Хмельницький національний університет
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання
Назва теми

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

КРМКІ.015051.19.01.02 ПЗ

Виконав: студент 2 курсу, група КІ1м-19-1 Підпис Даценко В.С.

Керівник доц., к. т. н, доцент кафедри КбКСМ Підпис Тітова В.Ю.

Нормоконтролер доц., к. т. н, доцент кафедри КбКСМ Підпис Муляр І.В.

До захисту допускаю:
Зав. кафедри КбКСМ, к.т.н., доц Підпис Кльоц Ю.П.

10 12 2020 р.

Хмельницький, 2020

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ПРОГРАМУВАННЯ ТА ЗАХИСТ КОМП`ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ЗАТВЕРДЖУЮ

Зав. кафедри Ю.П. Кльоц

"1" 09 2020 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Даценко Владислав Сергійович
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання

Керівник роботи Тітова Вєра Юріївна
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання
кандидат технічних наук, доцент

Затверджена наказом № 118 ректора університету додаток №23 від 01.09.2020

2. Строк подання студентом проекту (роботи) на кафедру 20.11.2020

3. Вихідні дані до проекту (роботи) Удосконалена модель комп'ютерної системи для ідентифікації вторгнень.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Вступ. Способи застосування технологій нейронних мереж в системах виявлення вторгнень. Моделі нейромережі з метою їх застосування до виявлення вторгнень. Методи реалізації. Проектування програмного додатку. Висновки.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) Тема, мета магістерської, наукова новизна, практична значення, публікації. Дослідження систем виявлення вторнень на основі нейронних мереж. Розробка моделі нейромережі для виявлення вторгнень. Реалізація нейромережної моделі засобами Matlab. Структурна та функційна схеми системи виявлення вторгнень на базі нейронної мережі. Проектування програмного додатку. Висновки.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийня
Нормоконтроль	Муляр І.В. Доцент кафедри КбКСМ		

7. Дата видачі завдання «__» _____ 2020р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи	Термін виконання етапів роботи	Примітк
1	Вибір напряму дослідження та узгодження тематики КРМ з керівником	2.02.2020	
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	2.03.2020	
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	1.04.2020	
4	Робота над розділом 2 – розробка моделей і методів для вирішення поставленої задачі	1.05.2020	
5	Робота над науковою публікацією	1.06.2020	
6	Уточнення і затвердження теми	1.09.2020	
7	Робота над розділом 3 – розробка алгоритмів та технологій, їх аналіз	2.09.2020	
8	Робота над розділом 4 – апробація запропонованих рішень	1.10.2020	
9	Узгодження отриманих; оформлення пояснювальної записки згідно вимог	1.11.2020	
10	Попередній захист роботи	8.11.2020	
11	Попередній захист роботи	10.11.2020	
12	Захист роботи на засіданні ЕК	5.12.2020	

Студент Підпис Ініціали, прізвище

Керівник проекту (роботи) Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання

Автор роботи: Даценко Владислав Сергійович

Керівник роботи: к.т.н., доц. Тітова Вєра Юріївна

Загальний обсяг роботи: 108 сторінок, 19 рисунків, 2 таблиці, 2 додатків, 92 посилань.

Ключові слова: нейронна мережа, методи нейронної мережі, виявлення вторгнень, глибоке навчання.

Метою кваліфікаційної роботи є удосконалення моделі комп'ютерної системи для ідентифікації вторгнень.

Дана кваліфікаційна робота присвячена для удосконалення методу реалізації систем ідентифікації вторгнень на основі використання нейронних мереж глибокого навчання.

Дата

5. 12. 2020

Підпис студента

ANNOTATION

Theme of qualification work: Method of realization of systems of identification of intrusions on the basis of neural networks of deep training

Author of the work: Datsenko Vladislav Sergeevich

Mentor: Ph.D. Titova Vera Yuriyivna

Total volume of work: 108 pages, 19 figures, 2 tables, 2 appendices, 92 links.

Keywords: neural network, neural network methods, intrusion detection, deep learning.

The purpose of the qualification work is to improve the model of the computer system for identification of intrusions.

This qualification work is devoted to improving the method of implementation of intrusion identification systems based on the use of neural networks of deep learning.

Date

5.12.2020

Student's signature

ЗМІСТ

ВСТУП..... 7
1 СПОСОБИ ЗАСТОСУВННЯ ТЕХНОЛОГІЙ НЕЙРОННИХ МЕРЕЖ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ..... 15
1.1 Технології нейромереж..... 15
1.2 Поняття нейронної мережі..... 22
1.3 Проблеми навчання глибоких мереж..... 25
1.4 Переваги систем виявлення вторгнень на основі нейромереж..... 26
1.5 Недоліки систем виявлення вторгнень на основі нейромереж..... 27
1.6 Постановка задачі..... 29
2 МОДЕЛІ НЕЙРОМЕРЕЖІ З МЕТОЮ ЇХ ЗАСТОСУВАННЯ ДО ВИЯВЛЕННЯ ВТОРГНЕНЬ..... 30
2.1 Моделі нейронних мереж..... 30
2.2 Моделі глибинного навчання..... 35
2.3 Алгоритми глибинного навчання..... 43
2.4 Розробка моделі глибинної нейромережі для виявлення вторгнень..... 46
2.5 Висновки..... 54
3 МЕТОДИ РЕАЛІЗАЦІЇ..... 55
3.1 Потенційні методи реалізації..... 55
3.2 Інструмент NNT пакету Matlab..... 56
3.3 Реалізація нейронних мереж NNT..... 66
3.4 Висновки..... 70
4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ..... 71
4.1 Розмежування функцій виявлення вразливостей вторгнення нейромереж та системи прийняття рішень..... 71
4.2 Структурна та функційна схеми виявлення вторгнень на основі нейронної мережі..... 72
4.3 Опис програмних модулів інтелектуальної системи для виявлення мережевих вторгнень..... 75

4.4 Висновки……..... 75
ВИСНОВКИ…….. 77
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ…. 78
ДОДАТОК А Копії публікацій 87
ДОДАТОК Б Презентація..... 96

ВСТУП

Актуальність роботи. На сьогоднішній день важливість системи виявлення вторгнень у комп’ютерних мережах одне із головних питань які мають вирішуватись. Бездротові мережі в наші дні швидко розвиваються і розповсюджуються, надаючи великій кількості користувачів можливість бути мобільним 24/7. В той же час постає питання про те що кожен користувач хоче бути захищеним від посягань на особисту інформацію. Тому, за даними опитувань, це і є основним викликом для бездротових мереж. Для підвищення продуктивності системи виявлення вторгнень було використано багато методів, найбільш перспективним є розгортання машинного навчання. Саме тому ми і будемо висвітлювати корисність останніх моделей машинного навчання, які називаються глибоким навчанням, для покращення продуктивності системи виявлення вторгнень, особливо як підхід до функціонального навчання. Ми також пояснюємо мотивацію опитування системи виявлення вторгнень на основі глибокого навчання. Комп’ютерні мережі та Інтернет невід’ємні від людського життя сьогодні. Часто програми покладаються на Інтернет, включаючи життєво важливі програми в галузі охорони здоров’я та військових. Більше того, екстравагантні фінансові операції працюють через Інтернет щодня. Це стрімке зростання Інтернету призвело до значного збільшення трафіку бездротової мережі за останні роки. За даними всесвітнього консорціуму телекомунікації, організаціям, що надають послуги мобільного та бездротового зв'язку інформаційного суспільства (METIS) [1], поширення мереж 5G та Wi-Fi очікується в найближчі десятиліття. Вони вірять, що лавина обсягу мобільного та бездротового трафіку відбудеться через розвиток суспільства, яке потрібно виконати. Такі програми, як електронне навчання, електронний банкінг та електронна система охорони здоров'я, поширюватимуться та стануть більш мобільними. Зросла велика кількість вторгнень, особливо через величезний ринок продажів речей через інтернет.

IBM [2] повідомив про величезний викрадений обліковий запис протягом

2016 року. Спам-повідомлення в чотири рази перевищують показники попереднього року. Стали поширеними напади, зазначені в тому ж звіті, зловживання рекламою, фішинг, SQL, DDoS, шкідливе програмне забезпечення тощо. Більшість шкідливих програм є вимагаючими (85% шкідливих програм, що існували протягом року, є вимогаючими). Ці атаки можуть призвести до витоку конфіденційних даних або порушення нормальної роботи, що призведе до величезних фінансових втрат. Найпопулярнішими компаніями, на яких були здійснені напади, є компанії, пов'язані з фінансовими послугами, за ними йдуть інформація та комунікації, виробництво, роздрібна торгівля та охорона здоров'я [2]. Бездротові мережі, такі як IEEE 802.11, широко розгорнуті, щоб забезпечити користувачам мобільність та гнучкість у вигляді високошвидкісного підключення до локальної мережі. Це висвітило і інші питання, такі як конфіденційність та безпека. Швидке поширення пристроїв з підтримкою IoT призвело до того, що бездротові мережі стали схильні як до пасивних, так і активних атак, кількість яких різко зросла[3]. Прикладами таких атак є атаки на видавання себе за іншу особу, захоплення та викрадення данних. Широке та швидке розповсюдження обчислювальних пристроїв, що використовують мережі Wi-Fi, створює складні та надвеликі розміри даних, які викликають плутанину при визначенні властивостей атак і змушують нас посилити наші заходи безпеки в нашій системі.

Були проведені комплексні дослідження, щоб уникнути нападів, як згадувалося раніше. Система виявлення вторгнень є одним із найпоширеніших компонентів кожної інфраструктури мережевої безпеки [4], включаючи бездротові мережі[5]. Методи машинного навчання добре себе зарекомендували, як основний алгоритм виявлення в системи виявлення вторгнень, завдяки їх безмодельним властивостям та навчанню[6]. Використовуючи недавній розвиток таких методів машинного навчання, як глибоке навчання[7], можна очікувати, що вони принесуть значні вигоди від вдосконалення існуючих систем виявлення вторгнень, особливо для виявлення атак уособлення у великомасштабних мережах. На основі методу системи виявлення вторгнень можна класифікувати

на три типи: системи виявлення вторгнень з неправильним використанням, аномалією та специфікацією системи виявлення вторгнень на основі зловживання, також відомі як системи виявлення вторгнень на основі підпису [8], виявляє будь-яку атаку, перевіряючи, чи відповідають характеристики атаки раніше збереженим підписам або шаблонам атак. Цей тип систем виявлення вторгнень підходить для виявлення відомих атак. Однак нові або невідомі атаки важко виявити.

Широкі дослідження із застосуванням методів машинного навчання в системи виявлення вторгнень проводились як в академічних колах, так і в промисловості. Однак експерти з питань безпеки все ще переслідують більш високу продуктивність системи виявлення вторгнень, яка має найвищий коефіцієнт виявлення і найнижчу частоту помилкових тривог. Крім того, очікується, що загальний аналіз загроз захистить їх мережі [9]. Покращень в системи виявлення вторгнень можна досягти шляхом охоплення недавнього прориву в машинному навчанні [6], так зване глибоке навчання. Програми глибокого навчання вигравали численні конкурси з розпізнавання зразків та машинного навчання [10]. Поглиблене навчання належить до класу методів машинного навчання, який використовує послідовні шари етапів обробки інформації в ієрархічних манерах для класифікації зразків та навчання ознак чи репрезентацій [11]. Існує три важливі причини [12], що спричинили глибоке навчання протягом останнього часу. По-перше, різко зросли обробні можливості (наприклад, модулі графічного процесора). По-друге, обчислювальне обладнання стає доступним, а третє - недавній прорив у дослідженнях машинного навчання. Неглибокі та глибокі навчальні мережі відрізняються глибиною їхніх шляхів присвоєння кредитів, які є ланцюжками можливих причин вивчення причинно-наслідкових зв'язків між діями та наслідками. Зазвичай глибоке навчання відіграє важливу роль у результатах класифікації зображень. Крім того, глибоке навчання також широко використовується для мови графічного моделювання, розпізнавання образів, мови, аудіо, зображення, відео, природної мови та обробки сигналів [11]. Існує багато методів глибокого навчання, таких як Deep

Belief Network (DBN), Machine Boltzmann (BM), Restricted Boltzmann Machine (RBM), Deep Boltzmann Machine (DBM), Deep Neural Network (DNN), auto-encoder, Deep Auto-Encoder (DAE), автоматичний кодер з накопиченням (SAE), автоматичний кодер з обмеженням шуму (SDAE), розподілене представлення та згорткова нейронна мережа (CNN).

Широке та швидке розповсюдження обчислювальних пристроїв, що використовують Інтернет, особливо мереж Wi-Fi, створює складні, великі та багатовимірні дані, які викликають неминучі плутанини при визначені виявлених атак. Навчання функції діє як важливий інструмент для вдосконалення процесу навчання моделі машинного навчання. Він складається з побудови об’єктів, вилучення та відбору. Конструкція об’єкта розширює оригінальні функції, щоб підвищити їх виразність, тоді як виділення об’єктів перетворює оригінальні об’єкти в нову форму, а вибір об’єктів усуває непотрібні функції [13]. Навчання функції - це ключ до підвищення ефективності існуючих IDS на основі машинного навчання.

Ми усвідомлюємо, що існує плутанина щодо того, як правильно застосувати глибоке навчання в програмах системи виявлення вторгнень, оскільки кожен попередній підхід застосовував різні підходи. Деякі дослідження використовують методи глибокого навчання лише частково, тоді як решта все ще використовують звичайні нейронні мережі. Складність методу глибокого навчання може бути однією з причин. Крім того, метод глибокого навчання вимагає багато часу для правильного тренування. Однак ми виявили, що кілька дослідників застосовують метод глибокого навчання для вивчення особливостей та класифікації для інтелектуальних систем виявлення вторгнень у своїй мережі. Ми порівнюємо ефективність системи виявлення вторгнень серед них.

Тран та ін. [14] подав один із прикладів того, як використовувати глибоке навчання в IDS. Класичні алгоритми машинного навчання, Naive Bayes та C4.5, за допомогою функцій високого рівня були створені за допомогою генетичного програмування. Цей підхід є загальним способом використання моделей глибокого навчання в IDS, де моделі глибокого навчання допомагають будь-

якому класичному машинному навчанню з функціями високого рівня. Цей підхід також прийнятий Aminanto та співавт. [15]. У цій монографії ми виділили деякі IDS, які використовують моделі глибокого навчання. Хамед та ін. [16] обстежив декілька методів попередньої обробки в дослідженнях IDS, як збирати дані з реального світу, як будувати набір даних із вихідних даних. Хоча більшість IDS використовують моделі глибокого навчання як свою техніку попередньої обробки даних, ця монографія зосереджена на огляді IDS на основі глибокого навчання.

Поглиблене навчання корисно для систем виявлення вторгнень, особливо для навчання функції. У цій монографії розглядаються методи глибокого навчання з їх плюсами та мінусами, щоб краще зрозуміти, як застосовувати глибоке навчання в системах виявлення вторгнень. Зрештою, ми розглядаємо майбутні виклики та вказівки щодо використання глибокого навчання в системах виявлення вторгнень відповідно.

Актуальність проведеного дослідження обумовлюється необхідністю вирішення завдань виявлення вторгнень превентивними методами. Значимість застосування штучного інтелекту в контексті виявлення атак полягає в його ефективності прийняття рішень в умовах невизначеності.

Сучасний світ не можна уявити без читання і обробки інформації. Обсяг інформації, яку отримує людина, росте у величезній кількості. І ця інформація може бути оброблена різними інформаційними системами. На даний час найпростіший для інженера-програміста спосіб, це нейронна мережа. Нейронними мережами обробляється будь-яка інформація, від графічної до величезних масивів даних.

Останні роки знаменуються швидким розвитком інформаційних технологій, пов'язаних з мережею Інтернет. Багато компаній вже не можуть обійтися без застосування автоматизованих систем, побудованих з використанням мережі. Через те , що мережеві технології дозволяють передавати інформацію з великою швидкістю і практично будь-якому одержувачу. З цієї причини поширення отримують такі системи, як системи електронної комерції,

Інтернет-магазини, системи документообігу та здійснення банківських транзакцій. Тому що передача інформації через Інтернет є єдиним оптимальним рішенням проблеми оперативності, так як мережеві рішення забезпечують необхідну швидкість.

Хоча існує зростаюча потреба в системі, здатної безпомилково виявляти вторгнення в мережі, на даний момент не існує альтернативи до системи виявлення вторгнень на основі правил. Цей метод зарекомендував себе порівняно ефективним, за умови, що точні характеристики атаки відомі. Проте, мережеві атаки постійно змінюються через індивідуальності підходів зловмисників і регулярних змін в програмному забезпеченні і апаратних засобах цільових систем. Через нескінченної кількості атак і безлічі зловмисників, навіть цілеспрямовані зусилля на постійне оновлення бази правил експертної системи ніколи не зможуть точно ідентифікувати різні вторгнення.

Постійно мінливий характер мережевих атак вимагає гнучку захисну систему, яка здатна аналізувати величезну кількість мережевого трафіку за методом, який менш структурований ніж той, що заснований на побудові певних правил. Система виявлення вторгнень на основі нейронної мережі може потенційно вирішити багато з проблем, які мають місце бути в системах, заснованих на правилах.

Одне з питань забезпечення безпеки інформації - метод кодування повідомлень. Для запобігання атак на інформацію, що володіє високим ступенем секретності, спробуємо застосувати технологію нейронних мереж. За допомогою даної технології буде проводитися зміна алгоритму шифрування, ключа шифрування, кодування сигналу. Зміна буде відбуватися через різні проміжки часу, вибрані випадковим чином.

Кількість інцидентів в області інформаційної безпеки постійно збільшується. Атаки на обчислювальні мережі вже давно перестали бути справою обраних фахівців. В даний час кошти для злому може знайти будь-яка людина, що не володіє глибокими знаннями в області інформаційної безпеки для їх ефективного використання. Тому атаки на корпоративні мережі і ПК

звичайних користувачів стають все більш буденною справою. Тому важливим завданням стає розробка і вдосконалення засобів захисту.

Отже, із вищесказаного випливає, що тема кваліфікаційного дослідження є актуальною.

Метою кваліфікаційного дослідження є удосконалення моделі комп'ютерної системи для ідентифікації вторгнень.

Для досягнення цієї мети під час навчання необхідно вирішити наступні завдання:

1. Проаналізувати способи застосування технологій нейронних мереж в системах виявлення вторгнень та їх проблеми з перевагами й недоліками, щоб визначити найпоширеніші способи виявлення вторгнень.

2. Розглянути моделі нейромережі з метою їх застосування до виявлення вторгнень.

3. Запропонувати метод покращення реалізації для систем виявлення вторгнень.

4. Розробити алгоритм, що дозволяє виконувати глибоке навчання в нейронній мережі для виявлення вторгнень.

5. Розробити модель глибинної нейромережі для виявлення вторгнень.

6. Впровадити та протестувати розроблений алгоритм.

7. Визначити сферу застосування розробленого методу.

Об'єктом дослідження є нейромережа з поглибленим навчанням для забезпечення виявлення вторгнень.

Предметом дослідження є алгоритми та методи глибинного навчання нейронної мережі.

Методи дослідження, що використовуються в даній роботі, базуються на алгоритмах та методах навчання нейромереж.

Наукова новизна: вдосконалено метод ідентифікації вторгнень за рахунок впровадження в його структуру нейронних мереж, розробка набутого та подальший розвиток моделі системи ідентифікації вторгнень за рахунок додання правил.

Практична цінність: розроблений додаток удосконалює модель комп’ютерної системи для ідентифікації вторгнень.

За темою кваліфікаційної роботи опубліковано 2 статті і 1 теза доповіді на міжнародній конференції.

1 СПОСОБИ ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ НЕЙРОННИХ МЕРЕЖ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ

1.1 Технології нейромереж

В даний час існує багато програмних продуктів, спрямованих на вирішення такого актуального завдання, як виявлення вторгнень. Практично у всіх цих засобах використовуються методи штучного інтелекту, так як про наявність атаки можна судити лише приблизно, оцінюючи параметри системи[17].

Завдання виявлення вторгнень в обчислювальні мережі зазвичай вирішуються із застосуванням:

- експертних систем;
- штучних нейронних мереж;
- нечітких систем;
- генетичних алгоритмів.

Розглянемо докладніше інструментарій штучного інтелекту, застосовуваний для виявлення вторгнень.

Найбільш часто в системах виявлення вторгнень застосовуються експертні системи. Даний факт можна пояснити тим, що сигнатурний метод аналізу мережевого трафіку є найбільш швидким і не вимагає великих обчислювальних потужностей. Найвідомішими представниками систем виявлення вторгнень на основі експертних систем є: McAfee, Tripwire, IBM ISS, Snort.

Найбільшим недоліком експертної системи як системи виявлення вторгнень є нездатність в принципі виявляти нові види атак. Крім того, відомо безліч технологій обходу систем виявлення вторгнень на основі експертних систем, наприклад, polymorphic shell code, insertion, exclusion і т.д.

Підхід до виявлення вторгнень, заснований на застосуванні експертних систем, широко застосовується в практичних додатках, тому в подальшому детально не розглядається і вважається класичним.

Штучна нейронна мережа – представляє собою математичну модель,

програмну або апаратну її реалізацію, яка створена за аналогією організації і функціонування біологічних нейронних мереж нервових клітин живого організму[17]. Обмеженість застосування нейронних мереж в системах виявлення вторгнень обумовлена вимогою великих обчислювальних потужностей і неможливість оперативного аналізу великих обсягів даних в умовах роботи в якості NIDS (мережева система виявлення вторгнень) великої корпоративної мережі. Як приклад розробок в області застосування нейронних мереж в NIDS розглянемо нижче кілька найбільш відомих досліджень.

Жигулін П.В. і Подворчан Д.Е. використовували для детектування атак двошаровий персептрон з одним прихованим шаром по схемі 38 вхідних, 38 прихованих, 10 вихідних нейронів[18] . І тому в експериментах 38 вхідних векторів представляли собою числові еквіваленти найбільш значущих ознак набору даних KDD CUP 99. У прихованому шарі функціонували 38 вирішальних нейронів. У вихідному шарі було використано 10 виходів, є ідентифікаторами різних типів атак. Точність типу атаки досягла 98%. Однак недоліком розробки є те, що нейронна мережа тестувалася всього на 20% тестового набору, навчання та калібрування мережі відбувалася на інших 80%, тобто мережа «знала» практично все про тестовий набір.

Дослідники МОРАДА М. і Зелкернін М. з університету Queen використовували кілька схем реалізації нейронної мережі і отримали наступні результати [19] :

Схема: 35 вхідних, 35 прихованих нейронів першого рівня, 35 прихованих нейронів другого рівня, 3 вихідних нейрона показала 91% вірних рішень на тестових прикладах;

Схема: 35 вхідних, 45 прихованих, 3 вихідних нейронів показала 87% вірних рішень на тестових прикладах;

Схема: 41 вхідних, 40 прихованих нейронів першого рівня, 40 прихованих нейронів другого рівня, 1 вихідний нейрон показала 99% вірних рішень на тестових прикладах.

У перших двох схемах в якості вихідних векторів використовувалися

схеми:

[1,0,0] - нормальний стан;

[0,1,0] - відмова в обслуговуванні;

[0,0,1] - сканування.

Третя схема з великою ймовірністю визначає наявність атаки, але не її тип.

Як видно з отриманих результатів [19], збільшення числа прихованих шарів не призводить до значного поліпшення якості роботи мережі (всього 4%) при експоненціально зростаючої складності схеми і, як наслідок, часу аналізу.

Дослідники Кліонскій Д.М., Большев А.К. і Геппенер В.В. розробили систему на основі HNIDS (Heuristic Network Intrusion Detection Systems), яка використовує одношаровий класифікатор на базі штучних нейронних мереж [20].

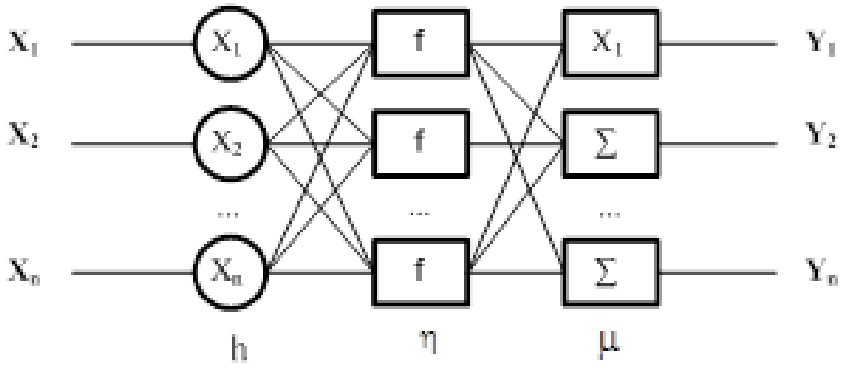


Рисунок 1.1 - Нейронна мережа, що представляє собою одношаровий класифікатор

На рисунку 1.1 - представлена штучна нейромережа, що реалізує роботу одношарового класифікатора. Як функції активації, використовується сигмоїдальна функція активації, де:

h - кількість нейронів прихованого шару;

η - коефіцієнт швидкості навчання;

μ - коефіцієнт інерційності.

Головною відмінністю такого типу систем від інших є те, що система навчається на «нормальному» трафіку цільової мережі і в разі виявлення

відхилень повідомляє про атаку або аномалії. Система працює на мережевому (транспортному) рівнях моделі відкритих систем і аналізує завершення TCP-сесії між хостами.

При тестуванні прототипу дослідниками в тестовій вибірці використовували 17 вторгнень мережевого рівня. За допомогою варіювання параметрів штучної нейромережі, проводилися мінімізації за критеріями помилкової тривоги (FP) і пропуску сигналів (FN). При мінімізації за критерієм FP, прототип виявив 12 атак при 2 помилковому спрацьовуванні. При мінімізації за критерієм FN, прототип виявив 16 атак при 1878 помилковому спрацьовуванні.

У своїх дослідженнях Жулька Є.В. запропонував розбити трафік на вектори і за допомогою системи виявлення вторгнень (СОВ), побудованої за модульним принципом, аналізувати трафік як вектори [21].

