

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Дніпровський національний університет залізничного транспорту
імені академіка В. Лазаряна

Кафедра «Електронні обчислювальні машини»

«ДО ЗАХИСТУ»

Завідувач кафедри
Жуковицький І. В.

(підпис) (ПІБ)
« ____ » ____ 20 ____ р.

ДИПЛОМНА РОБОТА
на здобуття освітнього ступеня «магістр»

Галузь знань ____ 12 Інформаційні технології

Спеціальність ____ 125 Кібербезпека
(код) (повна назва)

Тема «Визначення мережових атак з використанням методів штучного інтелекту»

Theme «Detection of network attacks using artificial intelligence methods»

Керівник дипломного проекту	_____ (посада)	_____ (підпис)	<u>Пахомова В.М.</u> (ПІБ)
Консультант розділу з БЖД	_____ (посада)	_____ (підпис)	<u>Музикін М. І.</u> (ПІБ)
Нормоконтролер	_____ (посада)	_____ (підпис)	<u>Шаповалов В. О.</u> (ПІБ)
Студент групи	_____ (група)	_____ (підпис)	<u>Биковська Д.Г.</u> (ПІБ)
Student	<u>Bikovska Daria</u> (family name)		

Дніпро
2020

**Дніпровський національний університет залізничного транспорту
імені академіка В. Лазаряна**

Факультет КТС кафедра ЕОМ

Спеціальність Кібербезпека

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ

до дипломної роботи на здобуття освітнього ступеня _____
(освітнього ступеня)

студента групи КБ1921 Биковської Дар'ї Григорівни
(номер групи) (ПІБ)

1 Тема дипломної роботи Визначення мережевих атак з використанням методів штучного інтелекту

затверджена наказом по університету від «16» _____ 12 _____ 2019 р. № 945

2 Термін подання студентом закінченої роботи _____

3 Вихідні дані до дипломної роботи відкрита база, що має параметри мережевого трафіку «KDD-99»

4 Зміст пояснювальної записки (перелік питань до розробки) Вступ 1 - Огляд методів штучного інтелекту для визначення мережевих атак 2 - Постановка задачі визначення мережевих атак 3 - Визначення мережевих атак на основі розробленого програмного комплексу з використанням нейромережної технології та імунного підходу 4 - Організація досліджень на створеному програмному комплексі та його використання в навчальному процесі 5 - Охорона праці та безпека в надзвичайних ситуаціях Висновки

5 Перелік креслень (демонстраційного матеріалу) _____

Огляд методів штучного інтелекту щодо визначення мережевих атак

Постановка задачі визначення мережевих атак

Загальна схема виявлення мережевих атак (дворівневий підхід)

SOM як основний метод розв'язання задачі

Алгоритм клональної селекції

MLP у якості математичного апарату

Структура створеного програмного комплексу

Формування вибірок для навчання нейронних мереж

Визначення оптимальних параметрів нейронних мереж

Дослідження показників якості визначення атак

Використання створеного програмного комплексу в навчальному процесі

6 Розділи та консультанти

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Музикін М. І.		

КАЛЕНДАРНИЙ ПЛАН

Назва розділу	Термін виконання	Обсяг розділу, %
Вступ		5
Огляд методів штучного інтелекту для визначення мережових атак		20
Постановка задачі визначення мережових атак		10
Визначення мережових атак на основі розробленого програмного комплексу з використанням нейромережної технології та імунного підходу		30
Організація досліджень на створеному програмному комплексі та його використання в навчальному процесі		20
Охорона праці та безпека в надзвичайних ситуаціях		10
Висновки		5

Дата видачі завдання: «19» 12 2019 р.

Керівник дипломної роботи

(підпис) (ПІБ)

Завдання прийняв до виконання

(підпис) (ПІБ)

РЕФЕРАТ

Биковська Д. Г. Визначення мережевих атак з використанням методів штучного інтелекту. Дніпровський національний університет залізничного транспорту ім. акад. В. Лазаряна, кафедра електронних обчислювальних машин. Дипломна магістерська робота. 130 с. 40 рис. 20 табл. 42 джерел. 5 додатків.

У дипломній магістерській роботі виконано огляд методів штучного інтелекту для визначення мережевих атак на комп'ютерну мережу. У якості математичного апарату використані SOM (Self Organizing Maps) та MLP (Multi Layer Perceptron). Для визначення мережевих атак створений програмний комплекс, в основу якого покладені наступні моделі: «SOM_Clone», що написана на C++, для визначення категорії мережевої атаки: DoS, Probe, R2L, U2R (на першому етапі); «MLP», що написана на «Python» з використанням бібліотек машинного навчання, для визначення класу мережевої атаки відповідно до категорії (на другому етапі). У програмній моделі «SOM_Clone» для формування навчальної вибірки використовуються дані KDD-99 та алгоритм клональної селекції. На базі створеного програмного комплексу проведені наступні дослідження: визначення оптимальних параметрів MLP-DoS, MLP-Probe, MLP-R2L, MLP-U2R (перше дослідження); визначення показників оцінки якості отриманих рішень (друге дослідження). Відповідно до першого дослідження проведена оцінка точності та середньоквадратичної логарифмічної помилки (Mean Squared Logarithmic Error, MSLE) від кількості епох навчання за різними функціями активації та різною кількістю прихованих нейронів при різних алгоритмах оптимізації навчання. Визначено, що для виявлення класу атак категорії DoS достатньо мати нейронну мережу конфігурації 29-1-25-6 з логістичною функцією у прихованому шарі та функцією Softmax на результуючому шарі, яка за алгоритмом AdaDelta за 25 епох надає точність в 99,82 % на основі навчальної вибірки із 849 прикладів. Відповідно до другого дослідження отримані значення показників якості (TPR, FPR, CCR та ICR) від довжини навчальної вибірки.

АТАКА, КАТЕГОРІЯ, КЛАС, SOM, MLP, ІМУННИЙ ПІДХІД, MSLE, ЯКІСТЬ.

RESUME

Bykovska D. G. Detection of network attacks using artificial intelligence methods. Dnipro National University of Railway Transport named after academician V. Lazaryan, Department of Electronic Computers. Master's thesis. 130 p. 40 pictures. 20 tables. 42 sources. 5 applications.

In the master's thesis we reviewed methods of artificial intelligence to determine network attacks on a computer network. SOM (Self Organizing Maps) and MLP (Multi Layer Perceptron) were used as mathematical apparatus. In order to identify network attacks, a software package was created, which is based on the following models: "SOM_Clon", written in C ++, to check the category of network attack: DoS, Probe, R2L, U2R (in the first stage); MLP, written in Python using machine learning libraries, was applied to determine the network attack class according to the category (in the second stage). The software model "SOM_Clon" uses KDD-99 data and a clonal selection algorithm to form a training sample. On the basis of the created software complex the following researches were carried out: definition of optimal parameters of MLP-DoS, MLP-Probe, MLP-R2L, MLP-U2R (the first research); determination of indicators of quality assessment of the received decisions (second research). According to the first study, the accuracy and mean square logarithmic error (MSLE) of the number of learning epochs for different activation functions and different numbers of hidden neurons with different learning optimization algorithms were evaluated. It was determined that in order to detect the class of DoS attacks it is enough to have a neural network configuration 29-1-25-6 with a logistic function in the hidden layer and Softmax function on the resulting layer, which according to the algorithm AdaDelta for 25 epochs provides an accuracy of 99.82% based on training sample of 849 examples. According to the second study, the values of quality indicators (TPR, FPR, CCR and ICR) were obtained from the length of the training sample.

ATTACK, CATEGORY, CLASS, SOM, MLP, IMMUNE APPROACH, MSLE, QUALITY.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК.....	10
1.1 Постановка проблеми	10
1.2 Нейронні мережі для визначення мережесих атак	11
1.3 Використання штучної імунної системи для виявлення мережесих атак.....	15
1.3.1 Біологічна модель імунної мережі	15
1.3.2 Моделі штучних імунних систем	17
1.4 Сучасні системи виявлення мережесих атак.....	19
1.5 Основні висновки	22
2 ПОСТАНОВКА ЗАДАЧІ ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК	24
2.1 Причини використання методів штучного інтелекту для визначення мережесих атак на основі імунного підходу	24
2.2 Основні висновки	26
3 ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК НА ОСНОВІ РОЗРОБЛЕНОГО ПРОГРАМНОГО КОМПЛЕКСУ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖНОЇ ТЕХНОЛОГІЇ ТА ІМУННОГО ПІДХОДУ	27
3.1 Визначення категорії мережесих атаки (перший етап).....	27
3.1.1 SOM як основний метод розв’язання задачі	27
3.1.3 Формування вибірок для навчання SOM (підготовчий етап).....	31
3.1.4 Програмна реалізація моделі «SOM_Clon».....	32
3.2 Визначення класу мережесих атаки (другий етап).....	34
3.2.1 MLP у якості математичного апарату	34
3.2.2 Формування вибірок для навчання MLP (підготовчий етап)	34
3.2.3 Програмна реалізація моделі «MLP»	37
3.3 Структура програмного комплексу та його тестування	40
3.4 Основні висновки	44
4 ОРГАНІЗАЦІЯ ДОСЛІДЖЕНЬ НА СТВОРЕННОМУ ПРОГРАМНОМУ КОМПЛЕКСІ ТА ЙОГО ВИКОРИСТАННЯ В НАВЧАЛЬНОМУ ПРОЦЕСІ ..	45
4.1 Визначення оптимальних параметрів нейронних мереж.....	45
4.1.1 Виявлення класу мережесих атак категорії «R2L»	45

4.1.2	Виявлення класу мережевих атак категорії «U2R»	47
4.1.3	Виявлення класу мережевих атак категорії «DOS».....	50
4.1.4	Виявлення класу мережевих атак категорії «Probe»	52
4.2	Дослідження показників якості визначення атак	55
4.3	Використання програмного комплексу в навчальному процесі	59
4.3.1	Інструкція по використанню моделі «SOM_Clon».....	59
4.3.2	Інструкція по використанню моделі «MLP»	61
4.3.3	Приклади завдань на основі програмного комплексу.....	62
4.4	Основні висновки.....	66
5	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	67
5.1	Вимоги безпеки при виконанні робіт на робочому місці	67
5.2	Шкідливі виробничі фактори на робочому місці	68
5.3	Дій працівників в аварійних ситуаціях	73
	ВИСНОВКИ.....	76
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	77
	ДОДАТКИ	
	Додаток А. Програмна реалізація моделі «SOM_Clon» Помилка! Закладку не визначено.	
	Додаток Б. Програмна реалізація моделі «MLP» Помилка! Закладку не визначено.	
	Додаток В. Дослідження параметрів якості на програмному комплексі..... Помилка! Закладку не визначено.	
	Додаток Г. Підготовка робочого місця для використання програмного комплексу в навчальному процесі..... Помилка! Закладку не визначено.	
	Додаток Д. Тези доповідей щодо конференцій Помилка! Закладку не визначено.	

ВСТУП

Все більше і більше нових мережевих атак з'являються кожен день, і це спонукає звернути увагу та занепокоїтись мережевою безпекою. Існуючі методики виявлення мережевих атак не завжди призводять до ефективного результату, тому доцільно використовувати методи штучного інтелекту, що підтверджує *актуальність* теми магістерської дипломної роботи.

Метою магістерської дипломної роботи є виявлення мережевих атак на комп'ютерну мережу з використанням методів штучного інтелекту. Відповідно до мети поставлені наступні *задачі*:

- виконати огляд методів штучного інтелекту для визначення мережевих атак на комп'ютерну мережу;
- створити програмний комплекс для виявлення мережевих атак з використанням нейромережної технології імунного підходу;
- визначити оптимальні параметри нейронних мереж для виявлення різних класів мережевих атак;
- провести на створеному програмному комплексі дослідження параметрів якості визначення мережевих атак

На сучасному етапі виявленням мережевих атак на комп'ютерну мережу з використанням методів штучного інтелекту займаються наступні вчені: Де Кастро, Тімміс Джон, Уоткінс Ендрю, Фон Зубен, Бурлаков М. Е., Васильев В. І., Жуков В. Г., Жуковицький І. В., Кораблев Н. М., Котов В. Д., Литвиненко В. І., Пахомова В. М., Саламатов Т. А., Фомичев А. А. та ін. Так, наприклад, Жуковицький І. В. і Пахомова В. М. проводили визначення мережевих атак на основі використання багатошарових нейронних мереж. У роботах Жукова В. Г. та Салматова Т. А. показано, що штучні імунні системи з клональною селекцією дозволяють виявити цілеспрямовані зміни у контрольованих даних. У роботах Кораблева Н. М. та Фомичева А. А. досліджується гібридизація імунних методів та моделей інтелектуальної обробки інформаційних можливостей для підвищення їх ефективності, що виражається в підвищенні швидкодії, або точності їх роботи. Досліджені можливості розробки гібридних методів

класифікації, функціонуючих на основі різних імунних моделей та інших підходах. Де Кастро і Фон Зубен розробили алгоритм клонального відбору CLONALG. Ендрю Уоткінс та Джон Тімміс запропонували систему розпізнавання штучного імунітету AIRS.

Представлена магістерська дипломна робота складається із вступу, п'яти розділів та висновків. У розділі 1 виконаний огляд існуючих підходів штучного інтелекту щодо виявлення мережевих атак, розглянута біологічна модель штучної та імунної мережі, а також огляд існуючих систем виявлення мережевих атак. У розділі 2 сформульована постановка задачі визначення мережевих атак на комп'ютерну мережу. У розділі 3 створений програмний комплекс, основу якого складає модель самоорганізованої карти з використанням клональної селекції для визначення категорії атаки (на першому етапі) та модель багатошарового перцептрону для визначення класу атаки відповідно до категорії (на другому етапі). У розділі 4 представлені дослідження оптимальних параметрів нейронних мереж. Отримана оцінка точності нейронних мереж від кількості епох навчання за різними функціями активації та різною кількістю прихованих нейронів. Крім того, на створеному програмному комплексі проведені дослідження показників якості визначення мережевих атак (TPR, FPR, CCR та ICR) від довжини навчальної вибірки, що згенерована за допомогою клональної селекції. У розділі 5 подані основні правила безпеки при роботі за комп'ютером, та дії при виникненні надзвичайної ситуації і правила надання домедичної допомоги.

Результати дипломної магістерської роботи доповідались на XIV Міжнародній науково-практичній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті», а також на Всеукраїнській конференції студентів та молодих вчених «Інформаційно-управляючі технології і системи на залізничному транспорті», що відбулись в Дніпровському національному університеті залізничного транспорту імені академіка В. Лазаряна в 2020 р. Тези доповідей опубліковані у відповідних збірниках до конференцій та представлені в додатку Д.

1 ОГЛЯД МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК

1.1 Постановка проблеми

Кожен день у всьому світі відбувається безліч випадків пов'язаних з порушенням інформаційної безпеки. Згідно [1] за 2018 рік 800 мільйонів дорослих людей стали жертвами кіберзлочинності і 38% мали фінансові втрати. Також згідно з [2] кількість веб-атак зросло на 56% у порівнянні з минулим роком. Лише за 2019 рік збитки від кіберзлочинності склали більше 2 трильйонів доларів по всьому світу. Постає гостра проблема захисту інформації не тільки для корпорацій, а і для громадян.

Основними цілями забезпечення інформаційної безпеки є зведення до мінімуму фінансових збитків, забезпечення конфіденційності, доступності інформації та забезпечення цілісності інформації.

Найбільш розповсюдженим та відносно ефективним рішенням для захисту мережі від зовнішніх загроз є мережевий екран. В більшості випадків його є достатньо для захисту від некваліфікованих злоумисників, але для професійних направлених атак мережевий екран не є серйозною перешкодою[3]. Тому для більш надійного виявлення мережевих атак використовують системи виявлення атак/вторгнень(СВА або СВВ).

Системи виявлення вторгнень(Intrusion Detection System, IDS) – це системи, які збирають інформацію з різних системних і мережевих джерел, а потім аналізують інформацію про ознаки вторгнення та зловживання [4].

Типовими прикладами атак, моніторинг яких здійснює СВВ є впровадження шкідливого коду, підбір пароллю, мережева активність троянських коней та вірусів, вичерпання смуги пропускання шляхом великої кількості з'єднань та інше. Згідно з [5] основними підходами до класифікації СВВ є класифікація по типу об'єкта моніторингу (вузлові та мережеві СВВ), за архітектурою(централізовані, розподілені), за технологією аналізу (без збереження стану, з збереженням стану), за методом виявлення атак(системи виявлення аномалій, системи виявлення зловживань, системи виявлення

порушень у протоколі), за шляхом реагування (пасивні, активні). Сучасні СВВ, як правило, активні, розподілені, та забезпечують моніторинг мережі у режимі близькому до режиму реального часу.

Відповідно до [6] серед методів визначень мережевих атак можна виокремити наступні методи: методи виявлення аномалій (статистичний аналіз, кластерний аналіз, нейронні мережі, експертні системи, поведінкова біометрія, Support vector machines (SVM)) та методи виявлення зловживань (аналіз систем станів, графи сценаріїв атак, нейронні мережі, штучні імунні системи, Support vector machines (SVM), експертні системи, методи засновані на специфікаціях, Multivariate Adaptive Regression Splines (MARS), сигнатурні методи).

У свою чергу нейронні мережі та штучні імунні системи можна віднести до біоінспірованих методів.

1.2 Нейронні мережі для визначення мережевих атак

Центральна нервова система має клітину будову. Одиниця – нервова клітина, нейрон.

Сучасні штучні нейронні мережі на даний момент не можуть зрівнятися з біологічним прототипом, але вже демонструють цінні властивості, такі як [7]: здатність навчатися, здатність до узгодження та до абстрагування.

Дослідження штучних нейронних мереж пов'язані з тим, що шлях обробки інформації людським мозком в корні відмінні від методів, що застосовуються звичайними цифровими комп'ютерами. Мозок являє собою дуже складний нелінійних, паралельний комп'ютер. Він має властивість організовувати свої структурні компоненти, так називаємо нейрони, так, щоб вони могли виконувати конкретні задачі в багато разів швидше, ніж можуть дозволити самі швидкодіючі сучасні комп'ютери [7].

Згідно за визначенням даним у [8] нейрона мережа це обчислювальна або логічна схема, побудована з нейронних процесорних елементів, які містять у собі правила рішень, та працює по відповідному нейрона мережевому алгоритму, який враховує таку властивість біологічного нейрона, як пластичність, здатність змінювати свої параметри у процесі навчання та здобуття нових знань.

Серед переваг використання штучних нейронних мереж для виявлення зловживань є:

- Гнучкість – можливість виявлення зловживань з використанням неповних або змінених даних ;
- Здатність виявляти маловідомі атаки, а також ймовірність появи атаки розподіленої у часі;
- Висока швидкість аналізу даних.

А до недоліків слід віднести:

Необхідність навчання нейронної мережі, що обумовлене застосуванням різних методів навчання і підготовки навчаючих даних для найкращого аналізу та виявлення зловмисницької діяльності

Суть процесів у нейронній мережі прихована, і якість аналізу залежить безпосередньо від навчання

На рисунку 1.1 наведена схема застосування штучних нейронних мереж для виявлення зловживань [8]

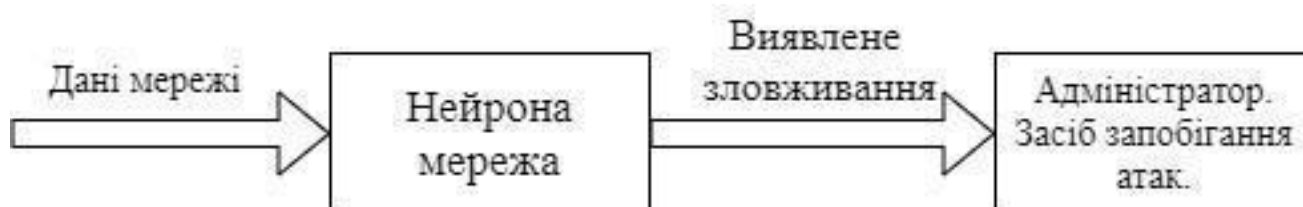


Рисунок 1.1 – Схема застосування НМ для виявлення зловживань

Розглянемо наступні методи нейронних мереж, які здобули широкого застосування в галузі захисту інформації: багатошаровий перцептрон та самоорганізовані карти SOM (Self Organizing Maps).

Багатошаровий перцептрон. Нейромережі можуть бути розділені на контрольовані (по типу алгоритму навчання) і неконтрольовані. У контрольованому варіанті мережу «дізнається» бажаний вихід (відомий правильний відгук мережі на заданий зразок) для кожного даного вхідного шаблону. Приклад цієї форми структури мережі – багаторівневий перцептрон MLP (Multilayered perceptron) [6].

Багатошаровий перцептрон вдало використовується для вирішення різних складних задач. При цьому навчання з викладачем виконується за допомогою такого популярного алгоритму, як алгоритм зворотного розповсюдження помилки [6, 9].

MLP – сильно взаємопов'язаний шар нейронів, що розміщуються у вхідному рівні, рівні виведення і одному або більше «прихованих» рівнях. У процесі навчання кожен вузол виконує зміщену зважену суму їх введів і передає цей рівень активації по функції перетворення на вихід в нашарувань топології прямого поширення. Мережа, таким чином, здійснює просту обробку за формою моделі введення / виведення з вагами і порогами, які є вільними параметрами моделі. В процесі навчання MLP прогресує ітераційно, проходячи стадії так званих «епох», тимчасової ряд появи сигналу. На кожній «епосі» кожен конкретний варіант циклічно обробляється по мережі – виконується порівняння цільових і поточних значень ефективних вихідних параметрів з оцінкою помилки, яка разом з градієнтом по топології простору помилок використовується для коригування вагових значень. Результат перетворення визначається відповідно до характеристик вузлів і ваг по їх взаємозв'язку. Змінюючи характер зв'язку між вузлами, можна адаптувати мережу до потрібного відгуку [6, 9, 10].

