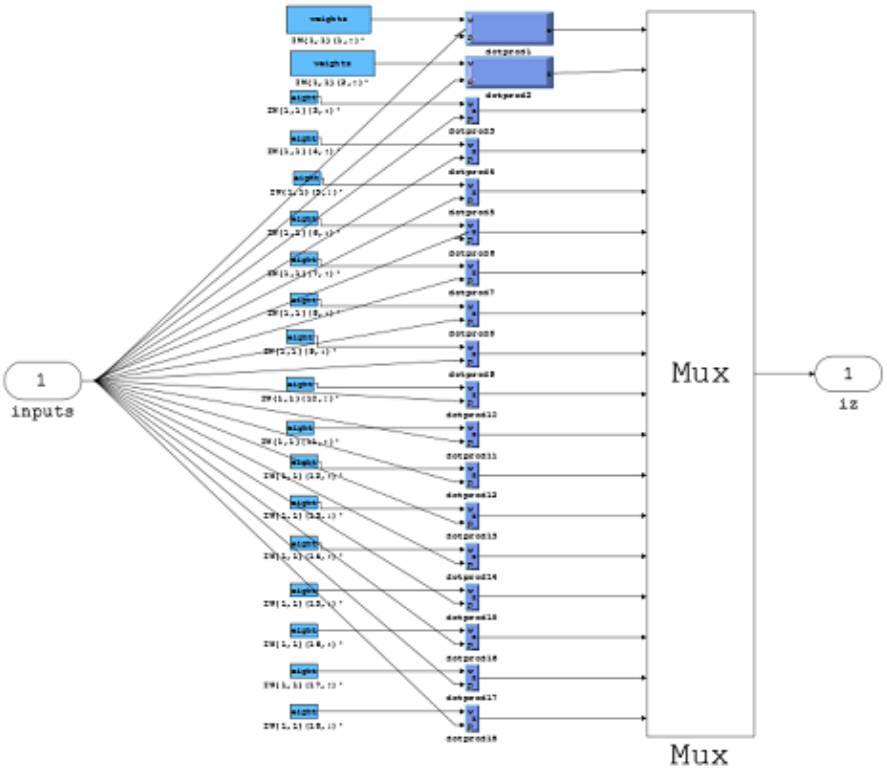


Рисунок 4.3 – Шар нейронної мережі

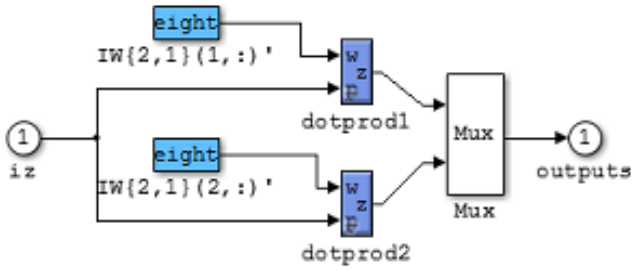


Рисунок 4.4 – Шар нейронної мережі

Загальна схема нейронної мережі типу FeedForward наведена на рис. 4.5.

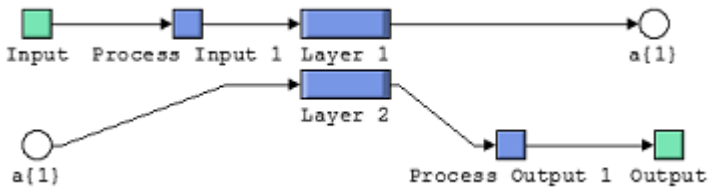


Рисунок 4.5 – Загальна схема нейронної мережі

Нейронна мережа складається із двох шарів: Layer 1 LW(1,1) показаного відповідно на рис. 4.6, із затримкою показаною на рис. 4.7 та Layer 2 LW(2,1) показаного відповідно на рис. 4.8.

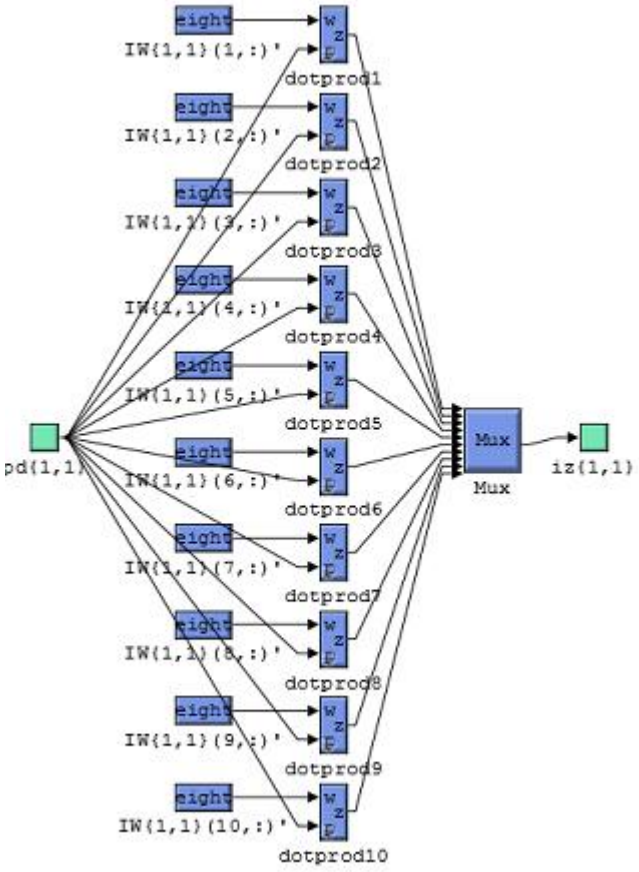


Рисунок 4.6 – Перший шар нейронної мережі

Рисунок 4.7 – Затримка на першому шарі нейронної мережі

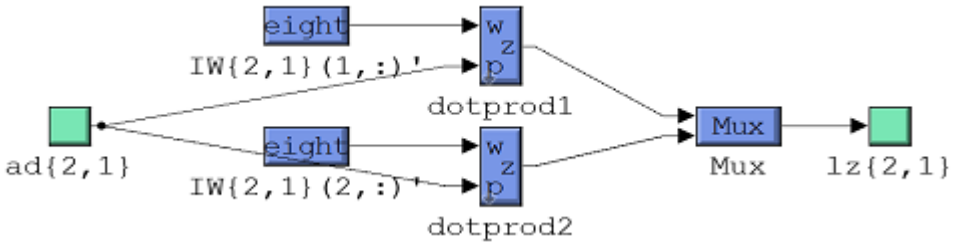


Рисунок 4.8 – Другий шар нейронної мережі Layer 2 LW{2,1}

Загальна схема нейронної мережі типу Recurent наведена на рис. 4.9.

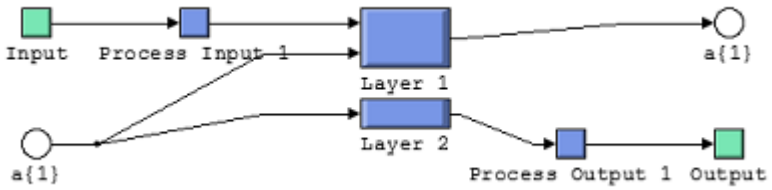


Рисунок 4.9 – Схема Recurent нейронної мережі

Нейронна мережа складається із двох шарів: Layer 1 LW(1,1) показаного відповідно на рис. 4.10, із затримками показаними на рис 4.11 та Layer 2 LW(2,1) показаного відповідно на рис 4.12 – 4.14.