Алгоритм пошуку вторгнень зводиться до послідовності наступних дій.

Підготовка бази даних атак. На підставі частини бази DARPA [22] яка не містить вторгнень, генерація бази для генератора шуму (система, що видає вектори нормального трафіку, використовується для навчання нейронної мережі).

Виділення параметрів міжмережевої взаємодії (основні параметри трафіку, використовувані в векторах). Створення і навчання нейронної мережі першого рівня, навчання другого рівня нейронної мережі вихідними даними першого рівня, тестування, робота СОВ. Ймовірність виявлення відомих атак склала 91%, ймовірність виявлення невідомих атак склала 86%.

Хафизов А.Ф. запропонував використовувати гібридну нейронну мережу для аналізу піфограмм атак [23]. На першому етапі роботи гібридної штучної нейронної мережі, на безлічі вхідних векторів навчається шар Кохонена. В результаті нейрони цього шару самоорганізуються таким чином, що вектори їх ваг найкращим чином відображають розподіл даних навчальних векторів. Далі ваги фіксуються і на вхід подається навчальна вибірка, потім відбувається фінальне коригування ваг нейронів. На другому етапі навчається персептрони

мережі. Навчання відбувається з учителем. Для даної мережі навчальні сигнали формуються з вихідних сигналів прошарку Кохонена і вектора очікуваних значень.

Результатом роботи такої нейронної мережі є віднесення вхідних даних до класу атак або до класу нормальних взаємодій. Ефективність системи полягає в тому, що розроблена методика виявлення мережевих вторгнень перевершує існуючі рішення на 15%.

Нечіткі системи знайшли своє застосування в якості компонента системи виявлення вторгнень, так як вони оперують «нечіткими» і «розмитими» даними [18], якими і є вектори атак на обчислювальні мережі. Як приклад застосування нечітких систем в IDS, розглянемо кілька робіт вчених.

Дослідники з Індії Шанмагавадіва та Нагаражан створили систему виявлення вторгнень на основі нечіткої логіки. Схема роботи мережі представлена на рисунку 1.2.

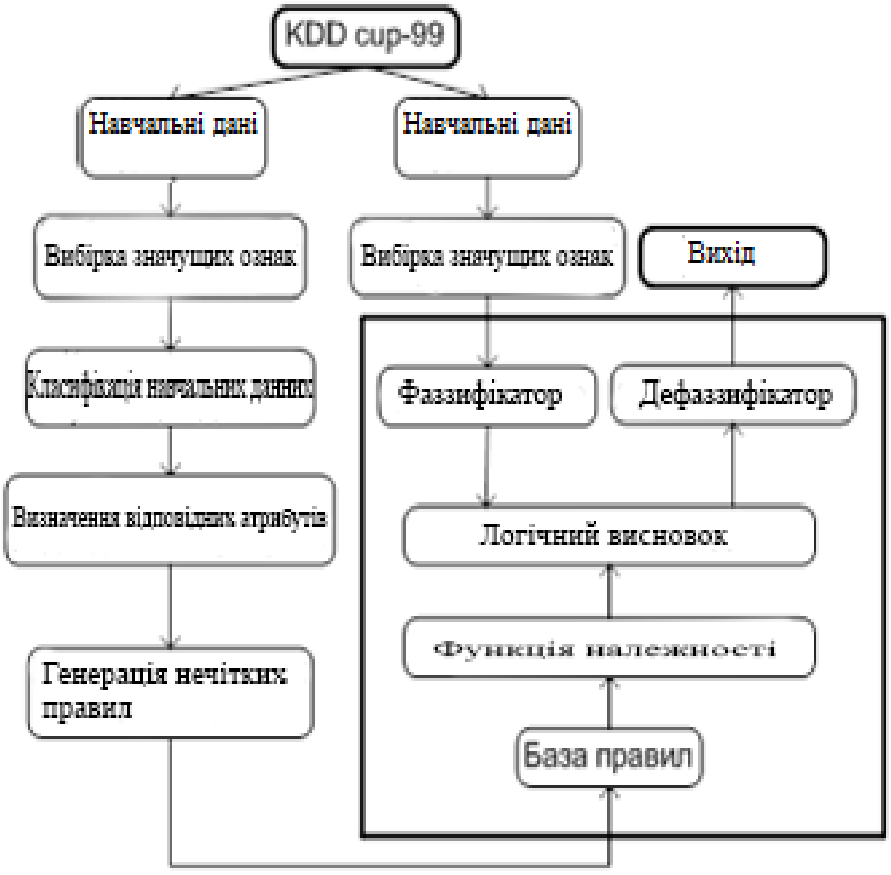


Рисунок 1.2 - Схема роботи СОВ на основі нечіткої логіки

Авторам [24] вдалося досягти більше 90% спрацьовувань, причому лише 10% набору використовувалося для створення бази нечітких правил.

Дослідники Слепович І.І., Ірматов П.В., Комарова М.С. і Бежін А.А. [25] розробили систему виявлення SYN Flood атак на основі нечіткої нейронної мережі. Метод навчання штучної нейромережі - метод зворотного поширення помилки.

На підставі розробленої моделі дослідниками була розроблена програма, яка, використовуючи математичний апарат нечіткої логіки та нейронних мереж, визначає ступінь впевненості в наявності атаки.

При синтезі алгоритмів активного аудиту інформаційної системи Кашаєва Т.Р. була розроблена система виявлення вторгнень на основі штучних імунних систем [26].

Розроблена система використовує нечіткі мережі Петрі. Система в загальному випадку працює наступним чином:

- Визначаються нормальні шаблони активності системи (безліч S) у вигляді рядків однакової довжини l, складених з літер кінцевого алфавіту.
- Генерується набір детекторів R, кожен з яких не збігається ні з однією з рядків з нормального шаблону активності. При цьому кандидат у детектори вважається схожим з нормальним шаблоном в тому і тільки в тому випадку, коли збігаються символи в R однакових позиціях. Величина R підбирається відповідно до розв'язуваного завдання.
- Дані контролюються шляхом зіставлення детекторів з поведінкою системи. Будь-який збіг на даному етапі означає зміну в роботі системи (аномалію).

На підставі розробленої моделі був реалізований прототип. Тестування показало, що система виявляє до 85% атак.

Свечников Л.А. [27] в рамках створення інтелектуальної системи

виявлення атак на основі імітаційного моделювання показав ефективність використання нечітких когнітивних карт в області виявлення атак [28].

Даний підхід заснований на виділенні сукупності основних факторів (концептів), що позначають різні поняття моделюється предметної області і побудові на їх основі орієнтованого графа, що відображає взаємозв'язки між концептами. Кожному і-му концепту нечіткої когнітивної карти ставиться у відповідність змінна стану Xi і вага Wij, що характеризує вплив і-го концепту на j-й концепт. Величина ваги Wij лежить в межах відрізка [0; 1] і характеризує ступінь значущості (впливу) відповідного концепту.

На підставі розробленого алгоритму був створений дослідний прототип системи виявлення атак на основі нечітких когнітивних карт, який реалізує запропоновані алгоритми виявлення атак за результатами моделювання ризиків ІС в режимі реального часу. У проведених експериментах розроблений прототип системи виявлення атак дозволяє розпізнати і блокувати до 97% атак на захищених компонентах інформаційної системи.

Хоча основною сферою застосування генетичних алгоритмів є оптимізація запитів бази даних [17], генетичні алгоритми також можуть бути частиною системи виявлення вторгнень. Розглянемо роботу, яка дозволяє говорити про перспективність цього напрямку.

Вчені Ануп Гоял і Четан Камар з Northwestern University створили систему виявлення вторгнень GA-NIDS, засновану на генетичному алгоритмі [29]. Схема роботи системи представлена на рисунку 1.3.

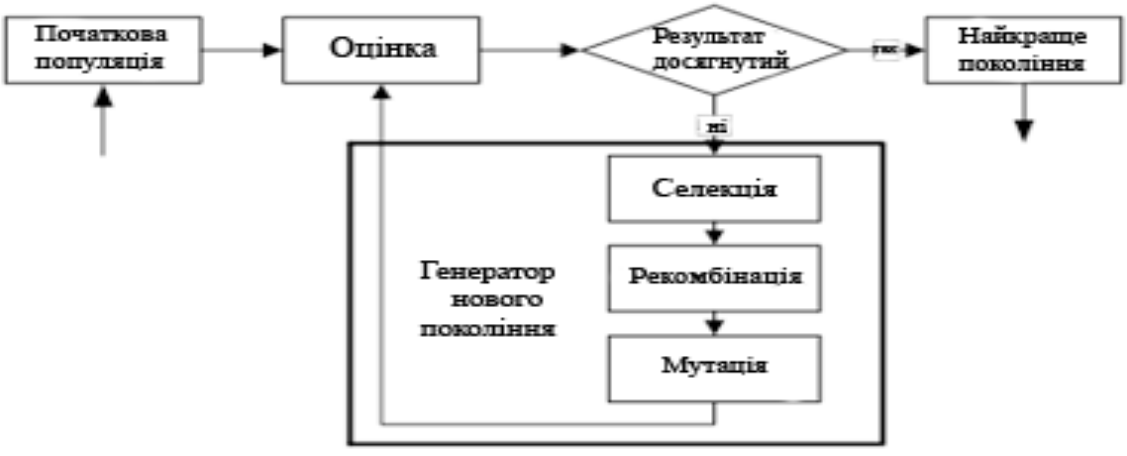
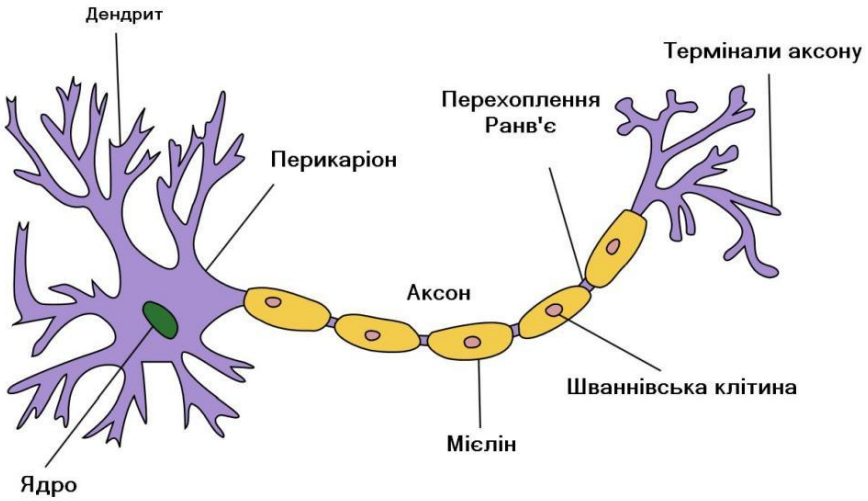


Рисунок 1.3 - Схема роботи GA-NIDS

Для генерації правил використовувалося 10% набору KDD CUP 99. Згенеровані правила показали більше 95% правильних рішень на тестових прикладах.

1.2 Поняття нейронної мережі

Штучні нейронні мережі були створені по аналогії біологічних нейронних мереж, що представляють собою мережі нервових клітин за фізіологічними функціями. Елементами нейронних мереж є нейрони які представлені на рисунку 1.4.



Рисунку 1.4 - Структура нейрона

Функції нейрона:

- Приймаюча функція: вхід інформації;
- Функція інтеграції: сигнал, який надає інформацію про всі підсумовувані в нейроні сигнали;
- Провідникова функція: до синапсів проходить інформація по аксону;
- Передаюча функція: імпульс, після досягнення закінчення аксона, передає збудження наступного

нейрона.

Синапсами називають збірку, за якою вихідні сигнали одних нейронів надходять на входи інших. Кожна збірка характеризується своєю вагою. Збірки з позитивною вагою називаються збудливими, а з негативними - гальмуючими. Аксоном називається вихідний нейрон. Штучний нейрон в штучній нейронній мережі - це нелінійна функція, якою є лінійна комбінація всіх вхідних сигналів, вона є активуючою. ЇЇ результи подаються на вихід нейрона. Об'єднання таких нейронів з іншими нейронами і утворюють штучну нейромережу.

Темп навчання: для початку гіпер-параметру мережі необхідно визначити кінцеве значення для η, коли миттєво знижується значення цільової функції без коливань. Значення оцінки встановлюється $\eta = 0.01$. Коли значення цільової функції знижується під час перших епох, то необхідно збільшувати темп навчання до тих пір, поки буде не знайдено значення коливання цільової функції. Якщо ж значення цільової функції коливаються при початковому темпі навчання, то необхідно його зменшувати. Розмір кроку в градієнтному спуску регулює темп навчання спостерігаючи за значеннями цільової функції, визначаючи, чи був занадто великим розмір кроку градієнтного спуску;

Рання зупинка (early stopping) для визначення розміру епох навчання, означає, що наприкінці кожної епохи необхідно обчислити вірогідність класифікації даних при перевірці (validation set). Процес навчання необхідно зупинити при поліпшенні точності, це допоможе запобігти перенавчанню.

Графік навчання зберігає темп навчання незмінним до моменту погіршення достовірності даних, при цьому необхідно зменшити темп навчання.

Параметр регуляризації λ: після визначення η, можна почати з $\lambda = 1.0$ і потім збільшувати або зменшувати значення;

Розмір пакетів: при занадто малих розмірах пакетів, переваги хороших матричних бібліотек неможливо повністю використовувати. При занадто великому розмірі пакетів ваги мережі будуть оновлюватись дуже рідко. Потрібно вибирати значення, яке максимально підвищить швидкість навчання.

Глибокими нейронними мережами називають мережі з кількома прихованими шарами. Кожен прихований шар визначає нелінійне перетворення попереднього шару. Глибока мережа теоретично матиме набагато більшу потужність (вона представлятиме більш складніші функції), ніж нейромережа з меньшою кількістю шарів. При навчанні глибокої мережі є сенс застосовувати цю нелінійну функцію активації кожного прихованого шару. Отже велика кількість шарів лінійних функцій самі б вираховували тільки лінійну функцію введення, і не були б більш виразними, ніж застосування тільки одного прихованого шару.

Головною перевагою глибинних мереж є стисле уявлення достатньо великої безлічі функцій. Можна сказати, що є функції, які К-шарова мережа може представляти стисло, (К-1) – якщо шарова мережа не має експоненціально великої кількості елементів в прихованих шарах, то вона не може цього зробити.

Метод доступності даних покладається тільки на марковані дані для навчання. Але буває недостатньо помічених даних і для багатьох завдань до відповідності параметрам складної моделі важко отримати достатню кількість прикладів. Наприклад, до перенавчання може привести високий ступінь виразності глибинних мереж.

До локального оптимума зближення параметрів з відповідними значеннями приводить навчання мало шарових мереж (з 1 прихованим шаром) із застосуванням контрольованого навчання. При навчанні глибокої мережі, це рідко спрацьовує. Також, навчання нейронної мережі із застосуванням навчання з учителем містить вирішення питання з неопуклою оптимізацією. Навчання з градієнтним спуском перестає працювати, тому що у глибокій мережі з'являється велика кількість локальних оптимумів.

Використовуючи метод зворотного поширення, можна визначити похибку для обчислення похідних, градієнти, які виходять від вихідного шару до більш ранніх шарів мережі, і в результаті вони швидко зменшуються в міру збільшення глибини мережі. Отже співвідношення між загальною вартістю та вагою в більш ранніх шарах стає незначною. При застосуванні градієнтного спуску ваги ранніх

шарів поволі змінюються і більш ранні шари не взмозі багато чому навчитися. Це називають "дифузією градієнтів" (diffusion of gradients).

1.3 Проблеми навчання глибоких мереж

Складність, яка виникає при навчанні штучних нейронних мереж із застосуванням методології навчання зворотного поширення і на основі градієнта, помилкою називається проблемою зникаючого градієнта. Нейронна мережа оновлюється пропорційно градієнту функції помилки щодо поточного ваги на кожній ітерації навчання. Метод зворотнього поширення помилки обчислює стандартні функції активації, такі як гіперболічний тангенс, які мають градієнти в діапазоні (-1, 1), по їх ланцюговому правилу. При множенні даних чисел для розрахунку градієнтів "фронтальних" шарів в n-шарової мережі, градієнт (сигнал помилки) експоненціально стає меньшим разом з n. Передні ж шари навчаються дуже повільно. При застосуванні функції активації є ймовірність виникнення проблеми exploding gradient problem, коли їхні похідні набувають великих значень.

Багаторівнева ієрархія: шар мережі попередньо навчається, використовуючи методи навчання без вчителя. При цьому рівень його навчання регулюється методом зворотного поширення помилки. Тому на наступний шар мережі передається стисле уявлення спостережень з попереднього шару.

Різновидом архітектури рекурентних нейронних мереж є довга короткострокова пам'ять. Помилка не випускається з пам'яті LSTM-блоку, якщо величини помилки рухаються у зворотному напрямку від вихідного шару. Доки кожен вентиль не буде навчений відкидати подібні значення вона безперервно буде передаватися назад до них.

Одним із найефективнішим методом вирішення питання зникаючого градієнта є застосування остаточних нейронних мереж (ResNets) є залишкові мережі (Residual networks). Малошарова мережа буде мати нижчу помилку навчання, ніж глибока мережа. Підрозділ компанії Microsoft Research виявив, що

розподіл глибокої мережі на частини допоміг усунути більшу частину проблеми зі зникненням градієнта і передачі вхідних даних від фрагмента до фрагмента. Ніяких змін або додаткових параметрів в алгоритмі навчання незнадобилось. ResNets продемонстрували нижчу помилку навчання, ніж їх аналоги з меньшою кількістю шарів.

1.4 Переваги систем виявлення вторгнень на основі нейромереж

Перша перевага у виявленні вторгнень нейронної мережі - це гнучкість, яку надає ця мережа. Нейронна мережа може аналізувати дані з мережі, навіть якщо вони неповні або перекручені. Більше того, мережа буде здатна проводити аналіз з даними в нелінійній формі. Ці характеристики мають важливе значення в мережевому середовищі, адже отримана інформація схильна до випадкових помилок системи. Особливо важлива здатність мережаі обробляти дані з декількох джерел в нелінійній формі, оскільки деякі атаки на мережу можуть бути проведені скоординованим вторгненням кількох зловмисників.

Ще однією перевагою цього підходу є одна з властивостей нейронних мереж, а саме швидкість . Адже захист обчислювальних ресурсів вимагає своєчасного виявлення атак. І саме швидкість обробки нейронної мережі повинна забезпечити швидке реагування на вторгнення до того, як буде завдано непоправної шкоди системі.

Результат роботи нейронної мережі виражається у вигляді ймовірності. Вона забезпечує можливість прогнозування виявлення випадків вторгнення. Система виявлення вторгнень на основі нейронних мереж сама визначить ймовірність того, що конкретна подія або ряд подій вказують про напад на систему. З часом, нейронна мережа буде покращувати здатність визначати, які події і де можуть статися в процесі атаки. Цю інформацію слід використовуватидля того, щоб згенерувати послідовність подій, які можуть відбутися, якщо має місце бути спроба вторгнення. Відстежуючи наступні виникнення подій, система буде здатна їх проаналізувати і, можливо, провести

захисні заходи, перш ніж така атака буде вдало виконана.

Найбільш важливою перевагою нейронних мереж у виявленні вторгнень є саме здатність нейронної мережі "навчатися" ознакам атак і визначати випадки, які не характерні для тих, що спостерігалися раніше. Нейронна мережа може бути навчена розпізнавати відомі підозрілі події з високим ступенем точності. Це дуже цінне її вміння дозволить застосовувати ці знання для виявлення фактів про напад, які не відповідають точним характеристикам попередніх вторгнень. Імовірність вторгнення в систему може бути передбачувана і позначена як потенційна загроза, коли ймовірність перевищує певний поріг.

1.5 Недоліки систем виявлення вторгнень на основі нейромереж

Чому нейронні мережі у минулому не застосовували завдання виявлення вторгнень? Цьому є дві основні причини. Перша причина пов'язана з вимогами до навчання нейронної мережі. Адже здатність штучної нейронної мережі до ідентифікації ознак вторгнення повністю залежить від правильного навчання системи. Дані для навчання і методи навчання, які використовуються при цьому, є критичними. Процедура ж навчання вимагає дуже великий обсяг даних, щоб гарантувати статистично точні результати. Навчання нейронної мережі з метою виявлення вторгнень може зажадати величезну кількість послідовностей індивідуальних атак, яку важко отримати.

Істотним недоліком застосування нейронних мереж для виявлення вторгнень є природа "чорної скриньки" нейронної мережі. Нейронні мережі адаптують свій аналіз даних у відповідь на отримане навчання,що відрізняє їх від експертних систем, які мають жорстко закодовані правила для аналізу подій. Як правило, вага зв'язку та передавальні функції різних мережевих вузлів, заморожуються після того, як мережа досягла прийнятного рівня успіху в ідентифікації подій. У той час як аналіз мережі досягає достатньої ймовірності успіху, не завжди відома основа для цього рівня точності. "Проблема чорної скриньки" переслідує нейронні мережі в ряді програм, що і сьогодні.

досліджується в нейронних мережах[18].

1.6 Постановка задачі

Аналізуючи зарубіжну і вітчизняну спеціалцізовану літературу з даного питання, наукові статті які розміщені на ресурах мережі інтернет, наведено поняття нейронної мережі, їх моделі з метою їх застосування до виявлення вторгнень, проаналізували технології нейромереж. На підставі досліджень відомих праць зарубіжних науковців виконали системний виклад проблем навчання глибоких мереж.

Сучасний підхід до побудови систем виявлення мережевих вторгнень і виявлення ознак комп'ютерних атак на інформаційні системи, має величезну кількість недоліків і вразливостей, як це ми побачили вивчаючи це питання. Це дозволяє зловмисникам мати успіх у подоланні системи захисту інформації. Шлях від пошуку сигнатур атак до викриття передумов виникнення загроз інформаційній безпеці, веде до того, щоб скоротити відставання в розвитку систем захисту від систем їх подолання. Розглянуті в даній роботі дослідники вивчали застосування різних методів штучного інтелекту в системах виявлення вторгнень. Однак мало уваги приділялося дослідженню складних технічних атак, які у сучасних умовах є основним джерелом загроз у великих підприємствах. Подібні висновки створюють платформу для подальшого вивчення і проектування альтернативних методів аналізу та ідентифікацій мережевих аномалій і пошуку мережевих вторгнень.

Сформулюємо задачу дослідження:

1. Вдосконалити метод ідентифікації вторгнень за рахунок впровадження в його структуру нейронних мереж.
2. Вдосконалити системи ідентифікації вторгнень за рахунок введення в її архітектуру підсистем прогнозування.
3. Розробити алгоритм, що дозволяє виконувати глибоке навчання в нейронній мережі для виявлення вторгнень.

4. Розробити модель глибинної нейромережі для виявлення вторгнень.
5. Впровадити та протестувати розроблений алгоритм.

2 МОДЕЛІ НЕЙРОМЕРЕЖІ З МЕТОЮ ЇХ ЗАСТОСУВАННЯ ДО ВИЯВЛЕННЯ ВТОРГНЕНЬ

2.1 Моделі нейромереж

Основи нейроматематики були закладені в 1943 році У. Маккалохом і його учнем У. Піттс. Вони розробили основні положення теорії діяльності головного мозку [54], і отримали такі результати:

- розроблена модель нейрона - найпростішого процесорного елемента, який виконує обчислення перехідної функції скалярного добутку вектора вхідних сигналів і вагових коефіцієнтів;
- запропоновали конструкцію мережі елементів для арифметичних та логічних операцій;
- припустили, що нейромережа може навчатися, узагальнювати отриману інформацію, розпізнавати образи.

Нейроматематика швидко розвивається, але багато тверджень Макклоха є актуальними і сьогодні. При великій різноманітності моделей нейронів, принцип дії закладений Макклохом і Піттсом, не змінюється.

Дефектом цієї моделі є сама модель нейрона - "пороговий" вид перехідної функції. У формалізмі У. Маккалоха і У. Піттса нейрони мають стан 0,1 і порогову логіку переходу зі стану в стан. Кожний нейрон в мережі вираховує зважену суму станів всіх інших нейронів і порівнює її з порогом, для визначення свого власного стану. Нейронна мережа не достатню гнучка граничним видом функції при навчанні та налаштуванні на задану задачу. Якщо значення обчисленого скалярного виробу, не доходить до даного порогу, то вихідний сигнал взагалі не формується і нейрон не виконує свої функції. Даний нейрон втрачає інтенсивність вихідного сигналу (аксона).

Серйозного розвитку нейрокібернетики було досягнуто в роботах американського нейрофізіолога Френсіса Розенблата (Корнельський

університет). Ще у 1958 році він запропонував свою модель нейронної мережі, ввівши в модель Маккаллока і Піттса здатність зв'язків до модифікації, що зробило її здатною до навчання. Ця модель була названа персептроном [55, 56, 57, 58]. Спочатку персептрон представляв собою одношарову структуру з жорсткою пороговою функцією процесорного елемента та бінарними і багатозначними входами. Перші персептрони мали здатність розпізнати деякі букви латинського алфавіту. Згодом модель персептрона була значно вдосконалена [58].

Персептрон застосовувався для вирішення проблеми автоматичної класифікації, що полягає в розділенні простору ознак між заданою кількістю класів. У двовимірному просторі слід провести лінію на площині, яка відділяє одну область від іншої. Персептрон зпроможний ділити простір тільки прямими лініями (площинами) [55,59].

Алгоритм навчання персептрона виглядає наступним чином:

- системі пред'являється еталонний образ;
- при правильному спрацюванні виходів системи, вагові коефіцієнти зв'язків не змінюються;
- при неправильному спрацюванні виходів, ваговим коефіцієнтам дається невелике збільшення в сторону підвищення якості розпізнавання.