Застосування та дослідження роботи нейронної мережі на багатошаровому перцептроні для виявлення мережевих вивчені та розглянуті у таких роботах як [11, 12, 13].

Основними перевагами використання виявлення атак на базі багатошарового перцептрону є [14]:

- гнучкість та адаптивність алгоритмів;
- здатність аналізувати дані з мережі навіть якщо вони неповні або змінені;
- висока швидкість обробки даних, яка забезпечує роботу системи в режимі реального часу;

- здатність «вивчення» характеристик атак та виокремлення елементів, що відрізняються від спостерігаємих раніше.

Одним із найбільш значимих недоліків багатошарового перцептронів є недостатня достовірність визначення допустимості відхилень параметрів поточного функціонування в області їх мінімальних значень. Показано, що вказаний недолік спричинений неадекватністю цільового функціоналу алгоритму зворотного поширення помилки, який застосовується для навчання багатошарового перцептронів [11]

Мережа Кохонена або самоорганізована карта (Self Organizing Maps, SOM) є новим, ефективним програмним інструментом для візуалізації багатовимірних даних.

У своєму основному варіанті SOM створює граф подібності вхідних даних. Вона перетворює нелінійні статистичні співвідношення між багатовимірними даними і прості геометричні зв'язки між зображують їх точками на пристрої відображення низької розмірності. зазвичай н вигляді регулярної двовимірної сітки вузлів. Оскільки SOM здійснює стиснення інформації зі збереженням в вихідному зображенні найбільш важливих топологічних і чи метричних зв'язків між первинними елементами даних, можна також вважати, що з її допомогою породжуються абстракції (узагальнення) деякого віща. Ці два характерних властивості SOM. візуалізацію і узагальнення, можна використовувати різними способами у вирішенні складних завдань таких, як аналіз процесів, машинне сприйняття, управління, передача інформації [15].

У процесі навчання SOM представлена поруч векторів даних і ітераційно проходить також через «епохи». У кожній ітерації SOM вибирає «виграє» нейрон, який є найближчим до вхідного вектора, і потім коригує його, збільшуючи функцію правдоподібності по вхідному сигналу. SOM мережі призначені для задач класифікації, але не для завдань розпізнавання образів [6].

SOM («мережі Кохонена») – використовує один шар (рівень) нейронів для подання відомості від окремого домену в формі геометрично організованою карти. Пропонована мережа була призначена для вивчення характеристик

діяльності звичайної системи та ідентифікації статистичних відхилень від норми, що може вказувати на вірус [6].

Відмінні риси застосування самоорганізованих карт Кохонена в області аналізу інцидентів інформаційної безпеки [15]:

- навчання нейромережі відбувається без вчителя на основі тільки вхідних даних, тобто для проведення аналізу даних не потрібно бути фахівцем в області штучного інтелекту;
- наочне представлення результатів у вигляді кольорових двовимірних карт, на яких схожі в вихідному просторі опиняються поруч;
- використання нейромережевих алгоритмів дозволяє виявляти приховані закономірності в даних, які можуть залишитися без уваги при використанні тільки статистичних методів.

1.3 Використання штучної імунної системи для виявлення мережевих атак

1.3.1 Біологічна модель імунної мережі

Постійна еволюція живих організмів привела до утворення особливого захисного біологічного механізму, здатного функціонувати на клітинномолекулярному рівні і успішно справлятися з безліччю сприятливих даного організму зовнішніх чинників, -природно імунної системи. Серед її відмінних рис можна виділити здатність розпізнавання будь-яких молекул незалежно від їх походження та приналежності до власних тканин організму або чужорідним мікроорганізмам (патогенів), наявність асоціативної пам'яті, що сприяє ідентифікації не тільки однієї певної форми патогена, а й структурно пов'язаних з нею інших шаблонних варіацій, відсутність централізованого контролю, що забезпечує розподілене поведінку компонент імунної системи без необхідності взаємодії з будь-яким єдиним керуючим органом, надання багаторівневої лінії оборони з використанням фізичного (слизові оболонки, епітеліальні клітини Лангерганса), фізіологічного (ферменти, лужно-кислотні і температурні середовища) і лейкоцитарного (клітини вродженої і адаптивної імунних систем) бар'єрів [3].

Імунітет виконує функцію розпізнавання і усунення чужорідного («не свого») матеріалу, який надходить в організм зазвичай у вигляді небезпечних для життя патогенних мікроорганізмів, але в той же час у формі життєво необхідного трансплантата, наприклад нирки [16].

Імунна система забезпечує [17]:

а) захист організму від впровадження чужорідних клітин і від виниклих в організмі модифікованих клітин (наприклад, злоякісних);

б) знищення старих, дефектних і пошкоджених власних клітин, а також клітинних елементів, не характерних для цієї фази розвитку організму;

в) нейтралізацію з подальшою елімінацією всіх генетично чужорідних для даного організму високомолекулярних речовин біологічного походження (белков, полісахаридів, липідів, полісахаридів в і т.д.).

Стійкість до інфекції може бути природною (тобто вродженої і незмінної) або придбаної в результаті адаптивного імунної відповіді.

Адаптивний імунітет заснований на властивостях Т- і В-лімфоцитів вибірково відповідати на тисячі чужорідних речовин (антигенів) з утворенням специфічної пам'яті і реагування, індивідуального для кожного конкретного антигену адаптування до навколишнього середовища). Проти деяких антигенів адаптивні механізми здатні діяти автономно. У більшості випадків відбувається взаємодія з елементами природного імунітету: антитіла – з комплементом і фагоцитарних клітинами. Т-лімфоцитів з макрофагами. [16]

Лімфоцити відрізняються між собою не тільки специфікою своїх рецепторів, але і по їх функціональним властивостям. Розрізняють два основних класу лімфоцитів: В-лімфоцити, які слугують попередниками антитіл-утворюючих клітин, та Т-лімфоцити, які також відомі як тимусозалежні лімфоцити. Т-лімфоцити поділяються на ряд підкласів. Частина з яких виконує важливі регуляторні функції : може допомагати («хелпери») чи пригнічувати («супресори») розвиток імунної відповіді, в особливості утворення антитіл. Інші Т-лімфоцити виконують ефекторні функції, наприклад виробляють розтворимі речовини, які запускають різноманітні запальюючі реакції, або здійснюють пряме

руйнування клітин, які несуть у собі антигени («кіллерна» функція). У відповідності з цим ми розрізняємо Т-хелпери, Т-супресори, Т-кіллери та Т-клітини, що приймають участь у реакціях сповільненої гіперчутливості і пов'язані з нею імунних явищ [18].

1.3.2 Моделі штучних імунних систем

Генерація детекторів за допомогою негативного відбору. Імунна система людини здатна виявляти як відомі патогени, так і нові, інформації про яких у системи не було. В основі цієї здатності лежить механізм негативного відбору, суть якого полягає в тому, що створювані лімфоцити перевіряються на відповідність «своїм» клітинам і в разі позитивного результату - знищуються. Таким чином, залишаються тільки лімфоцити, які не реагують на «свої» клітини, тільки на «чужі». Цей механізм покладено в основу алгоритму негативного відбору. Критерієм схожості в вихідному описі алгоритму виступає часткове відповідність, тобто відповідність n послідов символів в рядках. Випадковим чином генеруються детектори, які перевіряються на «схожість» з контрольованим рядками і відкидаються в разі збігу зі «своєю» рядком. В ході перевірки, якщо рядок схожий з детектором, реєструється інцидент. [19]

Модель розпізнавання «свій-чужий» базується на процесі дозрівання Т-клітин в тимусі. Для того щоб описати формування і функціонування таких клітин (детекторів), застосовується алгоритм негативного відбору, заснований на припущенні про те, що апріорно відома інформація лише про позитивні об'єктах, які розглядаються як «свої» (що відрізняються від них об'єкти іменуються «чужими»). Даний алгоритм в класичному виконанні складається з двох етапів: генерація детекторів і виявлення «чужих» об'єктів. [20]

Головним недоліком, «успадкованим» від базового алгоритму негативного відбору, є випадковість генерації детекторів. Через велику кількість можливих варіантів, потрібно, відповідно, велике число згенерованих детекторів для виявлення інцидентів інформаційної безпеки. Для зменшення стохастичності результатів, доцільно все детектори поділити на дві групи [19]:

1) Детектори, згенеровані випадковим чином, для виявлення нових, невідомих загроз.

2) Детектори, отримані шляхом модифікації відомих записів про інциденти інформаційної безпеки. Такі записи можуть бути відомі в системі спочатку або бути виявлені в ході роботи системи. Застосування таких детекторів дозволить виявляти відомі загрози і їх варіації.

Клональна селекція. Принцип клональної селекції моделює поведінку В-клітин в процесі імунної реакції на антигенний стимул. Така взаємодія супроводжується виробленням клонів В-клітини, які можуть зазнавати мутації різного ступеня залежно від сили їх зв'язування (заходи аффіності) з будь-яким епітопом антигену. Як і алгоритм негативного відбору, алгоритм клональної селекції відноситься до сімейства популяційних алгоритмів, в яких особини, які описують поточний рішення задачі, піддаються поліпшень і замін і борються один з одним за право бути відібраними в кращі кандидати[20].

Першою реалізацією алгоритму клональної селекції є CLONALG[21]. Автори CLONALG підкреслюють такі важливі особливості алгоритму клональної селекції, що відрізняють його від генетичного алгоритму: виконання операторів мутації і відбору пропорційно афінності особин, відсутність оператора схрещування і кодування особин у вигляді двоїчних рядків замість матеріально значного уявлення [20].

Клональна селекція є теорією, яка використовується для пояснення механізмів імунної відповіді при розпізнаванні імунними клітинами образів чужих антигенів В-клітинами. Гіпотеза, висловлена Н. Ерне, згідно з якою роль антигену зводиться до відбору до нього клітин, отримала подальший розвиток в теорії селекції клонів Франка Бернета, що постулює чотири основних положення [22]:

- 1) лімфоїдна тканина організму містить велику кількість клітин;
- 2) популяція лімфоїдних клітин гетерогенна і складається з великої кількості клонів, що виникли в результаті мутацій; постійні процеси мутацій лімфоїдних клітин забезпечивають достатню кількість окремих клонів

лімфоїдних клітин, специфічних по відношенню до можливої кількості антигенних детермінант; клонування може бути обумовлено генетичним кодом;

3) мала кількість антигену стимулює специфічний клон лімфоїдних клітин до розмноження і диференціювання в плазмочити, чим і забезпечується імунна відповідь -вироблення антитіл;

4) велика кількість антигену елімінує відповідний клон лімфоїдних клітин, ніж обумовлюється загибель виникли в ембріональному періоді лімфоїдних клітин, здатних реагувати проти власних антигенів, і формується природна толерантність до них.

1.4 Сучасні системи виявлення мережових атак

Для забезпечення захисту даних у рамках локальних та глобальних мереж існує безліч рішень, до таких рішень відносяться системи виявлення та запобігання вторгнень (Intrusion detection systems, IDS і Intrusion prevention systems, IPS).

Під системою виявлення вторгнень (СВВ) розуміється програмне або апаратне засіб, призначений для виявлення фактів несанкціонованого доступу до комп'ютерної / інформаційну систему або мережу або несанкціонованого управління ними злоумисником [23].

IDS/IPS-системи використовуються для виявлення аномальних дій в мережі, які можуть негативно вплинути на безпечність і конфіденційність даних, наприклад: спроби використання вразливостей програмного забезпечення; спроби підвищення привілеїв; несанкціонований доступ до конфіденційних даних; активність шкідливих програм і т.д.

Використання IPS-систем переслідує кілька цілей [24]:

- виявити вторгнення або мережеву атаку і запобігти їй;
- спрогнозувати можливі майбутні атаки і виявити вразливості для запобігання їх подальшого розвитку;
- виконати документування існуючих загроз;
- забезпечити контроль якості адміністрування з точки зору безпеки, особливо в великих і складних мережах;

- отримати корисну інформацію про проникнення, які мали місце, для відновлення і коригування викликали проникнення факторів;
- визначити розташування джерела атаки по відношенню до локальної мережі (зовнішні або внутрішні атаки), що важливо при прийнятті рішень про розташування ресурсів в мережі.

У цілому, IPS аналогічні IDS. Головна відмінність полягає в тому, що вони функціонують в реальному часі і можуть в автоматичному режимі блокувати мережеві атаки. Кожна IPS включає в себе модуль IDS. У свою чергу, IDS зазвичай складається з:

- системи аналізу зібраних подій;
- системи збору подій;
- сховища, в якому накопичуються зібрані події та результати їх аналізу;
- бази даних про вразливість (цей параметр є ключовим, тому що чим більше база у виробника, тим більше загроз здатна виявляти система);
- консолі управління, яка дозволяє налаштовувати всі системи, здійснювати моніторинг стану мережі, що захищається, переглядати виявлені порушення і підозрілі дії.

Загальна класифікація сучасних підходів до детектування мережевих атак може бути представлена наступним чином [25,26]:

- 1) За типом об'єкта моніторингу – хостової СВВ та мережеві СВВ.
- 2) По архітектурі – централізовані та розподілені.
- 3) За технологією аналізу – без збереження стану та зі збереженням стану.
- 4) За методом виявлення атак – системи виявлення зловживань, системи виявлення аномалій та системи виявлення порушень в протоколі.
- 5) За способом реагування – пасивні та активні.

Найважливішою перевагою використання IDS є те, що IDS надає інформацію, якщо відбувається атака за певною системою. Через користувачів IDS відомо про те, в якому ризику і загрозі вони знаходяться. Багато інструментів може бути використано для захисту певної системи. Це такі, як брандмауери, IDS

SNORT, Suricata і т.д. Дана стаття присвячена аналізу таких систем, виявленню особливостей та побудові системи, що буде належати до цього класу систем і яку в подальшому можна буде досліджувати та використовувати на практиці [27].

Розглянемо деякі IDS та IPS, що на даний момент широко застосовані, та технології, які в них використовуються.

Порівняння деяких IDS наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз існуючих IDS

Назва системи	Переваги	Недоліки
Snort	Дуже простий в установці, запуску та роботі. Велике ком'юніті користувачів, багато підтримуваних ресурсів, доступних в Інтернеті	Поставляється без графічного інтерфейсу, хоча існують додатки, розроблені ком'юніті. Обробка пакетів може бути повільною
Suricata	Може використовувати набори правил Snort. Має розширені функції.	Схильні до помилкових спрацьовувань. Системна та мережева ресурсомісткість
Bro IDS	Платформа може бути адаптована до різних випадків використання мережевої безпеки, в доповненні до NIDS	Необхідний певний досвід програмування. Отримання кваліфікації в Bro DSL може потребувати певних зусиль
Security Onion	Комплексний стек безпеки, що складається з декількох провідних рішень із відкритим кодом. Забезпечує простий інструмент налаштування для встановлення всього стеку	Як платформа, що складається з декількох технологій, Security Onion наслідуює недоліки кожного компонента

Серед широковідомих та використовуваних IPS слід згадати такі: Cisco IPS, Sourcefire IPS, Adaptive IPS, McAfee Network Security Platform, Stonesoft StoneGate IPS, IBM Proventia Network Intrusion Prevention System

Порівняння деяких IPS наведено у таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз існуючих IPS

Назва системи	Переваги	Недоліки
Cisco IPS	Запобігання вторгнення більш 30000 відомих експлойтів; автоматичне оновлення сигнатур з глобального сайту Cisco Global Correlation для динамічного розпізнавання і запобігання вторгнень атак з боку Internet; передові дослідження і досвід Cisco Security Intelligence Operations; взаємодія з іншими мережевими компонентами для запобігання вторгнень; підтримка широкого спектру варіантів розгортання в режимі, близькому до реального часу.	Не може відстежити атаки, що виходять за рамки налаштованих правил. Часом, необхідно відключати підпису, які забороняють потрібний трафік. Необхідно періодичне оновлення шаблонів.
IBM Proventia Network Intrusion Prevention System	Не вносить затримки в мережі; деталізоване налаштування політик; превентивні заходи по запобігання вторгнень; підтримка користувацьких сигнатур; карантин	Недоліком IDS є єдино можливий метод блокування атак - посилка пакетів TCP Reset

1.5 Основні висновки

1 Сучасні штучні нейронні мережі на даний момент не можуть зрівнятися з біологічним прототипом, але вже демонструють ціні властивості, такі як: здатність навчатися, здатність до узгодження та до абстрагування. Серед методів нейронних мереж для виявлення мережових атак можна відокремити використання багатошарового перцептрону та самоорганізованої карти. У магістерській дипломній роботі перевагу віддано самоорганізованої карті, через здатність до самонавчання та гнучкому алгоритму класифікації даних.

2. В основі методів штучної імунної системи покладені реальні біологічні моделі В-лімфоцитів та Т-лімфоцитів, та взагалі робота імунної системи. Основними моделями штучної імунної системи є алгоритми клональної селекції та алгоритм генерації детекторів за допомогою негативного відбору, в основі якого лежить принцип «свій-чужий». Принцип клональної селекції моделює

поведінку В-клітин в процесі імунної реакції на антигенний стимул. Така взаємодія супроводжується виробленням клонів В-клітини, які можуть зазнавати мутації різного ступеня залежно від сили їх зв'язування (заходи аффіності) з будь-яким епітопом антигену. У магістерській дипломній роботі перевагу алгоритм клональної селекції використовувався для формування навчальних вибірок самоорганізованої карти.

3. Для захисту глобальних та локальних мереж є безліч сучасних системних виявлення мережевих атак. Найбільш застосованими та найбільш популярними є системи, в основі яких лежить використання методів штучного інтелекту, а саме біо-інспірованих методів, здатних до гнучкого навчання та адаптації. На основі виконаного огляду прийнято рішення використовувати імунний підхід для виявлення мережевих атак, до переваг якого можна зазначити: адаптивність, здатність до навчання, гнучкість, швидко здатність аналізу даних та здатність виявляти маловідомі атаки. Ці переваги вигідно відрізняють імунний підхід, наприклад, від експертних систем, та є найбільш актуальними та потрібними, так як щоденно збільшується кількість нових мережевих атак.

2 ПОСТАНОВКА ЗАДАЧІ ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК

2.1 Причини використання методів штучного інтелекту для визначення мережесих атак на основі імунного підходу

Повідомлення про проникнення в комп'ютерні мережі та атаки на Web-сервера останнім часом з'являються все частіше. У багатьох випадках зловмисники обходять встановлені захисні засоби. Атаки здійснюються за дуже короткий термін, різноманіття загроз постійно збільшується, що не дозволяє виявити ці загрози та запобігти їм стандартними захисними засобами. Існуючі підходи мають певні особливості, які перешкоджають їх використанню (невисока швидкість роботи, низька точність). Цих недоліків позбавлена нейромережна технологія: багат шаровий перцептрон; мережа Кохонена; нейронечітка мережа (гібридна система) [28].

Мережесі атаки - це вторгнення в операційну систему віддаленого комп'ютера. Зловмисники роблять мережесі атаки, щоб захопити управління над операційною системою, привести її до відмови в обслуговуванні або отримати доступ до захищеної інформації. Мережесими атаками називають шкідливі дії, які виконують самі зловмисники (такі як сканування портів, підбір паролів), а також дії, які виконують шкідливі програми, встановлені на атакованому комп'ютері (такі як передача захищеної інформації зловмиснику).

Атаки поділяються на чотири основні категорії [2]:

- DoS (Back, Land, Neptune, Pod, Smurf, Teardrop);
- U2R (Buffer_overflow, Loadmodule, Perl, Rootkit);
- R2L (Ftp_write, Quess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster);
- Probe (Ipsweep, Hmap, Portsweep, Satan).

Атака типу «відмова в обслуговуванні» (DoS) - це спроба завдати шкоди, зробивши недоступною цільову систему, наприклад веб-сайт або додаток, для звичайних кінцевих користувачів. Зазвичай зловмисники генерують велику кількість пакетів або запитів, які в кінцевому рахунку перевантажують роботу цільової системи. Для здійснення атаки типу «розподілена відмова в

обслуговуванні» (DDoS) зловмисник використовує безліч зламаних або контрольованих джерел. Виділяють шість DoS атак: back, land, neptune, pod, smurf, teardrop.

U2R атаки передбачають отримання зареєстрованим користувачем привілеїв локального суперкористувача (мережевого адміністратора). Виділяють чотири типи U2R атак: buffer_overflow, loadmodule, perl, rootkit.

R2L атаки характеризуються отриманням доступу незареєстрованого користувача до комп'ютера з боку віддаленого комп'ютера. Виділяють вісім типів R2L атак: ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster.

Probe атаки полягають в скануванні мережевих портів з метою отримання конфіденційної інформації. Виділяють чотири типи Probe атак: ipsweep, nmap, portsweep, satan

Конкретні різновиди мережевих атак представлені в базі даних (БД) KDD-99[29]. Як навчальної множини виступає база KDD-99. Ця БД містить близько 5000000 записів про з'єднання. З'єднання – це послідовність TCP-пакетів за обмежений період, моменти початку та завершення якого чітко визначені й протягом якого дані передаються від IP-адреси відправника на IP-адресу приймача, використовуючи певний протокол. Кожен запис являє собою образ мережевого з'єднання, включає 29 параметр мережевого трафіку і промаркована як «атака» або «не атака». У базі представлені 22 типу атаки. При цьому атаки діляться на 4 основні категорії: DoS, U2R, R2L і Probe.

Для вирішення проблеми класифікації мережевих атак було прийняте рішення використовувати генеротивно-змагательний підхід: генерувати на основі даних, що є наявними, а нові дані за допомогою алгоритму клональної селекції. Отриманні дані додати до тих, що вже є наявними, та подати на вхід SOM, який визначає нормальний стан мережі чи визначає категорію атак. Класифікації класів атак категорій DoS, U2R, R2L, Probe будуть здійснюватися за допомогою багатошарових перцептронів (рис. 2.1) [30].