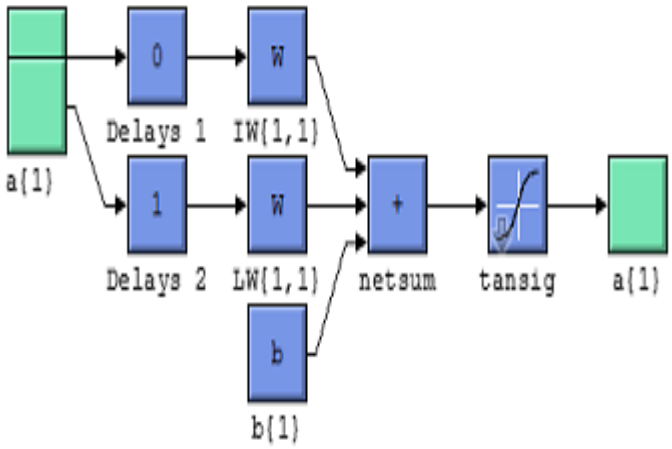


Рисунок 4.10 – Шар Layer 1 нейронної мережі

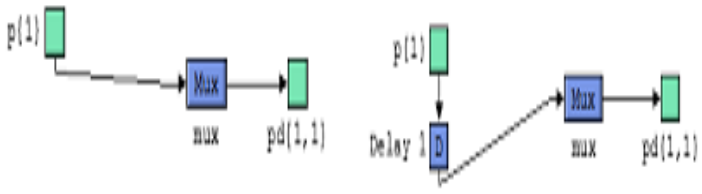


Рисунок 4.11 – Затримки на першому шарі нейронної мережі

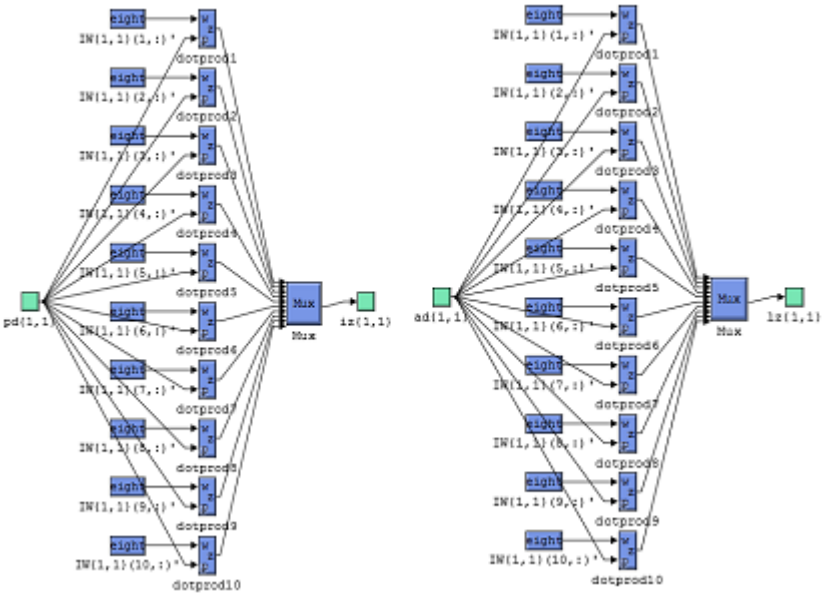


Рисунок 4.12 – Шар нейронної мережі Layer 1 IW{1,1} та LW{1,1}

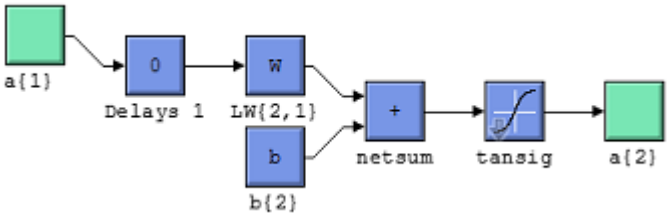


Рисунок 4.13 – Шар Layer 2 нейронної мережі

Рисунок 4.14 – Затримка на другому шарі нейронної мережі Layer 2 Delays 1

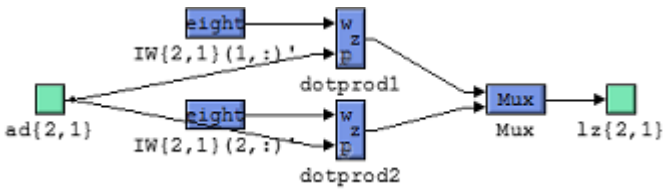


Рисунок 4.15 – Шар нейронної мережі Layer 2 LW{2,1}

Нейронні мережі мають ваги, параметри яких зображені на рис. 4.16.

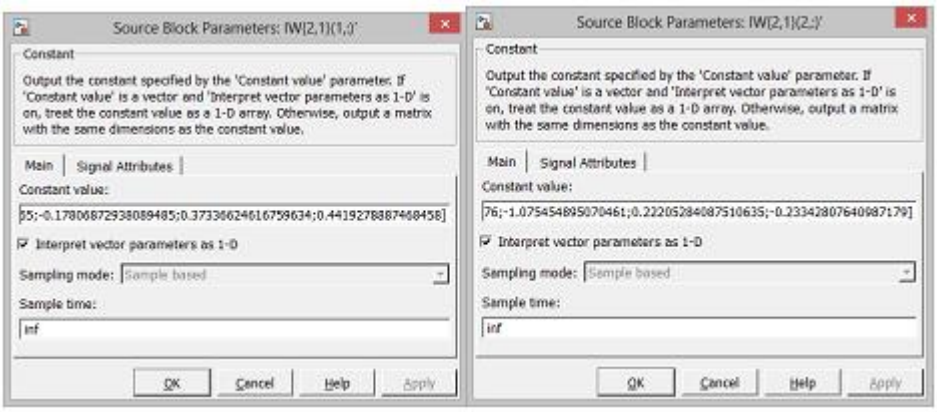


Рисунок 4.16 – Параметри завдання ваг нейронної мережі

Блоки та складові нейронних мереж:

– Inputs – створює вхідний порт для підсистеми або моделі верхнього рівня ієрархії.

Параметри:

– Port number – номер порту.

– Port dimensions – розмірність вхідного сигналу. Якщо цей параметр дорівнює-1, то розмірність вхідного сигналу буде визначатися автоматично.

– Sample time – крок модельного часу.

– Data type – тип даних вхідного сигналу: auto, double, single, int8, uint8, int16, uint16, int32, uint32 або boolean.

– Signal type – тип вхідного сигналу:

– auto – автоматичне визначення типу.

– real – дійсний сигнал.

– complex – комплексний сигнал.

– Interpolate data (прапорець) – інтерполювати вхідний сигнал.