Вада персептрона полягає в тому , що не завжди є така комбінація вагових коефіцієнтів, за якої наявна безліч образів буде спроможна розпізнаватися даним персептроном. Причиною цієї вади є те, що лише мала кількість завдань передбачає, що лінія, розділяюча еталони, буде прямою. Часто це досить складна крива, замкнута або роззімкнута. Якщо врахувати, що одношаровий персептрон реалізує тільки лінійну розділюючу поверхню, використання його там, де необхідна нелінійна, веде до невірного розпізнавання, ця проблема зветься лінійною нероздільністю простору ознак. Щоб вийти з цього положення використовують багатошаровий персептрон, який здатний будувати ламаний кордон між розпізнаваними образами. Описане дане питання не є єдиною

проблемою, що виникають при роботі з персептронами, також слабо формалізований метод навчання персептрона. Персептрон створює ряд питань, над вирішенням яких робота призводить до створення більш "розумних" нейронних мереж і розробки методів, що знайшли застосування не тільки в нейрокібернетиці.

У 70-ті роки минулого століття інтерес до нейронних мереж дуже знизився, проте роботи по їх дослідженню тривали. Було запропоновано ряд розробок, таких, як когнітрон, він здатний добре розпізнавати досить складні образи (ієрогліфи і т.д.) незалежно від повороту і змінювання масштабу зображення. Винахідником когнітрону є японський вчений І. Фукушіма.

Швидкий розвиток моделей нейронних мереж пов'язаний з роботами Амарі, Андерсона, Карпентера, Кохена, [60, 61, 62] та інших, і особливо, Хопфілда [63, 64, 65, 66, 67], а також під впливом обіцяючих успіхів оптичних технологій [68, 69] і зрілої фази розвитку НВІС [62] для побудови нових архітектур.

Початок сучасного математичного моделювання нейронних обчислень розпочато роботами Хопфілда в 1982 році, в яких була сформульована математична модель асоціативної пам'яті на нейронній мережі з використанням правила Хеббіана [70] для програмування мережі. Але не стільки сама модель спонукала до появи робіт інших авторів на цю тему, скільки введена Хопфілдом функція обчислювальної енергії нейронної мережі. Це аналог функції Ляпунова в динамічних системах. Для одношарової нейронної мережі зі зв'язками типу "все на всіх", є характерна збіжність до однієї з кінцевої безлічі рівноважних точок, які є локальними мінімумами функції енергії, що має в собі всю структуру взаємозв'язків в мережі. Розуміння такої динаміки нейронної мережі було і у інших дослідників. Але, Хопфілд і Тенк [63] показали, як конструювати функцію енергії для конкретної оптимізаційної задачі і як застосовувати її для відображення завдання в нейромережі. Цей підхід використовували і для вирішення інших комбінованих оптимізаційних задач. Підход Хопфілда полягає в тому, що нейронна мережа для даного завдання може бути запрограмована без

навчальних ітерацій. Ваги зв'язків обчислюються на підставі виду функції енергії, сконструйованих для цього питання. Нейронна мережа для конкретного завдання може бути запрограмованою без навчальних ітерацій.

Розвитком моделі Хопфілда для вирішення комбінаторних оптимізаційних задач, задач штучного інтелекту є машина Больцмана, запропонована і досліджена Джефері Е. Хінтоном і Р. Земелом [71,72, 73, 74]. У ній, як і в інших моделях, нейрон має стан 1,0 і зв'язок між нейронами має вагу. Кожен стан мережі може характеризуватися певним значенням функції консенсусу (аналог функції енергії). Максимум функції консенсусу відповідає оптимальному вирішенню завдання.

Моделювалася асинхронна робота мережі Хопфілда. Мережа працює без помилок, якщо відновлює еталонні образи з випадкових, коли в неї записується не більше 15% еталонних образів. Випробування проводилися для 30 нейронів і для 100 нейронів у мережі. Бралося декілька випадкових векторів в якості еталонних і створювалась відповідна матриця ваг зв'язків. Моделювання при 100 нейронах було повільнішим процесом, ніж при 30 нейронах, хоча результат і в тому і в іншому випадках був однаковий. Приблизно 88% випробувань закінчувалися в еталонних станах, 10% - в стійких станах, близьких до еталонних. При відстані <= 5 між початковим і еталонним векторами, еталонний стан досягався в 90% випадків. Коли збільшували відстань, ймовірність попадання в найбільш близький еталонний стан падала. При відстані 12 ймовірність дорівнювала 0,2. Стійкі стану, занадто близькі один до одного, та мають тенденцію "зливатися", вони потрапляють в одну западину на енергетичній поверхні. Програмувались завдання комівояжера на основі мережі Хопфілда. Мережею з 100 нейронів для 20 різних випадкових початкових станів були встановлені маршрути, 16 з яких були прийнятними, половина спроб дали 2 шляхи: 2.83 і 2.71 (цифри наводяться, щоб показати як вони схожі) при найкоротшому 2.67. Це є результати моделювання роботи мережі з безперервною моделлю нейрона. Моделювалася також завдання комівояжера, але для мережі типу машини Больцмана, проводилася при таких значеннях

керуючих параметрів: A = 0. 95, L = 10, M = 100 (A - позитивне число, яке менше одиниці, але близьке до неї, L - число випробувань, які проводяться без змін, M - число послідовних випробувань, що не веде до зміни стану машини, як критерію закінчення процесу). Помітимо, що імовірнісний механізм функціонування машини Больцмана надає можливість отримати на ній кращі результати оптимізації, ніж на моделі Хопфілда. Процес запускався 100 разів для n = 10 (разом в мережі N = n ^ 2 нейронів) і 25 раз для n = 30 при різних нормальних станах машини Больцмана. Для n = 10 отримано оптимальний результат, для n = 30 - рішення на 14% гірше оптимального.

Методом зворотного поширення (back propogation) називається спосіб навчання багатошарових нейромереж. У таких нейромережах зв'язок між собою мають тільки сусідні шари, при цьому кожен нейрон попереднього шару пов'язаний з усіма нейронами наступного [75, 76, 77, 78, 79, 80, 81, 82]. Зазвичай нейрони мають сигмоїдальну функцію збудження. Перший шар нейронів називають вхідним, який містить кількісь нейронів відповідно до розпізнавання образів. Останній шар нейронів називається вихідним і містить кількість нейронів, який дорівнює кількості класів образів що розпізнаються. Між вхідним і вихідними шарами розташовується один або декілька прихованих (тіньових) шарів. Визначення числа прихованих шарів і числа нейронів в кожному шарі для конкретного завдання є неформальним завданням.

Принцип навчання такої нейронної мережі заснований на розрахунках відхилень значень сигналів на вихідних процесорних елементах від еталонних і зворотному "прогоні" цих відхилень з метою корекції помилки. В 1974 році Поль Дж. Вербос [75] винайшов значно більш ефективну процедуру для обчислення величини, так званої похідної помилки по вазі, коли працював над докторською дисертацією в Гарвардському університеті. Процедура, відома тепер як метод зворотного поширення, стала найважливішим інструментом у навчанні нейронних мереж [75, 77, 78, 79, 80, 81, 82]. Однак цей алгоритм має й недоліки, головний з яких це відсутність скільки-небудь прийнятних оцінок часу навчання. Алгоритм зворотного поширення має найширше застосування.

Наприклад, успіх фірми NEC у розпізнаванні букв був досягнутий саме завдяки алгоритму зворотного поширення.

2.2 Моделі глибинного навчання

Моделі глибокого навчання складаються з різноманітних глибинних мереж. Серед них глибокі короткі мережі (DBN), глибокі нейронні мережі (DNN), згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN) є контрольованими моделями навчання. Моделі навчання без нагляду це автокодери, обмежені машини Больцмана (RBM) та генеративні змагальні мережі (GAN). З 2015 року кількість досліджень IDS, що базуються на глибокому навчанні почала швидко зростати, що відбувається і сьогодні. Моделі глибокого навчання безпосередньо вивчають подання функцій із вихідних даних, таких як зображення та тексти, без необхідності ручної інженерії особливостей. Таким чином, методи глибокого навчання можуть виконуватися наскрізними способами. Для великих наборів даних методи глибокого навчання мають величезну перевагу перед неглибокими моделями. При вивченні глибокого навчання основними акцентами є мережева архітектура, вибір гіперпараметрів та стратегія оптимізації. Порівняння різних моделей глибокого навчання показано в таблиці 2.2.

Таблиця 2.2 - Порівняння різних моделей глибокого навчання

Алгоритми	Відповідні типи даних	Під наглядом або без нагляду	Функції
Автокодер	Необроблені дані; Вектори характеристик	Без нагляду	Видобуток особливостей; Особливість зменшення; Шумить
RBM	Вектори характеристик	Без нагляду	Видобуток особливостей; Особливість зменшення; Шумить
DBN	Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
DNN	Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
CNN	Необроблені дані; Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
RNN	Необроблені дані; Вектори характеристик;	Під наглядом	Видобуток особливостей, класифікація
GAN	Необроблені дані; Вектори характеристик	Без нагляду	Збільшення даних; Змагальний тренінг

Автокодер містить два симетричні компоненти- кодер та декодер. Це показано на рисунку 2.1. Кодер витягує функції з необроблених даних, а декодер відновлює дані з вилучених об’єктів. Під час навчання розбіжність між входом кодера та виходом декодера поступово зменшується. Коли декодеру вдається реконструювати дані за допомогою вилучених функцій, це означає, що функції, витягнуті кодером, представляють суть даних. Весь цей процес не вимагає контрольованої інформації,що дуже важливо. Є багато відомих варіантів автоенкодерів, таких як шумозаглушувачі [30, 31] та розріджені автокодери [32].

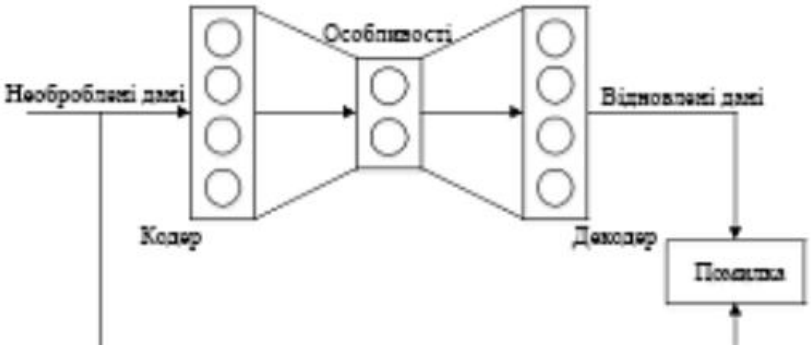


Рис 2.1 - Структура автокод ера

Обмежена машина Больцмана (RBM) - це рандомізована нейронна мережа, в якій одиниці підкоряються розподілу Больцмана. RBM складається з видимого та прихованого шару. Одиниці в одному шарі не з'єднані; однак блоки в різних шарах повністю з'єднані, як показано на рисунку 2.2 де v_i є видимим шаром, і h_iє прихованим шаром. RBM не розрізняють прямий і зворотний напрямки. Отже,таким чином, ваги в обох напрямках однакові.

RBM - це моделі без нагляду, що навчаються за алгоритмом контрастивної дивергенції [33], і їх зазвичай застосовують для вилучення ознак або зменшення шуму.

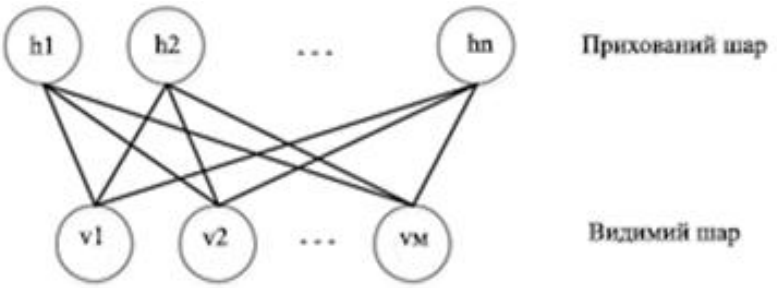


Рисунок 2.2 - Структура RBM

Deep Brief Network (DBN) - складається з декількох шарів RBM та рівня класифікації softmax, як показано на рисунку 2.3. Навчання DBN включає два етапи: попередню підготовку без нагляду та точну настройку під контролем [34,35]. Кожен RBM тренується з використанням жадібної попередньої підготовки. Потім, вага шару softmax визначається за маркованими даними. При виявленні атак DBN використовуються і для вилучення ознак, і для класифікації [36, 37, 38].

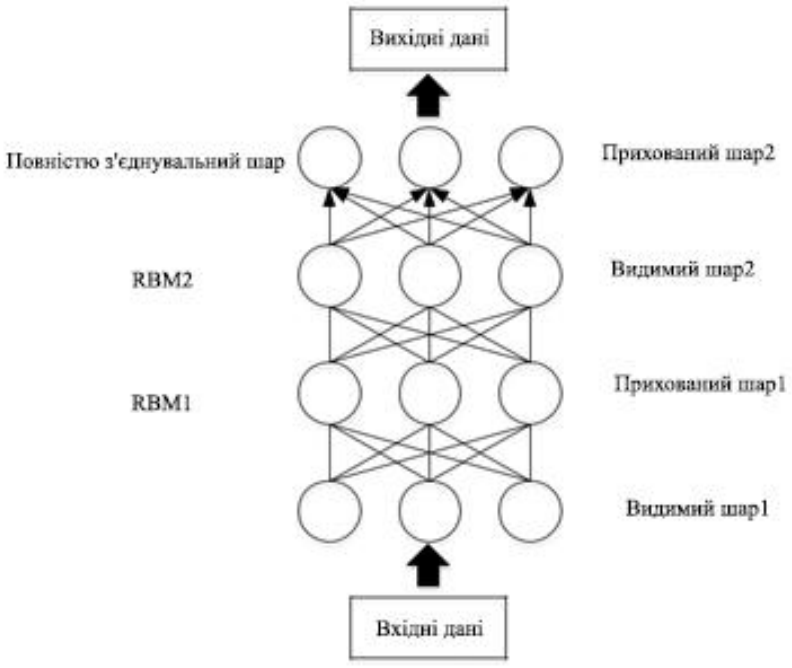


Рисунок 2.3 - Структура DBN

Глибока нейронна мережа (DNN) - попередня підготовка та точна настройка пошарово дозволяє побудувати DNN з декількома шарами, як показано на рисунку 2.4. Під час навчання DNN параметри вивчаються спочатку з використанням немічених даних. Це є етапом вивчення особливостей без нагляду. Мережа налаштовується на марковані дані. Це є етапом навчання під контролем. Дивовижні досягнення DNN в основному зумовлені стадією вивчення особливостей без нагляду.

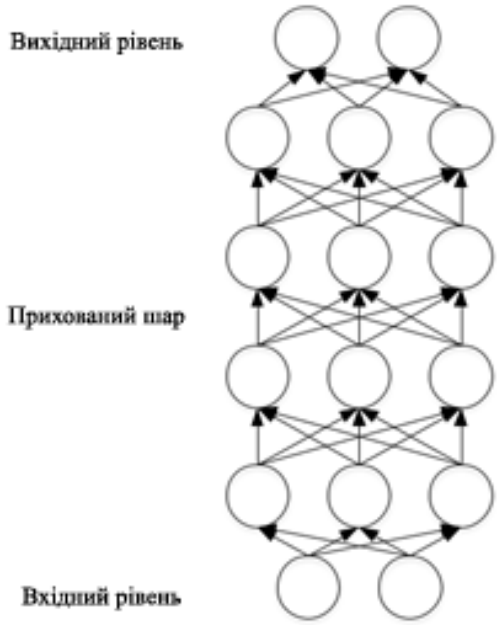


Рисунок 2.4 - Структура DNN

Згорткова нейронна мережа (CNN) - призначені для імітації зорової системи людини (HVS). І вона досягли великих досягнень у галузі комп'ютерного зору [39, 40, 41] CNN складається з альтернативних згорткових та об'єднуючих шарів, як показано на рисунку 2.5.

Згорнуті шари використовуються для вилучення об’єктів, а шари об’єднання використовуються для посилення узагальнення ознак. CNN працюють над двовимірними (2D) даними, тому вхідні дані повинні бути перетворені в матриці для виявлення атак.

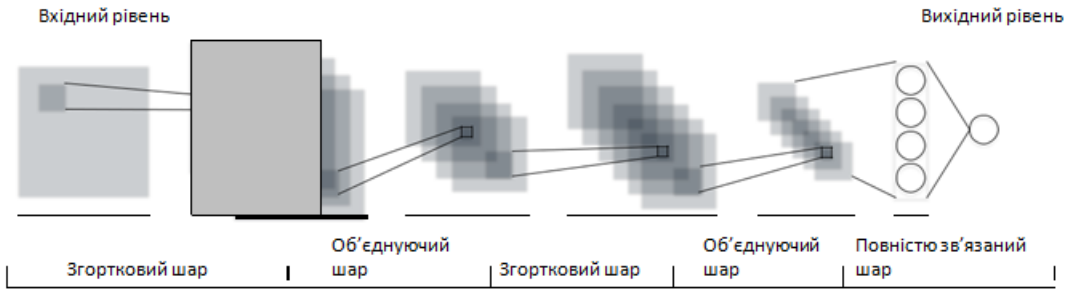


Рисунок 2.5 - Структура CNN

Нейронна мережа (RNN) - послідовні дані, які використовуються в обробці природними мовами (NLP) [42, 43, 44] Характеристики послідовних даних є контекстуальними. Аналіз ізольованих даних із послідовності не має сенсу. Кожна одиниця в RNN отримує не лише поточний стан, а й попередні. Всі елементи W є однакові. Ця характеристика змушує РНН часто страждати від зникаючих або вибухаючих градієнтів. Насправді стандартні RNN мають справу лише з послідовностями обмеженої довжини. Для вирішення проблеми довгострокової залежності було запропоновано багато варіантів RNN, таких як довга короткочасна пам'ять (LSTM) [45], закрита рекурентна одиниця (GRU) [46], і бі-RNN[47].

У 1997 р. модель LSTM була запропонована Хохрейтером та Шмідхубером [45]. Кожен блок LSTM містить три ворота: ворота забуття, вхідні ворота та вихідні ворота. Ворота забуття усуває застарілу пам’ять, вхідний шлюз отримує нові дані, а вихідний шлюз поєднує короткочасну пам’ять з довгостроковою пам’яттю для формування поточного стану пам’яті. У 2014 р. GRU було запропоновано Чунгом та ін. [46]. Модель GRU об'єднує ворота забуття та вхідні ворота в єдиний шлюз оновлення, що простіше, ніж LSTM.

Генеративна змагальна мережа (GAN). Модель GAN включає дві підмережі, а саме: генератор і дискримінатор. Генератор генерує синтетичні дані, подібні до реальних даних, а дискримінатор відрізняє синтетичні дані від реальних даних. Отже, генератор і дискримінатор вдосконалюють один одного. В даний час GAN є актуальною темою досліджень, яка використовується для

збільшення даних при виявленні атак, що частково полегшує проблему нестачі набору даних IDS. GAN належать до змагальних підходів, які можуть підвищити точність виявлення моделей, додаючи змагальні зразки до навчального набору.

Глибоке навчання - це галузь машинного навчання. Ефекти моделей глибокого навчання, перевершують ефекти традиційних методів машинного навчання у більшості сценаріїв застосування. Відмінності між поверхневими моделями та глибокими моделями в основному відображаються в наступних аспектах:

- Тривалість роботи. Час бігу включає як час тренувань, так і час випробувань. Через високу складність глибоких моделей і час їх навчання, час випробувань набагато довший, ніж у неглибоких моделей.
- Кількість параметрів. Є два типи параметрів: параметри, які можна дізнатись, та гіперпараметри. Параметри, що підлягають вивченню, обчислюються на етапі навчання, а гіперпараметри встановлюються вручну перед початком навчання. Параметри та гіперпараметри, що вивчаються, у глибоких моделях значно перевершують такі, як у поверхневих моделях. Можна зробити висновок, що навчання та оптимізація глибоких моделей займає більше часу.
- Представлення функції. Вхідні дані до традиційних моделей машинного навчання - це вектор функцій, а конструювання особливостей - важливий крок. Моделі ж глибокого навчання здатні вивчати подання об’єктів із необроблених даних і не залежать від інженерних можливостей. Методи глибокого навчання можуть виконуватися наскрізним способом, надаючи їм видатну перевагу перед традиційними методами машинного навчання.
- Навчальна здатність. Структури моделей глибокого навчання складні і містять величезну кількість параметрів. Отже, моделі глибокого навчання мають сильніші здібності до пристосування, ніж моделі з поверхневим вивченням. Однак моделі глибокого навчання також

стикаються з вищим ризиком переобладнання, вимагають набагато більшого обсягу даних для навчання. Отже і ефект від моделей глибокого навчання кращий.

- Інтерпретабельність. Моделі глибокого навчання - це чорні скриньки [48,49,50,51], результати яких майже не піддаються інтерпретації. Це є критичним моментом у глибокому навчанні. Однак деякі традиційні алгоритми глибокого навчання, такі як дерево рішень та наївний Байєс, мають сильну інтерпретацію.

Розробка функцій залежить від знань в області, якість ж функцій іноді є вузьким місцем ефектів виявлення. Методи виявлення на основі глибокого навчання вивчають функцію автоматично. Ці типи методів працюють наскрізно. Вони поступово стають основним підходом у дослідженнях IDS.

Методи глибокого навчання можуть обробляти необроблені дані, що дозволяє їм одночасно вивчати особливості та виконувати класифікацію. Потлурі та ін. [52] запропонував метод виявлення на основі CNN. Вони провели експерименти на наборах даних NSL-KDD та UNSW-NB 15. Тип даних у цих наборах даних є вектором ознак. Оскільки CNN добре обробляють двовимірні (2D) дані, вони перетворили вектори об’єктів на зображення. Номінальні функції були кодовані одним гарячим способом, а розміри ознак зросли з 41 до 464. Потім кожен 8-байтовий фрагмент був перетворений в один піксель. Пусті пікселі заповнювались 0. Кінцевим результатом було те, що вектори об’єктів перетворювались на зображення розміром 8 * 8 пікселів. Це дало змогу побудувати тришаровий CNN для класифікації атак. Вони порівняли свою модель з іншими глибокими мережами (ResNet 50 та GoogLeNet), і запропонована CNN показала найкращі результати, досягнувши точності 91,14% для NSL-KDD та 94,9% для UNSW-NB 15.

Моделі глибокого навчання без нагляду також можуть бути використані для вилучення особливостей. Для класифікації можна використовувати неглибокі моделі. Чжан та ін. [53] витягували функції за допомогою розрідженого автокодера та виявляли атаки за допомогою моделі XGBoost. Вони

використовували дані з набору даних NSL-KDD. Через незбалансований характер цього набору даних, вони взяли вибірку набору даних за допомогою SMOTE. Алгоритм SMOTE перевизначає класи меншості та розділяє класи більшості на багато підкласів, щоб кожен клас був збалансованим. Розріджений автокодер вводить обмеження розрідженості в оригінальний автокодер, підвищення його здатності виявляти невідомі зразки. Вони класифікували дані за допомогою моделі XGBoost. Їхня модель досягла точності для класів Normal, DOS, Probe, R2L та U2R - 99,96%, 99,17%, 99,50%, 97,13% та 89,00% відповідно.

Моделі глибокого навчання досягли значних успіхів в аналізі великих даних. Але їх результати не є ідеальними для малих або незбалансованих наборів даних. Недоліком є суперечливі підходи до навчання, що може покращити точність виявлення на малих наборах даних. Чжан та ін. [54] провели збільшення даних за допомогою GAN. Набір даних KDD99 є незбалансованим і не має нових даних, що призводить до поганої узагальненості моделей машинного навчання. Для вирішення цих проблем вони використали GAN для розширення набору даних. Модель GAN генерувала дані, подібні до даних потоку KDD99. Додавання цих сформованих даних до навчального набору дозволяє виявити варіанти атак. Вони обрали 8 типів атак та порівняли точність, досягнуту на вихідному наборі даних, порівняно з розширеним набором даних. Експериментальні результати показали, що змагальне навчання покращило 7 точностей у 8 типах атак.

2.3 Алгоритми глибинного навчання

Глибока нейронна мережа для реального завдання може мати більше десяти прихованих шарів. Її топологія може бути простою або досить складною. Чим більше шарів в мережі, тим більше характеристик вона може розпізнати, але тим більше часу буде потрібно для розрахунку, і тим самим складніше буде навчання.

Як говорилося раніше, «найбільш глибинне» навчання здійснюється за

допомогою глибоких нейронних мереж. Згорткові нейронні мережі (CNN) часто використовуються для комп'ютерного зору. Рекурентні нейронні мережі (RNN) часто використовуються в задачах природної мови і для обробки інших послідовностей, як і мережі з довгої короткостроковою пам'яттю (Long short-term memory; LSTM) і нейронні мережі з механізмом уваги[83]. Випадкові лісу (вони ж - ліси випадкових рішень), нейронними мережами не є, корисні для цілого ряду завдань класифікації і регресії.

Згорткові нейронні мережі в основному використовують згорткові шари, шари об'єднання, ReLU шари, як повністю пов'язані і втрачені шари для імітації зорової кори. Згортковий шар в основному вважається інтегралами багатьох невеликих перекриваючих областей. Шар об'єднання виконує форму нелінійної понижувальної дискретизації.

Шари ReLU застосовують функцію активації f (x) = max (0, x). У повністю пов'язаному шарі нейрони зв'язані з усіма активаціями в попередньому шарі. Рівень втрат обчислюється, як мережеве навчання виправляє відхилення між передбаченими і істинними мітками, використовуючи функцію Softmax або функцію втрат перехресної ентропії для класифікації або функцію втрат для регресії.

У нейронних мережах прямий зв'язок інформації тече від входу через приховані шари і до виходу. Це обмежує мережу в зверненні до кожного окремого стану за раз. У рекурентних нейронних мережах інформація проходить через цикл, що дозволяє мережі запам'ятовувати минулі виходи. Це дає можливість проводити аналіз послідовностей і часових рядів.

Мережа з довгою короткостроковою пам'яттю здатна забувати попередню інформацію або запам'ятовувати її. LSTM може працювати з послідовностями з сотень минулих входів.