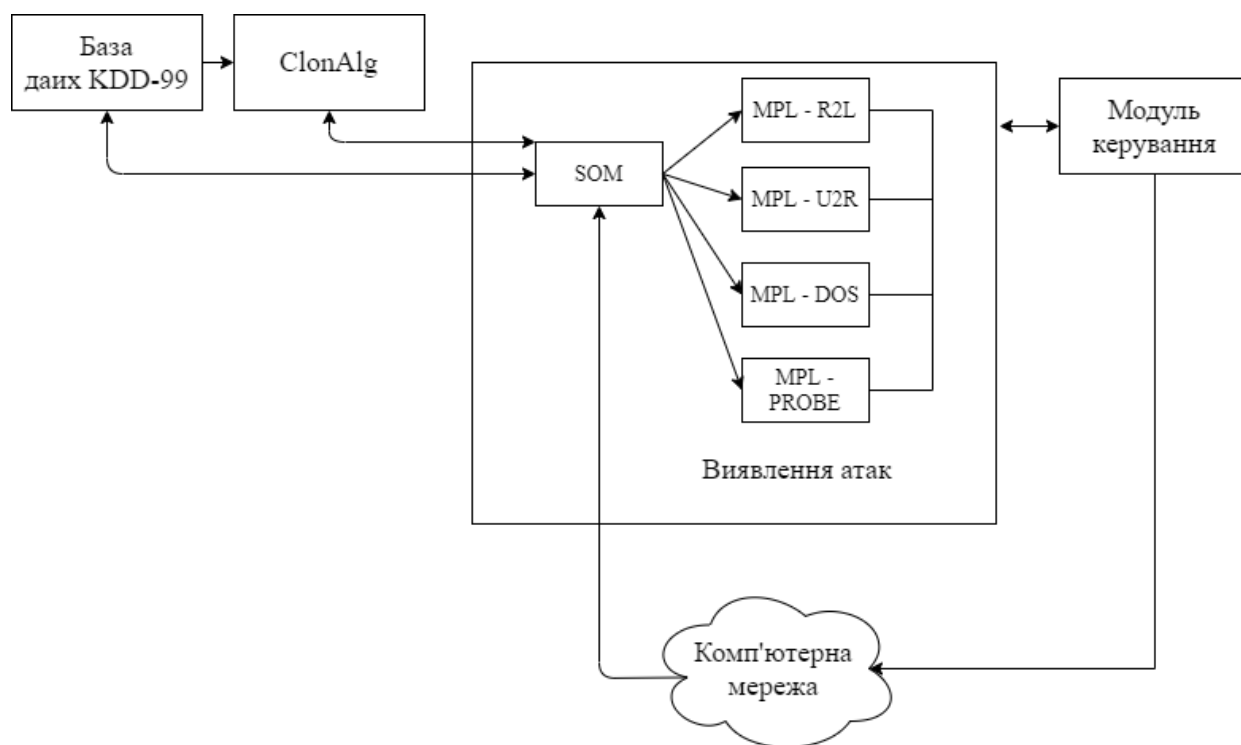


Рисунок 2.1 –Загальна схема виявлення мережеских атак

2.2 Основні висновки

Визначення атак на комп'ютерну мережу пропонується провести в два етапи: визначення категорії атаки, на основі використання SOM та алгоритму клональної селекції (перший етап); визначення типу атаки на відповідних MLP-DoS, MLP-U2R, MLP-R2L, MLP-Probe (другий етап). У якості початкових даних використовується відкрита база даних KDD-99.

3 ВИЗНАЧЕННЯ МЕРЕЖЕВИХ АТАК НА ОСНОВІ РОЗРОБЛЕНОГО ПРОГРАМНОГО КОМПЛЕКСУ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖНОЇ ТЕХНОЛОГІЇ ТА ІМУННОГО ПІДХОДУ

3.1 Визначення категорії мережевої атаки (перший етап)

3.1.1 SOM як основний метод розв'язання задачі

Самоорганізована карта Кохонена – нейронна мережа з навчанням без учителя, що виконує завдання візуалізації і кластеризації. Ідея мережі запропонована фінським вченим Т. Кохоненом. Вхідний шар мережі Кохонена (рис. 3.1) містить число нейронів, така ж кількість ознак набору даних. Кількість нейронів у результуючому шарі обчислювальних кластерів, які формуються моделлю. Кожен нейрон вхідного шару пов'язаний з усіма нейронами результуючого.

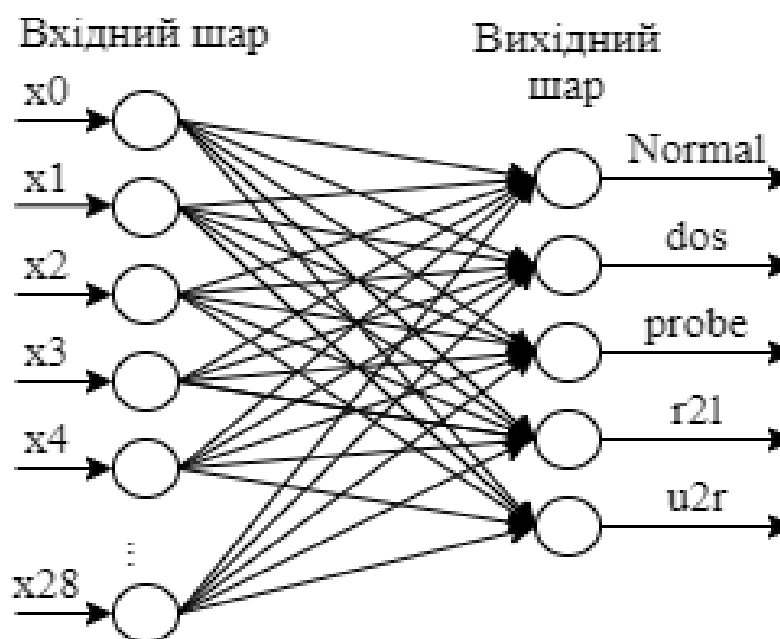


Рисунок 3.1 – Структура мережі Кохонена

SOM є комбінацією алгоритму кластеризації на основі мережі Кохонена і візуалізації її результатів на основі саморганізованих карт, які формуються за допомогою проектування зі збереженням топологічного подібності. Алгоритм навчання SOM наведено на рис. А.1 (у додатку А).

Відповідно до джерела [8] для виявлення і класифікації 9 з 22 типів атак досить 29 параметрів, що характеризують мережеві з'єднання. У таблиці 3.1 наведено співвідношення параметрів у реалізації і атрибутів KDD-99 [29].

Таблиця 3.1 – Співвідношення параметрів і атрибутів KDD-99

Позначення вхідного нейрону	Атрибут	Позначення вхідного нейрону	Атрибут
x0	Duration	x15	Srv error rate
x1	Protocol type	x16	Same srv rate
x2	Service	x17	Diff srv rate
x3	Flag	x18	Srv diff host rate
x4	Source bytes	x19	Dst host count
x5	Destination bytes	x20	Dst host srv count
x6	Land	x21	Dst host same srv rate
x7	Wrong fragment	x22	Dst host diff srv rate
x8	Urgent	x23	Dst same src port rate
x9	Hot	x24	Dst host srv diff host rate
x10	Count	x25	Dst host error rate
x11	Srv count	x26	Dst host srvb error rate
x12	Error rate	x27	Dst host error rate
x13	Srv error rate	x28	Dst host srv error rate
x14	Error rate		

$$\text{Нехай } X^m = (x_1^m, x_2^m, \dots, x_n^m) \quad m = 1, M \quad n = 1, N, \quad (3.1)$$

де X^m – початковий вектор;

M – кількість векторів;

N – довжина вектора.

Тоді роботу SOM можна описати наступним чином.

Крок 1. Нормалізація початкових даних за стовпцями:

Знаходимо максимум та мінімум кожного стовпця

$$Max_n = \max_n X_n^m \quad (3.2)$$

$$Min_n = \min_n X_n^m \quad (3.3)$$

Знаходимо допоміжні значення a та b за формулами (3.4) та (3.5)

$$a_n = \frac{1}{Max_n - Min_n}; \quad (3.4)$$

$$b_n = \frac{-Min_n}{Max_n - Min_n}, \quad (3.5)$$

де Min_n – мінімальне значення у даному стовпці, а Max_n – максимальне значення.

Обчислюємо нове значення x_n^m за формулою (3.6)

$$x_n^m = a_n \times x_n^m + b_n, \quad (3.6)$$

де x_n^m – нормалізоване значення початкового вектора.

Крок 2. Заповнення випадковим чином вектора ваг.

$$w_n^k = rand[0; 0.3] \quad k = 1, K, \quad (3.7)$$

де k - кількість класів.

Крок 3. Навчання векторів ваг

Крок 4. Для кожного x^i знаходимо найближчий вектор.

Крок 5. Обчислюємо нове значення w^j за формулою (3.8) та (3.9)

$$w^j = w^j + \lambda(x^i - w^j); \quad (3.8)$$

$$\lambda = \lambda - \Delta\lambda, \quad (3.9)$$

де λ – параметр навчання;

$\Delta\lambda$ – крок зменшення параметру навчання.

Крок 6. Якщо λ більше нуля переходимо до кроку 4, інакше – завершуємо навчання.

Класифікація нових даних, що поступають, відбувається наступним чином: нормалізуємо ці дані та шукаємо найближчий вектор ваг. Номер найближчого вектору ваг і буде класом нових даних.

3.1.2 Клональна селекція як додатковий метод розв'язання задачі

Клональний алгоритм відбору – клас алгоритмів, заснованих на теорії клонової селекції набутого імунітету, що пояснює, як Б- і Т-лімфоцити покращують їх реакцію на антигени з плином часу, що називається affinity maturation. Першою реалізацією алгоритму клональної селекції є CLONALG, що складається з наступних кроків, представлених на рис. А.2 (у додатку А).

Де Кастро і Фон Зубен розробили алгоритм клонального відбору CLONALG на основі теорії клонального відбору імунної системи [11, 12]. Клональний відбір заснований на способі адаптації В-клітин і Т-клітин для відповідності і знищення чужорідних клітин. Цей алгоритм може виконувати розпізнавання образів і адаптуватися для вирішення завдань мультимодальної оптимізації та може бути описаним наступним чином.

Крок 1. Генерація (випадковим чином) набору P можливих антитіл:

$$P = Pr + M, \quad (3.10)$$

де M – елементи пам'яті; Pr – Популяція.

Крок 2. Виберіть n кращих антитіл P_n на основі афінності;

Крок 3. Клонувати ці n кращих антитіл пропорційно їх афінності, використовуючи

$$N_c = \sum_{i=1}^n P \text{round}\left(\frac{\beta N}{i}\right), \quad (3.11)$$

де N_c – загальна кількість клонів, згенерованих для кожного з антигенів, таких як цільова функція,

β – множником,

N – загальним числом антитіл і оператором, який округлює свій аргумент до найближчого цілого числа .

Кожен член цієї суми відповідає розміру клону кожного обраного антитіла, наприклад, для $N = 100$, і антитіло з найбільшою спорідненістю продукуватиме 100 клонів; антитіло з другою за величиною спорідненістю продукує 50 клонів і т. д., приводячи до тимчасового набору клонів C .

Крок 4. Застосувати мутацію до тимчасових клонів. Ступінь мутації обернено пропорційна афінності. Дозрілі антитіла генеруються C^* .

Крок 5. Повторно виберіть кращі елементи з C^* , щоб скласти набір пам'яті M . Деякі члени P можуть бути замінені іншими поліпшеними членами C^* .

Крок 6. Замінити d антитіла новими, щоб ввести концепцію різноманітності.

3.1.3 Формування вибірок для навчання SOM (підготовчий етап)

Формування загальної вибірки, на якій здійснювалося навчання самоорганізованої карти Кохенена, виконано на основі даних [29].

Початкові навчальні вектори формуються у вигляді таблиці за допомогою пакету Excel 2019 у одному файлі – «traffic.txt». У цьому файлі містяться дані, що подаються на вхід у вигляді 369 вектора. Фрагмент змісту файлу «traffic.txt» наведено у додатку А.

Останній стовпець, що містить назву атаки, чи нормального стану було видалено. Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число.

Останній стовпець, що містить назву атаки, чи нормального стану було видалено. Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число. Аналогічно підготовлюємо тестові вектори, що зберігаються у вигляді таблиці за допомогою пакету Excel 2019 у одному файлі –

«test.txt». У цьому файлі містяться дані, що подаються на вхід у вигляді 1000 векторів. Фрагмент змісту файлу «test.txt» наведено на рис. 3.2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
88	0	0	33	6	1492	649186	0	0	0	0	0	0	0	0	0	0	0	0	0
89	31	0	33	9	1345	10036	0	0	0	0	0	1	16	0	0	16	0	0	0
90	41	0	33	9	1334	162	0	0	0	0	0	0	0	0	0	0	0	0	0
91	0	0	33	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
92	0	0	33	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
93	0	0	33	9	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
94	0	0	33	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	0	0	33	9	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
96	0	0	33	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
97	0	0	33	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
98	0	0	33	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
99	0	0	33	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
00	1	0	4	9	0	988002	0	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	4	9	0	467968	0	0	0	0	0	0	0	0	0	0	0	0	0
02	198	0	7	9	562	9139	0	0	0	3	0	1	22	1	0	39	4	2	0
03	192	0	5	9	119	426	0	0	0	2	0	1	0	0	0	0	1	0	0
04	179	0	5	9	87	319	0	0	0	1	0	1	0	0	0	0	1	0	0
05	0	0	4	9	866	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06	12	0	0	9	51	8127	0	0	0	2	0	1	0	1	0	0	0	0	1
07	0	0	0	9	51	8127	0	0	0	2	0	1	0	1	0	0	0	0	1
08	0	0	0	9	51	8127	0	0	0	2	0	1	0	1	0	0	0	0	1
09	10	0	4	9	0	5155468	0	0	0	0	0	0	0	0	0	0	0	0	0
10	9	0	4	9	0	5153771	0	0	0	0	0	0	0	0	0	0	0	0	0
11	9	0	4	9	0	5153460	0	0	0	0	0	0	0	0	0	0	0	0	0
12	10	0	4	9	0	5153385	0	0	0	0	0	0	0	0	0	0	0	0	0
13	10	0	4	9	0	5153154	0	0	0	0	0	0	0	0	0	0	0	0	0
14	9	0	4	9	0	5151049	0	0	0	0	0	0	0	0	0	0	0	0	0
15	9	0	4	9	0	5150938	0	0	0	0	0	0	0	0	0	0	0	0	0
16	10	0	4	9	0	5150877	0	0	0	0	0	0	0	0	0	0	0	0	0
17	10	0	4	9	0	5150841	0	0	0	0	0	0	0	0	0	0	0	0	0

Рисунок 3.2 – Фрагмент змісту файлу «test.txt»

3.1.4 Програмна реалізація моделі «SOM_Clon»

Прийняте рішення здійснювати програмну реалізацію алгоритму самоорганізованої карті та алгоритму клональної селекції з використанням мови програмування C++ через ряд переваг: C++ більш швидкодіюча, що має суттєве значення для навчання та виявлення можливих атак; компілятори C++ є на кожній ОС і більшість програм легко переносяться з платформи на платформу; має багато математичних бібліотек.; автор дипломної магістерської роботи, має знання та навички роботи з даною мовою програмування.

На рисунку 3.3. наведено основне меню програмної моделі «SOM_Clon»

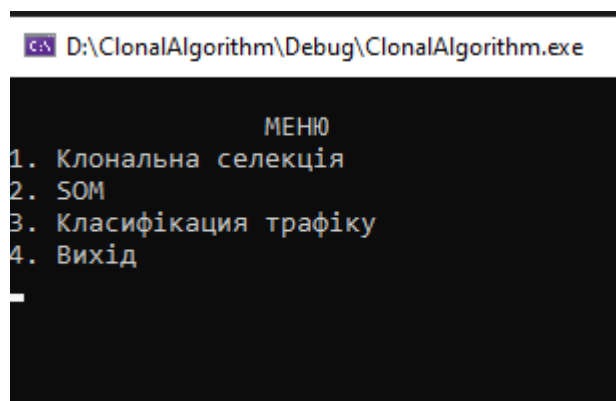


Рисунок 3.3 – Вікно програми «SOM_Clon»

До складу створеної програми надходять наступні модулі: «SOM» та «ClonAlg». Програмний модуль «ClonAlg» реалізований за алгоритмом клональної селекції з використанням наступних функцій:

- *void ReadingAntiGen()* – для зчитування з файлу «антигенів»;
- *vector<double> RandomGenAB()* – генерує випадкове число та повертає його;
- *void GenerateAntiBodys()* – генерує «антитіло»;
- *void ComputationParamsAntiBody()* – виконує обчислення параметрів антитіла;
- *void ClassificationAB()* – виконує класифікацію отриманих «антитіл»;
- *void SortAB()* – виконує сортування «антитіл» за їх аффіністю;
- *void GenNewPopulation()* – генерує нову популяцію «антитіл» на базі вже існуючих;
- *void WritingDBAntiBody()* – виконує запис отриманих «антитіл».

Програмний модуль «SOM» реалізований за алгоритмом мережі Кохонена з використанням наступних функцій:

- *void Classification()* – здійснює безпосередньо класифікацію даних, та підрахунок якісних параметрів класифікації;
- *void FindAnAndBn()* – знаходить допоміжні значення a та b за формулами (3.4) та (3.5);
- *void GenerateWRandom()* – задає початкові випадкові ваги класів;
- *void NormalisationValue()* – виконує нормалізацію числових даних;
- *void ReadingTrafficAndFindMinMax(string NameFile)* – виконує зчитування даних, та знаходить мінімальне та максимальне значення.
- *void Studing()* – виконує навчання мережі Кохонена.

3.2 Визначення класу мережевої атаки (другий етап)

3.2.1 MLP у якості математичного апарату

Багатошаровий перцептрон – окремий випадок перцептрону Розенблатта, в якому один алгоритм зворотного поширення помилки навчає всі шари. Багатошаровий перцептрон – це клас штучних нейронних мереж прямого поширення, що складаються як мінімум з трьох шарів: вхідного, прихованого і результуючого. За винятком вхідних, всі нейрони використовують нелінійну функцію активації. MLP використовує навчання з вчителем і алгоритм зворотного поширення помилки. У якості активаційних функцій нейронів використовуються сигмоїдальні: логістична або гіперболічний тангенс.

Багато сучасних нейронних мереж сконструйовані з формальних нейронів, що віддалено нагадують свій біологічний прототип. Структура нейрона має вигляд, представлений на рис. 3.4, при наступних позначеннях: x_1, \dots, x_n – значення, що надходять на входи (синапси) нейрона; w_1, \dots, w_n – ваги синапсів, які можуть бути як гальмують, так і підсилюють; S – зважена сума вхідних сигналів, що визначається як $S = \sum_{i=1}^n w_i x_i - T$; T – поріг нейрона (у багатьох моделях обходяться без нього); F – функція активації нейрона, що перетворює зважену суму в вихідний сигнал $y = F(S)$. Вид функції активації (див. додаток А) може мати різне математичне вираз, вибір якого визначається характером вирішуваних завдань.

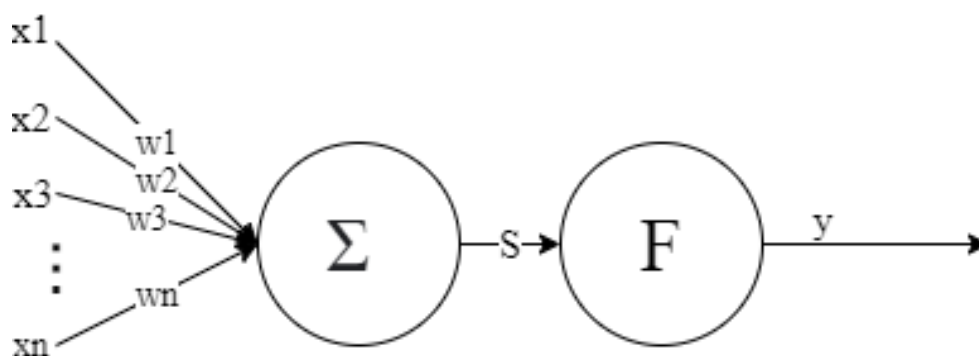


Рисунок 3.4 - Структура штучного нейрона

3.2.2 Формування вибірок для навчання MLP (підготовчий етап)

Для навчання MLP відбувається формування чотирьох навчальних вибірок, на основі [29].

Початкові навчальні вектори формуються у вигляді чотирьох таблиць за допомогою пакету Excel 2019 у форматі csv, що обумовлено зручністю роботи з даним форматом за допомогою бібліотеки «Keras» на мові «Python». Ці таблиці зберігаються у наступних чотирьох файлах: «r2l.1.csv», «u2r.1.csv», «dos.1.csv» та «probe.1.csv».

У файлі «r2l.1.csv» містяться дані, що подаються на вхід MLP-R2L у вигляді 103 векторів. Фрагмент змісту файлу «r2l.1.csv» наведено у додатку

Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число за допомогою функції «Categorical» бібліотеки «Pandas» на мові «Python» наступним чином:

```
train_r2l['protocol_type'] = pd.Categorical(train_r2l['protocol_type'])
train_r2l['protocol_type'] = train_r2l.protocol_type.cat.codes
train_r2l['service'] = pd.Categorical(train_r2l['service'])
train_r2l['service'] = train_r2l.service.cat.codes
train_r2l['flag'] = pd.Categorical(train_r2l['flag'])
train_r2l['flag'] = train_r2l.flag.cat.codes
```

У файлі «u2r.1.csv» містяться дані, що подаються на вхід MLP-U2R у вигляді 34 векторів. Фрагмент змісту файлу «u2r.1.csv» наведений у додатку Б.

Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число за допомогою функції «Categorical» бібліотеки «Pandas» на мові «Python».

У файлі «dos.1.csv» містяться дані, що подаються на вхід MLP-DOS у вигляді 849 векторів. Фрагмент змісту файлу «dos.1.csv» наведений у додатку Б.

Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число за допомогою функції «Categorical» бібліотеки «Pandas» на мові «Python».

У файлі «probe.1.csv» містяться дані, що подаються на вхід MLP-PROBE у вигляді 18070 векторів. Фрагмент змісту файлу «probe.1.csv» наведений у додатку Б.

Всю текстову інформацію, що залишилась, перетворюємо на числову, ставлячи у відповідність число за допомогою функції «Categorical» бібліотеки «Pandas» на мові «Python».

Аналогічно підготовлюємо тестові вектори, що зберігаються у вигляді таблиць за допомогою пакету Excel 2019 у чотирьох файлах: «r2l.test.csv», «u2r.test.csv», «dos.test.csv» та «probe.test.csv».

У файлі «r2l.test.csv» містяться дані, що подаються на вхід у вигляді 35 векторів. Фрагмент змісту файлу «r2l.test.csv» наведено на рис. 3.5.

r2l.test.csv - Excel																							
Что вы хотите сделать?																							
Общий											Стили												
Перенести текст											Условное форматирование												
Объединить и поместить в центре											Форматировать как таблицу												
Выравнивание											Стили												
Число											Ячейки												
Автосумма											Вставить Удалить Формат												
Заполнить											Сортировка и фильтр												
Очистить											Найти и выделить												
Редукция																							
duration																							
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	duration	protocol	service	flag	src_bytes	dst_bytes	land	wrong_fraction	urgent	hot	count	srv_count	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	
2	26	tcp	ftp	SF	116	451	0	0	0	0	2	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1	1.100	0.000	1.1
3	134	tcp	login	SF	100	39445	0	0	0	2	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	2	1.050	1.000	0.1
4	0	tcp	ftp_data	SF	613	0	0	0	0	0	0	1	2.000	0.000	0.000	0.000	1.000	0.000	1.000	1	84.100	0.000	1.4
5	0	tcp	ftp_data	SF	0	5	0	0	0	0	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	2	85.100	0.000	1.4
6	32	tcp	ftp	SF	104	449	0	0	0	0	2	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1	1.100	0.000	1.4
7	67	tcp	login	SF	157	2703	0	0	0	1	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	2	1.050	1.000	0.1
8	0	tcp	ftp_data	SF	676	0	0	0	0	0	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1	4.100	0.000	1.4
9	0	tcp	ftp_data	SF	0	5	0	0	0	0	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	2	5.100	0.000	1.4
10	23	tcp	telnet	SF	104	276	0	0	0	0	0	1	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1	2.100	0.000	1.4
11	60	tcp	telnet	SF	125	179	0	0	0	0	1	1	1.100	1.000	0.000	0.000	1.000	0.000	0.000	1	1.100	0.000	1.4
12	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.050	0.500	0.500	0.500	1.000	0.000	0.000	2	2.100	0.000	0.1
13	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.000	0.000	1.000	1.000	1.000	0.000	0.000	3	3.100	0.000	0.1
14	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	1	1.000	0.000	1.000	1.000	1.000	0.000	0.000	4	4.100	0.000	0.1
15	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.000	0.000	1.000	1.000	1.000	0.000	0.000	5	5.100	0.000	0.1
16	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.000	0.000	1.000	1.000	1.000	0.000	0.000	6	6.100	0.000	0.1
17	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.000	0.000	1.000	1.000	1.000	0.000	0.000	7	7.100	0.000	0.1
18	0	tcp	telnet	RSTO	125	179	0	0	0	0	1	2	2.000	0.000	1.000	1.000	1.000	0.000	0.000	8	8.100	0.000	0.1

Рисунок 3.5 – Фрагмент змісту файлу «r2l.test.csv»

У файлі «u2r.test.csv» містяться дані, що подаються на вхід у вигляді 10 векторів. Фрагмент змісту файлу «u2r.test.csv» наведено на рис. 3.6.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	duration	protocol	service	flag	src_bytes	dst_bytes	land	wrong_fraction	urgent	hot	count	srv_count	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio	error_ratio
2	0	tcp	ftp_data	SF	0	5696	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	tcp	ftp_data	SF	0	5928	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0	tcp	ftp_data	SF	0	5020	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	0	tcp	ftp_data	SF	0	2072	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
6	60	tcp	telnet	SF	2328	4551	0	0	0	0	3	0	1	1	1	0	0	0	0	0	0	0
7	158	tcp	telnet	SF	1567	3095	0	0	0	0	3	0	1	4	1	0	0	1	0	0	0	0
8	103	tcp	telnet	SF	302	8876	0	0	0	0	2	0	1	4	1	0	3	4	2	1	0	0
9	0	tcp	ftp_data	SF	0	5921	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
10	0	tcp	ftp_data	SF	0	5014	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
11																						
12																						
13																						
14																						
15																						

Рисунок 3.6 – Фрагмент змісту файлу «u2r.test.csv»

У файлі «dos.test.csv» містяться дані, що подаються на вхід у вигляді 200 векторів. Фрагмент змісту файлу «dos.test.csv» наведено на рис. 3.7.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
duration	protocol	service	flag	src_bytes	dst_bytes	land	wrong_fraction	urgent	hot	count	srv_count	error_rate	srv_error	error_rate	same_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv
1	0	tcp	http	SF	54540	8314	0	0	0	2	1	2	0.00	0.00	0.00	0.50	1.00	0.00	1.00	1	1	1.00
2	0	tcp	http	SF	54540	8314	0	0	0	2	2	3	0.00	0.00	0.00	0.33	1.00	0.00	0.67	2	2	1.00
3	0	tcp	http	SF	54540	8314	0	0	0	2	3	4	0.00	0.00	0.00	0.25	1.00	0.00	0.50	3	3	1.00
4	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	4	4	1.00
5	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	5	5	1.00
6	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	6	6	1.00
7	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	7	7	1.00
8	0	tcp	http	RSTR	54540	8314	0	0	0	2	4	4	0.00	0.00	0.25	0.25	1.00	0.00	0.00	8	8	1.00
9	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.20	0.20	1.00	0.00	0.00	9	9	1.00
10	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.20	0.20	1.00	0.00	0.00	10	10	1.00
11	0	tcp	http	SF	54540	8314	0	0	0	2	6	6	0.00	0.00	0.17	0.17	1.00	0.00	0.00	11	11	1.00
12	0	tcp	http	SF	54540	8314	0	0	0	2	7	7	0.00	0.00	0.14	0.14	1.00	0.00	0.00	12	12	1.00
13	0	tcp	http	SF	54540	8314	0	0	0	2	6	6	0.00	0.00	0.17	0.17	1.00	0.00	0.00	13	13	1.00
14	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	14	14	1.00
15	0	tcp	http	SF	54540	8314	0	0	0	2	6	6	0.00	0.00	0.00	0.00	1.00	0.00	0.00	15	15	1.00
16	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	16	16	1.00
17	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	17	17	1.00
18	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	18	18	1.00
19	0	tcp	http	RSTR	54060	7300	0	0	0	1	4	4	0.00	0.00	0.25	0.25	1.00	0.00	0.00	19	19	1.00
20	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.20	0.20	1.00	0.00	0.00	20	20	1.00
21	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.25	0.25	1.00	0.00	0.00	21	21	1.00
22	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.25	0.25	1.00	0.00	0.00	22	22	1.00
23	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.20	0.20	1.00	0.00	0.00	23	23	1.00
24	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	24	24	1.00
25	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	25	25	1.00
26	0	tcp	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	26	26	1.00
27	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	27	27	1.00
28	0	tcp	http	SF	54540	8314	0	0	0	2	4	4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	28	28	1.00
29	0	trn	http	SF	54540	8314	0	0	0	2	5	5	0.00	0.00	0.00	0.00	1.00	0.00	0.00	29	29	1.00

Рисунок 3.7 – Фрагмент змісту файлу «dos.test.csv»

У файлі «probe.test.csv» містяться дані, що подаються на вхід у вигляді 5000 векторів. Фрагмент змісту файлу «probe.test.csv» наведено на рис. 3.8.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
duration	protocol	service	flag	src_bytes	dst_bytes	land	wrong_fraction	urgent	hot	count	srv_count	error_rate	srv_error	error_rate	same_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv	diff_srv
1	0	tcp	ftp_data	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	1	1	1.00
2	1	tcp	ftp	RSTO	0	133	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	2	2	1.00
3	7	tcp	ssh	RSTO	0	15	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	3	3	1.00
4	0	tcp	telnet	RSTO	0	15	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	4	4	1.00
5	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	5	5	1.00
6	0	tcp	smtp	SF	0	83	0	0	0	0	1	1	0.00	0.00	0.00	0.00	1.00	0.00	0.00	6	6	1.00
7	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	7	7	1.00
8	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	8	8	1.00
9	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	9	9	1.00
10	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	10	10	1.00
11	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	11	11	1.00
12	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	12	12	1.00
13	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	13	13	1.00
14	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	14	14	1.00
15	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	15	15	1.00
16	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	16	16	1.00
17	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	17	17	1.00
18	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	18	18	1.00
19	1	tcp	time	RSTO	0	4	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	19	19	1.00
20	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	20	20	1.00
21	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	21	21	1.00
22	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	22	22	1.00
23	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	23	23	1.00
24	0	tcp	name	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	24	24	1.00
25	0	tcp	whois	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	25	25	1.00
26	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	26	26	1.00
27	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	27	27	1.00
28	0	tcp	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	28	28	1.00
29	0	trn	private	REJ	0	0	0	0	0	0	1	1	0.00	0.00	1.00	1.00	1.00	0.00	0.00	29	29	1.00

Рисунок 3.8 – Фрагмент змісту файлу «probe.test.csv»

3.2.3 Програмна реалізація моделі «MLP»

Прийняте рішення здійснити реалізацію багат шарового перцептрона за допомогою мови програмування «Python» та бібліотек для машинного навчання: «Keras» [31], «TensorFlow» [32], «Pandas» [33] через їх зручність та ряд переваг.

«Python» – це найпопулярніша високорівнева мова програмування з динамічною семантикою. Вона досить проста для роботи і читання: її використання знижує вартість розробки та обслуговування програм. Одна з основних причин – чому «Python» використовується для машинного навчання полягає в тому, що у нього є безліч фреймворків, які спрощують процес написання коду і скорочують час на розробку.

«TensorFlow» – відкрита програмна бібліотека для машинного навчання, розроблена компанією «Google» для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття [32].

До переваг використання «TensorFlow» можна віднести: має велику кількість посібників та документації; пропонує потужні засоби моніторингу процесу навчання моделей і візуалізації (Tensorboard); підтримується великою спільнотою розробників і технічними компаніями; забезпечує обслуговування моделей; підтримує розподілене навчання.

Перевагами використання «Keras» є: прототипування швидке та просте; має простий та інтуїтивно-зрозумілий інтерфейс, відповідно зручний для новачків; має вбудовану підтримку для навчання на декількох GPU; може бути налаштований в якості оцінювачів для TensorFlow і навченості на кластерах GPU на платформі Google Cloud.

Бібліотека «TensorFlow» надає користувачу багато можливостей для створення та дослідження різних методів штучного інтелекту. Розглянемо функції для створення нейронної мережі, що використовувались у дипломній магістерській роботі.

```
tf.keras.Sequential(layers=None, name=None)
```

Ця функція послідовно групує лінійний стек шарів нейронної мережі. Sequential має метод `add`, що додає шар нейронної мережі у стек. Сам шар задається за допомогою наступної функції [31, 32]:

```
tf.keras.layers.Dense(
    units,
```

```

activation=None,
use_bias=True,
kernel_initializer="glorot_uniform",
bias_initializer="zeros",
kernel_regularizer=None,
bias_regularizer=None,
activity_regularizer=None,
kernel_constraint=None,
bias_constraint=None,
**kwargs )

```

де `units` – ціле додатне число, розмірність вихідного простору; `activation` – функція активації для використання. Якщо ви нічого не вказуєте, активація не застосовується (тобто лінійна активація: $f(x) = x$); `use_bias` – логічне значення, чи використовує шар зміщення; `kernel_initializer` – ініціалізатор для матриці ваг ядра; `bias_initializer` – ініціалізатор для вектора зміщення; `kernel_constraint` – функція обмеження, застосована до матриці ваг ядра; `bias_constraint` – функція обмеження, застосована до вектора зміщення.

Бібліотека «Keras» надає користувачу широкий вибір функцій активації, а саме [31]: логістична (сігмоїдна або м'який крок); TanH; Softsign; Exponential linear unit (ELU); Scaled exponential linear unit (SELU); SoftPlus; Softmax.

Бібліотека «TensorFlow» має функцію навчання нейронної мережі `fit`, що має наступну структуру [32]:

```

fit(x=None, y=None, batch_size=None, epochs=1, verbose=1, callbacks=None,
    validation_split=0.0, validation_data=None, shuffle=True,
    class_weight=None,
    sample_weight=None, initial_epoch=0, steps_per_epoch=None,
    validation_steps=None, validation_batch_size=None, validation_freq=1,
    max_queue_size=10, workers=1, use_multiprocessing=False)

```

де `x` – вхідні дані; `y` – цільові дані; `batch_size` – кількість зразків за оновлення градієнта. якщо не вказано, для `batch_size` буде встановлено значення

за умовчанням; `epochs` – кількість епох для навчання моделі; `verbose` – режим багатослів'я: 0 = беззвучний, 1 = індикатор прогресу, 2 = один рядок за епоху; `callbacks` – список екземплярів `keras.callbacks.callback`. список зворотних звертань, які слід застосувати під час навчання; `validation_split` – частка навчальних даних, що використовуватимуться як дані перевірки; `validation_data` – дані, за якими оцінюють втрати та будь-які показники моделі в кінці кожної епохи; `shuffle` – чи потрібно перемішувати дані навчання перед кожною епохою; `sample_weight` – необов'язковий масив ваг `numpy` для тренувальних зразків, що використовується для зважування функції втрат (лише під час тренування); `initial_epoch` – епоха, з якої слід розпочати тренування (корисно для відновлення попереднього тренувального циклу).

3.3 Структура програмного комплексу та його тестування

Вхідні параметри. Окремо в меню можливо власноруч задати бажані розміри створюваної популяції антигенів, та кількість кращих антигенів. Кращі з популяції будуть доповнювати вибірку для навчання мережі Кохонена. Є можливість задавати кількість класів та кількість ознак – параметрів, що досліджуються. Програма розроблена гнучкою, для можливості використання функцій з новими параметрами (наприклад іншою кількістю класів). Також вхідними даними є дані, що подаються на вхід нейронної мережі. Вони представлені у вигляді вектору, що складається з 29 параметрів, які перелічені у таблиці 3.1. Вхідні параметри програмного комплексу наведені у таблиці 3.2.

Результуючі характеристики. По завершенню роботи програми у файл «BDclonalg.txt» заносяться нові елементи для навчання, що згенеровані за допомогою клональної селекції. Елементи, що генеруються за допомогою клональної селекції, мають таку ж структуру, що і вхідні дані. Вони представлені у вигляді вектора з 29 параметром. Приклад результату наведено у додатку А. Результатом виконання навчання мережі Кохонена, є отримані вектора ваг класів, за допомогою яких надалі виконується класифікація. У даному випадку результатом навчання буде 5 векторів ваг класів, що відповідають наступним категоріям: нема атаки, dos, probe, r2l, u2l. Структура програмного комплексу

Таблиця 3.2 – Вхідні параметри програмного комплексу

Назва	Тип	Значення	Призначення
POPULATION_SIZE	int	1000	Розміри створюваної популяції антитіл
QUANTITY_BEST_ANTI_BODY	int	100	Кількість кращих антитіл
SIZE_CLASS_SIGNS	int	29	Кількість ознак класу
QUANTITY_CLASSES	int	5	Кількість класів
Ab	vector < vector<double> >	Зчитані з файлу «AntiGen.txt»	Початкова навчальна вибірка – антигени
TrafficValues	vector< std::vector<double> > >	Зчитані з файлу «traffic.txt»	Вектори вхідного трафіку

представлена на рис. 3.8. Його основу складають наступні програмні модулі: Main, ClonAlg, SOM та MLP. Модуль Main виконує функції меню для завдання початкових параметрів роботи програми та об'єднує роботу двох програмних модулів ClonAlg та SOM. У модулі ClonAlg розташовані функції, що відповідають за роботу клональної селекції (генерації нової популяції та обрання найкращих «антитіл»). Модуль SOM містить функції для навчання нейронної мережі, класифікації та нормалізації вхідних даних. У модулі MLP здійснюється класифікація відповідного до категорії клас атаки.

Дані, що поступають, проходять перевірку на наявність можливих загроз – виявлення аномалій. Якщо аномалія виявлена, то вона проходить класифікацію за чотирма класами можливих загроз: dos, probe, r2l, u2l, які в свою чергу мають ще підкласи загроз. У рамках даної магістерської дипломної роботи виконується класифікація по виявленню нормальної поведінки мережевого трафіку, та класифікація загроз на чотири категорії: dos, probe, r2l, u2l за допомогою SOM, після чого здійснюється класифікація відповідного до категорії класу атак, за допомогою модуля «MLP». Виявлені загрози за допомогою алгоритму клональної селекції отримують додаткові «антитіла», що запам'ятовуються.

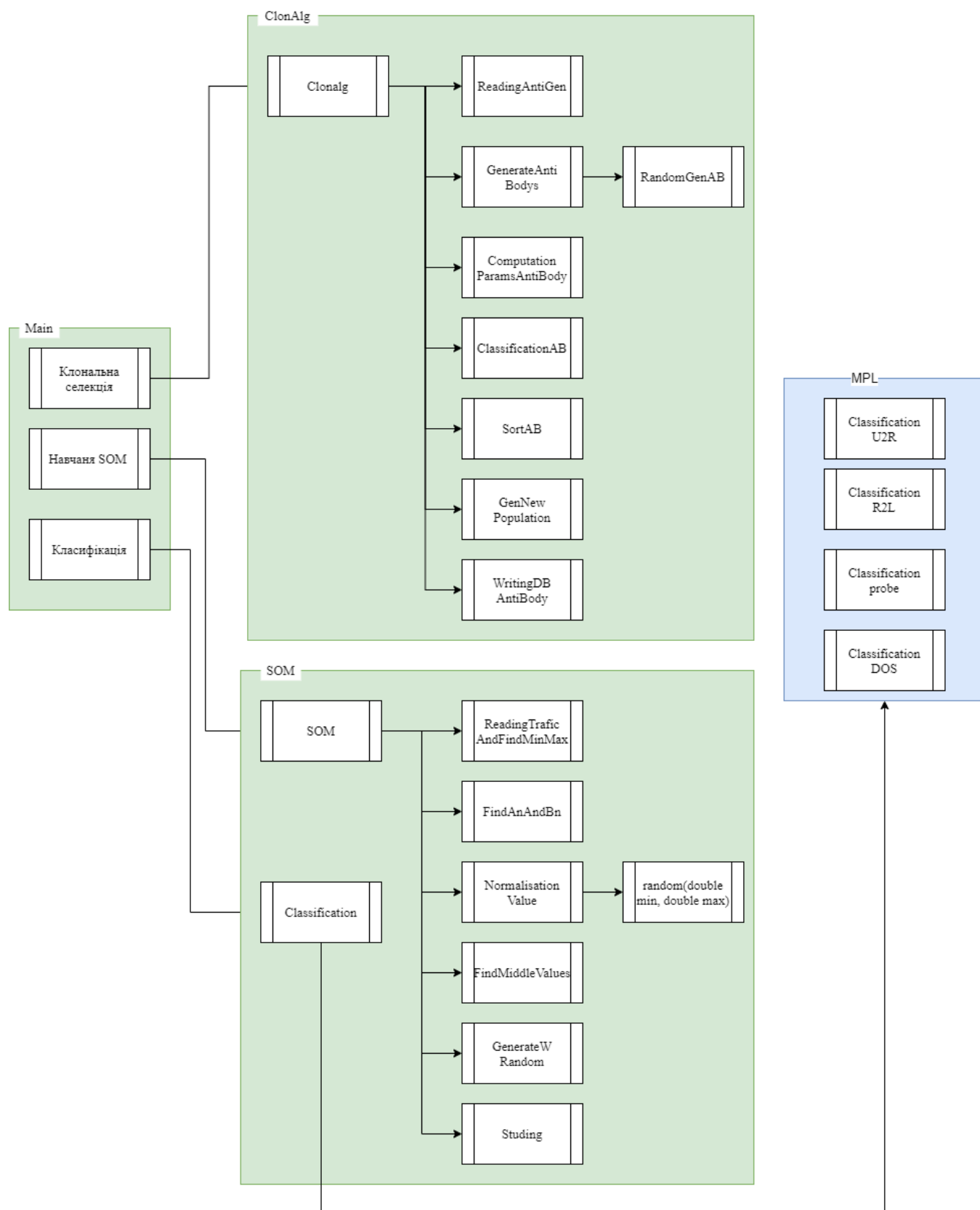


Рисунок 3.8 – Структура створеного програмного комплексу: модель «SOM_Clone», що створена на C++ та складається із Main, ClonAlg, SOM; модель «MLP», що створена на Python

Відбувається процес перенавчання векторів ваг з рахунком нових даних, які поступили, та утворених антигенів.