У випадку, якщо тимчасові відліки вхідного сигналу зчитувального з робочої області MATLAB не збігаються з модельним часом, то блок буде виконувати інтерполяцію вхідного сигналу. При використанні блоку Inport у підсистемі даний параметр не доступний. Блоки Inport підсистеми являються її входами. Сигнал, подаваємі на вхідний порт підсистеми через блок Inport, передається в середину підсистеми. Назву вхідного порта буде показано на зображенні підсистеми як мітка порта. Блоки Inport підсистеми є її входами. Сигнал, що подається на вхідний порт підсистеми через блок Inport, переда-

ється усередину підсистеми. Назва вхідного порту буде показано на зображенні підсистеми як мітка порту.

При створенні підсистем і додаванні блоку Inport у підсистему Simulink використає наступні правила:

При створенні підсистеми за допомогою команди Edit/Create subsystem вхідні порти створюються й нумеруються автоматично починаючи з 1.

Якщо в підсистему додається новий блок Inport, то йому присвоюється наступний один по одному номер.

Якщо який-небуть блок Inport видаляється, то інші порти перейменовуються таким чином, щоб послідовність номерів портів була безперервною.

Якщо в послідовності номерів портів є розрив, то при виконанні моделювання Simulink видасть повідомлення про помилку й зупинить розрахунок. У цьому випадку необхідно вручну перейменувати порти таким чином, щоб послідовність номерів портів не порушувалася.

Вхідний порт у системі верхнього рівня використається для передачі сигналу з робочої області MATLAB у модель.

Для передачі сигналу з робочго простору MATLAB потрібно не тільки встановити в моделі вхідний порт, але й виконати установку параметрів уведення на вкладці Workspace I/O вікна діалогу Simulation parameters... (повинен бути встановлений прапорець для параметра Input і задане ім'я змінної, котра містить вхідні дані). Тип даних, що вводять: Array (масив), Structure (структура) або Structure with time (структура з полем "час") задається на цій же вкладці.

Outputs – Створює вихідний порт для підсистеми або для моделі верхнього рівня ієрархії.

Параметри:

– Port number – номер порту.

– Output when disabled – вид сигналу на виході підсистеми, у випадку якщо підсистема виключена. Використається для керованих підсистем. Може приймати значення (вибираються зі списку):

– held – вихідний сигнал підсистеми дорівнює останньому розрахованому значенню.

– reset – вихідний сигнал підсистеми дорівнює значенню задаючому параметром Initial output.

Initial output – значення сигналу на виході підсистеми до початку її роботи й у випадку, якщо підсистема виключена. Використається для керованих підсистем.

Блоки Outport підсистеми є її виходами. Сигнал, що подається в блок Outport усередині підсистеми, передається в модель (або підсистему) верхнього рівня. Назва вихідного порту буде показано на зображенні підсистеми як мітка порту.

При створенні у підсистеми за допомогою команди Edit/Create subsystem вихідні порти створюються й нумеруються автоматично починаючи з 1.

Якщо в підсистему додається новий блок Outport, то йому привласнюється наступний один по одному номер.

Якщо який або блок Outport віддаляється, то інші порти перейменовуються таким чином, щоб послідовність номерів портів була безперервною.

Якщо в послідовності номерів портів є розрив, то при виконанні моделювання Simulink видасть повідомлення про помилку й зупинить розрахунок. У цьому випадку необхідно вручну перейменувати порти таким чином, щоб послідовність номерів портів не порушувалася.

У тому випадку, якщо підсистема є керованої, то для її вихідних портів можна задати вид вихідного сигналу для тих тимчасових інтервалів, коли підсистема заблокована. Для першого вихідного порту підсистеми параметр Output when disabled заданий як held, а для другого - як reset, причому величина початкового значення задана рівною нулю. Графіки сигналів показують, що коли підсистема заблокована, сигнал першого вихідного порту залишається незмінним, а сигнал другого стає рівним заданому початковому значенню (нулю).

Вихідний порт у системі верхнього рівня використається у двох випадках:

- для передачі сигналу в робочий простір MATLAB;
- для забезпечення зв'язку функцій аналізу з виходами моделі.

Для передачі сигналу в робочий простір MATLAB потрібно не тільки встановити в моделі вихідні порти, але й виконати установку параметрів висновку на вкладці Workspace I/O вікна діалогу Simulation parameters... (повинен бути встановлений прапорець для параметра Output і задане ім'я змінної для збереження даних). Тип даних, що зберігають – Array масив, Structure (структура) або Structure with time (структура з полем “час”) задається на цій же вкладці. NetSum – функція підсумовування входів. Параметри функції показані на рис. 4.17.

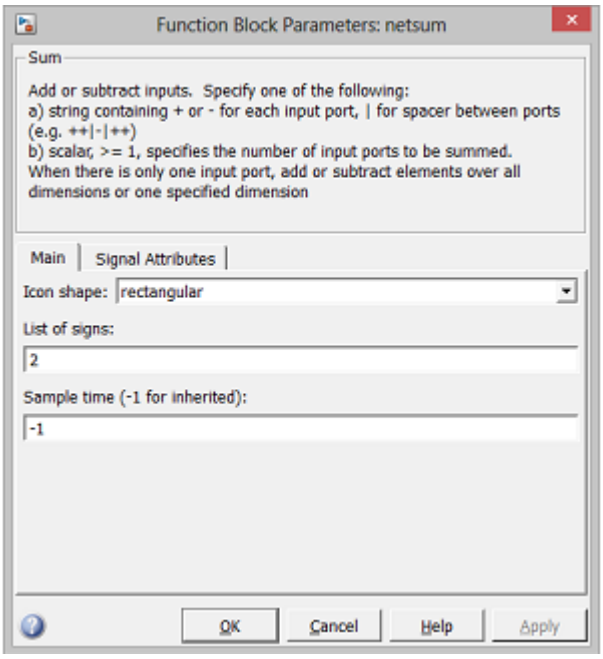


Рисунок 4.17 – Параметри блоку NetSum RadBas

RadBas – функція активації. Функція активації обчислює вихід шарую-чи по його входу. Вона має один вхідний аргумент – N – Sx матрицю вхідних векторів (стовпців) і повертає вихідні вектори зі значеннями в діапазоні від -1 до 1. RADBAS(code) – повертає інформацію про цю функцію.

Значення "code":

– 'deriv' – вид похідної від функції активації;
– 'name' – повна назва;
– 'output' – діапазон вихідних значень;
– 'active' – діапазон вхідних значень.

Для того, щоб використати в мережі функцію, RADBAS необхідно ви-кликати NEWPNN або NEWGRNN.

Logsig – функція активації. Функція активації обчислює вихід шарів його входу.

Функція LOGSIG (N) має один вхідний аргумент – N – Sx матрицю вхі-дних векторів (стовпців) і повертає вихідні вектори зі значеннями в діапазоні від 0 до 1.

LOGSIG (code) – повертає інформацію про цю функцію.

Значення “code":

– 'deriv' – вид похідної від функції активації;
– 'name' - повна назва;
– 'output' – діапазон вихідних значень;

– 'active' – діапазон вхідних значень.