Модулі уваги - це узагальнені елементи, які застосовують ваги до вектору входів. Ієрархічним нейронним кодувальником уваги використовується декілька рівнів модулів уваги для роботи з десятками тисяч минулих входів.

Випадковим лісом вважається вид алгоритму глибинного навчання, який

не є глибокою нейронною мережею. Випадковий ліс будується з багатьох шарів, але замість нейронів він будується з дерев рішень і виводить статистичне середнє (режим для класифікації або середнє для регресії) передбачення окремих дерев. Рандомізовані аспекти випадкових лісів це результат використання бутстрап-агрегування для окремих дерев і випадкових підмножин ознак.

Писати програми глибинного навчання можна і з нуля, але набагато ефективніше використовувати фреймворки глибинного навчання, особливо з огляду на те, що вони оптимізовані для використання з графічними процесорами і іншими прискорювачами. Вірогідно, кращим фреймворком є Google TensorFlow. Переважно високорівнева API для TensorFlow - Keras; його також можна використовувати з іншими серверними фреймворками.

PyTorch, створений в Facebook за участю ряду інших організацій, що є хорошою альтернативою TensorFlow, виділяється підтримкою динамічних нейронних мереж, в яких топологія мережі може змінюватися від епохи до епохи. Fastai - це високорівневий сторонній API, який використовує PyTorch як серверні додатки.

Amazon MXNet є ще однією доброю альтернативою TensorFlow, з претензією на кращу маштабованість. Gluon є кращим високорівневим імперативним API для MXNet.

Chainer розробки IBM і Intel послужив в деякому роді джерелом натхнення для PyTorch, враховуючи, що він визначає нейронну мережу шляхом запуску і підтримує динамічні нейронні мережі.

Всі згадані фреймворки в основному - для Python, а Deeplearning4j (спочатку створений Sky Mind, а тепер є проектом Apache) в першу чергу - для Java і Scala. DL4J сумісний з Apache Spark і Hadoop. ONNX спочатку був запропонований як відкритої екосистеми для взаємозамінних моделей штучного інтелекту, а зараз в добавок має і виконуючу середу.

Nvidia TensorRT - ще одне середовище виконання для моделей штучного інтелекту, зокрема, для використання графічних процесорів Nvidia. ONNX може використовувати TensorRT в якості плагіна.

2.4 Розробка моделі глибинної нейромережі для виявлення вторгнень

На сьогоднішній день мережеві системи виявлення вторгнень є найважливішим інструментом захисту даних численних комп'ютерних систем та мереж. Такі системи використовуються протягом декількох десятиліть, вони розробляються кваліфікованими фахівцями, численні роботи присвячені розробці науково-методичної бази, проте практичний досвід демонструє [85], що системи виявлення вторгнень характеризуються певними недоліками. Основними з них є недостатня точність виявлення загального діапазону мережевих вторгнень. Про цей недолік свідчать як практичні результати [86], так і відомі випадки успішних вторгнень на комп’ютерні системи та мережі в США та країнах ЄС.

Загальновизнано, що недостатня точність виявлення системи виявлення вторгнень головним чином пов'язана з недосконалістю програмного забезпечення таких систем. Одним з головних підходів до підвищення точності системи виявлення вторгнень є застосування програмного забезпечення на базі штучних нейронних мереж [85]. Такі системи виявлення вторгнень довели свою ефективність, наприклад, у жорстких та програмних комплексах захисту даних фірми Cisco Systems, Inc. Слід зазначити, що апробовані системи в основному застосовують моделі нейронних мереж на основі двошарового персептрона, карт Кохонена та асоціативних нейронних мереж. Поряд з цим розвиток теорії штучних нейронних мереж пов'язаний головним чином з так званими глибокими нейронними мережами. В даний час глибокі нейронні мережі підтвердили свою перевагу в порівнянні з класичними моделями нейронних мереж у складних випадках виявлення, управління якими вимагає великих обчислювальних ресурсів. Це аналіз мови, розпізнавання почерку, аналіз зображення. Прикладом цього може служити система розпізнавання голосу в браузері Google. У той же час глибокі нейронні мережі в системах виявлення вторгнень в даний час широко не застосовуються. Це можна досягти шляхом розробки відповідної моделі нейронної мережі та апробації при виявленні мережевих кібератак.

Згідно з [85, 86], в сучасних системах виявлення вторгнень нейронні мережі використовуються як для виявлення відхилення контрольованих параметрів комп'ютерної безпеки від нормального стану, так і для виявлення відповідності розглянутих параметрів сигнатурам вторгнень.

Слід зазначити, що виявлення мережевих вторгнень шляхом аналізу відхилення параметрів безпеки від нормального стану залежить від властивостей захищеної комп'ютерної системи, що означає, що модель нейронної мережі, пристосована для виявлення мережевих вторгнень на зазначену комп'ютерну систему, повинна бути змінена для виявлення вторгнень на іншій комп'ютерній системі. У той же час параметри моделі нейронної мережі, пристосованої до виявлення підписів, насправді не залежать від складу та структури захищеної комп'ютерної системи. Відповідно, отримані результати матимуть універсальний характер і дозволять оцінити потенціал застосування глибоких нейронних мереж для захисту різних комп’ютерних систем. Наслідком є те, що увага зосереджується на розробці моделі глибокої нейронної мережі, призначеної для виявлення підписів мережевих вторгнень.

Як правило, глибока нейронна мережа - це штучна нейронна мережа з більш ніж двома прихованими шарами [87,88,89]. Подібно до звичайних нейронних мереж, глибокі нейронні мережі здатні імітувати складні нелінійні зв'язки між елементами. Під час навчання глибокої нейронної мережі, отримана модель є об’єктом у вигляді комбінації простих примітивів. Наприклад, при виявленні обличчя, такі примітиви можуть бути представлені окремими частинами обличчя: носом, очима, ротом тощо. При виявленні вторгнень такі примітиви можуть бути представлені комбінаціями різних параметрів мережевого трафіку. Додаткові шари дозволяють імітувати абстракції вищих рівнів, тим самим полегшуючи розробку моделей для виявлення складних об'єктів реального часу. Ще однією особливістю глибоких нейронних мереж є їх навчання. Глибоке навчання - це набір алгоритмів, що імітують абстракції вищого рівня в аналізованих даних на основі архітектур, що складаються з численних нелінійних перетворень.

Існує безліч архітектур глибоких нейронних мереж. Більшість з них походять з класичної архітектури. Одночасне порівняння ефективності різних архітектур не завжди можливо, оскільки не всі вони були оцінені на основі подібних наборів даних. Поглиблене навчання це швидко прогресуюча тенденція, і нові архітектури, варіанти чи алгоритми з'являються досить часто. Однак, згідно з [89, 90, 91], більшість сучасних методів навчання поділяються на два основні етапи: попередня підготовка та сама підготовка, реалізована в основному за допомогою алгоритму зворотного розповсюдження. При цьому сучасні методи варіюються шляхом здійснення попередньої підготовки, заснованої на застосуванні автокодувальника. Таким чином, для оцінки потенціалів була використана одна з базових моделей глибокої нейронної мережі на основі тришарового персептрона, який був попередньо навчений з використанням розрідженого автокодувальника.

Структура автокодувальника проілюстрована на рисунку 2.6 [89, 92]. Слід зазначити, що на рисунку 2.6 вихідні сигнали вхідних нейронів позначені «x», приховані нейрони - «a», виходи - «y», а зміщений блок - «+1».

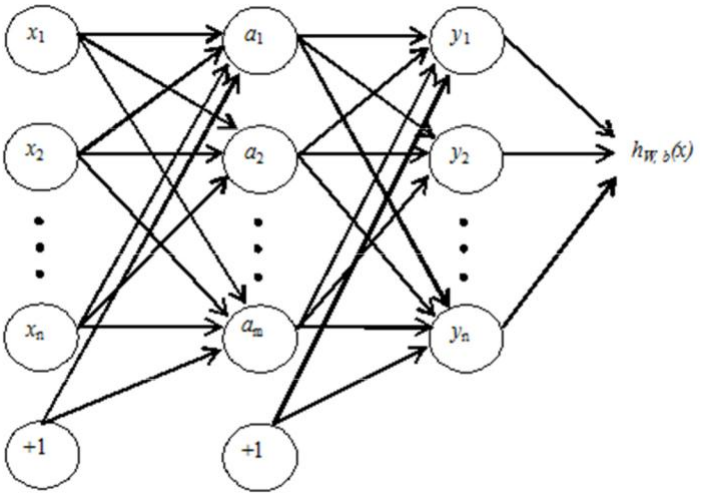


Рисунок 2.6 - Архітектура автокодувальника

Вхідні дані автоматичного кодувальника представлені неформатованим навчальним набором $x = (x_1, x_2, \dots, x_i)$. Функція активації сигмоїдів використовується в прихованих і вихідних нейронах:

$$f(z_k) = \frac{1}{1+e^{-zk}} \quad (2.1)$$

Де z_k - сукупний вхідний сигнал k-го нейрона в прихованому або вихідному шарі. У свою чергу:

$$z_k = \sum_{i=1}^{n}(W_{i,k}x_{i,k} + x_0 b_k) \quad (2.2)$$

Де $W_{i,k}$ маса зв'язку від і-го нейрона попереднього шару до k-го нейрона в прихований або вихідний шар, $x_{i,k}$ вхідний сигнал і-го нейрона попереднього шару до k-го нейрону, $x_0 = 1$ маса зв'язку нейрона з самим собою, b_k зміщення k-го нейрона (коефіцієнт ваги зв'язку k-го нейрона з самим собою).

Вихідним сигналом авто кодувальника з l нейронними шарами є:

$$h_{W,b}(x) = a^{(l)} \quad (2.3)$$

Де W - масив коефіцієнтів ваги, b - масив зсувів, $a^{(l)}$ масив вихідних значень нейронів шару l.

Як застосовано до рисунку 2.6

$$a^{(l)} = y \quad (2.4)$$

Де y - масив вхідних значень нейронів останнього (l-го шару)

Особливістю автокодувальника є навчання без викладача з використанням алгоритму зворотного розповсюдження. З цією метою цільова функція навчання автокодувальнику визначається наступним чином:

$$h_{W,b}(x) \approx x \quad (2.5)$$

Використання рівняння (2.5) передбачає рівність вихідного сигналу автокодувальника до вхідного сигналу.

Тому навчання класичному автокодувальнику зводиться до пошуку таких значень вагових коефіцієнтів за алгоритмом зворотного розповсюдження, коли вихідний сигнал дорівнює вхідному сигналу [88, 90]. При цьому навчальні приклади можуть бути без позначень, тобто не містити очікуваного вхідного сигналу. Пошук оптимального значення вагових коефіцієнтів проводиться за допомогою градієнтного спуску за рахунок мінімізації функції втрат:

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(0{,}5\left\|h_{W,b}\left(x^{(i)}\right) - y^{(i)}\right\|^2\right)\right] + 0{,}5\lambda\sum_{l=1}^{m}\sum_{i=1}^{s_{l-1}}\sum_{j=1}^{s_l}(W_{j,i}^{(l-1)})^2 \quad (2.6)$$

Де m - кількість прихованих шарів, s^l - кількість нейронів у шарі l,

$W_{j,i}^{(j-1)}$ - вага зв'язку між нейроном i у шарі l та нейроном j у шарі $(l-1)$.

Перша частина функціоналу - це усереднена квадратична похибка за всіма прикладами тренувань, друга частина - це регуляризація (або контроль за зниженням ваги), яка контролює порядок ваг та запобігає повторному тренуванню. Параметр λ, який контролює зниження ваги, регулює відносну значимість двох частин функціоналу.

Навчання виконується до:

$$J(W,b) < \theta \quad (2.7)$$

Де θ - попередньо визначений коефіцієнт (поріг).

Що стосується класичного варіанту, то особливістю розрідженого автокодувальника є обмеження кількості одночасно активних нейронів у проміжних шарах. Передбачається, що завдяки цьому розріджений автокодувальник автоматично навчається виділяти загальні особливості у вхідних даних, які відображаються у вагових коефіцієнтах. З цією метою до функції додається додатковий компонент:

$$P = \sum_{j=1}^{h}\left(p\log\frac{p}{\hat{p}j} + (1-p)\log\frac{(1-p)}{(1-\hat{p}j)}\right) \quad (2.8)$$

Де $\hat{p}j$ - середнє значення функції активації нейрона j протягом усього

тренування. Наприклад, $p \approx 0{,}05$ - параметр розрідженості. Слід зазначити, що нейрон вважається активним, якщо його вихідний сигнал близький до 1, а неактивним - до 0. З урахуванням рівняння (8) оптимізована функція втрат розрідженого автокодувальника є наступною:

$$J_s(W, b) = J(W, b) + \beta P \tag{2.9}$$

Де β - заданий коефіцієнт (у першому наближенні $\beta \approx 3$).

Глибока нейронна мережа з m нейронними шарами попередньо навчена наступним чином:

1) Вагові коефіцієнти всіх синаптичних ланок ініціюються випадковим чином.

2) На основі необхідної точності тренувань коефіцієнт θ є попередньо визначеним.

3) Кількість навченого шару встановлено заздалегідь $l = 2$ (кількість вхідного шару - 1).

4) Новий додатковий шар з'єднаний з -шаром нейронів.

5) Набір навчальних прикладів подається на вхід l-го шару.

6) Використання рівнянь. (2.1-2.9) обчислюється матриця коефіцієнтів ваги посилань l-го шару нейронів.

7) Нейронний шар, з'єднаний на стадії 2.4, видаляється.

8) Якщо $l < m$, то $l = l + 1$ і здійснюється перехід на стадію 2.5. В іншому випадку попередня підготовка припиняється.

Після етапу попередньої підготовки два останні рівні глибокої нейронної мережі тренуються на маркованих даних. Розроблена модель реалізована у вигляді відповідного програмного забезпечення, написаного мовою програмування Python. Вибір мови програмування обумовлений її апробацією в завданнях машинного навчання. Крім того, під час розробки програмного забезпечення була використана додаткова бібліотека TensorFlow (розроблена Google). Ця бібліотека дає можливість автоматизувати більшість операцій,

пов’язаних з навчанням та виявленням різних типів нейромережевих моделей. Додатковими перевагами бібліотеки є її безкоштовна основа та відкритий вихідний код. Набір навчальних матеріалів був сформований на основі бази даних NSL-KDD, яка є модифікацією відомої бази даних KDD-99. Короткий опис атрибутів NSL-KDD узагальнено в таблиці 2.4. Ці атрибути слугували вхідними параметрами для моделі нейронної мережі. Отже, кількість вхідних нейронів дорівнює 40, що відповідає кількості атрибутів.

Таблиця 2.4 - Характеристики атрибутів бази даних NSL-KDD

Номер	Атрибут	Опис
1	тривалість	Час з'єднання в секундах
2	тип_протоколу	Протокол (TCP, UDP)
3	обслуговування	Мережева служба (http, telnet тощо)
4	прапор	Стан підключення (підключення, помилка)
5	src_bytes	Обсяг даних, переданий від джерела до одержувача в байтах
6	dst_bytes	Обсяг даних, переданий від одержувача до джерела в байтах

Згадані атрибути об'єднані в чотири групи:

- Основні атрибути - параметри з'єднання TCP / IP (1-10).
- Атрибути часу руху - вони оцінюються через дві секунди після встановлення з'єднання (22-31).
- Атрибути вмісту (11-21).
- Атрибути трафіку хосту (32-41).
- База даних містить значення кожного атрибута для виявлення наступних типів мережевих вторгнень:
- Розподілена відмова в обслуговуванні (DDoS) - вторгнення, спрямовані на блокування мережі.

- Зонд - вторгнень, спрямовані на сканування даних або виявлення вразливості мережі для наступних атак на інші мережі.
- Віддалений до локального (U2L) - вторгнення, спрямовані на встановлення несанкціонованого з'єднання шляхом відправлення пакетів до цієї мережі.
- User to Root (U2R) - вторгнення, спрямовані на отримання прав адміністратора загальним користувачем.

Кількість вхідних нейронів дорівнює 4, що відповідає кількості виявлених вторгнень. Відповідно до [85] кількість нейронів у кожному прихованому шарі становить 300. Після тренування розроблену модель нейронної мережі використовували для виявлення прикладів, які не застосовувались для навчання. Тестовий набір містив приклади, що описують мережеві підключення за 24 години. Результати виявлення проілюстровані на рисунку 2.7. Слід зазначити, що на рисунку 2.7 показано кількість та типи виявлених вторгнень за кожну годину. Середня точність виявлення становить близько 90%, що відповідає точності виявлення вторгнень за допомогою відомих систем виявлення вторгнень [85, 86].

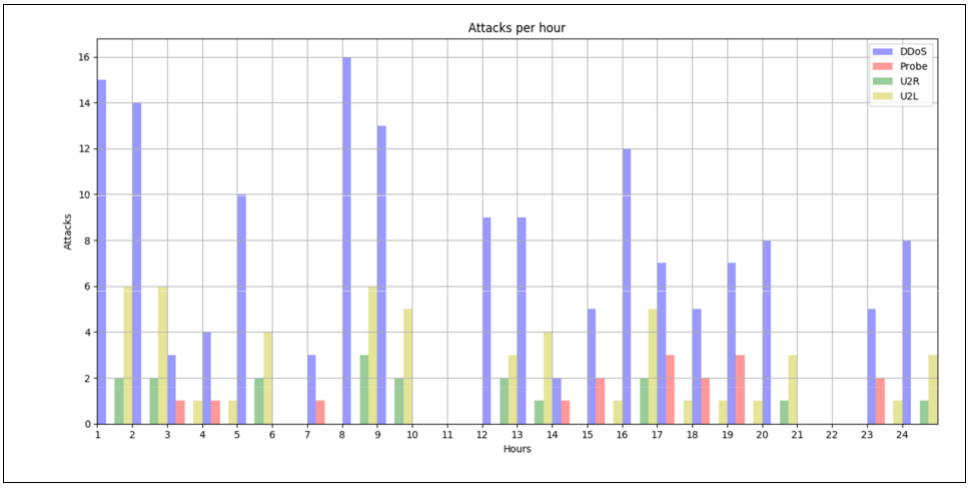


Рисунок 2.7 - Схема виявлення вторгнень моделі нейронної мережі

2.5 Висновки

Проаналізувавши моделі та алгоритми навчання нейронної мережі, стало зрозуміло, що проектування систем ідентифікації вторгнень стане одним із найперспективніших підходів до розробки систем виявлення мережевих вторгнень за допомогою вдосконалення їх програмного забезпечення шляхом застосування сучасних моделей на базі глибоких нейронних мереж. Цьому сприяла розробка відповідної моделі нейронної мережі, яка була попередньо навчена розрідженим автокодером. Запропонована модель була підтримана програмним забезпеченням з подальшим апробацією для виявлення мережевих вторгнень, параметри цих атак зберігались у базі даних NSL-KDD. Результати модельного тестування продемонстрували, що точність базового варіанту порівнянна з точністю сучасних систем виявлення мережевих вторгнень.

У розділі розглянуто основні моделі нейронних мереж та алгоритми глибокого навчання, що дозволяють розробити та покращити систем для виявлення вторгнень.

3 МЕТОДИ РЕАЛІЗАЦІЇ

3.1 Потенційні методи реалізації

В нейронних мережах в системах виявлення вторгнень існує дві основні реалізації. Перша передбачає включення їх в існуючі або модифіковані експертні системи. На відміну від попередніх спроб використовування нейронні мережі в виявлення аномалій, використовують їх в якості заміни для існуючих компонентів статистичного аналізу, цей варіант пов'язаний з використанням нейронної мережі для фільтрації вхідних даних з метою виявлення підозрілих подій, які можуть вказувати на вторгнення і направляти ці події експертної системі. Дана конфігурація повинна поліпшити ефективність системи виявлення за рахунок зменшення помилкових тривог експертної системи. Оскільки нейронна мережа визначає ймовірність того, що певна подія є показником атаки, можна встановити поріг, при якому подія направляється в експертну систему для додаткового аналізу. Оскільки експертна система тільки отримує дані про події, які розглядаються як підозрілі, чутливість експертної системи може бути збільшена, (зазвичай, чутливість експертних систем повинна бути низькою, щоб зменшити частоту помилкових тривог). Ця конфігурація буде корисна для організацій, які інвестували в технології експертних систем на основі правил, за рахунок підвищення ефективності системи при збереженні інвестицій, які були зроблені в існуючі системах виявлення вторгнень. Недоліком цього підходу є те, що, в той час, як нейронна мережа розширила свої можливості для виявлення нових атак, експертну систему необхідно буде оновити для того, що б вона так само розпізнавала ці загрози[84].

Другий підхід визначає нейронну мережу як автономну систему виявлення вторгнень. У цій конфігурації, нейронна мережа буде отримувати дані з мережевого потоку і аналізувати інформацію на наявність вторгнення. Будь-які випадки, які визначені як показник атаки будуть спрямовані адміністратору безпеки або використані автоматизованою системою реагування на вторгнення.

Цей підхід передбачає велику швидкість виявлення в порівнянні з попереднім підходом, оскільки задіяний тільки один шар аналізу. Крім того, ця конфігурація повинна поліпшити ефективність з плином часу, так як мережа вивчає нові ознаки атак. На відміну від першого підходу, ця концепція не обмежена аналітичними здібностями експертної системи, і, як наслідок, вона зможе розширитися за межі бази правил експертної системи.

3.2 Інструмент NNT пакету Matlab

Пакет для роботи з нейронними мережами Neural Network Toolbox являє собою повноцінне середовище MATLAB для вирішення прикладних завдань, забезпечуючи підтримку проектування, навчання та моделювання безлічі відомих мережевих парадигм, від класичних моделей персептрона до найновіших асоціативних і самоорганізуючих мереж. Пакет можна використовувати для дослідження і застосування нейромереж для обробки сигналів, нелінійного управління та фінансового моделювання.

Основні можливості пакету NNT:

- Керовані мережеві парадигми: персептрон, лінійні, зворотного поширення, Левенберга, радіальний базис, Елмана, Хопфилда і самонавчанавчання квантування векторів;
- некеровані мережі: Хебом, Кохонен, конкурентні, карти ознак і самоорганізуючі карти;
- конкурентні, граничні, лінійні і сигмоїдальні передавальні функції;
- необмежена кількість елементів і взаємозв'язків;
- настроюються на користувача архітектури та передавальні функції;
- модульна організація – пакет використовує узгоджену, модульну реалізацію, яка полегшує дослідження і спрощує настройку на користувача. Він не накладає штучні обмеження на розмір мережі або зв'язність (необмежена кількість нейронів в шарі або в типі передавальної функції).

- архітектури і навчальні правила - у пакет включені більше 15 відомих типів мереж і навчальних правил, що дозволяють користувачеві вибирати найбільш підходящу для конкретного додатка або дослідницької мети парадигму. Для кожного типу архітектури та навчальних правил є функції ініціалізації, навчання, адаптації, створення і моделювання, демонстрації та приклад програми мережі.
- керовані і некеровані мережі - для керованих мереж можна вибрати пряму або рекурентну архітектуру, використовуючи безліч навчальних правил і методів проектування, таких як персептрон, зворотне поширення, зворотне поширення Левенберга, мережі з радіальним базисом і рекурентні мережі.
- для некерованих мереж можна вибрати асоціативні або самоорганізовуючі мережі, такі як конкурентні, карти властивостей і самоорганізуючі карти. Асоціативні мережі можна використовувати як складові блоки для більш складних мереж з використанням навчальних правил Хебба, Кохонена, внутрішніх або зовнішніх (instar or outstar) асоціативних навчальних правил.
- Neural Network як інженерне середовище. Пакет Neural Network надає доступ до повного набору засобів для дослідження, проектування і моделювання нейронних мереж. Засоби аналізу і моделювання MATLAB дозволяють швидко оцінювати поведінку мережі і її якість в сенсі остаточного результату проектування.

В склад пакету Neural Networks Toolbox входять більше 150 різних функцій, які розбиті на декілька груп, надаючи користувачу широкі можливості по створенню, навчання і застосування різних нейромереж.

Ввівши в вікні «Command Window» команду Help nnet можна отримати перелік вхідних в пакет NNT функцій. Для отримання довідки щодо необхідної функції використовується команда:

Help ім’я_функції

Функції активації (transfer functions) і пов'язані з ними функції:

- compet(X) - «змагальна» функція. Як аргумент використовується матриця X, стовпці якої асоціюються з векторами входів. Повертає розріджену матрицю з одиничними елементами, індекси яких відповідають індексам найбільших елементів кожного стовпця.

  В вигляді запиту compet (code) повертається службова інформація. Мінлива code може приймати значення “deriv” (ім’я похідною функції), “name” (повне ім’я), “output” (діапазон виходу), “active” (можливий діапазон входів).

  Ця функція використовується при створенні нейронної мережі із «змагальним» шаром нейронів (як в мережах зустрічного розповсюдження).
- hardlim(X) - порогова функція активації з порогом P = 0; аргумент має таке ж значення, що і для попередньої команди. Матриця повертається, якщо її розмір дорівнює розміру матриці X, а елементи дорівнюють 0 або 1 в залежності від знака відповідного елемента X. В вигляді hardlim (code) функція поверне інформацію, аналогічну розглянутої для функції compet.
- hardlims (X) - знакова або сигнатурна функція активації, працює так як функція hardlim (X), але повертає значення -1 або +1.
- logsig (X) - сигмоїдальна логістична функція. Результат: матриця, елементи якої є значеннями логістичної функції від аргументів, якими служать елементи матриці X.
- poslin(X) - повертає матрицю значень напівлінійної функції.
- purelin(X) - повертає матрицю значень лінійної функції активаціі.
- radbas(X) - повертає матрицю значень радіальної базисної функції.
- satlin(X) - повертає матрицю значень напівлінійної функції з насиченням.
- satlins(X) - повертає матрицю значень лінійної функції з насиченням.
- softmax(X) - повертає матрицю, яка обчислюються за формулою $\frac{e^{x_{ij}}}{\sum_{i=1}^{N} e^{x_{ij}}}$, де N - число рядків матриці - аргументу X.