Тестування програмного комплексу. Для тестування програмного комплексу в моделі «SOM_Clon» реалізований механізм підрахунку якісних параметрів під час класифікації. Для кожної категорії мережевих атак підраховувались наступні показники: кількість неправильної класифікації; загальна кількість представника категорії у вибірці; кількість правильно класифікованих екземплярів категорії; кількість неправильно прийнятих за нормальний стан екземплярів (помилка першого роду); кількість помилково прийнятих за атаку нормальних станів (помилка другого роду). Ці показники отримані шляхом співставлення класу екземпляра з класом, встановленим алгоритмом. Крім того, під час тестування програмного комплексу в моделі «MPL» використовувались вбудовані функції бібліотеки «TensorFlow» та «Keras», а також на етапі compile моделі додати ряд параметрів, основні із яких:

Функція втрат (*Loss function*) – вимірює точність моделі під час навчання (необхідно мінімізувати цю функцію, щоб «направити» роботу моделі в правильному напрямку).

Метрики (*Metrics*) – використовуються для моніторингу тренування і тестування моделі. Наш приклад використовує метрику *accuracy*, що дорівнює частині правильно класифікованих зображень.

```
modelDos.compile(loss='sparse_categorical_crossentropy',
                 metrics=['accuracy'])
```

«Keras» надає великий спектр використання різних функцій втрат, наприклад:

- *sparse_categorical_crossentropy* – обчислює рідкісну категоріальну втрату кросцентропії;
- *categorical_crossentropy* – обчислює категоріальну втрату кросцентропії;
- *MeanSquaredError* – обчислює середнє значення квадратів помилок між мітками та прогнозами;

- *MeanAbsoluteError* – обчислює середнє значення абсолютної різниці між мітками та прогнозами;
- *MeanAbsolutePercentageError* – обчислює середню абсолютну процентну помилку між *y_true* та *y_pred*;
- *MeanSquaredLogarithmicError* – обчислює середню квадратичну логарифмічну помилку між *y_true* та *y_pred*.

Після навчання моделі із заданими параметрами отримана оцінка роботи моделі на тестовій вибірці з використанням функції *evaluate*:

```
loss, accuracy = model_dos.evaluate(test_dos)
print("Loss: ", loss)
print("Accuracy: ", accuracy)
```

3.4 Основні висновки

1. Для визначення мережевих атак на комп'ютерну мережу створений програмний комплекс, в основу якого покладені наступні моделі: «SOM_Clon» для визначення категорії мережевої атаки (на першому етапі); «MLP» для визначення класу мережевої атаки відповідно до категорії (на другому етапі).

2. З використанням алгоритму клональної селекції (для навчальної вибірки) створена в C++ модель «SOM_Clon», на вхід якої подаються: розмір сформованої «популяції» навчальних даних; кількість класів; вектор із 29 параметрів, що сформований на основі відкритої бази KDD-99.

3. З використанням наступних бібліотек: «TensorFlow»; «Keras»; «Pandas» створена в Python модель «MLP», на вхід якої подаються: *x0* - Duration, *x1* - Protocol type, *x2* - Servicem, *x3* - Flag, *x4* - Source bytes, *x5* - Destination bytes, *x6* - Land, *x7* - Wrong fragment, *x8* - Urgent, *x9* - Hot, та інші.

4 ОРГАНІЗАЦІЯ ДОСЛІДЖЕНЬ НА СТВОРЕННОМУ ПРОГРАМНОМУ КОМПЛЕКСІ ТА ЙОГО ВИКОРИСТАННЯ В НАВЧАЛЬНОМУ ПРОЦЕСІ

4.1 Визначення оптимальних параметрів нейронних мереж

4.1.1 Виявлення класу мережевих атак категорії «R2L»

Для визначення оптимальної структури багатошарового перцептрону для класифікації атак категорії «R2L» було проведено ряд експериментів. У ході яких досліджувалися: точність та середньоквадратична логарифмічна помилка (Mean Squared Logarithmic Error, MSLE), яку можна інтерпретувати як міру співвідношення між істинними та прогнозованими значеннями.

У ході експерименту встановлено, що кращі показники якості визначення мережевих атак категорії «R2L» надали багатошарові перцептрони з двома прихованими шарами.

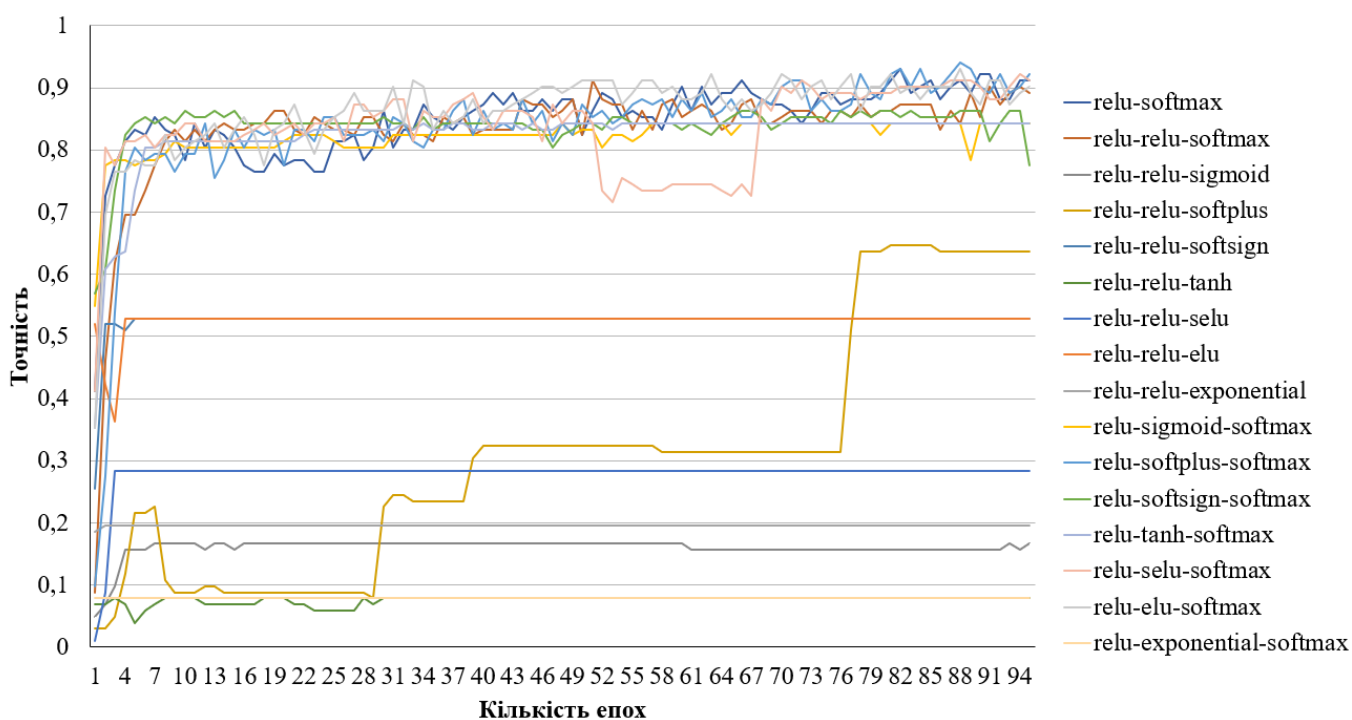


Рисунок 4.1 – Дослідження точності MLP- R2L від кількості епох навчання за різними функціями активації

На рис. 4.1 бачимо, що найкращі показники мають нейронні мережі з наступними функціями активації: логістичною функцією на першому прихованому шарі; функцією SoftPlus на другому прихованому шарі; функцією

На рис. 4.4 показано дослідження значення MSLE від кількості епох навчання для отриманої структури MLP- R2L, що побудована на основі даних табл. В.1 (у додатку В).

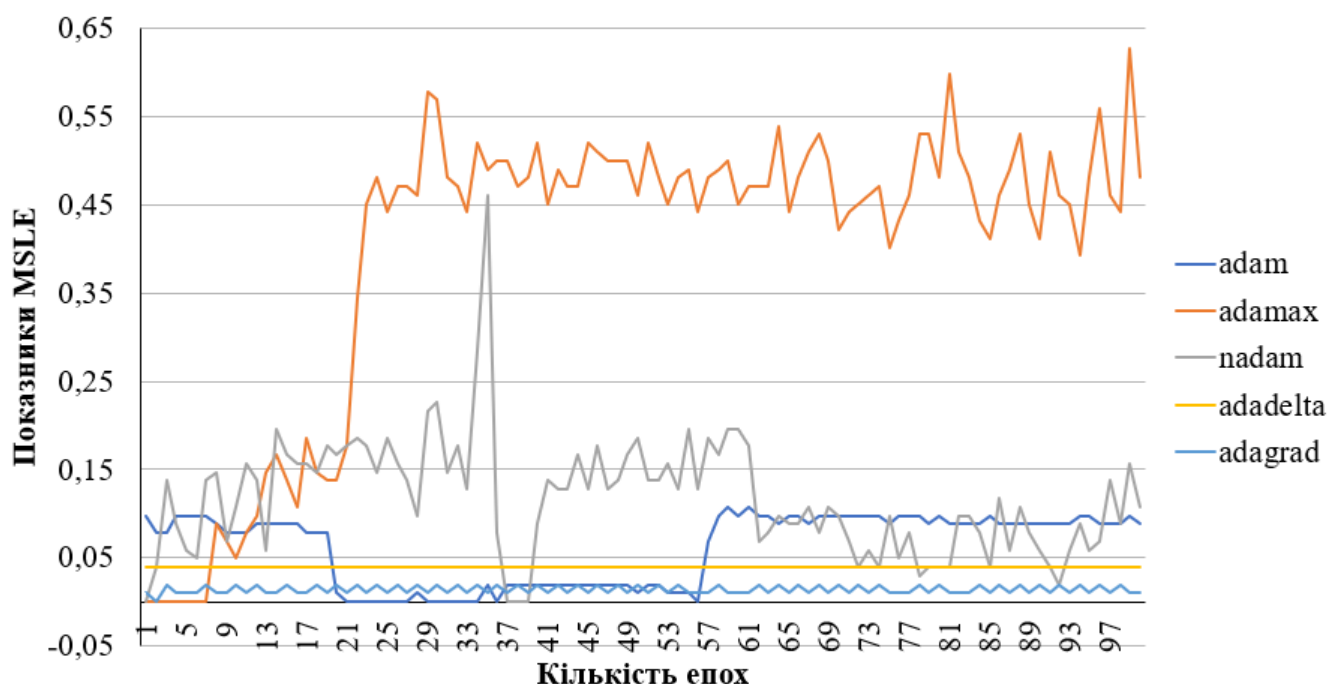


Рисунок 4.4 –Дослідження MSLE від кількості епох за різними методами оптимізації навчання MLP- R2L

Відповідно до отриманих даних (низькі показники MSLE) можна зробити висновки, що для MLP-R2L кращим методом оптимізації є метод «adam». Для досягнення достатнього рівня якості роботи алгоритму достатньо 25 епох навчання: зі збільшенням епох навчання не відбувається покращення роботи НМ, а починаючи з 57-ї епохи, спостерігається навпаки регрес і погіршення показників якості.

4.1.2 Виявлення класу мережевих атак категорії «U2R»

У ході експерименту встановлено, що кращі показники якості визначення мережевих атак категорії «U2R» в багатошарових перцептронах з одним прихованим шаром. На рис. 4.5 бачимо, що найкращі показники якості мають НМ, що мають наступні функції активації: логістичну функцію на першому прихованому шарі; функцію активації Softmax на результуючому шарі.

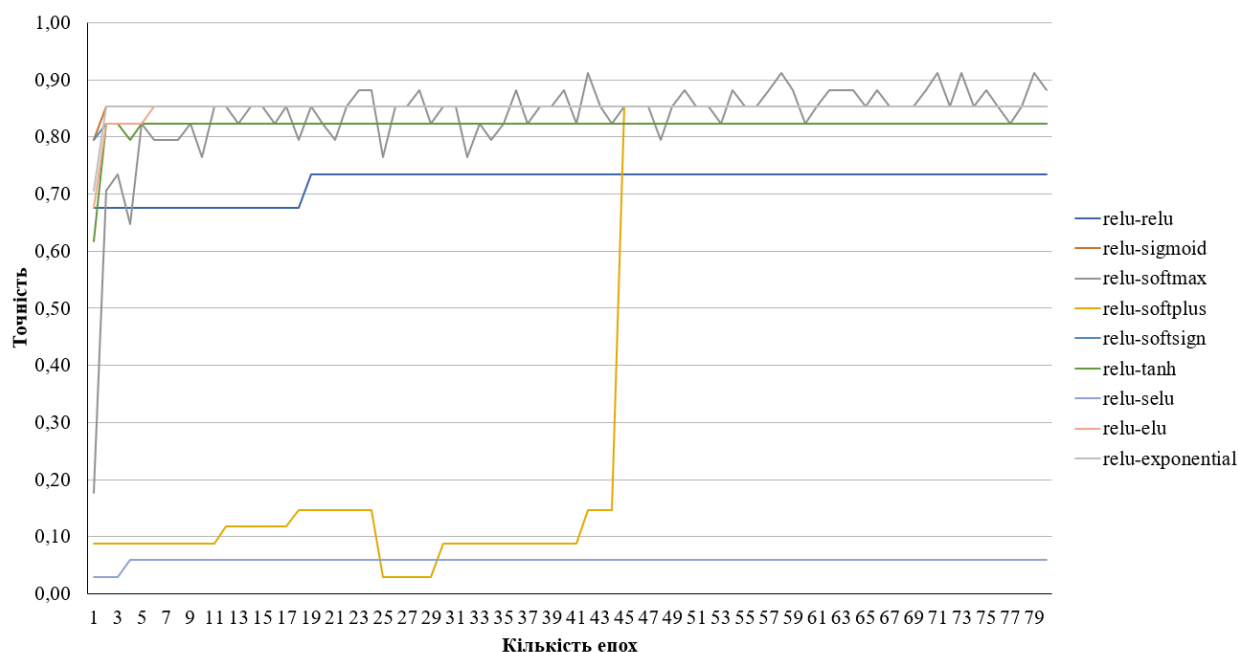


Рисунок 4.5 – Дослідження точності MLP-U2R від кількості епох навчання за різними функціями активації

Дослідження точності MLP-U2R від кількості епох навчання за різною кількістю прихованих нейронів наведено на рис. 4.6. Із рисунку видно, що найкращі показники якості показали НМ конфігурації 29-1-10-4.

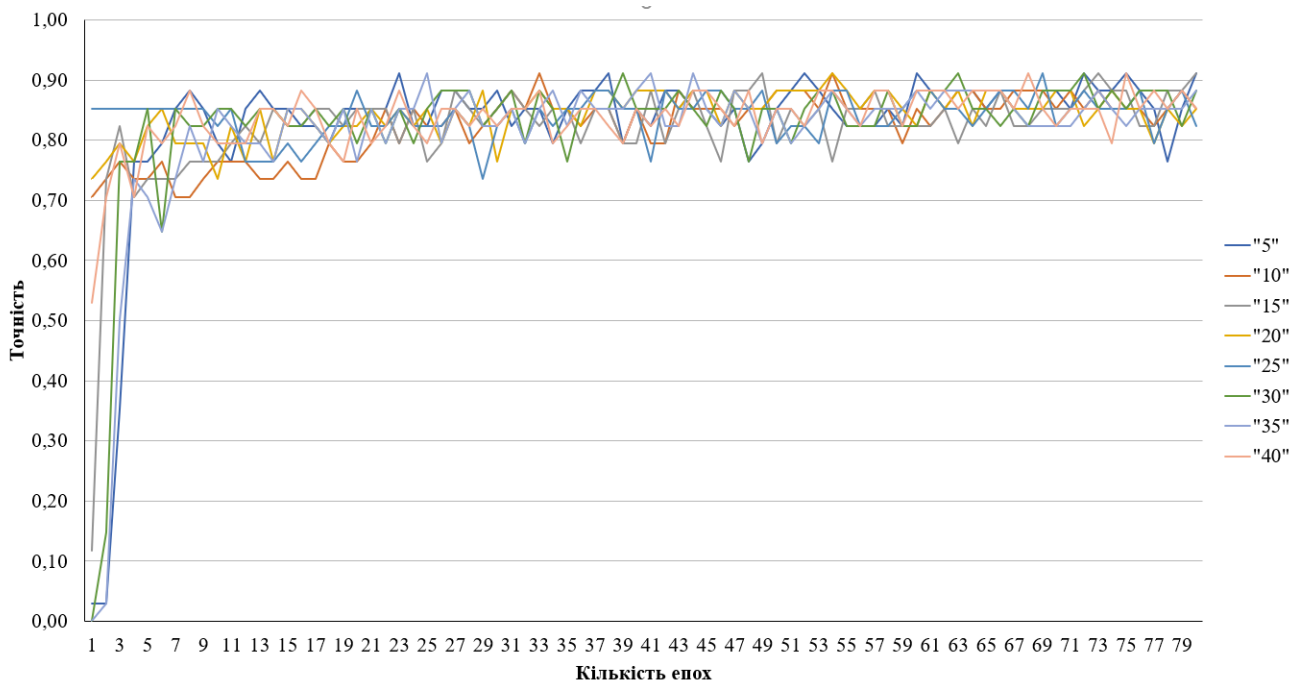


Рисунок 4.6 – Дослідження точності MLP-U2R від кількості епох навчання за різною кількістю прихованих нейронів

На рисунку 4.7 наведена отримана структура багатошарового перцептрону «U2R», де y_0 відповідає Buffer_overflow, y_1 – Loadmodule, y_2 – Perl, y_3 – Rootkit. Опис x_0, \dots, x_{28} наведено у розділі 3 (див. таблицю 3.1).

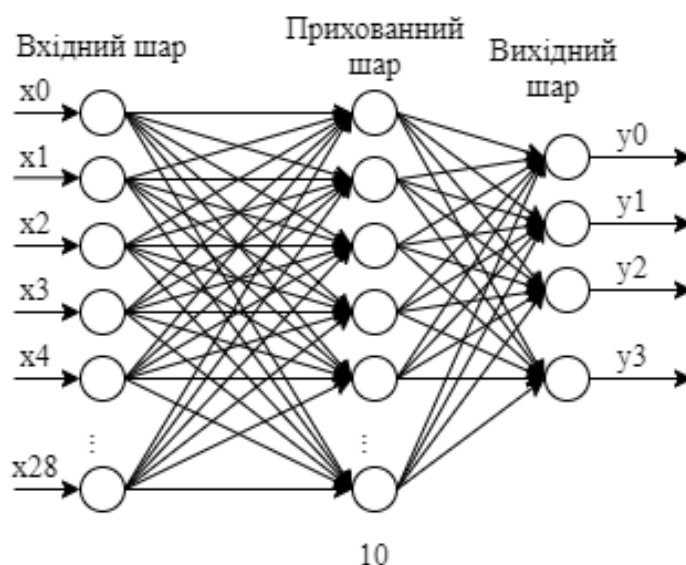


Рисунок 4.7 – Структура багатошарового перцептрону MLP-U2R

На рис. 4.8 показано дослідження MSLE від кількості епох навчання для отриманої структури MLP-U2R, що побудована на основі даних що представленні в таблиці В.2 (у додатку В). На підставі низьких показників MSLE зроблений висновок, що для MLP-U2R кращим методом є метод «adam». Для досягнення достатнього рівня якості достатньо 73 епох навчання.

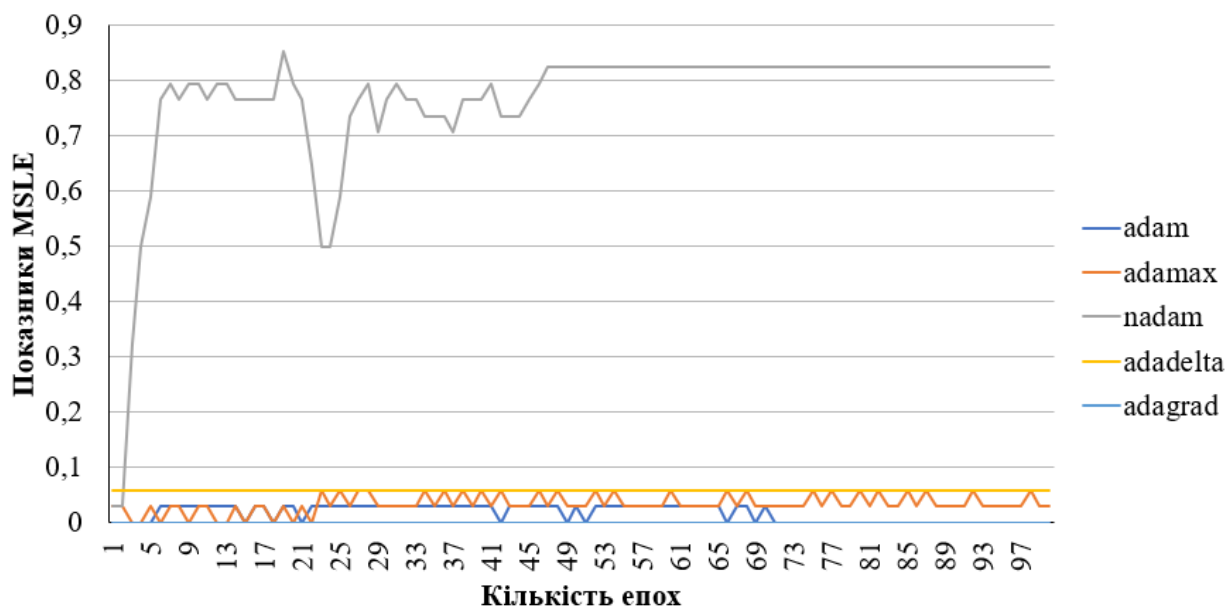


Рисунок 4.8 – Дослідження MSLE від кількості епох навчання за різними методами оптимізації навчання MLP-U2R

4.1.3 Виявлення класу мережевих атак категорії «DOS»

У ході експерименту встановлено, що кращі показники якості визначення мережевих атак категорії «DOS» в багатошарових перцептронах з одним прихованим шаром нейронів. На рис. 4.9 видно, що найкращі показники мають НМ з наступними функціями активації: логістичною функцією на першому прихованому шарі; функцією Softmax на результуючому шарі.