Для того, щоб використати в мережі функцію, LOGSIG необхідно викликати NEWFF або NEWCF.

Для того, щоб використати в мережі функцію LOGSIG, необхідно виконати наступні установки:

NET.layers{i,j}.transferFcn установити як 'logsig'.

Блок Biases – визначає вектори зсувів для кожної верстви зі зсувом.

net.b – масив Nl x1 осередків, де Nl – число верстов у мережі (net.numLayers).

Вектор зсуву для і-го шару (або нульова(порожня) матриця []) визначений в net.b{i} якщо відповідне з'єднання зсуву net.biasConnect(i)- 1 (або 0). Число елементів у векторі зсуву завжди дорівнює розміру слоя, з яким він зв'язаний (net.layers{i}.size). Цей розмір може також бути отриманий із властивостей зсуву: net.biases{i}.size..

layerWeights – містить структури властивостей для кожного з ваг шаруючи мережі.

net.layerWeights – масив NlxNl осередків, де Nl – число верстов мережі (net.numLayers).

Структура, що визначає властивості зв'язків і-го шару с j м верствою (або нульова матриця) визначена в: net.layerWeights{i,j}, якщо відповідний зв'язок шару net.layerConnect(i,j) – 1 (або 0).

Структура пергого шар IW(1,1) IW– визначає матриці вхідних ваг.

net.IW – масив NlxNi осередків, де Nl – число верстов у мережі (net.numLayers), а Ni- число входів мережі (net.numInputs). Матриця ваг для зв'язку і-го шару с j м входом (або нульова матриця []) описується в: net.IW{i,j}, якщо відповідне вхідне з'єднання. net.inputConnect(i,j) – 1 (або 0).

Матриця ваг має кількість рядків, рівна розміру шаруючи, до якого вона ставиться (net.layers{i}.size). Містить кількість стовпців рівне добутку розмірності входу на кількість затримок, асоційованих з вагою: net.inputs{j}.size * length(net.inputWeights{i,j}.delays).

Ці розміри можуть також бути отримані із властивостей вхідних ваг:

net.inputWeights{i,j}.size.

Перша верства нейромережі складається з наступних компонентів:

– inputs – вхідні змінні;

– output – вихідні змінні;

– weights (1-18) – містить структури властивостей для кожного з ваг шару мережі;

– блоки doprod(1-18) - Вагові функції у вигляді скалярного добутку.

Вагові функції використаються, коли викликається функція sim для моделювання мережі [Y,Pf,Af] = sim(net,P,Pi,Ai)

Mux – поєднує 18 вхідних сигналів в одну загальну шину. Параметри блоку Inputs values – Display. Цифровий дисплей Display. Призначення – відображає значення сигналу у вигляді числа. Параметри: Format – формат відображення даних. Параметр Format може приймати наступні значення:

– short – 5 значущих десяткових цифр;

– long – 15 значущих десяткових цифр;

– short_e – 5 значущих десяткових цифр і 3 символи ступеня десяти;

– long_e – 15 значущих десяткових цифр і 3 символи ступеня десяти;

– bank – "грошовий" формат. Формат з фіксованою крапкою й двома десятковими цифрами в дробовій частині числа;

– Decimation – кратність відображення вхідного сигналу.

При Decimation = 1 відображається кожне значення вхідного сигналу, при Decimation = 2 відображається кожне друге значення, при Decimation = 3 – кожне третє значення тощо.

Sample time – крок модельного часу. Визначає дискретність відображення даних.

Floating display (прапорець) – переклад блоку в “вільний” режим. У даному режимі вхідний порт блоку відсутній, а вибір сигналу для відображення виконується щигликом лівої клавіші “миші” на відповідні линій зв'язку. У цьому режимі для параметра розрахунку Signal storage reuse повинне бути встановлене значення off (вкладка Advanced у вікні діалогу Simulation parameters).

Кожний компонент (крім, бінарних) тестових вибірок $x_j^{(1t)} \in x^{(1t)}$ і $x_j^{(2t)} \in x^{(2t)}$ сформована за правилом

$$x_j^{(1t)} \in x_j \pm 0{,}15\, x_j \mathrm{rand}() \qquad (4.1)$$

де rand() – функція MATLAB, що генерує випадкові числа в інтервалі [0, 1].

В табл. 4.1 наведені параметри моделювання трафіка, по яких відбувалося навчання мережі. Конкретний набір тестових даних (діапазон коливань значень 1010) не дозволяє в реальних умовах одержати «чисті» значення 0 і 1 на заданому наборі x (1t) даних. Реальне навчання нейромережі завершувалося при значеннях набору, ріх (2t) вних 0,95 і 0,05, а для – відповідно 0,05 і

0,95. Зона невизначеності становить 5 – 10 %, підвищення точності класифікації можливо за рахунок використання нових ознак. Слід зазначити, що навчена мережа може правильно класифікувати ситуації при неповних або нечітких (розмитих) значеннях вхідного вектора. Схема, що реалізує дані подані в табл. 4.1, показана нижче, на рис. 4.18.

Таблиця 4.1 – Параметри моделювання трафіка

	Y	0,1	1,0
Protocol ID	X1	0	0
Source Port	X2	2314	1611
Source Address	X3	80	6101
Destination Port	X4	1573638018	8801886082
Destination Address	X5	-1580478590	-926176166
CMP Type ID	X6	1	1
CMP Code ID	X7	1	1
Raw Data	X8	401	0
Length Data	X9	3758	2633

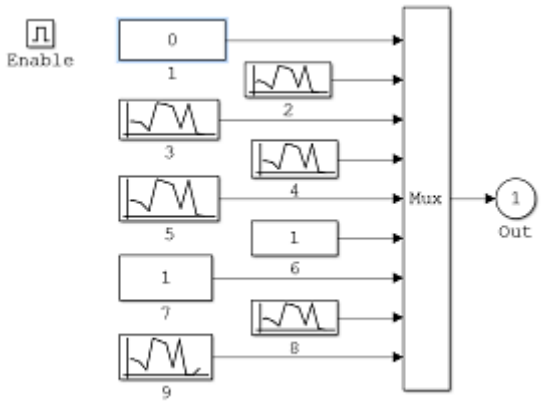


Рисунок 4.18 – Схема утворення з базових тестових наборів

4.3 Аналіз реального трафіка

Для аналізу підозрілого трафіку, подій і різного роду мережних атак, оцінки ризиків й управління ними необхідно мати зручну систему моніторингу мережевого трафіку. Інформація, що збирається системою Tcpdump є достатньою для виявлення аномалій.

Як приклад підозрілого трафіка, розглянемо фрагмент реального файлу дампу відобразивши його у вигляді таблиці Додаток А.

Для реалізації навчання нейронних мереж необхідно відділити нормальний стан трафіку від аномального (Додатки Б і В) та по можливості зменшити кількість параметрів, видаливши стовпці з даними, що не змінюються в будь-якій із ситуацій.

В нашому файлі представлені різні варіанти аномалій, що мають відповідні назви (apache2, guess_passwd, Httptunnel, Mailbomb, Mscan, Neptune, Portsweep, Processtable, Satan, Smurf, Snmpgetattack, Sqlattack, Warezmaster) та характерні відхилення.