- tansig(X) - повертає матрицю значень сигмоїдальної функції (гіперболічний тангенс).
- tribas(X) - повертає матрицю значень трикутної функції приналежності.
- dhardlim(X, Y) - похідна порогової функції активації. Аргументами є матриця входів X і матриця виходів Y. Матриці мають однаковий розмір. Результат: матриця такого ж розміру з нульовими елементами.
- dhardlms(X, Y) – похідна знакової функції активації. Повертається матриця з нульовими елементами.
- dlogsig(XY) - похідна сигмоїдальної логістичної функції. Результат - матриця з елементами $y_{ij}(1 - y_{ij})$.
- dposlin(X, Y) - похідна напівлінійної функції. Результатом служить матриця з елементами, рівними 1 для відповідних позитивних елементів матриці-аргументу Y і рівними 0 в протилежному випадку.
- dpurelin(X, Y) - похідна лінійної функції активації. Повертається матриця з одиничними елементами.
- dradbas(X, Y) - похідна радіальної базисної функції. Буде повернута матриця з елементами $-2x_{ij}y_{ij}$.
- dsatlin(X, Y) - повертає матрицю значень похідної напівлінійної функції з насиченням. Відповідні елементи матриці Y належать інтервалу (0, 1), і нулі протилежному випадку, якщо елементи цієї матриці - одиниці.
- dsatlins(X, Y) - повертає матрицю значень похідної лінійної функції з насиченням. Відповідні елементи матриці Y належать інтервалу (-1, 1), інакше це нулі, якщо елементи матриці - одиниці.
- dtansig(X, Y) - повертає матрицю значень похідної сигмоїдальної функції – гіперболічного тангенса. Елементи матриці обчислюються як $(1 - y_{ij}^2)$.
- dtribas(X, Y) - повертає матрицю значень похідної трикутної функції активації. Елементи матриці визначаються наступним чином: це 1, якщо $-1 < y_{ij} < 0$; якщо $0 < y_{ij} < 1$; інакше 0.

Функції створення та навчання нейронних мереж (training functions)

network створює нейронної мережі користувача.

синтаксис:

net = network

net = network (numInputs, numlayers, biasConnect, inputConnect, layerConnect, outputConnect, targetConnect)

Функція network повертає створену нейронну мережу з ім’ям net з наступними характеристиками (в дужках дані значення за замовчуванням):

- numInputs - кількість входів (0);
- numLayers - кількість шарів (0);
- biasConnect - булевий вектор з числом елементів, що дорівнює кількості шарів (нулі);
- inputConnect - булева матриця з числом рядків, що дорівнює кількості шарів, і числом стовпців, що дорівнює кількості входів (нулі);
- layerConnect - булева матриця з числом рядків і стовпців, що дорівнює кількості шарів (нулі);
- outputConnect - булевий вектор-рядок з числом елементів, що дорівнює кількості шарів (нулі);
- targetConnect - вектор-рядок така ж, як попередня (нулі).

net = newc (PR, S, KLR, CLR) – функція створення шару Кохонена.

Опис аргументів:

- PR – R * 2 - матриця мінімуму і максимуму значень для R вхідних елементів;
- S - число нейронів;
- KLR - коефіцієнт навчання Кохонена (за замовчуванням 0,01);
- CLR - коефіцієнт «справедливості» (за замовчуванням 0,001). Функція повертає шар Кохонена із заданим ім'ям.

net = newcf (PR, [S1 S2 ... SN1], {TF1 TF2 ... TFN1}, BTF, BLF, PF) - функція

створення різновиду багатошарової нейромережі зі зворотним поширенням помилки каскадної нейромережі. Така мережа містить N1 прихованих шарів, використовує вхідні функції типу dotprod і netsum, ініціалізація мережі здійснюється функцією initnw. Аргументи функції:

- PR – R * 2 - матриця мінімальних і максимальних значень R вхідних елементів;
- Si - розмір i-го прихованого шару, для N1 шарів;
- TFi - функція активації нейронів r-го шару, за замовчуванням “tansig”;
- BTF - функція навчання мережі, за замовчуванням “traingd”;
- BLF - функція настройки ваг і зміщень, за замовчуванням “Learngdm”;
- PF - функція помилки, за замовчуванням “mse”.

```
»P = [012345678910];
»T = [0 123432123 4];
»Net = newcf ([0 10], [5 1], { 'tansig' 'purelin'});    % Створення нової мережі
»Net.trainParam.epochs = 50; % Завдання кількості циклів навчання
»Net = train (net, P, T);                % Навчання нейромережі
TRAINLM, Epoch 0/50, MSE 7.77493 / 0, Gradient 138.282 / le-010
TRAINLM, Epoch 25/50, MSE 4.01014e-010/0, Gradient 0.00028557 / le010
TRAINLM, Epoch 50/50, MSE 1.13636e-011/0, Gradient 1.76513e-006 / le-
010TRAINLM, Maximum epoch reached, performance goal was not met.
»Y = sim (net, P);                % Використання нейромережі

»Plot (P, T, P, Y, 'o')            % Графічне зображення роботи мережі
```

На рисунку 3.1 елементи навчальної вибірки відображені точками, суцільна лінія - вихід мережі.

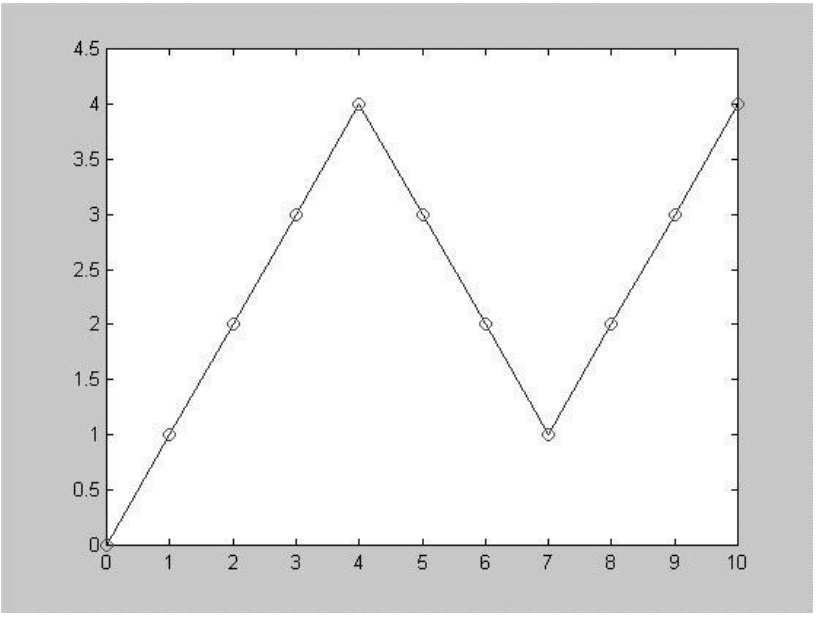


Рисунок 3.1 - Робота нейромережі

- net = newelm (PR, [S1 S2 ... SN1], {TF1 TF2 ... TFN1}, BTF, BLF, PF) - функція створення мережі Елмана. Аргументи - такі самі, як і у попередньої функції.
- net = newff (PR, [S1 S2 ... SN1], {TF1 TF2 ... TFN1}, BTF, BLF, PF) - функція створення «класичної» багатошарової нейромережі з навчанням за методом зворотнього розповсюдження помилки.
- net = newfftd (PR, ID, [S1 S2..SN1], {TF1 TF2 ... TFN1}, BTF, BLF, PF) - виконує те саме, що і попередня функція, з затримкою по входах. Аргумент ID тут - це вектор вхідних затримок.
- net = newgrnn (P, T, spread) - функція створення узагальнено регресивної мережі. Значення аргументів:
- P – R^x Q - матриця Q вхідних векторів;
- T – S^x Q - матриця Q цільових векторів;
- spread - відхилення (за замовчуванням 1,0).
- net = newhop (T) - функція створення мережі Хопфілда. має тільки

Один аргумент:

- T – R^x Q - матриця Q цільових векторів (значення елементів повинні бути +1 або -1).

- net = newlin (PR, S, ID, LR) - функція створення шару лінійних нейронів. аргументи:
- PR - R x 2 - матриця мінімуму і максимуму значень для R вхідних елементів;
- S - число елементів у вихідному векторі;
- ID - вектор вхідний затримки (за замовчуванням [0]);
- LR - коефіцієнт навчання (за замовчуванням 0,01).

Як результат функція повертає новий лінійний шар. При запису в формі

net = newlin (PR, S, 0, P) використовується аргумент P - матриця вхідних векторів, при цьому повертається лінійний шар з максимально можливим коефіцієнтом навчання при заданій матриці P.

- net = newlind (P, T) - функція проектування нового лінійного шару. Ця функція за матрицями входу і виходу векторів методом найменших квадратів встановлює ваги і зміщення лінійної нейромережі.
- net = newlvq (PR, S1, PC, LR, LF) - функція створення мережі зустрічного поширення.

Аргументи:

- PR - R x 2 - матриця мінімуму і максимуму значень R вхідних елементів;
- S1 - число прихованих нейронів;
- PC - S2 елементів вектора, які задають частки приналежності до різноманітних класів;
- LR - коефіцієнт навчання, за замовчуванням 0,01;
- LF - функція навчання, за замовчуванням 'learnlv1'.
- net = newp (PR, S, TF, LF) - функція створення персептрона.

Аргументи:

- PR - R x 2 - матриця мінімальних і максимальних значень R вхідних елементів,
- S - число нейронів;
- TF - функція активації, за замовчуванням 'hardlim';

- LF - функція навчання, за замовчуванням 'learnp'.
- net = newpnn (P, T, spread) - функція створення імовірнісної нейромережі.

Аргументи - як у функції newgrnn.

- net = newrb (P, T, goal, spread) - функція створення мережі з радіальними базисними елементами.

Аргументи P, T, spread - такі ж, як у функції newgrnn;

Аргумент goal - задана середньоквадратична помилка.

- net = newrbe (P, T, spread) - функція створення мережі з радіальними базисними елементами з нульовою помилкою на навчальній вибірці.
- net = newsom (PR, [D1, D2, ...], TFCN, DFCN, OLR, OSTEPS, TLR, TND) - функція створення самонавчальної карти з наступними аргументами:
- PR - R ^{x} 2 - матриця мінімальних і максимальних значень R вхідних елементів;
- I - розміри і-го шару, за замовчуванням [5 8];
- TFCN - топологічна функція, за замовчуванням 'hextop';
- DFCN - функція відстані, за замовчуванням 'linkdist';
- OLR - коефіцієнт навчання фази упорядкування, за замовчуванням 0,9;
- OSTEPS - число кроків фази упорядкування, за замовчуванням 1000;
- TLR - коефіцієнт навчання фази настройки, за замовчуванням 0,02;
- TND - відстань для фази настройки, за замовчуванням 1.

Функції розміщення нейронів використовуються для створення самоорганізуючих карт.

ridtop (diml, dim2, ..., dimN) - функція розміщення N шарів нейронів в вузлах регулярної прямокутної N-мірної решітки, diml, dim2, ..., dimN - число нейронів в шарах. Повертає матрицю, що містить N рядків і (diml^{x} dim2 ^{x} ... ^{x} dimN) стовпців з координатами нейронів.

```
» pos = gridtop (2,3)
pos =
```

0 1 0 1 0 1

0 0 1 1 2 2

hextop (dim1, dim2, ..., dimN) - функція аналогічна попередній, але розміщення нейронів проводиться в вузлах гексагональної (шестикутної) решітки (дивитись рисунок 3.2).

» pos = hextop (8,5);

» plotsom (pos)

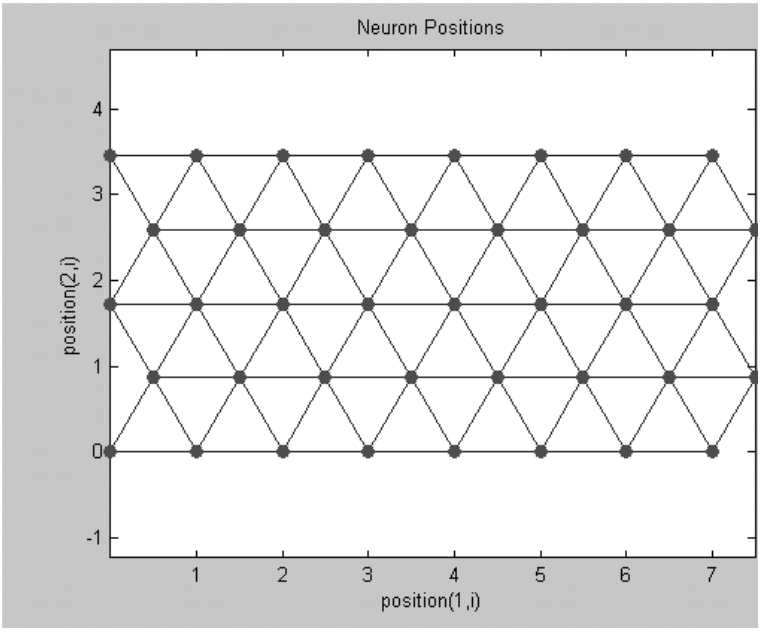


Рисунок 3.2 - Ілюстрація роботи команди hextop

randtop (dim1, dim2, ..., dimN) - аналогічна функції gridtop (diml, dim2, ..., dimN), але координати нейронів вибираються випадковим чином рисунок 3.3):

» pos = randtop (16,12);

» plotsom (pos)

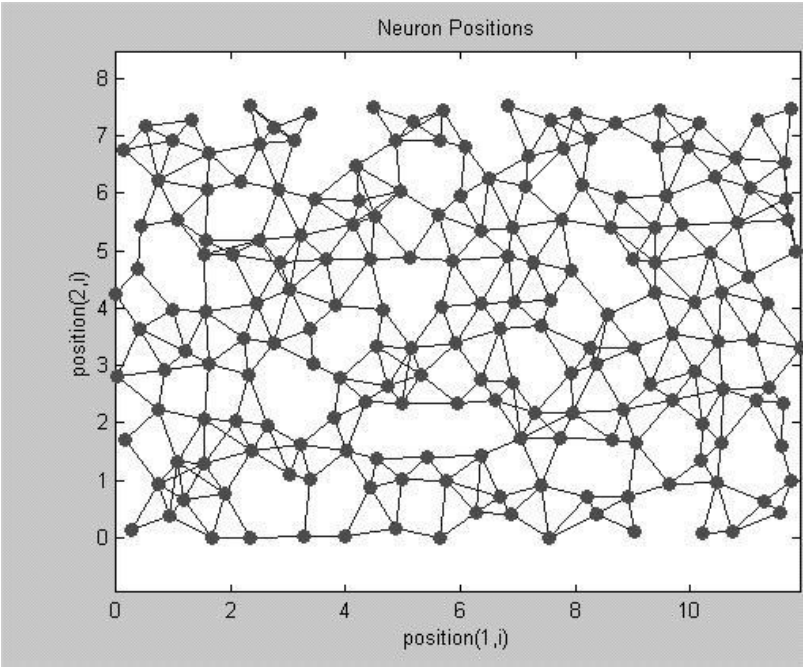


Рисунок 3.3 - Ілюстрація роботи команди randtop

Як бачимо, нейрони тут розташовані хаотично.

3.3 Реалізація нейронних мереж NNT

Апроксимуємо функцію $y = x^2$ зі зміною аргументу на відрізку [-1, 1]. Для цього створимо узагальнено-регресійну нейронну мережу під ім'ям «a». Апроксимація проводиться за наступними даними:

x = [-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1],

y = [1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1]

Реалізація (створення і використання даної мережі). У командному вікні пакета Matlab послідовно наберемо:

» % Задати вектор вхідних значень

» P = [-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1];

» % Задати вектор вихідних значень

» T = [1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1];

» a = newgrnn (P, T, 0.01); %Створення нейромережі з відхиленням 0.01

» Y = sim (a, [- 0.9 -0.7 -0.3 0.4 0.8])% Опитування нейромережі

Y =

0.8200 0.6400 0.0400 0.0900 0.8100

Як бачимо, точність апроксимації в даному випадку вийшла недостатньо високою.

Можна спробувати поліпшити якість апроксимації варіацією величини відхилення, але тут непоганий результат можна легко досягнути використовуючи мережі з радіальними базисними елементами:

» a = newrbe (P, T);

» Y = sim (a, [- 0.9 -0.7 -0.3 0.4 0.8]) % Опитування нейромережі

1. Y =
2. 0.8100 0.4900 0.0900 0.1600 0.6400

Щоб зберегти побудовану мережу, в командному рядку треба набрати оператор *save ('a')*; при цьому буде створено файл *a.mat*, що містить ім'я нейромережі, з розширенням *mat*. У наступних сеансах роботи збережену мережу можна завантажити, використовуючи функцію *load ('a').*

Розглянемо використання лінійної нейромережі для вирішення аналогічної задачі. Нехай експериментальні дані такі:

x = [+1.0 +1.5 +3.0 -1.2], y = [+0.5 +1.1 +3.0 -1.0]

Процес створення, навчання і використання лінійної нейромережі з ім'ям «b» ілюструється наведеним нижче лістингом і рисунок 3.3.

» P = [+1.0 +1.5 +3.0 -1.2];

» T = [+0.5 +1.1 +3.0 -1.0];

% Визначення величини коефіцієнта навчання

» maxlr = maxlinlr (P, 'bias');

» B = newlin ([- 2 + 2], 1, [0], maxlr); % Створення лінійної нейромережі з ім'ям «b»

» b.trainParam.epochs = 15; % Завдання кількості циклів навчання

» B = train (b, P, T); % Навчання нейромережі

TRAINWB, Epoch 0/15, MSE 2.865 / 0.

TRAINWB, Epoch 15/15, MSE 0.0730734 / 0.

TRAINWB, Maximum epoch reached.

» p = -1.2;

» y = sim (b, p)% Опитування мережі

y = -1.1803

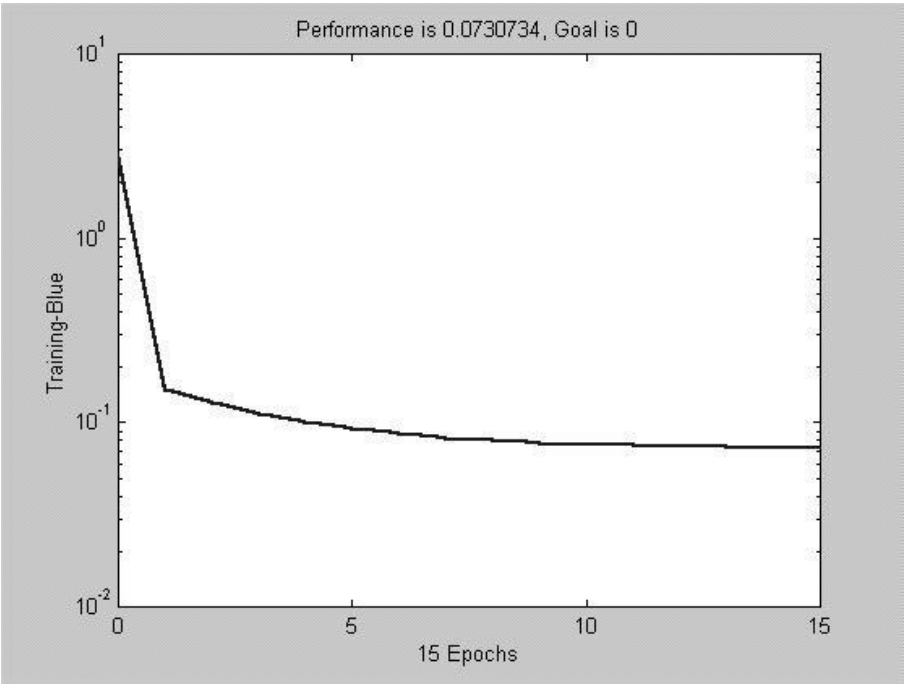


Рисунок 3.3 - Процес навчання нейромережі

Це ілюстрацією того, як зменшується помилка в процесі навчання нейромережі. У цьому розділі ми представляємо деталі нашого методу виявлення вторгнень в мережу. Пропозиція застосовує DNN у два етапи. Перший етап витягує абстраговані часові особливості поведінки користувачів за допомогою LSTM і виводить вектори характеристик. Потім вектори ознак перетворюються на матриці функцій xed-size. На другому етапі ці матриці функцій xed-size подаються на CNN, щоб класифікувати їх як нормальні або як аномалії.

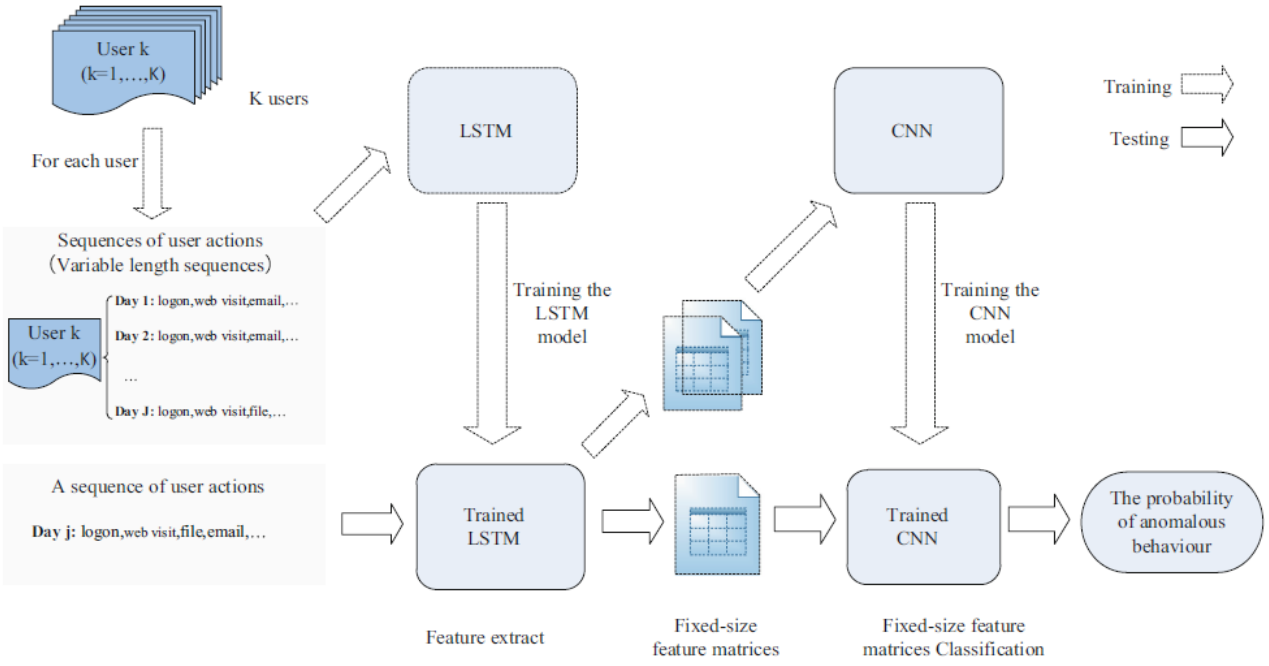


Рисунок 3.4 - Огляд способу

Огляд нашого методу виявлення внутрішньої загрози наведено на рисунок 3.4. Індивідуальна дія (наприклад, вхід на призначений комп'ютер поза робочим часом) являє собою роботу користувача; дії, здійснені користувачем за один день, представляють поведінку користувача. Подібно до мов моделювання, дія відповідає слову, а послідовність дій відповідає реченню. З цієї причини ми намагаємось вивчити мову поведінки користувачів як новий метод виявлення внутрішньої загрози. LSTM використовується для вилучення особливостей поведінки користувачів. CNN використовує ці функції для пошуку аномальної поведінки.

Нехай $U = \{u_1, u_2, \dots, u_K\}$, буде набором K користувачів.

Для користувача $u_k (1 \le k \le K)$, ми можемо отримати його послідовності дій протягом J дня.

$S = \left\{S_{u_{k,1}}, S_{u_{k,2}}, \dots, S_{u_{k,j}}\right\}$, де $S_{u_{k,j}} (1 \le j \le J)$ є вектором, який означає послідовність дій у дні, проіндексованому j. На етапі навчання, ми спочатку отримуємо послідовність дій $S_{u_{k,j}}$, яку користувач u_K виконав протягом дня індексованому j. Потім, послідовність дій $S_{u_{k,j}}$, яка подається в LSTM і LSTM,

навчається будувати екстрактор ознак для отримання абстрагованих векторів ознак у глибокому шарі. На наступному етапі, вектори характеристик перетворюються на матрицю фіксованого розміру $M^{u_{k,j}}$. Матриця функцій фіксованого розміру потенційно містить різні абстраговані часові ознаки, які представляють поведінку користувачів. Нарешті, ми використовуємо ці матриці фіксованого розміру, анотовані нормальним або аномальним тренуванням CNN. На етапі тестування ми оцінюємо підхід за допомогою навченого LSTM та CNN.

3.4 Висновки

У цьому розділі ми запропонували до розгляду потенційні методи реалізації виявлення вторгнень за допомогою глибокої нейронної мережі, пов'язані з використанням нейронної мережі для фільтрації вхідних даних з метою виявлення підозрілих подій, які можуть вказувати на вторгнення і направляти ці події експертної системі. Сскільки загроза вторгнень проявляється у різних формах, її потрібно по різному моделювати. Для реалізації систем ідентифікації вторгнень ми розглянули інструмент nnt пакету середовища matlab, де розглянули основні можливості пакету та його функції для створення та навчання нейромереж. також реалізували нейронну мережу з nnt в два етапи, часові особливості поведінки користувачів за допомогою lstm і виводить вектори характеристик і класифікація їх як нормальні або як аномалії.

4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

4.1 Розмежування функцій виявлення вразливостей вторгнення в нейромережу та системи прийняття рішень.

Для можливості створення виявлення вразливостей вторгнення в нейромережу було проведено розмежування задач, які виконуються нейромережею після надходження інформації про виникнення вторгнення, виконання більшої частини цих задач було перенесено на систему виявлення вразливостей вторгнення в нейромережу.