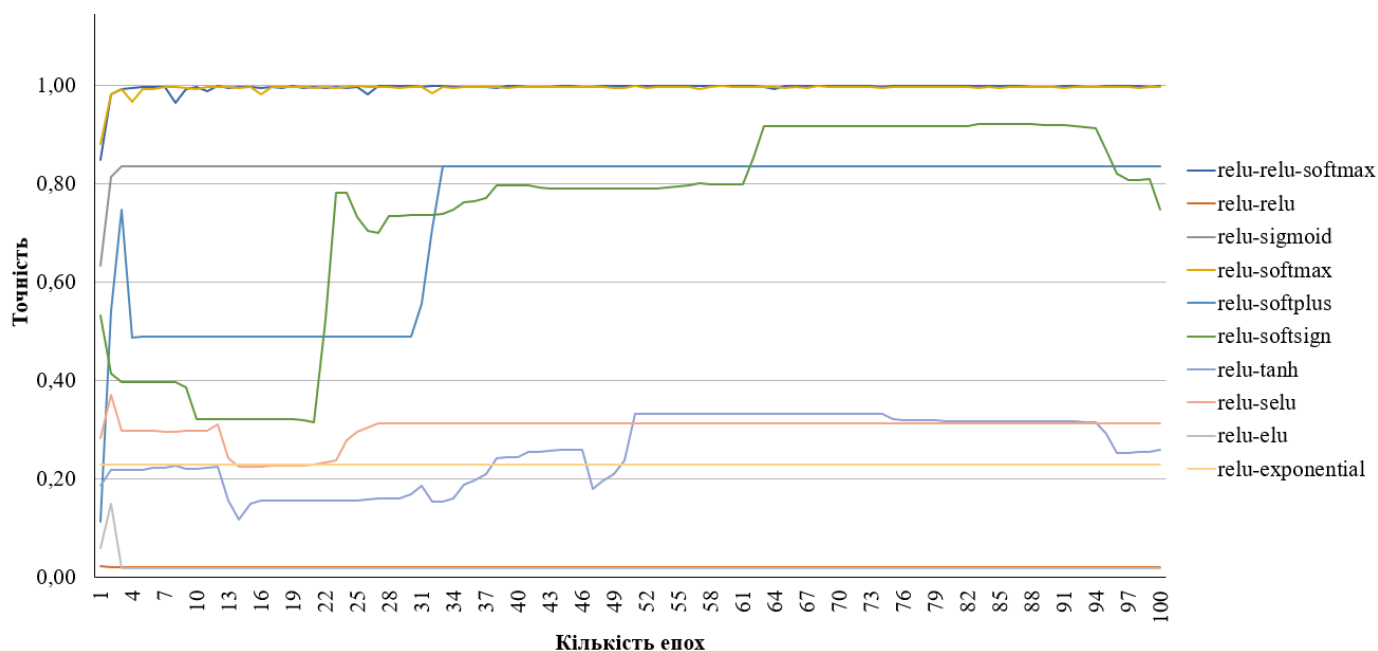


Рисунок 4.9 – Дослідження точності MLP-DOS від кількості епох навчання за різними функціями активації

Дослідження точності MLP-DOS від кількості епох навчання за різною кількістю прихованих нейронів наведено на рис. 4.10. Із рисунку видно, що найкращі показники якості показали НМ конфігурації 29-1-25-6. На рис. 4.11 наведена отримана структура MLP-DOS, де y_0 відповідає Back, y_1 – Land, y_2 – Neptune, y_3 – Pod, y_4 – Smurf, y_5 – Teardrop. Опис x_0, \dots, x_{28} наведено у розділі 3 (див. таблицю 3.1).

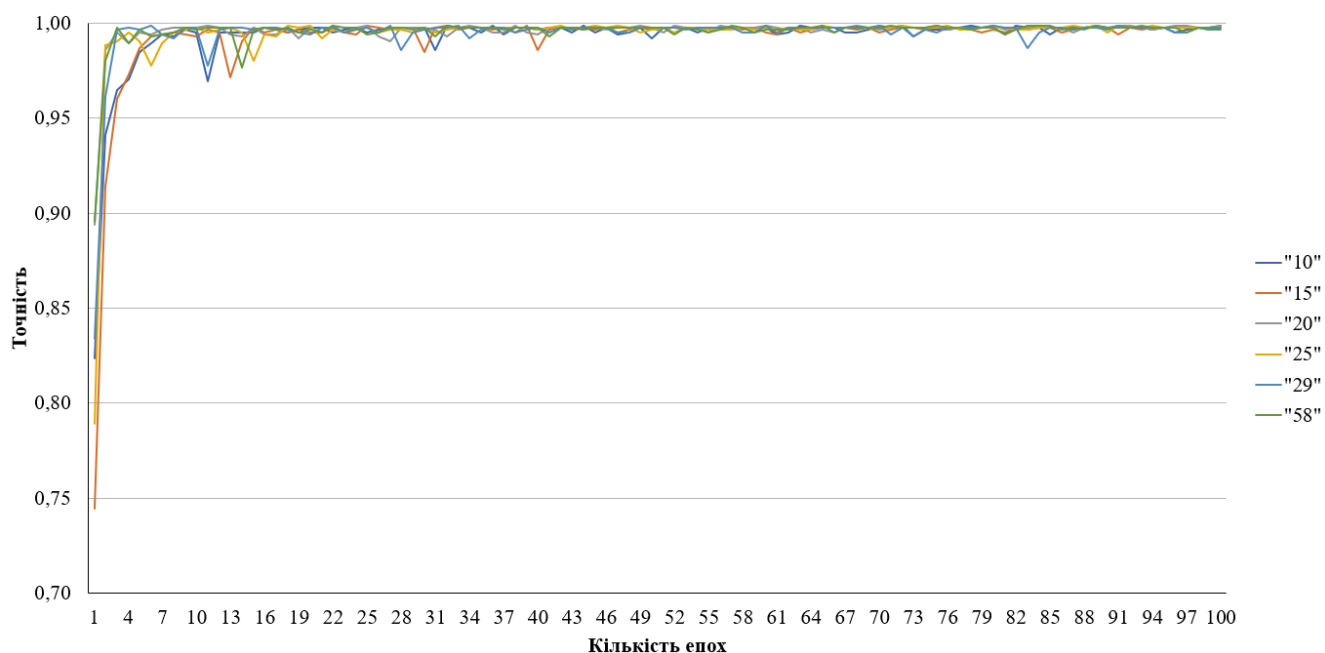


Рисунок 4.10 – Дослідження точності MLP-DOS від кількості епох навчання за різною кількістю прихованих нейронів

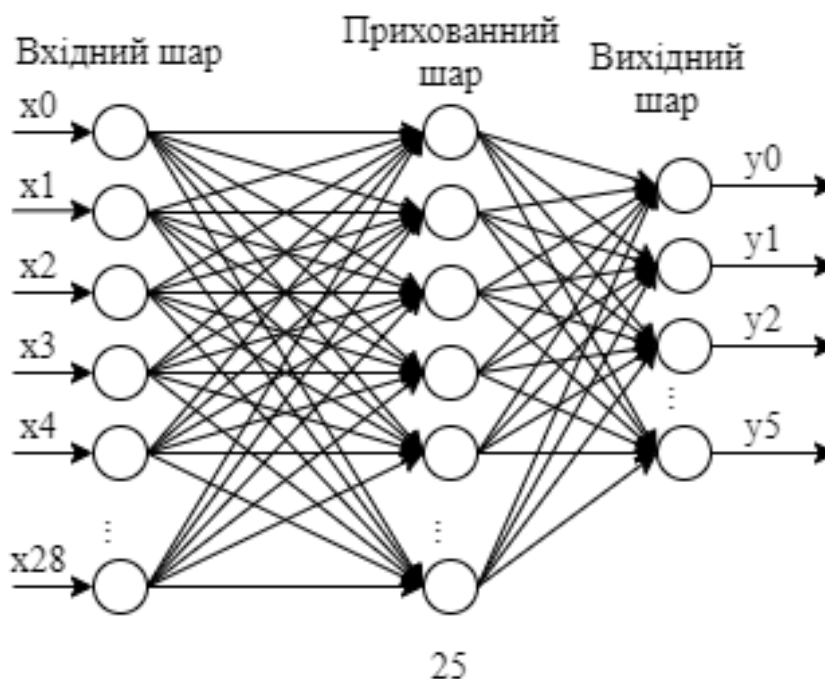


Рисунок 4.11 – Структура багатошарового перцептрону MLP-DOS

На рис. 4.12 представлено дослідження MSLE від кількості епох навчання для отриманої структури MLP-DOS, що побудована на основі даних таблиці В.3 (у додатку В). На підставі низьких показників MSLE зроблений висновок, що для MLP-DOS кращим методом оптимізації є метод «adadelata». Для досягнення достатнього рівня якості роботи алгоритму достатньо 25 епох навчання.

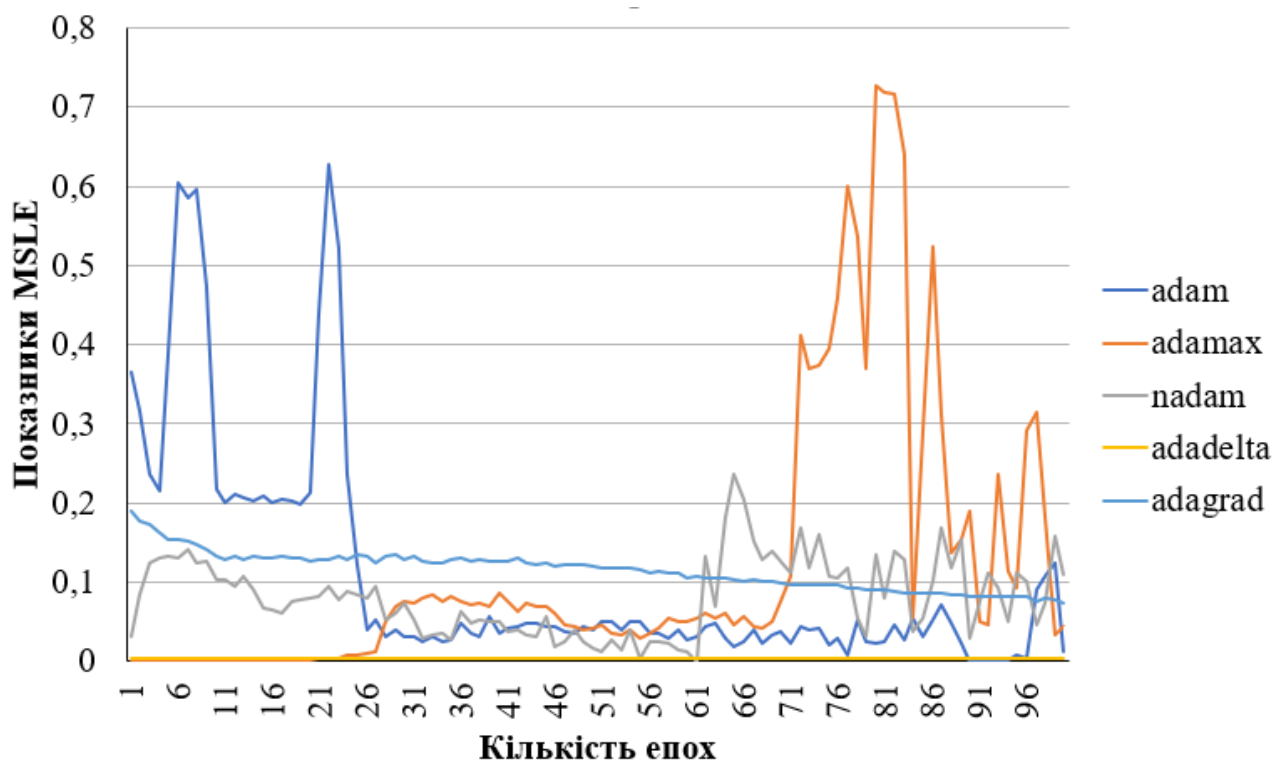


Рисунок 4.12 – Дослідження MSLE від кількості епох навчання за різними методами оптимізації навчання MLP-DOS

4.1.4 Виявлення класу мережевих атак категорії «Probe»

У ході експерименту встановлено, що кращі показники якості визначення мережевих атак категорії «Probe» в багатошарових перцептронах з одним прихованим шаром нейронів. На рисунку 4.13 видно, що найкращі показники якості мають НМ з наступними функціями активації: логістичною функцією на першому прихованому шарі; функцією Softmax на результуючому шарі. Дослідження точності MLP-Probe від кількості епох навчання за різною кількістю прихованих нейронів наведено на рис. 4.14. Із рисунку видно, що найкращі показники надала НМ конфігурації 29-1-58-6.

На рисунку 4.15 наведена отримана структура багатошарового перептрону класифікатора «Probe», де y_0 відповідає Ipsweep, y_1 – Hmap, y_2 – Portsweep, y_3 – Satan. Опис x_0, \dots, x_{28} наведено у розділі 3 (див. таблицю 3.1).

На рисунку 4.15 наведена отримана структура багатошарового перептрону класифікатора «Probe», де y_0 відповідає Ipsweep, y_1 – Hmap, y_2 – Portsweep, y_3 – Satan. Опис x_0, \dots, x_{28} наведено у розділі 3 (див. таблицю 3.1).

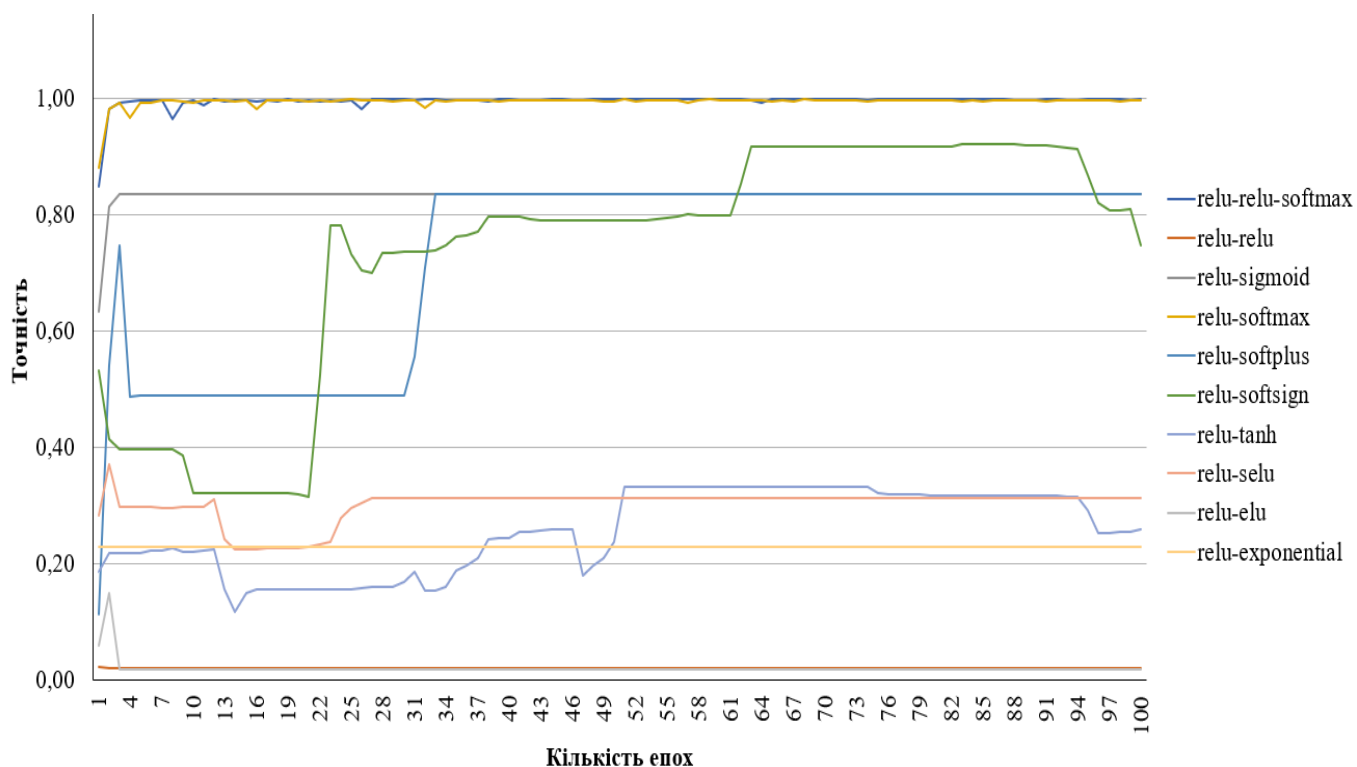


Рисунок 4.13 – Дослідження точності MLP- Probe від кількості епох навчання за різними функціями активації

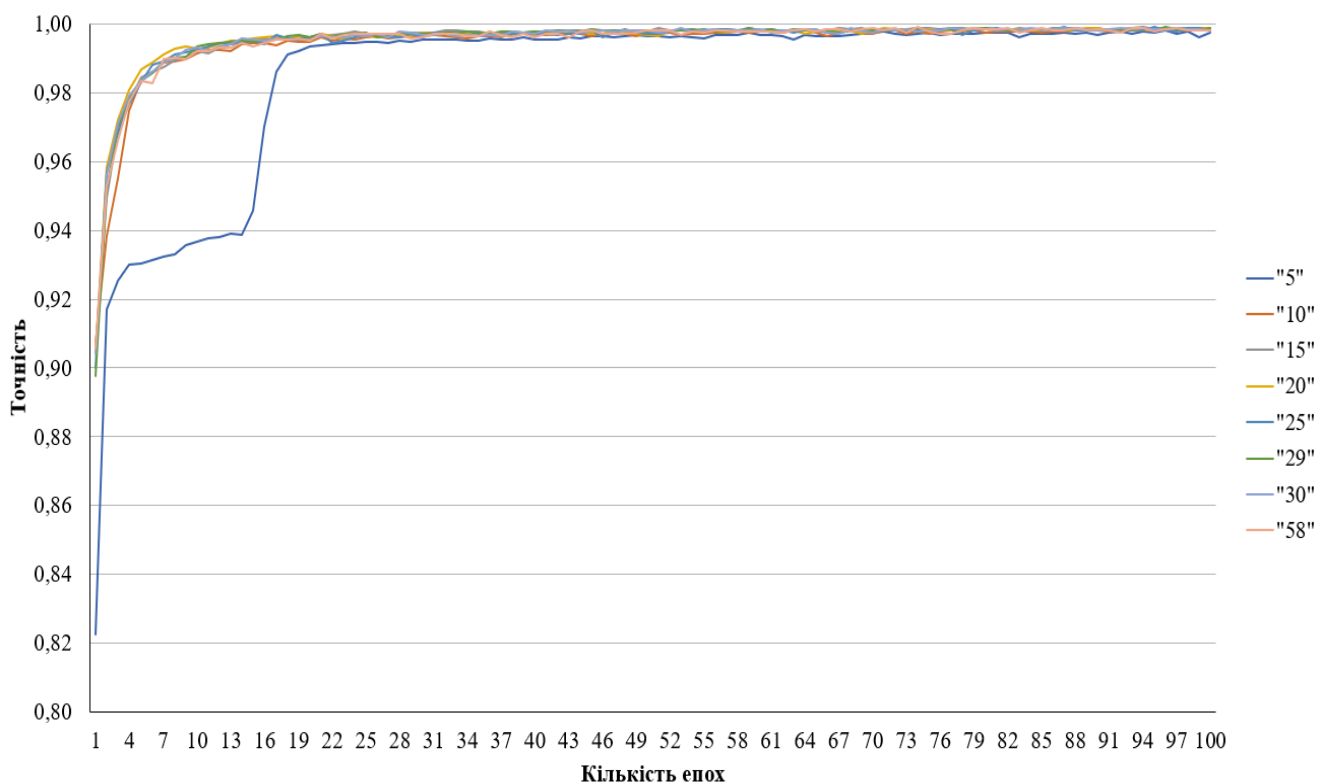


Рисунок 4.14 – Дослідження точності MLP- Probe від кількості епох навчання за різною кількістю прихованих нейронів

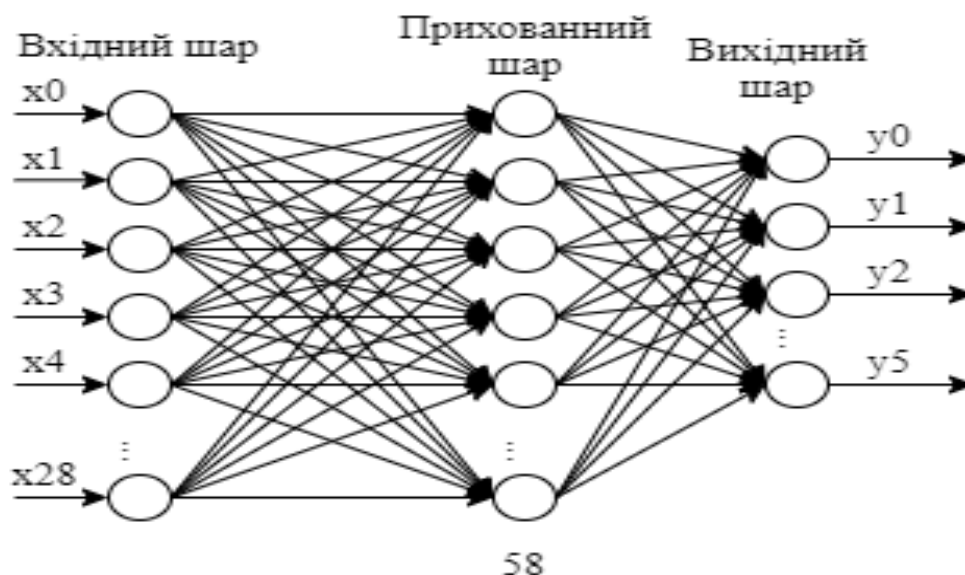


Рисунок 4.15 – Структура багатошарового перцептрону MLP- Probe

На рисунку 4.16 показано дослідження MSLE від кількості епох навчання для отриманої структури MLP-Probe, яка побудована на основі даних таблиці В.4 (у додатку В). На підставі низьких показників MSLE зроблений висновок, що для MLP-Probe кращим методом оптимізації є метод «adagrad». Для досягнення достатнього рівня якості роботи алгоритму достатньо 20 епох навчання.