Виконавши всі необхідні зміни ми отримали 2 вибірки з нормальнім та аномальнім трафіком, що описується 28 параметрами. (Додатки Г і Д).

Структура кожного із дев’яти елементів даної схеми зображено на рис. 4.19 (а-з).

4.4 Аналіз отриманих результатів

Для проведення аналізу отриманих результатів представимо отримані данні у табл. 4.2.

Таблиця 4.2 – отримані результати

	Тип нейромережі	Вихідні дані			
		Аномальний стан		Відсутність атаки	
1	Персептрон	0,0009878	0,999	0,999	0,0009878
2	Мережа прямого поширення	5,344e-0.5	1	1	5,344e-0.5
3	Рекурентна мережа	0,0002424	0,9997	0,9997	0,0002424

Проаналізувавши отримані дані можна досить точно зробити висновок, що всі нейронні мережі надали досить точні дані, їх значення настільки близькі до одиниці, що можна сказати, що обрані типи нейронних мереж, незважаючи на різницю їх структурної реалізацій, впоралися з поставленою задачею виявлення аномалій.

Але все ж необхідно виділити нейронну мережу прямого поширення, яка дала найточніші результати.

Для більшої наочності на рис. 4.19 покажемо поруч структури використаних мереж.

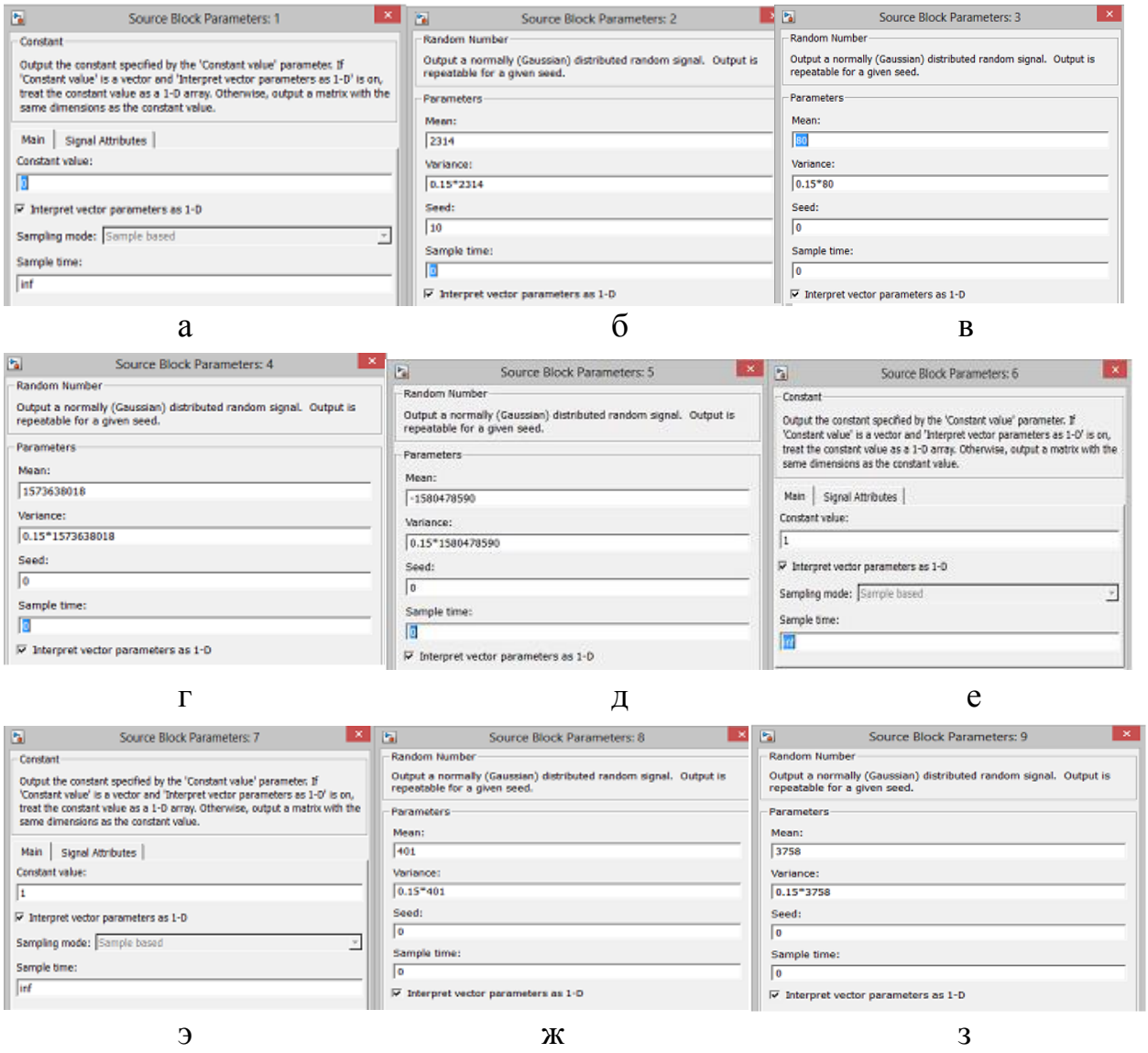


Рисунок 4.19 – Структура кожного тестового набору: а – Source BlockParametrs Protocol ID; б – Source Block Parametrs Source Port; в – Source Block Parametrs Source Address;г – Source Block Parametrs Destination Port; д – Source Block Parametrs Destination Address; е – Source Block Parametrs CMP Type ID; э – Source Block Parametrs CMP Code ID; ж – Source Block Parametrs Raw Data; з – Source Block Parametrs Length Data

Представлена програма дозволяє проаналізувати роботу запропонованої нейромережевої технології для ідентифікації атак. Автоматично генеруються випадкові вхідні набори, які відносяться до одного із класів, на екран одночасно виводиться сгенерований набір і рішення нейромережею завдання класифікації.

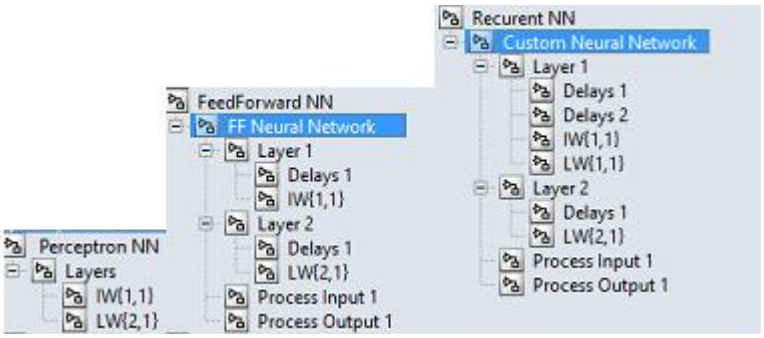


Рисунок 4.19 – Структури використаних мереж

Результат роботи віртуального нейромережевого процесора у випадку наявності та відсутності аномалій у трафіку зображено на рис. 4.20 і рис. 4.21.