На систему покладено виконання таких функцій:

- прийняття повідомлення про виникнення вторгнення;
- введення інформації про виниклу ситуацію у виявленні вразливостей вторгнення в нейромережу за допомогою діалогових вікон;
- керування та взаємодія з системою у процесі її роботи, отримання результатів (множини первинних рішень, наслідків виконання цих рішень, прогнозованого сценарію загрози, остаточного рішення) від системи;
- визначення остаточного рішення (на основі рекомендацій системи або на основі власних переваг) та його реалізація.

На виявлення вразливостей вторгнення в нейромережу покладено виконання таких функцій:

- аналіз інформації про ситуацію, яка була виявлена в мережі, її доповнення необхідними для подальшої роботи з данними;
- розпізнавання вторгнення, визначення множини первинних рішень для її вирішення та набору критеріїв для їх оцінювання;
- прогнозування розвитку вторгнення, визначення наслідків виконання кожного з первинних рішень;
- оцінювання кожної з первинних альтернатив за відібраними критеріями та наслідками виконання;

- за отриманими результатами рекомендація остаточного рішення для вирішення ситуації;
- демонстрація усієї вищезгаданої інформації про виявлення вторгнення у зрозумілому для сприйняття вигляді.

4.2 Структурна та функційна схеми виявлення вторгнень на основі нейронної мережі.

На основі концептуальної моделі та проведеного у 4.1 розмежування задач було побудовано структурну схему виявлення вторгнення в нейронну мережу (рисунок 4.1).

Рисунок 4.1 - Структурна схема виявлення вторгнення в нейронну мережу.

На “Діалогову підсистему” покладено задачу спілкування з користувачем та демонстрація йому результатів роботи системи.

На “Підсистему збір чуттєвої інформації” покладено аналіз інформації, що надходить від нейромережі та доповнення її відомостями, пов'язаними з

згаданими в інформації про вторгнення.

На “Підсистему прийняття рішень” покладено вирішення задач виявлення вторгнень та прийняття первинних рішень.

На “Підсистему прогнозування сценарію загрози” покладено виконання задач прогнозування сценарію загрози та визначення наслідків первинних рішень.

На “Підсистема оцінювання ефективності рішень” покладено вирішення задачі оцінювання ефективності рішень.

Використовуючи за основу структуру виявлення вразливостей вторгнення в систему, було побудовано функціональну схему такої системи (рисунок 4.2).

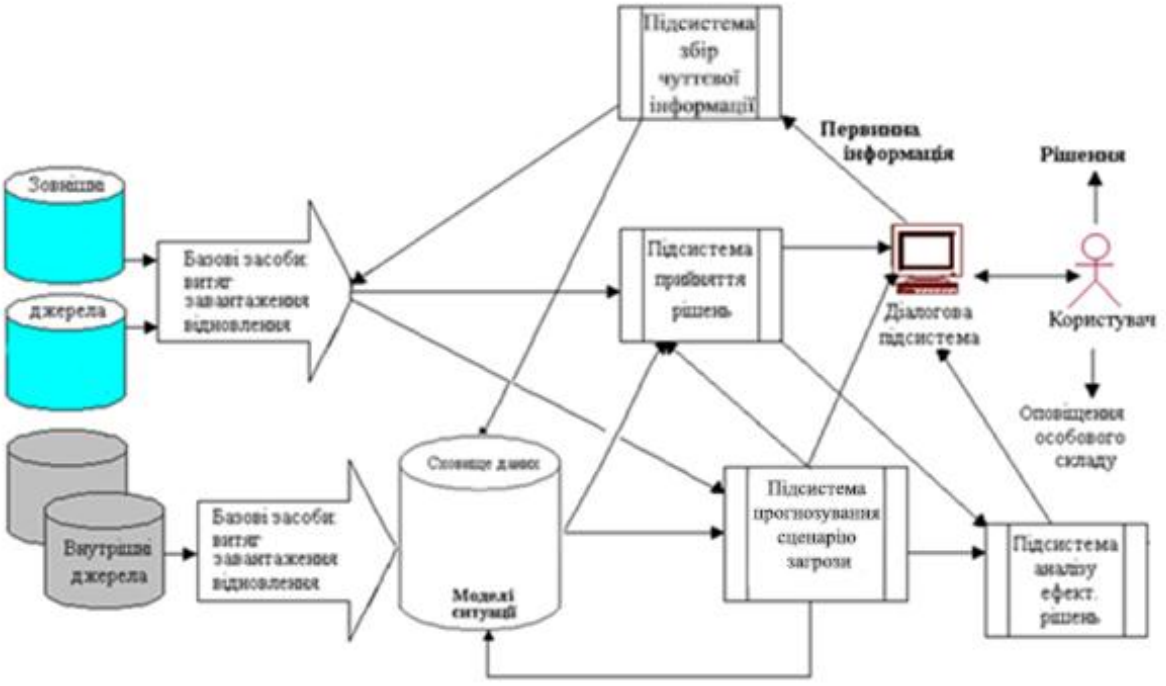


Рисунок 4.2 - Функціональна схема системи.

Діалогова підсистема є інтерфейсом користувача для зручного спілкування із нейромережею.

Підсистема збір чуттєвої інформації виконує аналіз данних, що надходить від нейромережі та доповнювання її відомостями, пов'язаними з згаданими в данних про вторгнення. Для цього нейромережа інтегрує бази даних, накопичені

в органах внутрішніх справ за особами, які вже створювали вторгнення в мережі. Зазначені бази даних було зроблено при допомозі програмного продукту MySQL.

Підсистема прийняття первинних рішень розрізняє поточну ситуацію і пропонує набір можливих альтернатив поведінки, а також набір критеріїв для того, щоб оцінюванити ефективність альтернатив. При цьому ситуація є зростаючою в плані поповнення в неї нових варіантів дій. Зростання проходить через додавання множини даних створенням нових типів ситуацій і прогнозами, набутими в процесі роботи “Підсистеми прогнозування сценарію загрози”. Інформація стосовно можливих альтернатив та критеріїв демонструється “Діалоговою підсистемою” за запитом користувача.

Підсистема прогнозування сценарію загрози забезпечує прогнозування розвитку оперативної обстановки у часі та передбачає наслідки альтернатив, запропонованих “Підсистемою прийняття первинних рішень”. Цю інформацію вона використовує для демонстрації сценарію загрози “Діалоговою підсистемою” за запитом нейромережі. Передбачені наслідки можливих альтернатив використовуються “Підсистемою оцінювання ефективності рішень” для оптимізації попередньо названих альтернатив.

Підсистема оцінювання ефективності рішень проводить оцінку визначених “Підсистемою прогнозування розвитку вторгнення” наслідків альтернатив поведінки за наданими “Підсистемою прийняття первинних рішень” критеріями. Отримана у результаті оцінювання краща альтернатива – остаточне рішення, демонструється користувачу за допомогою “Діалогової підсистеми”.

Моделі ситуації являють собою структуровану інформацію про всі обставини що відбуваються, альтернативні сценарії і прийняті остаточні рішення.

Система накопичує такі моделі з трьох основних джерел:

- опрацьовані і сформовані спеціальні плани;
- моделі подій, що реально відбувалися;
- результати роботи підсистеми прогнозування сценарію загрози стосовно подій, що ще не відбулися, але можуть відбутися.

4.3 Опис програмних модулів інтелектуальної системи для виявлення мережевих вторгнень.

Після виявлення вторгнення, система покаже на екрані вікно (рисунок 4.3).

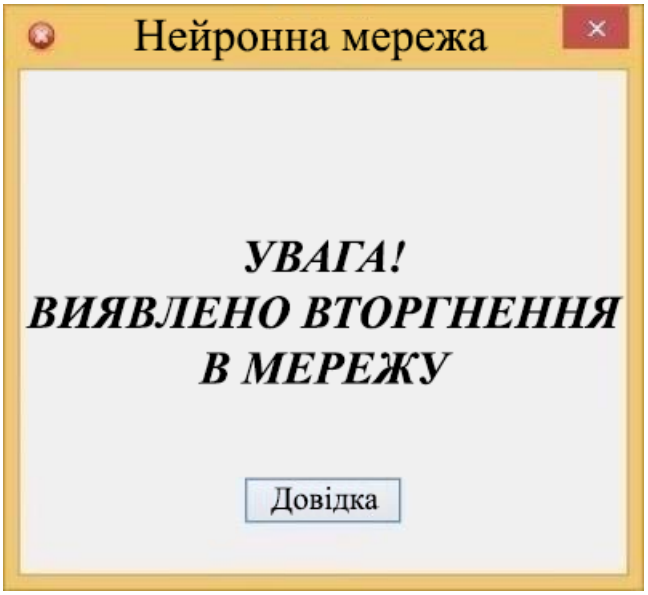


Рисунок 4.3 - Сигнал від нейромережі про вторгнення.

При натисканні на кнопку “Довідка” користувач отримує інформацію про вторгнення. Підсистема “Збір чуттєвої інформації” визначає показники небезпечності вторгнення. Абсолютний час ситуації визначається за годинником на комп’ютері, де встановлена програма. Сформовані дані подаються на опрацювання до підсистем “Прийняття первинних рішень” та “Прогнозування сценарію загрози”.

Виявлення вразливостей вторгнення в нейромережу та системи прийняття рішень, та взаємодія і обмін даними між її підсистемами було реалізовано за допомогою програмного продукту C++ Builder 6.0.

4.4 Висновки

В даному розділі було досліджено функції виявлення вразливостей вторгнення в нейромережу та системи прийняття рішень. Побудувано структурну та функційну схеми виявлення вторгнень на основі нейромережі. Було виявлено вразливості вторгнення в нейромережу через розмежування задач, які виконуються нейромережею після надходження інформації про виникнення вторгнення, виконання більшої частини цих задач було перенесено на систему виявлення вразливостей вторгнення в нейромережу.

ВИСНОВКИ

У кваліфікаційній роботі було описано метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання. Запропонований метод дозволяє удосконалити модель комп'ютерної системи для ідентифікації вторгнень.

Аналізуючи світові тенденції ,ми можемо стверджувати, що в найближчі роки дане питання буде ще більше актуальне, цьому буде сприяти стрімкий розвиток інформаційних технологій, яке ми спостерігаємо вже сьогодні.

Основні результати кваліфікаційної роботи:

1. Досліджено способи застосування технологій нейронних мереж в системах виявлення вторгнень та їх проблеми , з перевагами і недоліками, щоб визначити найпоширеніші способи виявлення вторгнень.

2. Розглянуто моделі нейромереж з метою їх застосування до виявлення вторгнень.

3. Розроблено метод покращення реалізації для систем виявлення вторгнень.

4. Розроблено алгоритм, що дозволяє виконувати глибоке навчання в нейронній мережі для виявлення вторгнень.

5. Розроблено модель глибинної нейромережі для виявлення вторгнень.

6. Впроваджено та протестовано розроблений алгоритм.

7. Визначено сферу застосування розробленого методу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus and M. Fallgren "Scenarios for 5G mobile and wireless communications: The vision of the metis project," IEEE Commun. Mag., vol. 52, no. 5, pp. 26–35, May 2014.

2. M. Alvarez, N. Bradley, P. Cobb, S. Craig, R. Iffert, L. Kessem, J. Kravitz, D. McMilen , S. Moore "IBM X-force threat intelligence index 2017," IBM Corporation, pp. 1–30, 2017.

3. C. Kolias, A. Stavrou, J. Voas, I. Bojanova, R. Kuhn "Learning internet-of-things security" hands-on"," IEEE Security Privacy, vol. 14, no. 1, pp. 37–46, 2016.

4. C. Kolias, G. Kambourakis, M. Maragoudakis "Swarm intelligence in intrusion detection: A survey," Computers & Security, vol. 30, no. 8, pp. 625–642, 2011.

5. A. G. Fragkiadakis, V. A. Siris, N. E. Petroulakis, A. P. Traganitis "Anomaly-based intrusion detection of jamming attacks, local versus collaborative detection," Wireless Communications and Mobile Computing, vol. 15, no. 2, pp. 276–294, 2015.

6. R. Sommer and V. Paxson "Outside the closed world: On using machine learning for network intrusion detection," in Proc. Symp. Security and Privacy, Berkeley, California. IEEE, 2010, pp. 305–316.

7. G. Anthes "Deep learning comes of age," Communications of the ACM, vol. 56, no. 6, pp.13–15, 2013.

8. A. H. Farooqi , F. A. Khan "Intrusion detection systems for wireless sensor networks: A survey," in Proc. Future Generation Information Technology Conference, Jeju Island, Korea. Springer, 2009, pp. 234–241.

9. R. Zuech, T. M. Khoshgoftaar, R. Wald "Intrusion detection and big heterogeneous data: a survey," Journal of Big Data, vol. 2, no. 1, p. 3, 2015.

10. J. Schmidhuber "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.

11. L. Deng "A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA Transactions on Signal and Information Processing, vol. 3, 2014.

12. L. Deng, D. Yu,et al. “Deep learning: methods and applications,” Foundations and Trends R _in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.

13. H. Motoda , H. Liu “Feature selection, extraction and construction,” Communication of IICM (Institute of Information and Computing Machinery), Taiwan, vol. 5, pp. 67–72, 2002.

14. B. Tran, S. Picek, B. Xue “Automatic feature construction for network intrusion detection,” in Asia-Pacific Conference on Simulated Evolution and Learning. Springer, 2017, pp. 569–580.

15. M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, K. Kim “Deep abstraction and weighted feature selection for Wi-Fi impersonation detection,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 3, pp. 621–636, 2018.

16. T. Hamed, J. B. Ernst, S. C. Kremer “A survey and taxonomy on data and pre-processing techniques of intrusion detection systems,” in Computer and Network Security Essentials. Springer, 2018, pp. 113–134.

17. Демидова Л.А., Пилькін А.Н. Методи і алгоритми прийняття рішень в задачах багатокритеріального аналізу. Телеком, 2007. 232

18.Демидова Л.А., Кіраковский В.В., Пилькін А.Н. Прийняття рішень в умовах невизначеності. Телеком, 2012. 288 с.

19. МОРАДА М., Зелкернін М. Стаття в інформаційному порталі університету Queen's University [Електронний ресурс]. - Кінгстон: http://queensu.ca. - «A Neural Network Based System for Intrusion Detection and Classification of Attacks» .Режим доступу: http://research.cs.queensu.ca/~moradi/148-04-MM-MZ.pdf+ 01.12.2013.

20. Кліонский Д.М., Большев А.К., Геппенер В.В. Стаття в Національному дослідницькому ядерном університеті «МІФІ» [Електронний ресурс]. - Москва: http://library.mephi.ru. - «Застосування штучних нейронних мереж в мережевих технологіях і інтелектуальному аналізі даних». Режим доступу: http://library.mephi.ru/data/scientific-sessions/2011/neiroinform/ch3/2-1-6.doc;

21. Жигулін П.В., Подворчан Д.Е. Стаття в інформаційному порталі університету ТУСУР [Електроний ресурс]. - Томськ: www.tusur.ru. - «Аналіз мережевого трафіку за допомогою нейронних мереж». Режим доступу

http://storage.tusur.ru/files/425/КИБЭВС1005_Жигулин_П.В.__Подворчан_Д.Э.pdf.

22. Набір даних KDD Cup тисяча дев'ятсот дев'яносто дев'ять Data, [Електронний ресурс]. - http://kdd.ics.uci.edu «KDD Cup тисячу дев'ятсот дев'яносто дев'ять Data». Режим доступу: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

23. Хафизов А. Ф. Дисертація в електронній бібліотеці РДБ [Електронний ресурс]. - Москва: http://dlib.rsl.ru. - «Нейросетевая система виявлення атак на WWW-Сервер». Режим доступу: http://dlib.rsl.ru/01002663345;

24. Шанмагавадіва Р., Нагаражан Н. INDIAN JOURNAL OF COMPUTER SCIENCE AND ENGINEERING [Електронний ресурс]. - Таміл Наду, Індія: www.ijcse.com, 2011. - «network intrusion detection system using fuzzy logic». Режим доступу: http://www.ijcse.com/docs/IJCSE11-02-01-034.pdf+01.12.2013.

25. Слепович І.І., Ірматов П. В., Комарова М. С., Бежін А. А. Виявлення DDoS атак нечіткої нейронної мережею // Изв. Сарат. ун-ту. Нов. сер. Сер. Математика. Механіка. Інформатика. 2009. № 9: 3. С. 84-89.

26. Кашаев Т. Р. Дисертація в електронній бібліотеці РДБ [Електронний ресурс]. - Москва: http: //Dlib.rsl.ru. - «Алгоритми активного аудиту інформаційної системи на основі технологій штучних імунних систем». Режим доступу: http://dlib.rsl.ru/01003447155.

27. Свечников Л. А. Дисертація в електронній бібліотека РДБ [Електронний ресурс]. - Москва: http://dlib.rsl.ru. - «Система виявлення атак на інформаційну систему з використанням динамічних моделей на основі нечітких когнітивних карт». Режим доступу: http://dlib.rsl.ru/01004730956.

28. Крошілін А.В., Крошіліна С.В. Огляд способів формування когнітивних карт в системах підтримки прийняття рішень // Програмні інформаційні системи. Рязань: РГРТУ. 2011. С.20-24.

29. Ануп Гоял, Четан Кумар. Стаття в інформаційному порталі університету Northwestern University [Електронний ресурс]. - Іллінойс: http://www.cs.northwestern.edu. - «GA-NIDS: A Genetic Algorithm based Network Intrusion Detection System». Режим доступу: http://www.cs.northwestern.edu/~ago210/ganids/GANIDS.pdf 01.12.2013.

30. Вінсент, П .; Ларошель, Г .; Бенджо, Ю .; Манзагол, Пенсільванія Видобування та складання надійних функцій за допомогою шумозаглушаючих автокодерів. У матеріалах 25-ї Міжнародної конференції з машинного навчання, Гельсінкі, Фінляндія, 5–9 липня 2008 р., С. 1096–1103.

31. Вінсент, П .; Ларошель, Г .; Lajoie, I .; Бенджо, Ю .; Манзагол, Пенсільванія Автоматичні кодектори шуму: вивчення корисних уявлень у глибокій мережі з локальним критерієм деноізування Дж. Мах. Навчіться. Рез. 2010, 11, 3371–3408.

32. Ден, Дж .; Чжан, З .; Марчі, Е .; Шуллер, Б. Розріджене навчання переносу функцій на основі автокодера для розпізнавання емоцій мовлення. У матеріалах конференції Асоціації Хумена 2013 року з питань афективних обчислень та інтелектуальної взаємодії, Женева, Швейцарія, 2–5 вересня 2013 р .; С. 511-516 Хінтон, Дж. Е. Практичний посібник з підготовки машин Больцмана з обмеженими можливостями у нейронних мережах: хитрощі ,торгівля. Спрінгер: Берлін, Німеччина, 2012; С. 599–619.

33. Хінтон, Джорджія; Осіндеро, С .; Teh, YW Алгоритм швидкого навчання для мереж глибоких переконань. Нейронні обчислення. 2006, 18, 1527–1554.

34. Буро, Іл; Кун, Ю.Л .; Ранзато, М.А. Розріджене вивчення особливостей для мереж глибоких переконань. У матеріалах 21-ї щорічної конференції з нейронних систем обробки інформації, Ванкувер, Британська Корея, Канада, 8–10 грудня 2008 р .; С. 1185–1192.

35. Чжао, Г.; Чжан, С.; Чжен, Л. Виявлення вторгнень за допомогою мережі глибоких вірувань та імовірнісної нейронної мережі. У матеріалах Міжнародної конференції IEEE 2017 року з обчислювальної науки та техніки (CSE) та Міжнародної конференції IEEE з питань вбудованих та повсюдних обчислень (EUC), Гуанчжоу, Китай, 21–24 липня 2017 р .; Том 1, с. 639–642.

36. Альравашде, К .; Перді, К. На шляху до онлайнової системи виявлення вторгнень, заснованої на глибокому навчанні. У матеріалах 15-ї Міжнародної конференції IEEE з машинного навчання та додатків (ICMLA), Анахайм, Каліфорнія, США, 18–20 грудня 2016 р .; С. 195–200.

37. Ян, Ю .; Чжен, К.; Ву, С .; Ніу, Х .; Ян, Ю. Створення ефективної системи

виявлення вторгнень із використанням модифікованого алгоритму кластеризації піків щільності та мереж глибоких переконань. Заяв. Наук. 2019, 9, 238.

38. Шариф Разавіан, А .; Азізпур, Х .; Салліван, Дж .; Карлссон, С. CNN пропонує готові: вражаючу базову лінію для визнання. У матеріалах конференції IEEE з питань семінарів з комп'ютерного зору та розпізнавання зразків, Коламбус, Огайо, США, 23–28 червня 2014 р .; С. 806–813.

39. Крижевський, А .; Суцкевер, І .; Хінтон, Г. Е. Класифікація Imagenet із глибокими згортковими нейронними мережами. У матеріалах 26-ї щорічної конференції з нейронних систем обробки інформації, озеро Тахо, Невада, США, 3–6 грудня 2012 р .; С. 1097–1105.

40. Лоуренс, С.; Giles, CL; Цой, змінного струму; Назад, AD Розпізнавання обличчя: згортковий нейромережевий підхід. IEEE Trans. Нейронна мережа. 1997, 8, 98–113.

41. Могили, А .; Мохамед, АР; Хінтон, Г. Розпізнавання мови за допомогою глибоких рекурентних нейронних мереж. У матеріалах Міжнародної конференції IEEE 2013 року з акустики, обробки мови та сигналів, Ванкувер, Британська Колумбія, Канада, 26–31 травня 2013 р .; С. 6645–6649.

42. Могили, А .; Джейтлі, Н. Назустріч наскрізному розпізнаванню мови за допомогою періодичних нейронних мереж. У матеріалах Міжнародної конференції з машинного навчання, Пекін, Китай, 21–26 червня 2014 р .; С. 1764–1772.

43. Суцкевер, І .; Віньялс, О .; Ле, QV Послідовність навчання послідовності за допомогою нейронних мереж. У матеріалах щорічної конференції з нейронних систем обробки інформації 2014 р., Монреаль, QC, Канада, 8–13 грудня 2014 р .; С. 3104–3112.

44. Гохрейтер, С.; Шмідхубер, Дж. Довга короткочасна пам'ять. Нейронні обчислення. 1997, 9, 1735–1780.

45. Чунг, Дж .; Гюльчере, С .; Чо, К.; Бенджо, Ю. Емпірична оцінка керованих рекурентних нейронних мереж при моделюванні послідовностей. arXiv 2014, arXiv: 1412.3555.

46. Шустер, М .; Палівал, К. К. Двонаправлені рекурентні нейронні мережі. IEEE Trans. Процес сигналу. 1997, 45, 2673–2681.

47. Рібейро, МТ; Сінгх, С.; Гострін, К. Чому я маю вам довіряти ?: Пояснення прогнозів будь-якого класифікатора. У матеріалах 22-ї Міжнародної конференції ACM SIGKDD з питань виявлення знань та видобутку даних, Сан-Франциско, Каліфорнія, США, 13–17 серпня 2016 р .; С. 1135–1144.

48. Лундберг, С.М .; Лі, С.І. Єдиний підхід до інтерпретації модельних прогнозів. У матеріалах щорічної конференції з нейронних систем обробки інформації 2017, Лонг-Біч, Каліфорнія, США, 4–9 грудня 2017 р .; С. 4765–4774.

49. Лі, Дж .; Монро, штат Вікторія; Юрафський, Д. Розуміння нейронних мереж шляхом стирання репрезентацій. arXiv 2016, arXiv: 1612.08220.

50. Фонг, РК; Ведальді, А. Інтерпретовані пояснення чорних скриньок значущим збуренням. У матеріалах Міжнародної конференції IEEE з комп'ютерного зору, Венеція, Італія, 22–29 жовтня 2017 р .; С. 3429–3437.

51. Потлурі, С .; Ахмед, С .; Дідріх, К. Згорткові нейронні мережі для багатокласної системи виявлення вторгнень. У видобутку розвідки та дослідження знань; Спрінгер: Чам, Швейцарія, 2018; С. 225–238.

52. Чжан, Б .; Ю, Ю .; Лі, Дж. Виявлення вторгнень у мережу на основі складеного розрідженого автокодера та методу двійкового дерева. У матеріалах Міжнародної конференції IEEE 2018 року з питань комунікаційних майстер-класів (ICC Workshop), Канзас-Сіті, Міссурі, США, 20–24 травня 2018 р .; С. 1–6.

53. Чжан, Х .; Ю, Х .; Рен, П .; Лоо, С .; Мін, Г. Глибоке змагальне навчання у виявленні вторгнень: вдосконалена система збільшення даних. arXiv 2019, arXiv: 1901.07949.

54. Маккалох Дж., Питтс У. Логические исчисления идей, относящихся к нервной деятельности.// Автоматы. М.: ИЛ, 1956. https://works.doklad.ru/view/9On-gYSK_X8.html

55. Минский М., Пейперт С. Персептроны. М.: МИР, 1971. С. 261.

56. Розенблат Ф. Аналитические методы изучения нейронных сетей.//

Зарубежная радиоэлектроника. - 1965 - N 5 - с. 40-50.

57. Rosenblatt F. The peseptron: a probabilistic model for information storage and organization in the brain//Psychol. Rev.

58. Rosenblatt F. Principles of neurodynamics. Spartan., Washington, D. C., 1962.

59. Иванченко А. Г. Персептрон - системы распознавания образов.// К.: Наукова думка, 1972.

60. Amari S. Field theory of self-organizing neural networks//IEEE Trans. Syst., Man, Cybern. 1983. V. 13. P. 741

61. Carpenter G. A., Grossberg S. A massively parallel architecture for a self-organizing neural pattern recognition machine.//Comput. Vision Graphics Image Process. 1986. V. 37. p. 54-115.

62. Cohen M. A., Grossberg S. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks//IEEE Trans. Syst., Man, Cybern. 1983. V. 13. N 5.