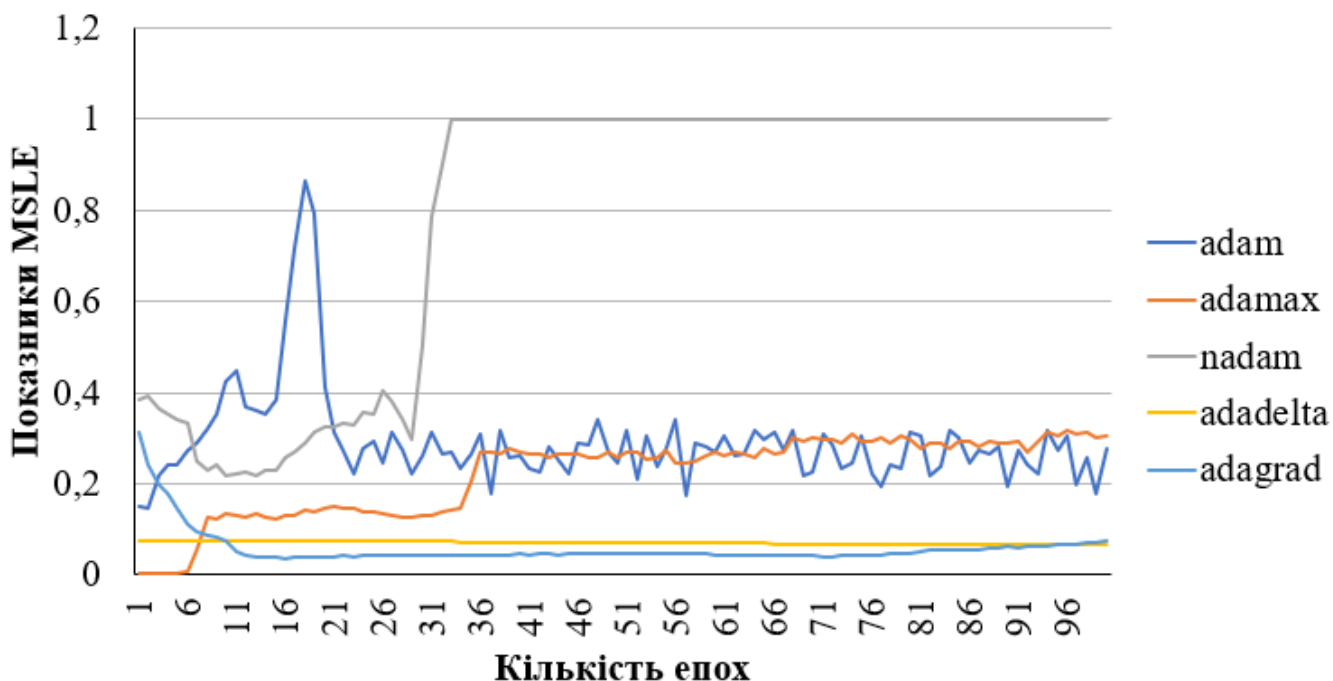


Рисунок 4.16 – Дослідження MSLE від кількості епох навчання за різними методами оптимізації навчання MLP- Probe

4.2 Дослідження показників якості визначення атак

1) TPR – показник коректності виявлення мережових атак визначається за формулою (4.1)

$$TPR = \frac{TP}{TP+FN}, \quad (4.1)$$

де TP – кількість правильно розпізнаних аномальних з'єднань;

FN – кількість помилок другого роду.

1) FPR – показник хибних спрацювань визначається за формулою (4.2)

2)

$$FPR = \frac{FP}{FP+TN}, \quad (4.2)$$

де FP – кількість помилок першого роду;

TN – кількість правильно розпізнаних нормальних з'єднань.

3) CCR – показник коректності класифікації з'єднань визначається за формулою (4.3)

$$CCR = \frac{CC}{TP+TN+FN+FP}, \quad (4.3)$$

де CC – загальна кількість експериментів.

4) ICR – показник не коректності класифікації з'єднань визначається за формулою (4.4)

$$ICR = \frac{IC}{TP+TN+FN+FP}, \quad (4.4)$$

де IC – кількість випадків некоректної класифікації.

Порівняємо отримані значення для початкової вибірки з 369 векторів та отриманої за допомогою клональної селекції вибірки у розмірі 937 векторів. Обчислено TPR , FPR , CCR та ICR для початкової та отриманої вибірки за

формулами (4.1), (4.2), (4.3) та (4.4) відповідно. Результати виконаних обчислень зведено до таблиці 4.1.

На основі отриманих даних, що представлені в вигляді графіків на рисунках 4.17 – 4.19, можна стверджувати якісну залежність роботи алгоритму від кількості елементів в навчальних вибірках. Збільшуючи вибірку за допомогою клональної селекції – ми покращуємо таким чином алгоритм роботи розпізнавання вторгнень. Також це значно покращує розпізнавання маловідомих атак, вибірка яких складає зовсім мало елементів.

Помилка першого роду (False Positive, FP) – це кількість невірно виявлених атак, коли нормальний стан був прийнятий за атаку.

Помилка другого роду (False Negative, FN) – це кількість пропусків атак, коли атака була помилково прийнята за нормальний стан мережі.

На основі даних що представлені у таблиці 4.1, встановимо відсоткові значення помилок першого та другого роду на різних етапах експериментів. Результати наведені у вигляді таблиці 4.2.

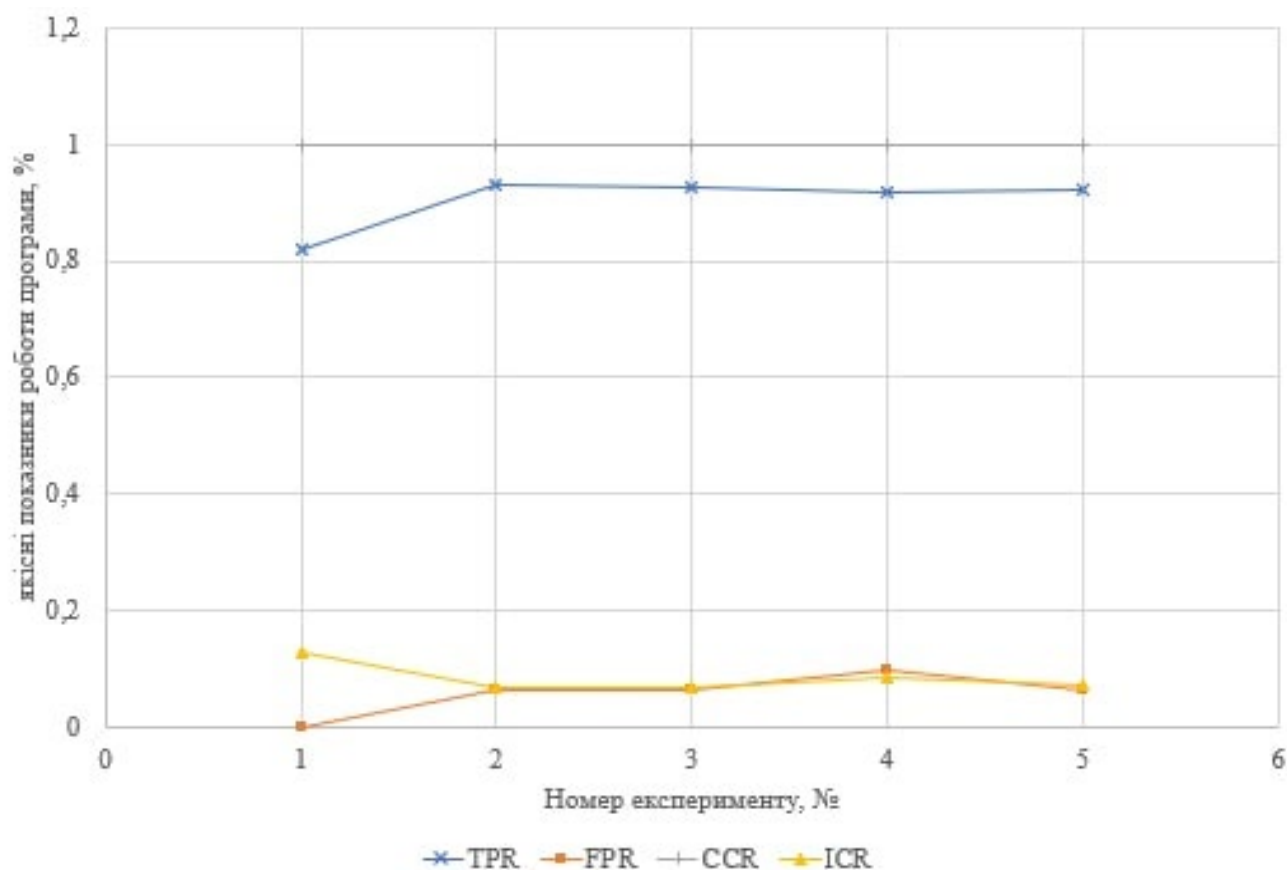
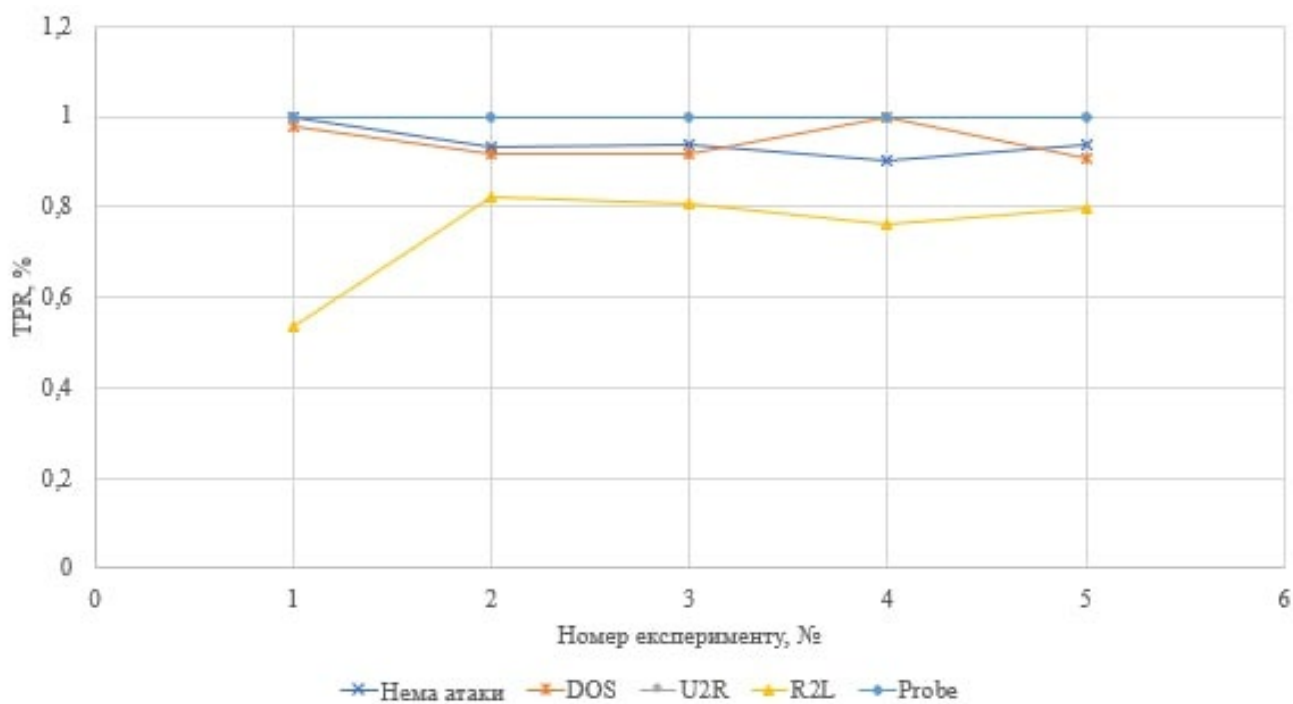


Рисунок 4.17 – Дослідження показників якості від довжини вибірки

Таблиця 4.1 – Результати дослідження параметрів якості

Об'єм загальної вибірки	Досліджувані параметри								
	Клас, що розпізнається	Кількість у вибірці	Кількість правильної класифікації	Кількість хибної класифікації	Кількість хибно прийнятих за нормальний стан	Загальні показники якості для всієї вибірки			
						TPR	FPR	CCR	ICR
362	Немає атак	104	104	0	-	0,82	0	1	0,13
	DOS	50	49	1	0				
	U2R	9	9	0	0				
	R2L	99	53	46	12				
	Probe	100	100	0	0				
1009	Немає атак	395	369	0	-	0,93	0,06	1	0,07
	DOS	344	315	29	26				
	U2R	9	9	0	0				
	R2L	83	68	15	8				
	Probe	178	178	0	0				
1986	Немає атак	610	571	39	-	0,93	0,06	1	0,07
	DOS	442	406	36	33				
	U2R	27	27	0	0				
	R2L	332	268	64	44				
	Probe	575	575	0	0				
4698	Немає атак	1484	1337	147	-	0,92	0,1	1	0,09
	DOS	1219	1217	2	0				
	U2R	162	162	0	0				
	R2L	1079	819	260	208				
	Probe	754	754	0	0				
9931	Немає атак	3051	2856	165	-	0,92	0,06	1	0,07
	DOS	2210	2005	205	190				
	U2R	135	135	0	0				
	R2L	1660	1320	340	220				
	Probe	2875	2875	0	0				



с

Рисунок 4.18 – Дослідження TPR від довжини вибірки для різних категорій

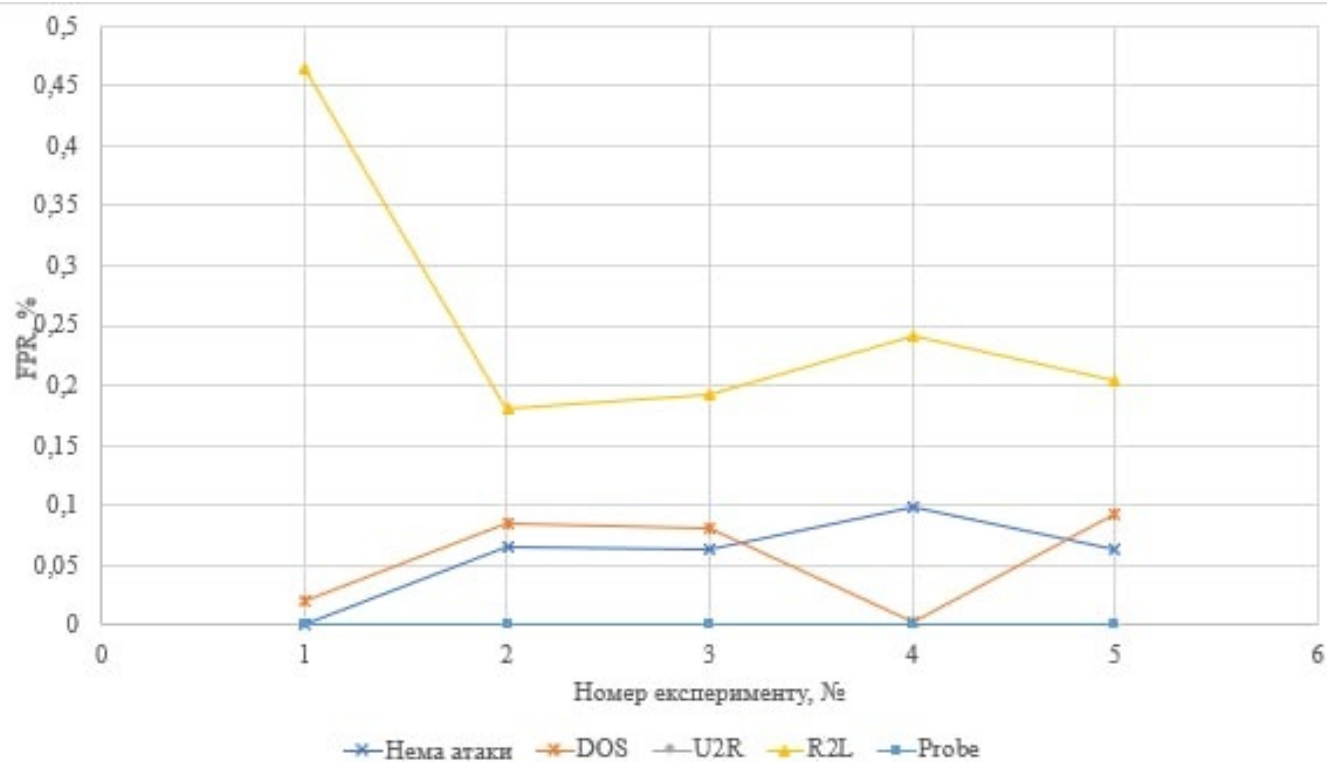


Рисунок 4.19 – Дослідження FPR від довжини вибірки для різних категорій

Таблиця 4.2 – Залежності відсоткових значень помилок першого та другого роду

Кількість тестових даних	Помилка першого роду, %	Помилка другого роду, %
362	0	12,98
1009	2,58	4,36
1986	1,964	5,04
4698	3,13	5,58
9931	1,96	5,49

Так, наприклад, із таблиці видно, що при збільшенні вибірки (з 1009 до 9931 прикладів) приблизно в 9 разів помилки першого (кількість невірно виявлених атак) та другого роду (кількість пропусків атак) зменшилися приблизно в 1,3 та 0,8 рази відповідно.

4.3 Використання програмного комплексу в навчальному процесі

4.3.1 Інструкція по використанню моделі «SOM_Clon»

Головне меню програми «SOM_Clon» представлене на рисунку 4.20. За допомогою програми ми можемо здійснювати генерацію додаткової навчальної вибірки з використанням алгоритму клональної селекції, здійснювати навчання SOM та класифікацію категорії атаки.

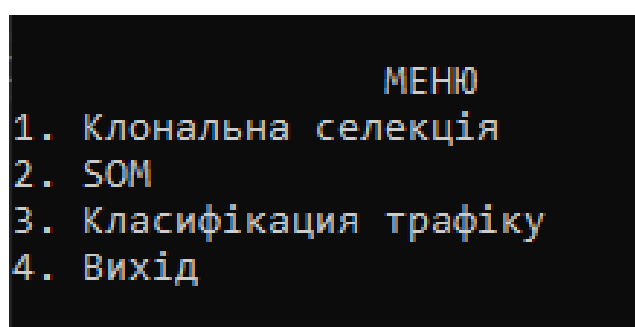


Рисунок 4.20 – Головне меню програмної моделі «SOM_Clon»

Перший пункт меню: Клональна селекція (рисунку 4.21), в якому задається кількість кращих антитіл та виконується безпосередньо клональна селекція. Дані, на основі яких алгоритм генерує «антитіла», повинні знаходитись у файлі «AntiGen.txt» у форматі, представленому на рисунку 4.22.

Умовне позначення: перший стовпець – кількість атак, що відповідає категорії, при цьому «0» відповідає нормальному стану мережі, «1» – категорії DoS, «2» – категорії Probe, «3» – категорії R2L, «4» – категорії U2R; другий – це кількість неправильної класифікації; третій – загальна кількість представника категорії у вибірці; четвертий – кількість правильно класифікованих екземплярів категорії; п'ятий – кількість неправильно прийнятих за нормальний стан екземплярів (помилка першого роду); шостий – кількість помилково прийнятих за атаку нормальних станів (помилка другого роду).

4.3.2 Інструкція по використанню моделі «MLP»

Для роботи з програмною моделлю «MLP», що написана на Python, та стандартними бібліотеками «Keras», «TensorFlow» та «Pandas» перш за все слід встановити на робочу машину Python версії 3.6.

Для зручної роботи з бібліотеками, щоб провести навчання штучного інтелекту, рекомендується встановити програму «Anaconda». Інструкція з встановлення представлена у додатку Г. Після чого необхідно створити у робочий простір та встановити всі необхідні пакети. Інструкція з роботи в «Anaconda» також представлена у додатку Г.

Запускаємо «Jupyter Notebook» через «Anaconda».

Активуємо в «Anaconda» робочий простір, потім задаємо директиву, де знаходяться робочі файли, а саме необхідні для навчання та тестування файли: «dos.1.csv», «dos.test.csv», «probe.1.csv», «probe.test.csv», «r2l.1.csv», «r2l.test.csv», «u2r.1.csv», «u2r.test.csv»; та виконавчий файл з програмою – «treining.ipynb».

Щоб виконати блок програми – обираємо його та натискаємо на значок «запуск» (рисунок 4.25).