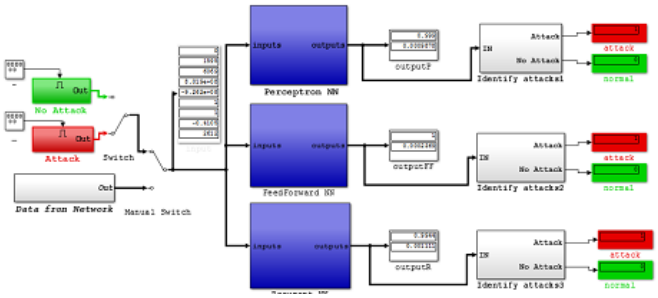


Рисунок 4.20 – Наявність аномалій

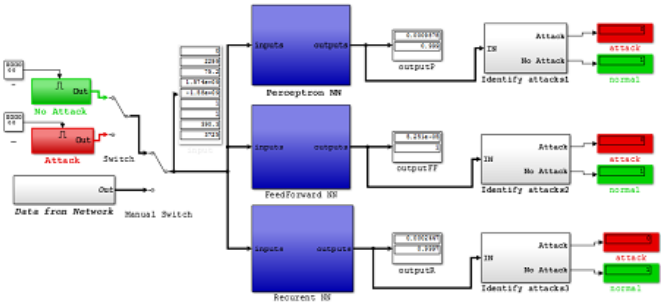


Рисунок 4.21 – Відсутність аномалій

ВИСНОВКИ

В ході виконання магістерської роботи розроблено нейромережевий процесор для ідентифікацій аномальних станів реального мережевого трафіку.

У роботі представлено новий підхід до процесу виявлення аномальних станів КМ, що використовує аналітичні можливості комітету нейромереж шляхом паралельної роботи мереж різної архітектури.

Створена модель віртуального процесору, що дає змогу розпізнати (ідентифікувати) аномальні стани в комп'ютерних мережах. Дослідження аномалій трафіку, дозволило створити багатофункціональний віртуальний процесор, на основі паралельної обробки трафіку, кожна складова процесора (нейромережа відповідної архітектури) може ідентифікувати визначений тип мережевих аномалій або створювати надлишкову систему для підвищення якості ідентифікації. Віртуальній процесор дозволяє проаналізувати роботу запропонованої нейромережевої технології для ідентифікації атак. Автоматично генеруються випадкові вхідні набори, які відносяться до одного із класів, на екран одночасно виводиться сгенерований набір і рішення нейромережею завдання класифікації.

Розв'язання задач ідентифікації аномалій трафіка на підставі представлення початкової задачі у формі задачі класифікацій, найбільш придатної для реалізації у НМ, дозволяє суттєво спростити головну задачу і підвищити ефективність функціонування КМ.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ажмухамедов И.М., Марьенков А.Н. Обеспечение информационной безопасности компьютерных сетей на основе анализа сетевого трафика// Вестн. Астрахан. гос. техн. ун-та. Сер. управление, вычисл. техн. информ., № 1, 2011. – С.137–141.

2. Астахов А. Актуальные вопросы выявления сетевых атак [Електронний ресурс] – URL: http://www.jetinfo.ru/2002/3/1/article1.3.2002.html (дата звернення: 27.09.2018).

3. Беляев А., Петренко С. Системы обнаружения аномалий: новые идеи в защите информации // Экспресс-Электроника №2, 2004. – С. 86–96.

4. Введение в сетевые атаки [Електронний ресурс] – URL: http://www.tshram.com/hacker/net_attacks.shtml#1 (дата звернення: 29.09.2018).

5. Воронцов К.В. Комбинаторные оценки качества обучения по прецеден-там // Докл. РАН. – 2004. – Т. 394, №2. – С. 175–178

6. Дюк В., Самойленко А. Data Mining: учебный курс, СПб: Питер, 2001. – 368 с.

7. Емельянова Ю. Г., Талалаев А. А., Тищенко И. П., Фраленко В. П. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы// Программные системы: теория и приложения : электрон. научн. журн. 2011, № 3(7), С. 3–15.

8. Жульков Е. Поиск уязвимостей в современных системах IDS// Открытые системы. СУБД – 2003. – N 7/8. С. 37 – 42.

9. Качановский Ю.П., Коротков Е.А. Предобработка данных для обучения нейронной сети// Фундаментальные исследования. – 2011. –№ 12 (часть 1). – С. 117–120.

10. Минаев Ю. Н., Филимонова О. Ю., Гузий Н. Н. «Интеллектуальные технологии в системах идентификации и прогнозирования атак на компьютерные сети»// Вестник НАУ. – 2006. – № 6. – С. 37–43.

11. Новіков О., Кащенко С. Розпізнавання сервісів tcp/ip за допомогою нейронних мереж// Періодичний науково-технічний збірник ” КПІ – 1, 2000. – С. 222–227.

12. KDD99 cup dataset. [Електронний ресурс] – URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (дата звернення: 15.10.2018).

13. Потемкин В.Г. Справочник по MATLAB [Електронний ресурс] – URL: http://matlab.exponenta.ru/ml/book2/index.php (дата звернення: 30.09.2018).

14. MathWorks [Електронний ресурс] – URL: http://matlab.ru/products/neural-network-toolbox (дата звернення: 18.10.2018).

15. Сапожников А. А. Обнаружение аномальной сетевой активности //Доклады Томского государственного университета систем управления и радиоэлектроники, 2009, № 1. – с. 79–80.

16. Тимофеев А., Браницкий А. Исследование и моделирование нейросетевого метода// International Journal "Information Technologies& Knowledge" Vol.6, Number 3, 2012. – С.257–265.

17. Черных И.В. Инструмент моделирования динамических систем [Електронний ресурс] – URL: http://matlab.exponenta.ru/simulink/book1 /index.php (дата звернення: 13.10.2018).

18. Perceptrons: anintroduction to computational geometry/ Marvin Minsky and Seymour Papert. Book ... Description, Cambridge, Mass.: MIT Press, 1988, p. 1969.