63. Тэнк Д. У., Хопфилд Д. Д. Коллективные вычисления в нейроноподобных электронных схемах.//В мире науки. 1988. N 2. С. 44-53.

64. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities.//Proc. Natl. Acad. Sci. 1984. V. 9. p. 147-169.

65. Hopfield J. J., Tank D. W. Neural computation of decision in optimization problems.//Biol. Cybernet. 1985. V. 52. p.

66. Hopfield J. J., Feinstein D. I., Palmer F. G. Unlearning has a stabilizing effect in collective memories//Nature. 1983. V. 304. P. 141-152.

67. Hopfield J. J., Tank D. W. Neural computation of decision in optimization problems//Biol. Cybernet. 1985. V. 52. P. 141-152.

68. Абу-Мустафа Я. С., Псалтис Д. Оптические нейронно-сетевые компьютеры//В мире науки, 1987. N 5. С. 42-50.

69. Athale R., Stirk C. W. Compact architectures for adaptive neuraal nets//Ibid. 1989. V. 28. N 4.

70. Hebb D. O. The organization of behaviour. N. Y.: Wiley, 1949.

71. Aarts E. H. L., Korst J. H. M. Boltzmann machines and their applications//Lect.

Notes Comput. Sci. 1987. V. 258. P. 34-50

72. Aarts E. H. L., Korst J. H. M. Boltzmann machines for travelling salesman problem//European J. Oper. Res. 1989. V. 39. P. 79-95.

73. Abu-Mostafa Y. S., Jaques J. N. St. Information capacity of the Hopfield model//IEEE Trans. Inform. Theory. 1985. V. 31. P.461.

74. Ackley D. H., Hinton G. E., Sejnowski T. J. A learning algorithm for Bolzmann machines//Cognit. Sci. 1985. V. 9. N 1. P. 147-169.

75. Ackley D. H., Hinton G. E., Sejnowski T. J. A learning algorithm for Bolzmann machines//Cognit. Sci. 1985. V. 9. N 1. P. 147-169.

76. Куссуль В. М., Байдык Т. Н. Разработка архитектуры нейроподобной сети для распознавания формы объектов на изображении.//Автоматика - 1990 - N 5 - с. 56-61.

77. Трикоз Д. В. Нейронные сети: как это делается? // Компьютеры + программы - 1993 - N 4(5) с. 14-20.

78. Bardcev S. I., Okhonin V. A. The algorithm of dual functioning (back-propagation) : general approach, vesions and applications. Krasnojarsk: Inst. of biophysics SB AS USSA - 1989.

79. Computing with neural circuits: a model.//Science, 1986. V. 233. p. 625-633.

80. Kuzewski Robert M., Myers Michael H., Grawford William J. Exploration of fourword error propagation as self organization structure.//IEEE Ist. Int. Conf. Neural Networks, San Diego, Calif., June 21-24,1987. V. 2. - San Diego, Calif., 1987.

81. Rumelhart B. E., Minton G. E., Williams R. J. Learning representations by back propagating error.// Wature, 1986. V. 323. p. 1016-1028.

82. Takefuji D. Y. A new model of neural networks for error correction.//Proc. 9th Annu Conf. IEEE Eng. Med. and Biol. Soc., Boston, Mass., Nov. 13-16,1987. V. 3, New York, N. Y., 1987 - p. 1709-1710.

83. Стаття «Глибинне навчання: основні поняття.»з матеріалів сайта. Режим доступу:https://www.osp.ru/articles/2019/0804/13055056

84. Люгер Ф. Искусственный интелект: стратегии и методы решения сложных проблем, 4-е издание.: Пер. с англ. – М.: Вильямс, 2003.

85. Терейковська Л. А. та Терейковський І. А. Використовуючи досвід у розробці моделі нейронної мережі для розпізнавання фонем у голосовому сигналі [Текст], матеріали II Міжнародної науково-практичної конференції Інформаційні та телекомунікаційні технології: освіта, наука та практика, Алмати, Казахстан , 2015, с. 258–261.

86. Ємельянова Ю.Г. Аналіз проблеми та перспективи створення інтелектуальної системи obnarugenia i predotvrashchenia setevykh atak na oblachnie vichisleniya. [Аналіз питань та можливостей розвитку інтелектуальної системи виявлення та запобігання мережевим атакам на хмарні обчислення]. Програмні системи: Теорія та застосування, 2011; 4: 17-31.

87. Айтчанов Б.Х. та Бапієв І.М. Разработка процесса обработки определения ожидаемого выходного сигнала нейросетевой модели нейросетевой модели распознавания кибератак: Автореф. Міжнародний журнал прикладних та фундаментальних досліджень, 2017;

88. Корченко А., Терейковський І., Карпінський Н. і Тинімбаєв С. Нейросетеві моделі, методи дослідження оцінки параметрів безпеки Інтернет-орієнтованих інформаційних систем, [Моделі нейронних мереж, методи та інструменти оцінки параметрів безпеки інформаційних систем інтерменту]. Київ: ТОВ НашФормат, 2016

89. Ахметов Б.Б., Корченко А.Г., Терейковський І.А., Алібієва Ж.М. та Бапієв І. М. Параметри оцінки ефективності нейронних мереж розпізнавання кібератак на мережевих ресурсах інформаційних систем. Звіти Національної академії наук Республіки Казахстан, 2017; 2: 28–37.

90. Бапієв І.М., Ахметов Б.С., Корченко А.Г. і Терейковський І.А. та захист інформації, Київ, Україна, 2016, с. 21–24.

91. Гришин А. В. Технологія нейронних мереж у задачах виявлення комп’ютерних атак. Інформаційні технології та комп’ютерні системи, 2011;

92. Мустафаєв А.Г.) Нейросетева система обробки комп'ютерних даних на основі аналізу сетевого трафіка. [Нейромережева система виявлення атак на основі мережевого трафіку]. 2016;

ДОДАТОК А

(обов’язковий)

Перелік наукових праць

УДК 004.832.2

В.Ю. ТІТОВА, В.С. ОРЛЕНКО, І.М. ШЕВЧУК, В.С. ДАЦЕНКО

Хмельницький національний університет

ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ РІШЕНЬ В СИСТЕМАХ ЗАХИСТУ ІНФОРМАЦІЇ

В даній статті розглянуто відомі методи оцінювання ефективності рішень. Проаналізовано можливість їх використання для оцінювання ефективності рішень, що приймаються системами захисту інформації стосовно класифікації та визначення загроз. На основі визначених переваг та недоліків існуючих методів запропоновано власний метод оцінювання ефективності рішень, який дозволяє підвищити відсоток прийнятих правильних рішень та оптимізувати ефективність системи захисту інформації в цілому.

Ключові слова: системи захисту інформації, багатокритеріальна оптимізація, метод аналізу ієрархій, метод Парето.

VERA YURIIVNA TITOVA, VIKTORIIA SERHIIVNA ORLENKO, ILLIA MYKOLAIOVYCH SHEVCHUK, VLADYSLAV SERHIIOVYCH DATSENKO

Khmelnytskyi National University

EVALUATION OF DECISIONS EFFICIENCY IN INFORMATION SECURITY SYSTEMS

The functioning of most information security systems is reduced to the recognition of many active processes, their classification in order to identify malicious and dangerous processes and make decisions to respond to them. The decision-making process is based on taking into account a large number of conflicting requirements and evaluating decision options according to many criteria. The inconsistency of the characteristics of the processes, the ambiguity of the evaluation of the process, the incompleteness of the information obtained greatly complicate the final decision and significantly affect its quality.

To increase the efficiency of the final decision, it is necessary to develop a method of multi-objective optimization of decisions, which is why this work was devoted.

To evaluate the efficiency of decisions in information security systems, the method was proposed that includes the advantages of the analytic hierarchy process and the Pareto efficiency. The proposed method provides increasing the percentage of identified correct decisions and has the following advantages: the result is always a single and effective decision; the possibility of compensation of values of partial criteria is eliminated. The method was implemented and tested in the subsystem for evaluating the decision efficiency of intrusion detection system based on in-depth learning networks and was used as the tools of multi-objective optimization of decisions for computer systems protection system.

Keywords: information security system, multi-objective optimization, analytic hierarchy process, Pareto efficiency.

Вступ. Забезпечення захисту інформації в комп’ютерних системах є однією з ключових проблем сьогодення. При цьому треба враховувати, що функціонування більшості систем захисту інформації, по факту, зводиться до розпізнавання множини активних в комп’ютері процесів, їх класифікації з метою визначення шкідливих та небезпечних процесів та прийняття рішень щодо їх блокування або ігнорування. Причому процес прийняття рішень ґрунтується на врахуванні великої кількості суперечливих вимог і оцінюванні варіантів рішень за багатьма критеріями. Суперечливість характеристик процесів, неоднозначність оцінювання процесу, неповнота отриманої інформації значною мірою ускладнюють прийняття остаточного рішення і суттєво впливають на його якість.

Для підвищення ефективності остаточного рішення необхідно ввести в структуру систем захисту інформації модуль, який забезпечить можливість вибору кращої альтернативи за допомогою методу оцінювання ефективності рішень, тобто буде реалізовувати багатокритеріальну оптимізацію рішень. Розроблення методу зазначеної багатокритеріальної оптимізації рішень і є метою даної роботи.

Характеристика предметної області. На сьогоднішній день методи вирішення задач багатокритеріальної оптимізації розділяють на два класи [1], [2]:

- методи, що дозволяють виділити деяку множину прийнятних варіантів;
- методи пошуку єдиного ефективного рішення.

До методів першого класу, наприклад, належить метод Парето. Але подібні методи не можуть бути використані в системах захисту інформації, оскільки головною метою захисту пошук єдиного ефективного рішення, яке б дозволило максимізувати захищеність та мінімізувати загрози інформації.

До методів другого класу, наприклад, відносяться методи з використанням узагальнюючого критерію (адитивний, мультиплікативний, максимінний) [1] та аналітична ієрархічна процедура (Analytic Hierarchy Process) Сааті або метод попарних порівнянь [2].

Перевагою перших методів є те, що завжди вдається визначити єдиний оптимальний варіант рішення. До недоліків відносять суб'єктивізм у визначенні вагових коефіцієнтів критеріїв та компенсацію значень часткових критеріїв [1], [3]. Останній недолік може призвести до того, що рішення, обране за найкращим сумарним результатом, має не найкращі результати за критеріями з найбільшими ваговими коефіцієнтами, які компенсуються найкращими результатами за критеріями з меншими ваговими коефіцієнтами. Як результат, обране рішення буде не самим ефективним, а це, в свою чергу, може призвести до ігнорування небезпечного або шкідливого процесу, реалізації сценарію його загрозу та, як наслідок, порушення конфіденційності, цілісності або доступності інформації.

Вищезазначені недоліки фактично ліквідовані аналітичною ієрархічною процедурою Сааті, але ця процедура має ряд недоліків, а саме: недосконалість шкали переваг та отримання результатів типу «критерій К1 важливіший за критерій К2» [2], не завжди враховуючи наскільки саме важливіший. Сааті пропонує таку шкалу переваг:

1 – рівноцінність;

3 – помірна перевага;

5 – велика перевага;

7 – дуже велика перевага;

9 – найвища перевага.

Розглянемо ситуацію, коли критерій К1 має дуже велику перевагу над критерієм К2, критерій К2 має дуже велику перевагу над критерієм К3. Що можна сказати про перевагу критерію К1 над критерієм К3?

Логічно зробити висновки, що критерій К1 має перевагу над К3 в 49 разів (7 помножити на 7), але цей висновок не входить у рамки даної шкали. Єдиним рішенням залишається зробити висновок, що критерій К1 має найвищу перевагу над критерієм К3, і в подальшому використовувати градацію шкали «9». Проте, при оцінювання ефективності рішень в системах захисту інформації через велику кількість прямих і зворотних зв'язків між характеристиками загроз, розглянути у [4] перевага кожного критерію над іншими має дуже велике значення, тому цей метод не може бути використаний.

Тому, для оцінювання ефективності рішень в системах захисту інформації використаємо переваги двох вищезазначених методів.

Метод оцінювання ефективності рішень в системах захисту інформації. Як вже зазначалося вище, в

ході свого функціонування система захисту інформації здійснює вибір рішення зі скінченої множини можливих рішень $R = \{r_j\}, j = \overline{1,q}$. Ці рішення є реакціями на діяльність одного з активних процесів *p* з множини усіх процесів $P = \{p_t\}, t = \overline{1,w}$. Щоб прийняти рішення r_j для процесу *p*, система має проаналізувати характеристики кожного процесу $A = \{a_{jm}\}, j = \overline{1,q}, m = \overline{1,n}$ для кожного критерію, обраного з відповідної множини $\{k_m\}, m = \overline{1,n}$, де *n* – максимальна можлива кількість критеріїв, та визначити для кожного рішення r_j його ефективність.

Причому треба враховувати, що в один момент часу можуть мати місце як один, так і кілька класів загроз, а характеристики можуть змінюватися в процесі прийняття рішення. І з їх зміною один клас загрози може перейти в інший або корелювати з ним [4]. Отже, формальний опис моделі задачі оцінювання ефективності рішень має такий вигляд:

$$E_{r_j} = M(A_p, k_p), \tag{1}$$

де E_{r_j} – ефективність рішення r_j, *M* – це метод, за яким ведеться пошук ефективного рішення; A_p – характеристики процесу *p*; k_p – множина критеріїв для оцінювання характеристик процесу *p*, згідно з якими оцінюється ефективність можливих рішень.

На основі наведених множин сформуємо три матриці. В першу (*A*) заносяться дані відношень критеріїв, в другу (*B*) і третю (*C*) значення, які відображають характеристики для кожного процесу за кожним з критеріїв.

Значення критеріїв обчислюються за допомогою методу попарних порівнянь з використанням шкали переваг Сааті. Попарно порівнюється лише один окремий критерій з усіма іншими. У результаті визначається перевага критерію k_i. Після цього дані заносяться у перший рядок матриці *A* (2). Всі подальші переваги обчислюються за математичними розрахунками. Таким чином, можна уникнути обмежень, що накладаються градацією шкали переваг.

$$A = \begin{bmatrix} 1 & kx_1 / kx_2 & \dots & kx_1 / kx_n \\ kx_2 / kx_1 & 1 & \dots & kx_2 / kx_n \\ \dots & \dots & \dots & \dots \\ kx_n / kx_1 & kx_n / kx_2 & \dots & 1 \end{bmatrix}, \tag{2}$$

де $kx_1 .. kx_n$ – відповідні критерії, *n* – максимальна кількість критеріїв, за якими виконується оцінювання.

Далі значення критеріїв нормуються таким чином, щоб їх сума дорівнювала одиниці, тобто визначається ваговий коефіцієнт кожного критерію. Для цього вони обчислюються за такими формулами:

$$\tilde{kx_j} = \sum_{i=1}^{n} kx_{ij}, j = \overline{1,m}$$

$$v_i = \tilde{kx_i} / \sum_{j=1}^{n} \tilde{kx_j}, i = \overline{1,m}$$

де $v_1 .. v_n$ – вагові коефіцієнти відповідних критеріїв.

В матрицю *B* (3) заносяться характеристики процесів за кожним обраним критерієм.

$$B = \begin{bmatrix} ax_{11} & ax_{12} & \dots & ax_{1n} \\ ax_{21} & ax_{22} & \dots & ax_{2_n} \\ \dots & \dots & \dots & \dots \\ ax_{f1} & ax_{f2} & \dots & ax_{f_n} \end{bmatrix}, \tag{3}$$

де $ax_{11}..ax_{fn}$ – значення відповідних характеристик процесів за відповідними критеріями, f – максимальна кількість рішень, для яких виконується оцінювання.

Далі дані нормуються таким чином, щоб сума значень у кожному стовпчику дорівнювала одиниці, і матриця B перетворюється в матрицю $B^{\sim}$.

$$B^{\sim} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{f1} & a_{f2} & \dots & a_{fn} \end{bmatrix}, \quad (4)$$

де $a_{11}..a_{fn}$ – нормовані характеристики процесів.

В матрицю C (5) також заносяться характеристики процесів за кожним обраним критерієм.

$$C = \begin{bmatrix} ax_{11} & ax_{12} & \dots & ax_{1n} \\ ax_{21} & ax_{22} & \dots & ax_{2n} \\ \dots & \dots & \dots & \dots \\ ax_{f1} & ax_{f2} & \dots & ax_{fn} \end{bmatrix}, \quad (5)$$

Перетворення над нею виконуються наступним чином. Якщо найкращим результатом для j-го критерію є максимальне значення наслідку рішення, то $n_{ij}^{o} = n_{ij} / n_{\max j}$, де n_{ij}^{o} – нормоване значення відповідного наслідку, $n_{\max j}$ – максимальне значення наслідку в j-му стовпці. Якщо для j-го критерію найкращим результатом є мінімальне значення наслідку рішення, то $n_{ij}^{o} = n_{ij} / n_{\min j}$, де $n_{\min j}$ – мінімальне значення наслідку в j-му стовпці. Матриця $C^{\sim}$ буде мати вигляд:

$$C^{\sim} = \begin{bmatrix} a^{o}_{11} & a^{o}_{12} & \dots & a^{o}_{1n} \\ a^{o}_{21} & a^{o}_{22} & \dots & a^{o}_{2n} \\ \dots & \dots & \dots & \dots \\ a^{o}_{f1} & a^{o}_{f2} & \dots & a^{o}_{fn} \end{bmatrix}, \quad (6)$$

де $a^{o}_{11}..a^{o}_{fn}$ – нормовані значення відповідних наслідків.

Після формування усіх матриць для кожного рішення обчислюється його ефективність за формулою:

$$E_{r_j} = \sum_{i=1}^{n} k_i * n_{ij} * n_{ij}^{o} \quad (7)$$

Висновки.

У статті було розглянуто задачу оцінювання ефективності рішень систем захисту інформації. Аналіз зазначеної задачі показав, що вона є задачею багатокритеріальної оптимізації і потребує для свого вирішення задіювання відповідних методів.

Виявлено, що існуючі методи оцінювання ефективності рішень не задовольняють вирішенню даної задачі, а тому не можуть бути використані. Було запропоновано удосконалений метод, який базується на

використанні матриці відношення критеріїв та врахуванні наслідків рішень. Запропонований метод оцінювання ефективності рішень дозволяє підвищити відсоток визначених правильних рішень та має наступні переваги:

- результатом завжди є єдине та ефективне рішення;
- усунена можливість компенсації значень часткових критеріїв.

Зазначений метод був реалізований та апробований у підсистемі оцінювання ефективності рішень системи виявлення вторгнень на базі мереж глибинного навчання та як засіб багатокритеріальної оптимізації рішень для системи захисту комп'ютерних систем.

Література

1. Штойер Р. Многокритериальная оптимизация. Теория вычислений и приложения/ Р. Штойер. – М.: Наука, 1992. – 204 с.

2. Саати Т. Принятие решений. Метод анализа иерархий/ Т. Саати. – М.: Радио и Связь, 1993. – 320 с.

3. Кини Р.Л. Принятие решений при многих критериях: предпочтения и замещения/ Р.Л. Кини, Х. Райфа. – М.: Радио и связь, 1981. – 560 с.

4. Тітова В.Ю. Класифікація моделей загроз в комп'ютерних системах/ В.Ю. Тітова, Ю.П. Кльоц, С.О. Савчук. – Вісник Хмельницького національного університету. – №2, 2020 (283). – С. 201-204.

Інформаційна модель захисту інформації.

Даценко В.С., Тітова В.Ю., Шевчук І.М.
Хмельницький національний університет

Розуміючи інформаційну безпеку як «стан захищеності інформаційного середовища суспільства, що забезпечує її формування, використання і розвиток в інтересах громадян та організацій», правомірно визначити загрози безпеки інформації, джерела цих загроз, способи їх реалізації та цілі, а також інші умови і дії, що порушують безпеку [1]. При цьому, природно, слід розглядати і заходи захисту інформації від неправомірних дій, що призводять до заподіяння шкоди.

Практика показала, що для аналізу такого значного набору джерел, об'єктів і дій доцільно використовувати методи моделювання. При цьому слід враховувати, що модель не копіює оригінал, а є простішою. При цьому, модель повинна бути досить загальною, щоб описувати реальні дії з урахуванням їх складності [2].

Можна запропонувати компоненти моделі захисту інформації на першому (інформаційному) рівні декомпозиції. На нашу думку, такими компонентами інформаційної моделі можуть бути:

- об'єкти загроз;
- загрози;
- джерела загроз;
- цілі загроз з боку зловмисників;
- джерела інформації;
- способи неправомірного оволодіння інформацією (способи доступу);
- напрямки захисту інформації;
- способи захисту інформації;
- засоби захисту інформації.

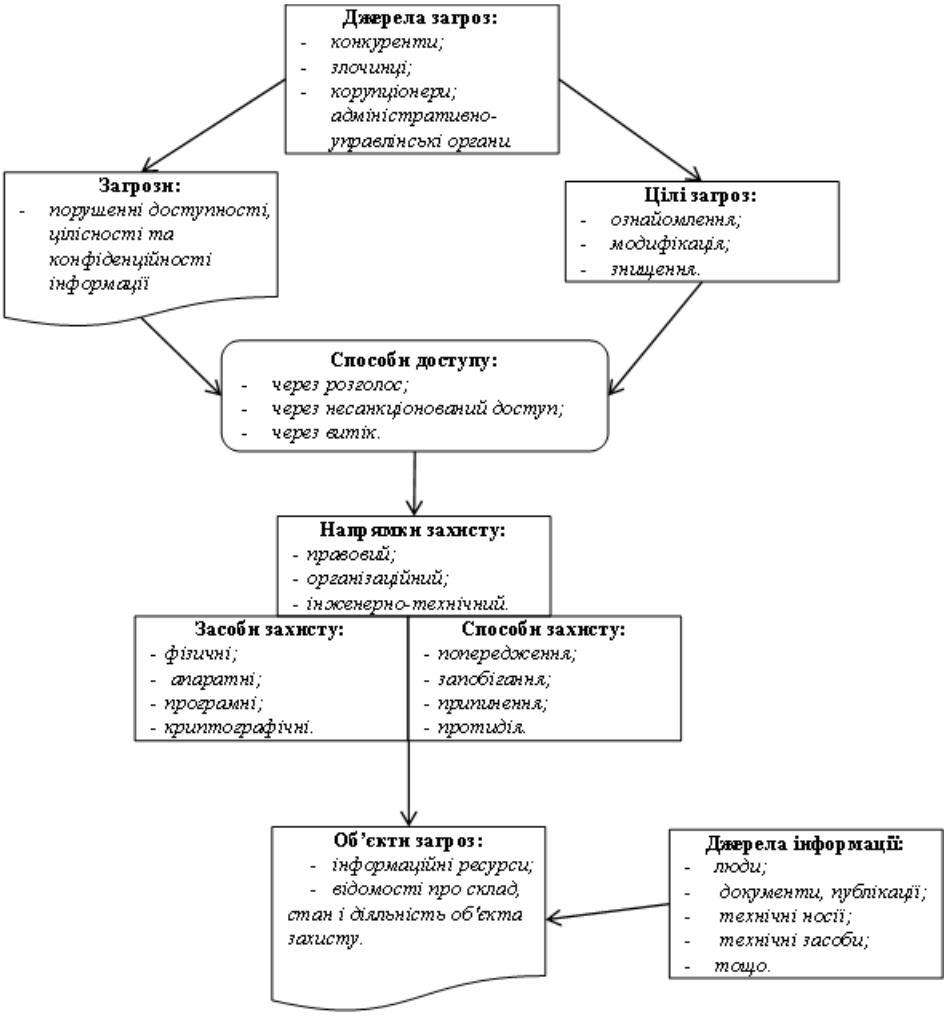


Рисунок 1 – Інформаційна модель захисту інформації

Об'єктами загроз інформаційної безпеки виступають відомості про склад, стан і діяльність об'єкта захисту (персоналу, матеріальних і фінансових цінностей, інформаційних ресурсів), тощо.

Загрози інформації виражаються в порушенні її доступності, цілісності і конфіденційності.

Джерелами загроз виступають конкуренти, злочинці, корупціонери, адміністративно-управлінські органи, тощо.

Джерела загроз переслідують при цьому наступні цілі: ознайомлення з відомостями, їх модифікація в корисливих цілях і знищення для нанесення прямих матеріальних збитків.

Неправомірне заволодіння відомостями можливо за рахунок їх розголошення джерелами інформації, за рахунок витоку через технічні засоби і за рахунок несанкціонованого доступу до відомостей. Джерелами інформації є люди, документи, публікації, технічні носії інформації, технічні засоби забезпечення виробничої та трудової діяльності, продукція і відходи виробництва.

Основними напрямками захисту інформації є правовий, організаційний та інженерно-технічний захист інформації, як показники комплексного підходу до забезпечення інформаційної безпеки.

Засобами захисту інформації є фізичні засоби, апаратні засоби, програмні засоби та криптографічні методи. Останні можуть бути реалізовані як апаратно, програмно, так і змішано-програмно-апаратними засобами.

В якості засобів захисту виступають всілякі заходи, шляхи, способи і дії, що забезпечують попередження протиправних дій, їх запобігання, припинення та протидія несанкціонованому доступу.

В узагальненому вигляді розглянуті компоненти у вигляді інформаційної моделі безпеки інформації наведені на наступній схемі (рисунок 1).

Співставлення об'єкта (фірма, організація) і суб'єкта (конкурент, зловмисник) в інформаційному процесі з протилежними інтересами можна розглядати з позиції активності, яка призводить до оволодіння інформацією.
У цьому випадку можливі такі ситуації:

- власник (джерело) не приймає ніяких заходів до збереження інформації, що дозволяє зловмисникові легко отримати цікаві для нього відомості;
- джерело інформації суворо дотримується заходів інформаційної безпеки, тоді зловмисникові доводиться докладати значних зусиль до здійснення доступу до потрібних йому відомостей, використовуючи для цього всю сукупність способів несанкціонованого проникнення;
- проміжна ситуація - це витік інформації по технічним каналам, при якій джерело ще не знає про це (інакше він прийняв би заходи захисту), а зловмисник легко, без особливих зусиль може їх використовувати в своїх інтересах.