Д О Д А Т К И

Додаток А
Фрагмент файлу дампу

0	tcp	http	SF	340	3641	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	19	19	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	udp	private	SF	105	147	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	tcp	http	SF	395	10364	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	4	0.00	0.00	0.00	0.00	1.00	0.00	0.50	7	255	1.00	0.00	0.14	0.05	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	323	4449	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	4	0.00	0.00	0.00	0.00	1.00	0.00	0.50	105	255	1.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	306	12949	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0.00	0.00	0.00	0.00	1.00	0.00	0.00	67	237	1.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	304	5096	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	18	18	0.00	0.00	0.00	0.00	1.00	0.00	0.00	77	237	1.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	normal
0	tcp	other	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	255	15	0.06	0.01	0.00	0.00	0.00	0.00	0.04	0.60	httptunnel
0	tcp	http	SF	319	1861	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	38	255	1.00	0.00	0.03	0.04	0.00	0.00	0.00	0.00	normal
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0.00	0.00	1.00	1.00	1.00	0.00	0.00	255	3	0.01	0.96	0.00	0.00	0.00	0.00	0.96	1.00	neptune
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	12	0.00	0.00	1.00	1.00	0.92	0.15	0.00	255	12	0.05	0.93	0.00	0.00	0.00	0.00	0.96	1.00	neptune
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	3	0.00	0.00	1.00	1.00	0.13	0.09	0.00	255	3	0.01	0.89	0.00	0.00	0.00	0.00	0.96	1.00	neptune
7402	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	206	0.81	0.19	0.00	0.00	0.50	0.62	0.18	0.01	processtable
7398	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	207	0.81	0.19	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7394	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	208	0.82	0.18	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7390	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	209	0.82	0.18	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7386	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	210	0.82	0.18	0.00	0.00	0.50	0.60	0.17	0.01	processtable
7382	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	211	0.83	0.18	0.00	0.00	0.50	0.60	0.17	0.01	processtable
7378	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	212	0.83	0.17	0.00	0.00	0.50	0.60	0.17	0.01	processtable
1	tcp	smtp	SF	2599	293	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00	mailbomb
1	tcp	smtp	SF	2599	293	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00	mailbomb
1	tcp	smtp	SF	2599	293	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	3	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00	mailbomb
1	tcp	smtp	SF	2599	293	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	3	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00	mailbomb
1	tcp	smtp	SF	2599	293	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	247	0.97	0.04	0.00	0.00	0.00	0.00	0.03	0.00	mailbomb
0	tcp	private	RSTR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	255	1	0.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	portsweep
0	tcp	http	SF	365	452	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal

Додаток Б
Формування навчаючої вибірки з нормальним трафіком

1	tcp	smtp	SF	1661	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	3	0.00	0.00	0.00	0.00	1.00	0.00	1.00	172	126	0.37	0.04	0.01	0.02	0.01	0.00	0.02	0.02	normal
20	tcp	ftp	SF	232	765	0	0	0	4	0	1	0	0	0	0	0	0	0	0	0	1	2	1	0.00	0.00	0.00	0.00	0.50	1.00	0.00	179	48	0.27	0.04	0.01	0.00	0.01	0.02	0.02	0.00	normal
0	tcp	http	SF	322	680	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6	8	0.00	0.00	0.00	0.00	1.00	0.00	0.25	6	255	1.00	0.00	0.17	0.04	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	321	2060	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	18	0.00	0.00	0.00	0.00	1.00	0.00	0.11	8	255	1.00	0.00	0.12	0.03	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	234	14497	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	24	0.00	0.00	0.00	0.00	1.00	0.00	0.12	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	218	261	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	32	0.00	0.00	0.00	0.00	1.00	0.00	0.09	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	216	7504	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	20	21	0.00	0.00	0.00	0.00	1.00	0.00	0.10	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	ftp_data	SF	615	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0.00	0.00	0.00	0.00	0.50	1.00	0.00	174	56	0.32	0.03	0.32	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	312	677	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	9	37	0.00	0.00	0.00	0.00	1.00	0.00	0.11	22	255	1.00	0.00	0.05	0.04	0.00	0.00	0.00	0.00	normal
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	307	1528	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	7	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	246	753	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	9	9	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	245	4199	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	smtp	SF	908	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	240	101	0.38	0.03	0.00	0.02	0.01	0.00	0.01	0.03	normal
0	tcp	http	SF	232	2367	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	3	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	235	695	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	13	13	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	249	198	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	23	23	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	235	3817	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	11	11	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	243	2211	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	21	21	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	236	3665	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	31	31	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	237	675	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	41	41	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	240	662	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	51	51	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	240	2172	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	59	59	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	315	1528	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	normal
0	tcp	http	SF	313	5381	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1	255	1.00	0.00	1.00	0.01	0.00	0.00	0.00	0.00	normal
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	normal

ДОДАТОК В

Тестова вибірка з нормальним трафіком

0	105	146	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	29	0	0	0	0	0	0	0	0	2	1	0.00	0.00	0.50	1.00	0.00	10	3	0.30	0.30	0.30	0.00	0.00	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	253	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	223	185	0	1	0	0	0	0	0	4	4	0.00	0.00	1.00	0.00	0.00	71	255	1.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
0	230	260	0	1	0	0	0	0	0	1	19	0.00	0.00	1.00	0.00	0.11	3	255	1.00	0.00	0.33	0.07	0.33	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00
1	3170	329	0	1	0	0	0	0	0	1	2	0.00	0.00	1.00	0.00	1.00	54	39	0.72	0.11	0.02	0.00	0.02	0.00	0.09	0.13
0	297	13787	0	1	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	177	255	1.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
0	291	3542	0	1	0	0	0	0	0	12	12	0.00	0.00	1.00	0.00	0.00	187	255	1.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
0	295	753	0	1	0	0	0	0	0	21	22	0.00	0.00	1.00	0.00	0.09	196	255	1.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
0	268	9235	0	1	0	0	0	0	0	5	5	0.00	0.00	1.00	0.00	0.00	58	255	1.00	0.00	0.02	0.05	0.00	0.00	0.00	0.00
0	223	185	0	1	0	0	0	0	0	3	3	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	227	8841	0	1	0	0	0	0	0	13	13	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	222	19564	0	1	0	0	0	0	0	22	23	0.00	0.00	1.00	0.00	0.09	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	740	0	0	0	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	77	33	0.34	0.08	0.34	0.06	0.00	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	35195	0	0	0	0	0	0	0	0	10	10	0.00	0.00	1.00	0.00	0.00	92	44	0.43	0.07	0.43	0.05	0.00	0.00	0.00	0.00
0	8325	0	0	0	0	0	0	0	0	20	20	0.00	0.00	1.00	0.00	0.00	103	54	0.49	0.06	0.49	0.04	0.00	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	559	336	0	1	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	84	176	0.75	0.08	0.01	0.01	0.01	0.01	0.08	0.05
0	227	182	0	1	0	0	0	0	0	8	8	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	105	146	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	252	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00
0	317	278	0	1	0	0	0	0	0	3	3	0.00	0.00	1.00	0.00	0.00	192	255	1.00	0.00	0.01	0.04	0.00	0.00	0.00	0.00
1	1661	330	0	1	0	0	0	0	0	1	3	0.00	0.00	1.00	0.00	1.00	172	126	0.37	0.04	0.01	0.02	0.01	0.00	0.02	0.02
20	232	765	4	1	0	0	0	0	0	2	1	0.00	0.00	0.50	1.00	0.00	179	48	0.27	0.04	0.01	0.00	0.01	0.02	0.02	0.00
0	322	680	0	1	0	0	0	0	0	6	8	0.00	0.00	1.00	0.00	0.25	6	255	1.00	0.00	0.17	0.04	0.00	0.00	0.00	0.00

ДОДАТОК Г

Формування навчаючої вибірки з аномальним трафіком

0	tcp	private	RSTR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	255	1	0.00	1.00	1.00	0.00	0.00	0.00	1.00	1.00	portsweep
7402	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	206	0.81	0.19	0.00	0.00	0.50	0.62	0.18	0.01	processtable
7398	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	207	0.81	0.19	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7394	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	208	0.82	0.18	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7390	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	209	0.82	0.18	0.00	0.00	0.50	0.61	0.18	0.01	processtable
7386	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	210	0.82	0.18	0.00	0.00	0.50	0.60	0.17	0.01	processtable
7382	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	211	0.83	0.18	0.00	0.00	0.50	0.60	0.17	0.01	processtable
7378	tcp	telnet	SF	0	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	212	0.83	0.17	0.00	0.00	0.50	0.60	0.17	0.01	processtable
0	udp	other	SF	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	1	0.00	0.00	0.00	0.00	0.05	0.40	0.00	255	1	0.00	0.50	0.97	0.00	0.00	0.00	0.03	0.00	satan
0	udp	other	SF	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	1	0.00	0.00	0.00	0.00	0.06	0.53	0.00	255	1	0.00	0.49	0.97	0.00	0.00	0.00	0.03	0.00	satan
0	udp	other	SF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	1	0.00	0.00	0.00	0.00	0.06	0.50	0.00	255	1	0.00	0.48	0.97	0.00	0.00	0.00	0.03	0.00	satan
0	udp	other	SF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	1	0.00	0.00	0.00	0.00	0.05	0.80	0.00	255	1	0.00	0.52	0.97	0.00	0.00	0.00	0.03	0.00	satan
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	508	508	0.00	0.00	0.00	0.00	1.00	0.00	0.00	42	41	0.98	0.05	0.98	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	509	509	0.00	0.00	0.00	0.00	1.00	0.00	0.00	31	30	0.97	0.06	0.97	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	510	510	0.00	0.00	0.00	0.00	1.00	0.00	0.00	23	22	0.96	0.09	0.96	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	510	510	0.00	0.00	0.00	0.00	1.00	0.00	0.00	51	50	0.98	0.04	0.98	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0.00	0.00	0.00	0.00	1.00	0.00	0.00	24	23	0.96	0.08	0.96	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0.00	0.00	0.00	0.00	1.00	0.00	0.00	5	4	0.80	0.40	0.80	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0.00	0.00	0.00	0.00	1.00	0.00	0.00	66	65	0.98	0.03	0.98	0.00	0.00	0.00	0.00	0.00	smurf
0	icmp	ecr_i	SF	1032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	506	506	0.00	0.00	0.00	0.00	1.00	0.00	0.00	99	98	0.99	0.02	0.99	0.00	0.00	0.00	0.00	0.00	smurf
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	255	1.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	252	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00	snmpgetattack
0	udp	private	SF	105	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0.00	0.00	0.00	0.00	1.00	0.00	0.00	255	254	1.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	snmpgetattack

ДОДАТОК Д
Тестова вибірка з аномальним трафіком

2065	55744	0	0	1	0	0	0	0	0	11	11	1.00	1.00	1.00	0.00	0.00	255	244	0.96	0.01	0.00	0.00	0.02	0.02	0.53	0.55
2065	55744	0	0	1	0	0	0	0	0	12	12	1.00	1.00	1.00	0.00	0.00	255	244	0.96	0.01	0.00	0.00	0.02	0.02	0.53	0.56
2065	55744	0	0	1	0	0	0	0	0	13	13	1.00	1.00	1.00	0.00	0.00	255	244	0.96	0.01	0.00	0.00	0.02	0.02	0.54	0.56
2064	55744	0	0	1	0	0	0	0	0	10	10	1.00	1.00	1.00	0.00	0.00	255	244	0.96	0.01	0.00	0.00	0.02	0.02	0.54	0.57
0	126	174	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	206	0.81	0.02	0.00	0.00	0.00	0.00	0.02	0.03
0	124	174	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	207	0.81	0.02	0.00	0.00	0.00	0.00	0.02	0.03
0	127	174	0	0	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	208	0.82	0.02	0.00	0.00	0.00	0.00	0.02	0.03
0	120	174	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	209	0.82	0.02	0.00	0.00	0.00	0.00	0.02	0.03
0	123	174	0	0	0	0	0	0	0	1	1	0.00	0.00	1.00	0.00	0.00	255	210	0.82	0.02	0.00	0.00	0.00	0.00	0.02	0.03
0	0	0	0	0	0	0	0	0	0	1	1	1.00	1.00	1.00	0.00	0.00	255	15	0.06	0.01	0.00	0.00	0.00	0.00	0.04	0.60
1	2599	293	0	1	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00
1	2599	293	0	1	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00
1	2599	293	0	1	0	0	0	0	0	3	3	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00
1	2599	293	0	1	0	0	0	0	0	3	3	0.00	0.00	1.00	0.00	0.00	255	246	0.96	0.04	0.00	0.00	0.00	0.00	0.04	0.00
1	2599	293	0	1	0	0	0	0	0	2	2	0.00	0.00	1.00	0.00	0.00	255	247	0.97	0.04	0.00	0.00	0.00	0.00	0.03	0.00
5	0	0	0	0	0	0	0	0	0	2	3	0.50	0.67	0.50	1.00	1.00	6	58	0.17	0.50	0.17	0.05	0.00	0.02	0.17	0.97
0	0	44	0	0	0	0	0	0	0	3	4	0.33	0.50	0.33	1.00	1.00	70	34	0.01	0.07	0.01	0.06	0.00	0.00	0.01	0.97
2	24	40	0	0	0	0	0	0	0	4	15	0.75	0.40	0.50	0.75	0.60	255	3	0.01	0.02	0.00	0.00	0.00	0.00	0.01	0.33
7	0	15	0	0	0	0	0	0	0	1	23	1.00	0.48	1.00	0.00	0.61	3	85	0.33	0.67	0.33	0.04	0.00	0.01	0.33	0.69
2	0	36	0	1	0	0	0	0	0	3	9	1.00	0.89	0.33	1.00	0.89	218	13	0.00	0.03	0.00	0.15	0.00	0.92	0.01	0.08
4	24	1254	0	0	0	0	0	0	0	4	35	0.75	0.51	0.50	0.75	0.66	6	85	0.33	0.67	0.17	0.04	0.00	0.01	0.50	0.67
0	0	0	0	0	0	0	0	0	0	1	2	1.00	1.00	1.00	0.00	1.00	73	34	0.01	0.10	0.01	0.06	0.00	0.00	0.04	1.00
6	0	44	0	0	0	0	0	0	0	1	10	1.00	1.00	1.00	0.00	1.00	4	59	0.25	0.75	0.25	0.05	0.00	0.00	0.50	0.97
9	24	109	0	0	0	0	0	0	0	3	16	0.67	0.25	0.33	1.00	0.75	6	83	0.33	0.67	0.17	0.04	0.00	0.00	0.50	0.70
0	0	0	0	0	0	0	0	0	0	3	3	1.00	1.00	1.00	0.00	0.00	255	3	0.01	0.96	0.00	0.00	0.00	0.00	0.96	1.00
0	0	0	0	0	0	0	0	0	0	13	12	1.00	1.00	0.92	0.15	0.00	255	12	0.05	0.93	0.00	0.00	0.00	0.00	0.96	1.00
0	0	0	0	0	0	0	0	0	0	23	3	1.00	1.00	0.13	0.09	0.00	255	3	0.01	0.89	0.00	0.00	0.00	0.00	0.96	1.00