Отже, на основі вищевикладеного можна зробити наступні висновки:

1. Інформація - це ресурс. Втрата інформації приносить моральні чи матеріальні збитки.

2. Умови, що сприяють неправомірному оволодінню інформацією, зводяться до її розголошенню, витоку і несанкціонованого доступу до її джерел.
3. У сучасних умовах безпека інформаційних ресурсів може бути забезпечена тільки системою захисту інформації, яка буде протидіяти загрозам через блокування неправомірних способів доступу та охоплювати усю множину існуючих способів за засобів захисту інформації.

Перелік посилань

1. Теоретичні засади поняття інформаційної безпеки та класифікація загроз системі інформаційного захисту/ О. В. Черевко. // Ефективна економіка. – 2014. – №5. – Режим доступу: http://nbuv.gov.ua/UJRN/efek_2014_5_103

2. Інформаційна безпека. Навчальний посібник. Ч.1/С.В. Кавун, В.В. Носов, О.В. Мажай. – Харків: Вид. ХНЕУ, 2008. – 352 с.

3. Основні поняття. НД ТЗІ 1.1-003-99: Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. – Київ: Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України. – 1999. – 30 с.

к.т.н., доц. Тітова В.Ю. (ХмНУ)

д.т.н., проф. Андрощук О.С.(ХмНУ)

Даценко В.С.(ХмНУ)

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У ВИЯВЛЕННІ ВТОРГНЕНЬ

В даний час в різних галузях науки і техніки підвищується інтерес до використання штучних нейронних мереж. На даний момент не існує альтернативи до даної системи на основі правил. Мережеві атаки постійно змінюються, тому постійно мінливий характер мережевих атак вимагає гнучку захисну систему, яка здатна аналізувати величезну кількість мережевого трафіку за методом, який менш структурований ніж той, що заснований на побудові певних правил.

Перевагами у використанні нейронної мережі у виявленні вторгнень є:

1. Гнучкість - яку надає ця мережа.

2. Швидкість - властива нейронним мережам.

3. Навчання - нейронна мережа може бути навчена розпізнавати відомі підозрілі події з високим ступенем точності.

Відстежуючи подальші виникнення цих подій, система буде здатна поліпшити аналіз подій і, можливо, провести захисні заходи, перш ніж атака буде вдало виконана.

Недоліками систем виявлення вторгнень на основі нейронних мереж є:

1. Вимоги до навчання нейронної мережі - здатність до ідентифікації ознак вторгнення повністю залежить від правильного навчання системи, дані для навчання і методи навчання.

2. "Чорний ящик" нейронної мережі - вага зв'язку і передавальні функції різних мережевих вузлів, заморожуються після того, як мережа досягла прийнятного рівня успіху в ідентифікації подій. "Проблема чорної скриньки" переслідує нейронних мереж в ряді додатків. Це постійна область досліджень в нейронних мережах.

Основними реалізаціями нейронних мереж в системах виявлення вторгнень є:

1. Включення їх в експертні системи. В той час, як нейронна мережа розширила свої можливості для виявлення нових атак, експертну систему необхідно буде оновити для того, що б вона так само розпізнавала ці загрози.

2. Нейронні мережі як автономні системи виявлення вторгнень, будуть отримуватись дані з мережевого потоку і аналізувати інформацію на наявність вторгнення.

Список використаних джерел:

1. Круглов в. в., Борисов В.В. штучні нейронні мережі. - М.: Гаряча лінія-Телеком, 2002.

2. Каллан р. основні концепції нейронних мереж.: Пер. з англ. - М.: Вільямс, 2003.

ДОДАТОК Б

Презентація

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ДАЦЕНКО ВЛАДИСЛАВ СЕРГІЙОВИЧ

МЕТОД РЕАЛІЗАЦІЇ СИСТЕМ ІДЕНТИФІКАЦІЇ ВТОРГНЕНЬ НА БАЗІ НЕЙРОМЕРЕЖ ГЛИБОКОГО НАВЧАННЯ

Науковий керівник
к.т.н., доцент Тітова В.Ю.

Тема Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання
Метою кваліфікаційної роботи є удосконалення комп'ютерної системи для ідентифікації вторгнень
Об'єкт дослідження: процес ідентифікації вторгнень в комп'ютерні системи.
Предмет дослідження: методи та моделі ідентифікації вторгнень в комп'ютерні системи.
Задачі досліджень у роботі формулюються наступним чином:

1) Проаналізувати способи застосування технологій нейронних мереж в системах ідентифікації вторгнень та їх проблеми з перевагами і недоліками, щоб визначити найпоширеніші способи виявлення вторгнень.
2) Розглянути моделі нейромережі з метою їх застосування до ідентифікації вторгнень.
3) Обрати модель глибинної нейромережі для ідентифікації вторгнень.
4) Вдосконалити метод ідентифікації вторгнень за рахунок ведення в його структуру нейронної мережі.
5) Вдосконалити функціонування системи ідентифікації вторгнень шляхом впровадження зазначеного методу.
6) Визначити сферу застосування вдосконаленого методу.

Наукова новизна:

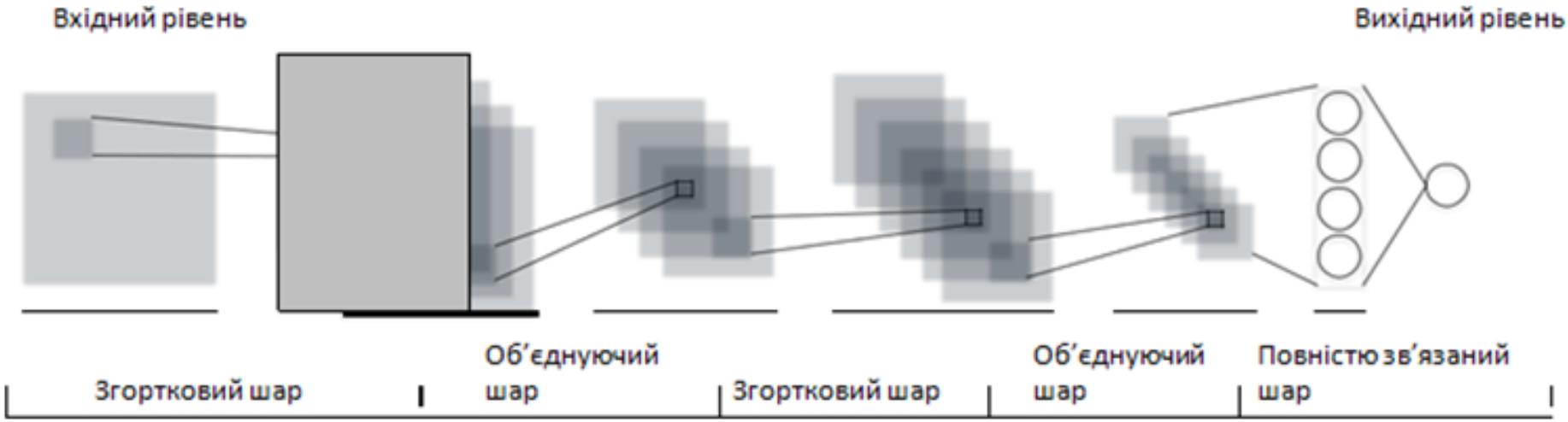
1) Вдосконалення методу ідентифікації вторгнень за рахунок впровадження в його структуру нейронних мереж.
2) Вдосконалення системи ідентифікації вторгнень за рахунок введення в її архітектуру підсистем прогнозування сценарію загроз та аналізу ефективності рішень.

Методи дослідження застосовані у даній роботі, базуються на алгоритмах та методах навчання нейромереж.
Практична цінність полягає в тому, що: розроблений додаток удосконалює комп'ютерну систему для ідентифікації вторгнень.
Публікації. По темі кваліфікаційної роботи опубліковано 1 стаття у фаховому журналі «Вісник ХНУ», №5, 2020. 1 стаття у нефаховому журналі (збірник НПК МНІС ІП-2020); 1 - теза доповідей на всеукраїнській конференції (Тези доповідей XVI Міжнародної науково-практичної конференції "Військова освіта і наука: сьогодення та майбутнє". К. : ВІКНУ, 2020);

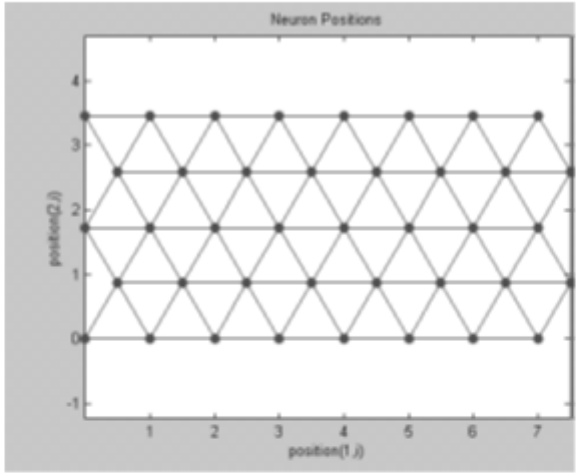
Порівняння різних моделей нейронних мереж глибокого навчання

Алгоритми	Відповідні типи даних	Під наглядом або без нагляду	Функції
Автокодер	Необроблені дані; Вектори характеристик	Без нагляду	Видобуток особливостей; Особливість зменшення; Шумить
RBM	Вектори характеристик	Без нагляду	Видобуток особливостей; Особливість зменшення; Шумить
DBN	Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
DNN	Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
CNN	Необроблені дані; Вектори характеристик	Під наглядом	Видобуток особливостей, класифікація
RNN	Необроблені дані; Вектори характеристик;	Під наглядом	Видобуток особливостей, класифікація
GAN	Необроблені дані; Вектори характеристик	Без нагляду	Збільшення даних; Змагальний тренінг

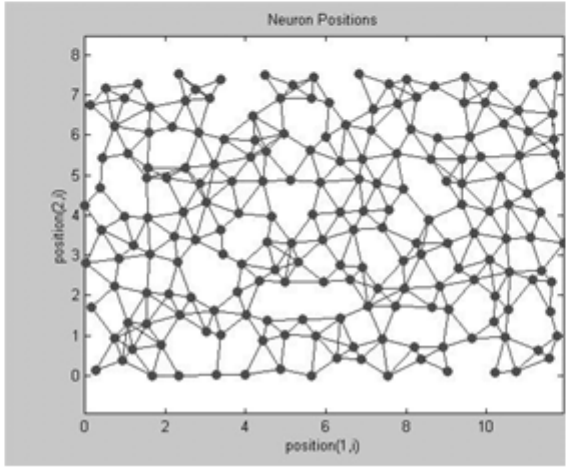
Графічна модель нейронної мережі глибинного навчання для ідентифікації вторгнень



Структура роботи нейромережі для ідентифікації вторгнень, представлена у вигляді гексагональної(шестикутної) решітки

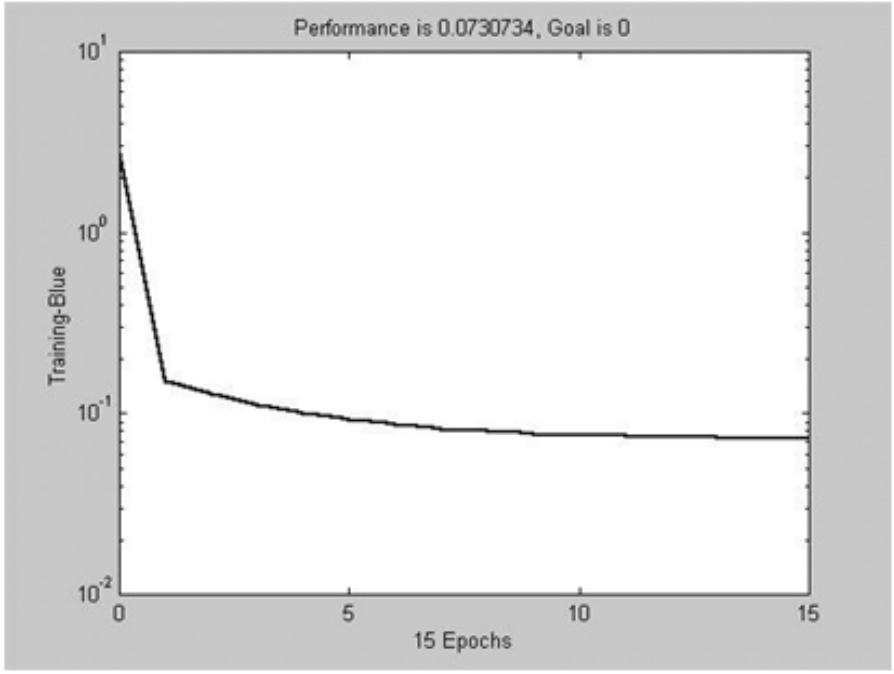


Структура роботи нейромережі для ідентифікації вторгнень, представлена у вигляді випадкових нейронів

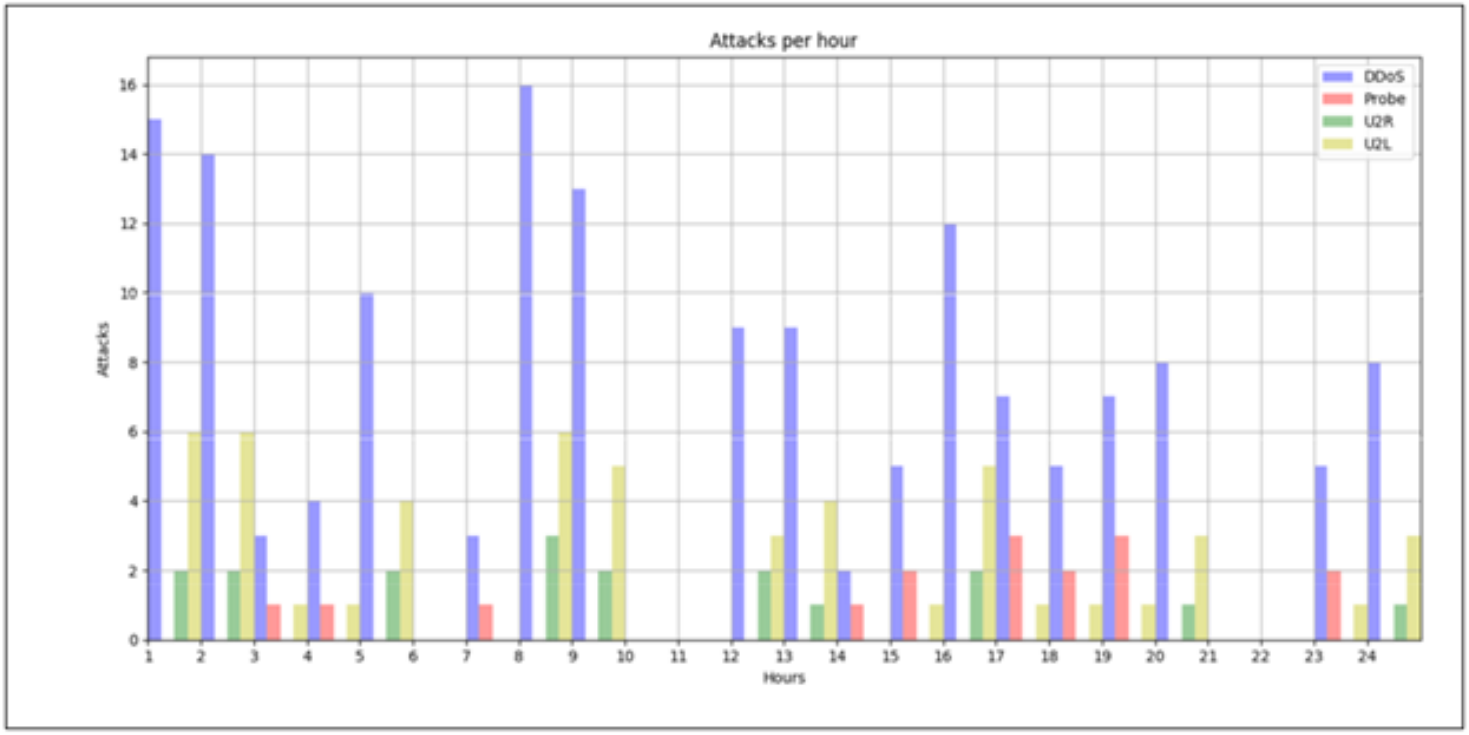


Процес навчання нейромережі

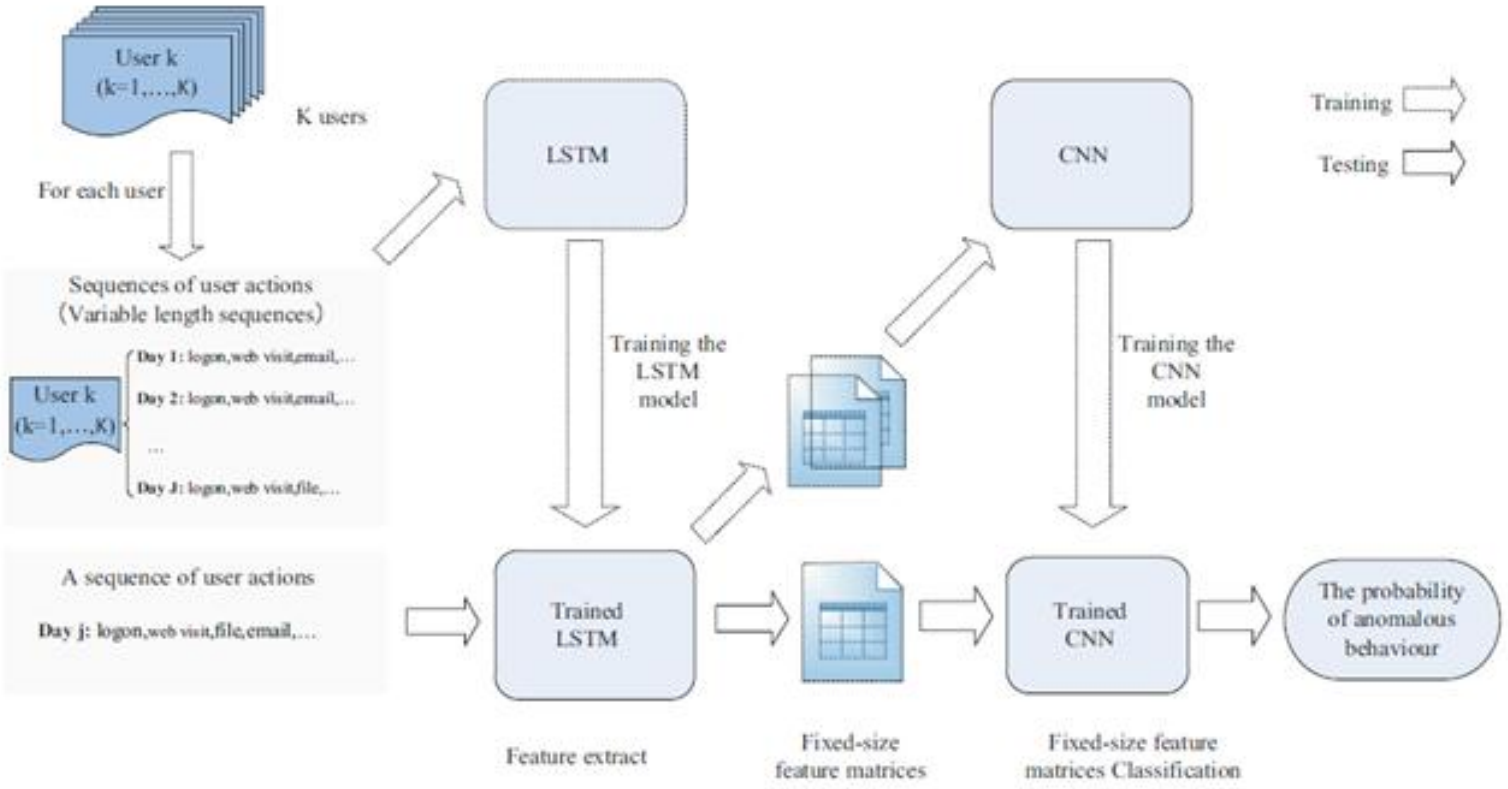
```
» P = [+1.0 +1.5 +3.0 -1.2];
» T = [+0.5 +1.1 +3.0 -1.0]; % Визначення величини коефіцієнта навчання
maxlr = maxlinlr (P, 'bias');
» B = newlin ([- 2 + 2], 1, [0], maxlr); % Створення лінійної нейромережі з ім'ям «b»
b.trainParam.epochs = 15; % Завдання кількості циклів навчання
» B = train (b, P, T); % Навчання нейромережі
TRAINWB, Epoch 0/15, MSE 2.865 / 0.
TRAINWB, Epoch 15/15, MSE 0.0730734 / 0.
TRAINWB, Maximum epoch reached.
» p = -1.2;
» y = sim (b, p) % Опитування мережі
» y = -1.1803
```



Результати роботи нейронної мережі ідентифікації вторгнень



Графічна структура вдосконаленого методу ідентифікації вторгнень за допомогою нейронних мереж



Функціональна схема системи виявлення вторгнень на основі нейронних мереж глибинного навчання

Приклад діалогового вікна системи виявлення вторгнень на основі нейронних мереж

Висновки

У кваліфікаційній роботі було описано метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання. Запропонований метод дозволяє удосконалити модель комп'ютерної системи для ідентифікації вторгнень.

Аналізуючи світові тенденції ,ми можемо стверджувати, що в найближчі роки дане питання буде ще більше актуальне, цьому буде сприяти стрімкий розвиток інформаційних технологій, яке ми спостерігаємо вже сьогодні.

Основні результати кваліфікаційної роботи:

1. Досліджено способи застосування технологій нейронних мереж в системах ідентифікації вторгнень та їх проблеми , з перевагами і недоліками, щоб визначити найпоширеніші способи ідентифікації вторгнень.
2. Розглянуто моделі нейромереж з метою їх застосування до ідентифікації вторгнень, визначено їх переваги та недоліки
3. На основі аналізу переваг та недоліків запропоновано модель глибинної нейромережі для ідентифікації вторгнень.
4. За допомогою впровадження зазначеної моделі вдосконалено метод виявлення вторгнень в комп'ютерних системах.
5. Вдосконалено систему виявлення вторгнень за рахунок введення в її архітектуру підсистем прогнозування сценарію загроз та аналізу ефективності рішень.
6. Визначено сферу застосування запропонованої у роботі моделі нейронної мережі для ідентифікації вторгнень та вдосконаленого методу.

Mon Dec 07 13:59:51 EET 2020, Муляра І.В, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 12%**

ID: 82752 Название: Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання Добавлено в БД: 2020-12-07 Авторы: Даценко В.С. Руководители: Тітова В.Ю. Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	113635	1026	5824 (5%)	72 (7%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

UNICHECK

User name:
Кафедра кибербезпеки

Check ID:
1005388961

Check date:
07.12.2020 14:56:58 EET

Check type:
Doc vs Internet

Report date:
07.12.2020 14:58:05 EET

User ID:
100005590

File name: **робота магістра Даценко -1**

Page count: **93** Word count: **17372** Character count: **133470** File size: **937.22 KB** File ID: **1005681005**

10.6% Matches

Highest match: **3.56%** with Internet source (https://otherreferats.allbest.ru/programming/00035304_2.html)

10.6% Internet sources	337	Page 95

No Library search was conducted

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters	7

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання

Автор: Даценко Владислав Сергійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: Програмування та захист комп`ютерних систем і мереж

Науковий керівник: Тітова ВіраІОріївна, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні і являють собою загальновживані терміни та загальновідому інформацію, або мають належним чином оформленні посилання;
3) в якості запозичень в окремих місцях системою зафіксовано послідовності стандартних процедур та функцій мови програмування нейронних мереж Matlab, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
4) також плагіатом не можна рахувати оформлені за вимогами літературні джерела.
5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 10,6 %, що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження.

Керівник роботи В.Ю. Тітова

Завідувач кафедри КбКСМ, гарант ОП Ю.П. Кльоц

Дата: 07.12.2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «магістр»

Магістр Даценко Владислав Сергійович

Тема Метод реалізації систем ідентифікації вторгнень на базі нейромереж глибокого навчання

Спеціальність 123 – Комп'ютерна інженерія

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень 11; кількість сторінок записки 90

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі удосконалено моделі комп'ютерної системи для ідентифікації вторгнень .

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подана загальна характеристика поставленої задачі, чітко визначено об'єкт, предмет та методи дослідження, сформульована актуальність. Визначені задачі, які необхідно вирішити для досягнення поставленої мети, практична цінність отриманих результатів, їхня новизна та наведені відомості про публікації. У першому розділі проведено огляд технологій нейронних мереж в системах виявлення вторгнень. В другому розділі розроблені моделі нейромережі з метою їх застосування до виявлення вторгнень. В третьому розділі визначено основні положення методу та розроблено алгоритми його реалізації. Четвертий розділ присвячено апробації методу та алгоритмів його реалізації моделюванням.

4. Позитивні сторони роботи Кваліфікаційна робота має комплексну наукову і практичну цінність. Наукова цінність полягає у вдосконаленні методу ідентифікації вторгнень за рахунок впровадження в його структуру нейронних мереж та розробці моделі системи ідентифікації вторгнень за рахунок додання множини експертних правил. Практична цінність результатів дослідження полягає у обґрунтуванні можливості використання алгоритмів і методів глибинного навчання нейронної мережі для підвищення ефективності функціонування систем виявлення вторгнень та захисту інформації в комп'ютерних системах в цілому.

5. Негативні сторони роботи В роботі неповністю наведено програмну реалізацію розробленого методу, що не знижує практичної та наукової цінності роботи

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження немає

9. Оцінка кваліфікаційної роботи враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «відмінно»/ A (4,75).

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Підченко Сергій Костянтинович
доктор технічних наук, професор,
завідувач кафедри телекомунікацій, медійних та інтелектуальних технологій

«9» 02 2020. (підпис)

ХНУ. Зам. 54, тир. 6000, 2007.