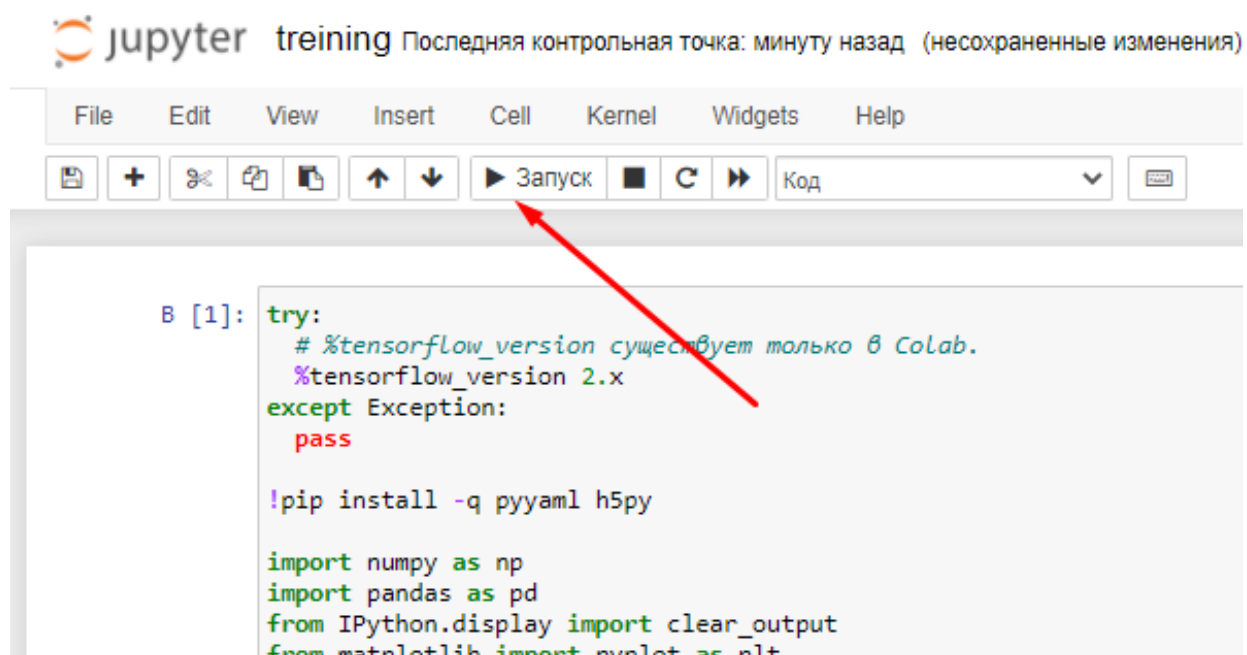


Рисунок 4.25 – Виконання блоку програми в «Jupyter Notebook»

4.3.3 Приклади завдань на основі програмного комплексу

Низький рівень

1) На основі програмної моделі «SOM_Clon» виконати навчання мережі Кохонена та дослідити значення помилок першого та другого роду. Отримані результати звести до таблиці 4.3.

Таблиця 4.3 – Дослідження значення помилок першого та другого роду

Об'єм загальної вибірки	Досліджувані параметри				
	Клас, що розпізнається	Кількість у вибірці	Кількість правильної класифікації	FP	FN
	Нема атаки				
	DOS				
	U2R				
	R2L				
	Probe				
	Нема атаки				
	DOS				
	U2R				
	R2L				
	Probe				

Середній рівень

1) На підставі отриманих даних (файл «results.txt») розрахувати показники якості роботи класифікації та звести до таблиці 4.4.

2) Провести дослідження показників якості від кількості екземплярів категорії у навчальній вибірці. Отримані результати звести до таблиці 4.4.

Таблиця 4.4 – Дослідження та розрахунок показників якості

Об'єм загальної вибірки	FN	TP	FP	TN	CC	IC	Загальні показники якості для всієї вибірки			
							TPR	FPR	CCR	ICR

Високий рівень

1) На основі KDD-99 сформувати навчальні вибірки відповідно до варіанту.

Таблиця 4.5 – Варіанти завдань

№ варіанту	Довжина першої навчальної вибірки	Довжина другої навчальної вибірки	№ варіанту	Довжина першої навчальної вибірки	Довжина другої навчальної вибірки
1	200	3800	6	700	3300
2	300	3700	7	800	3200
3	400	3600	8	900	3100
4	500	3500	9	1000	3000
5	600	3400	10	1100	2900

На основі програмної моделі «SOM_Clon» дослідити показники якості роботи класифікатора від довжини навчальної вибірки, отримані результати представити в графічному вигляді.

2) На основі програмної моделі «MLP» провести дослідження параметрів MLP (кількості епох, значення MSLE) та відсотка правильно класифікованих класів від кількості прихованих нейронів за різними алгоритмами оптимізації навчання: *Adam, AdaMax, Nadam, AdaDelta та AdaGrad*. Отримані дані занести у таблицю 4.6.

Таблиця 4.6 – Дослідження параметрів MLP від кількості прихованих нейронів за різними алгоритмами оптимізації навчання

Кількість прихованих нейронів	Алгоритм № 1			Алгоритм № n		
	Кількість епох	MSLE	Правильно класифіковані класи, %	Кількість епох	MSLE	Правильно класифіковані класи, %

3) На основі програмної моделі «MLP» провести дослідження параметрів MLP (кількості епох, значення MSLE) та відсотка правильно класифікованих класів від кількості прихованих нейронів за різними функціями активації: *relu, sigmoid, softmax, softplus, softsign, tanh, selu, elu, exponential*. Отримані дані занести у таблицю 4.7.

Таблиця 4.7 – Дослідження параметрів MLP від кількості прихованих нейронів за різними функціями активації

Кількість прихованих нейронів	Функція активації № 1			Функція активації № n		
	Кількість епох	MSLE	Правильно класифіковані класи, %	Кількість епох	MSLE	Правильно класифіковані класи, %

4) На основі програмної моделі «MLP» провести дослідження параметрів MLP (кількості епох, значення MSLE) та відсотка правильно класифікованих класів від довжини навчальної вибірки за різними алгоритмами оптимізації навчання: *Adam*, *AdaMax*, *Nadam*, *AdaDelta* та *AdaGrad*. Отримані дані занести у таблицю 4.8.

Таблиця 4.8 – Дослідження параметрів MLP від довжини навчальної вибірки за різними алгоритмами оптимізації навчання

Довжина навчальної вибірки	Алгоритм № 1		...	Алгоритм № n	
	MSLE	Правильно класифіковані класи, %	...	MSLE	Правильно класифіковані класи, %

5) На основі програмної моделі «MLP» провести дослідження параметрів MLP (кількості епох, значення MSLE) та відсотка правильно класифікованих класів від довжини навчальної вибірки за різними функціями активації: *relu*, *sigmoid*, *softmax*, *softplus*, *softsign*, *tanh*, *selu*, *elu*, *exponential*. Отримані дані занести у таблицю 4.9.

Таблиця 4.9 – Дослідження параметрів MLP від довжини навчальної вибірки за різними функціями активації

Довжина навчальної вибірки	Функція активації № 1		...	Функція активації № n	
	MSLE	Правильно класифіковані класи, %	...	MSLE	Правильно класифіковані класи, %

4.4 Основні висновки

1. На створеному програмному комплексі проведені наступні дослідження: визначення оптимальних параметрів нейронних мереж (MLP-R2L, MLP-U2R, MLP-DOS, MLP-Probe) для визначення класу мережевих атак відповідно до категорії (перше дослідження); визначення показників оцінки якості отриманих рішень (друге дослідження).

2. Відповідно до першого дослідження проведена оцінка точності та значення MSLE нейронних мереж від кількості епох навчання за різними функціями активації (relu, sigmoid, softmax, softplus, tanh, ...) та різною кількістю прихованих нейронів (10, 25, 30 та 55) при різних алгоритмах навчання (Adam, AdaMax, Nadam, AdaDelta та AdaGrad). Так, наприклад, для визначення класу мережевих атак категорії DOS необхідно мати нейронну мережу конфігурації 29-1-25-6 (з логістичною функцією у прихованому шарі та функцією Softmax на результуючому шарі), яка за алгоритмом adadelta (за 25 епох) надає 99,82 точність на основі навчальної вибірки із 849 прикладів.

3. Відповідно до другого дослідження отримані залежності показників якості визначення мережевих атак (TPR, FPR, CCR та ICR) від довжини навчальної вибірки (362, 1009, 1986, 4698 та 9931 прикладів), що згенерована за допомогою клональної селекції. Так, наприклад, при збільшенні вибірки приблизно в 9 разів помилки першого (кількість невірно виявлених атак) та другого роду (кількість пропусків атак) зменшилися приблизно в 1,3 та 0,8 рази відповідно.

4. Створений програмний комплекс може бути використаний в навчальному процесі для здобувачів ступеня «магістр» спеціальності «Кібербезпека» з дисципліни «Теорія проектування захищених комп'ютерних мереж».

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Вимоги безпеки при виконанні робіт на робочому місці

Питання охорони праці є дуже важливим і його необхідно вирішувати на всіх етапах трудового процесу незалежно від галузі професійної діяльності. Визначення основних положень що стосуються реалізації конституційного права громадянина на охорону її життя і здоров'я у процесі трудової діяльності міститься у законі України "Про охорону праці" згідно з Постановою Верховної Ради України № 345-VI від 2 вересня 2008 року [34].

Відповідно до нормативно правових актів з охорони праці «Правила безпечної експлуатації електроустановок споживачів», що затверджено: наказ Держнаглядохоронпраці України №4 від 9 січня 1998 року [35], та НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час робіт з екранними пристроями», затверджені наказом Міністерства соціальної політики України "Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" № 207 від 14 лютого 2018 року[36], вимоги безпеки під час роботи з електронно обчислювальною машиною з відео-дисплейними терміналами (далі - ВДТ) і периферійними пристроями (далі - ПП) наступні:

- Очищати перед початком роботи щодень екран від пилу та інших забруднень;
- Щодня перед початком роботи оператор ЕОМ повинен перевірити своє робоче місце на наявність ознак пошкодження обладнання;
- Перед початком роботи оператор ЕОМ повинен перевірити правильність підключення обладнання ЕОМ до електромережі;
- Перед початком роботи оператор ЕОМ повинен перевірити правильність організації робочого місця;
- Обладнання, принесене у холодну пору року з вулиці в робоче приміщення, можна підключати до електричної мережі тільки після того, як

температура обладнання зрівняється з температурою повітря відповідного робочого приміщення;

- Забороняється:
 - виконувати на робочому місці, ремонт та налагодження ЕОМ;
 - відключати захисні пристрої, самочинно проводити зміни у конструкції та складі ЕОМ або їх технічне налагодження;
 - працювати з ВДТ, у яких під час роботи з'являються нестабільне зображення на екрані, нехарактерні сигнали тощо;
 - зберігання біля ЕОМ дискет, паперу, інших носіїв інформації, запасних блоків, деталей тощо;
 - допускання попадання вологи на поверхню системного блоку;
 - доторкання до задньої панелі системного блоку при включеному живленні;
 - вимикання живлення під час виконання активного завдання;
 - приймання напоїв та їжі на робочому місці;
- Про виявлення несправності обладнання або інших факторів, які створюють загрозу для життя або здоров'я працівників, необхідно негайно інформувати свого безпосереднього керівника.

5.2 Шкідливі виробничі фактори на робочому місці

Правильна оцінка небезпечних і шкідливих виробничих факторів значно впливає на забезпечення безпечних умов праці. Фактори виробничого середовища, надмірне розумове та фізичне навантаження, нервово-емоційна напруга, а також різне сполучення цих причин можуть впливати на однакові по складності зміни в організмі людини.

У приміщенні на програміста можуть негативно впливати наступні фізичні та психофізіологічні фактори:

- підвищена або знижена температура;
- підвищена або знижена вологість;
- недостатня освітленість;
- підвищений рівень шуму;

- підвищена іонізація повітря;
- підвищений рівень електромагнітних випромінювань;
- нервово-психічні перевантаження.

Одним з найважливіших факторів, що впливають на здоров'я людини є організація робочого місця. Так у нормативно правовому акті з охорони праці НПАОП 22.1-1.01-96 «Правила охорони праці для видавництв і редакцій», що затверджено наказом Державного комітету України по нагляду за охороною праці № 122 від 18 липня 1996 року [37] йдеться мова про об'єм виробничих приміщень для програмістів. Відповідно [37] об'єм виробничих приміщень на одного працівника повинна складати 20 м^3 , а площа приміщень - не менше 6 м^2 з урахуванням максимального числа працівників в одну зміну.

Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99., що затверджено Постановою Головного державного санітарного лікаря України №42 від 1 грудня 1999 року [38] робота програміста за енерговитратами відноситься до категорії легких робіт, зокрема до категорій Ia та Ib.

Мікрокліматичні умови виробничих приміщень характеризуються наступними такими показниками, як: температура повітря, швидкість руху повітря, відносна вологість повітря, інтенсивність теплового (інфрачервоного) опромінення, температура поверхні.

Таблиця 5.1. – Оптимальні мікрокліматичні умови виробничого приміщення, для категорій Ia та Ib легких робіт

Період року	Категорія робіт	Температура повітря	Відносна вологість	Швидкість руху, м/сек.
Холодний період року	Легка Ia	22 – 24	60 – 40	0,1
	Легка Ib	21 – 23	60 – 40	0,1
Теплий період року	Легка Ia	23 – 25	60 – 40	0,1
	Легка Ib	22 – 24	60 – 40	0,2

У таблиці 5.1 наведено оптимальні мікрокліматичні умови виробничого приміщення, для категорій Ia та Ib легких робіт, відповідно до [36].

У таблиці 5.2 наведено допустимі мікрокліматичні умови виробничого приміщення, для категорій Іа та Іб легких робіт, відповідно до [35].

Таблиця 5.2. – Допустимі мікрокліматичні умови виробничого приміщення, для категорій Іа та Іб легких робіт.

Період року	Категорія робіт	Температура, град.С				Відносна вологість (%)	Швидкість руху, м/сек.
		Верхня межа		Нижня межа			
		На постійних робочих місцях	На непостійних робочих місцях	На постійних робочих місцях	На непостійних робочих місцях		
Холодний період року	Легка Іа	25	26	21	18	75	не більше 0,1
	Легка Іб	24	25	21	18	75	не більше 0,2
Теплий період року	Легка Іа	28	30	22	20	55 - при 28 град. С	0,2 - 0,1
	Легка Іб	28	30	21	19	60 - при 27 град. С	0,3 - 0,1

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ВДТ ЕОМ і ПЕОМ, мають відповідати вимогам що наведені у таблиці 5.3 – Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот для програміста відповідно до [35].

Таблиця 5.3 – Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот для програміста

Вид трудової діяльності	Рівні звукового тиску в дБ в октавних смугах із середньгеометричними частотами, Гц									
	31,5	63	125	250	500	1000	2000	4000	8000	Рівні звуку, еквівалентні рівні звуку, дБА/дБАекв.
Програмісти ЕОМ	86	71	61	54	49	45	42	40	38	50

Устаткування, що становить джерело шуму відповідно до [38], слід розташовувати поза приміщенням для роботи ЕОМ, а також для забезпечення допустимих рівнів шуму на робочих місцях слід застосовувати засоби звукопоглинання.

Відповідно до державних будівельних норм ДБН В.2.5-28:2018 «Природне і штучне освітлення», що затверджено наказом Мінрегіону №264 від 3 жовтня

2018 року [39] нормативним параметром природного освітлення є коефіцієнт природного освітлення (КПО). Коефіцієнт природного освітлення встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні бокового освітлення КПО=1,5%. Для IV розряду зорових робіт мінімальна освітленість складає 300-500 лк.

Розрахунок штучного освітлення для кімнати площею 16,165 м², ширина якої складає 3,05м, довжина – 5,3м, висота – 3м за методом коефіцієнта використання світлового потоку наведено далі.

Для визначення потрібної кількості світильників, які повинні забезпечити нормований рівень освітленості, необхідно визначити світловий потік, що падає на робочу поверхню за формулою (5.1):

$$F = \frac{ESKZ}{n}, \quad (5.1)$$

де F – світловий потік, що розраховується, Лм;

E – нормована мінімальна освітленість, Лк $E = 300$ Лк;

S – площа освітлюваного приміщення;

K – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації ($K = 1,5$)

Z – відношення середньої освітленості до мінімальної ($Z = 1,1$);

n – коефіцієнт використання світлового потоку, (залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ($\rho_{\text{ст.}}$) і стелі ($\rho_{\text{стелі}}$), значення коефіцієнтів дорівнюють $\rho_{\text{ст}} = 50\%$ і $\rho_{\text{стелі}} = 50\%$.)

Обчислимо індекс приміщення за формулою (5.2)

$$i = \frac{S}{H(A+B)}, \quad (5.2)$$

де S – площа приміщення, $S = 16,165\text{м}^2$;

h – розрахункова висота підвісу, $h = 2,9$ м;

A – ширина приміщення, $A = 3,05$ м;

B – довжина приміщення, $B = 5,3$ м.

Підставивши значення отримаємо: $i = 0,67$. Знаючи індекс приміщення, знаходимо $n = 0,22$. Підставимо всі значення у формулу для визначення світлового потоку F .

$$F = \frac{300 * 16.165 * 1.1 * 1.5}{0.22} = 33371.25 \text{ Лм}$$

Для освітлення використані люмінісцентні лампи типу ЛБ 40-1, світловий потік яких $F=4320$ Лм. Розрахуємо необхідну кількість ламп у світильниках за формулою (5.3)

$$N = \frac{F}{F_{\text{л}}}, \quad (5.3)$$

де N – кількість ламп, що визначається;

F – світловий потік;

$F_{\text{л}}$ – світловий потік ламп

$$N = \frac{33371.25}{4320} = 8,4 = 9$$

В приміщенні кожен світильник комплектується двома лампами, тобто необхідно використовувати 4 світильника з 2 працюючими лампами. Схема розташування світильників наведено на рисунку 5.1.

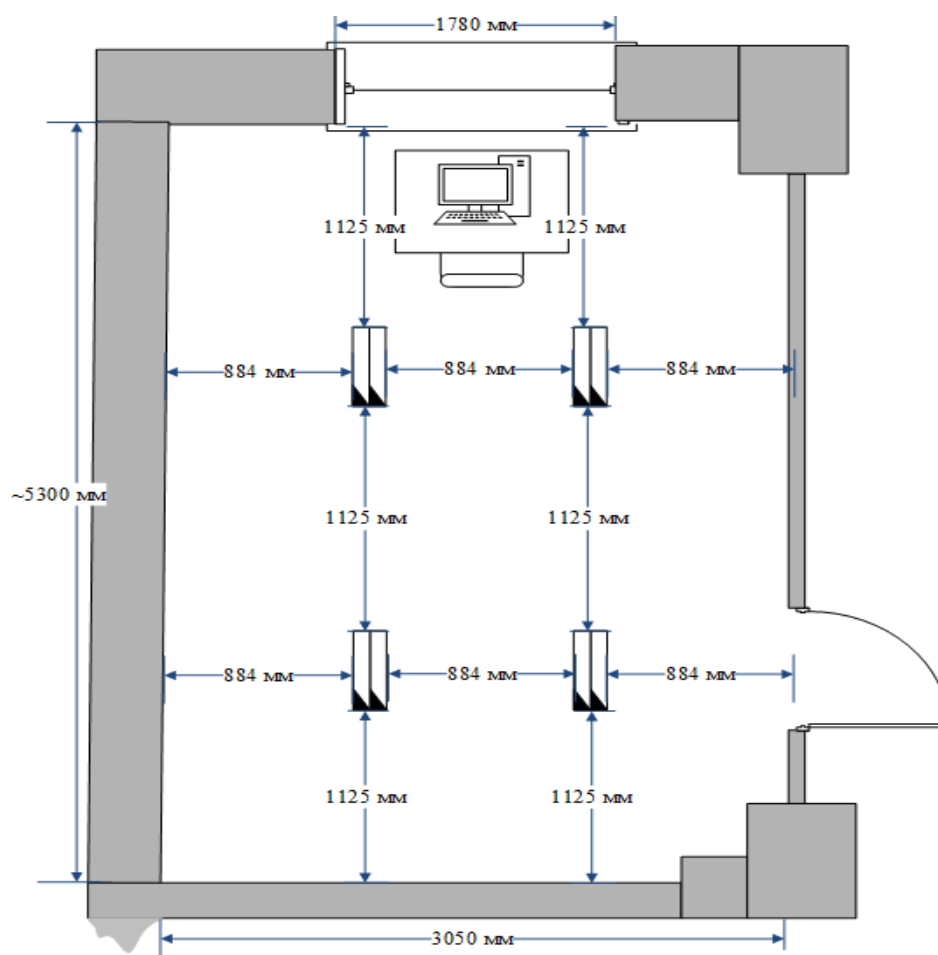


Рисунок 5.1 – Схема розташування світильників

Таким чином не задовольняється достатні умови штучного освітлення. Рекомендується додати ще один світильник.

5.3 Дій працівників в аварійних ситуаціях

У випадку Аварійної ситуації програміст зобов'язаний:

- при виявленні будь-яких неполадок в роботі персонального комп'ютера програміст повинен припинити роботу, вимкнути комп'ютер і повідомити про це безпосереднього керівника для організації ремонту;
- при попаданні людини під електричну напругу негайно вимкнути електричне живлення, до прибуття лікаря надати долікарську медичну допомогу;
- при будь-яких випадках порушень роботи технічного обладнання негайно викликати представника технічної служби;
- при нещасному випадку, отруєнні, раптовому захворюванні необхідно негайно надати першу допомогу потерпілому, викликати лікаря або допомогти

доставити потерпілого до лікаря, а потім повідомити керівника про те, що трапилося

- у випадку виникнення різі в очах, різкого погіршення зору, виникнення головного болю, больових почуттів у пальцях та кистях рук, посилення серцебиття – негайно припинити роботу з використанням ЕОМ, повідомити про те, що сталося, свого безпосереднього керівника й звернутися до медичної установи;
- при загорянні обладнання негайно відключити його від електромережі, ужити заходів щодо ліквідації вогню за допомогою вуглекислотного або порошкового вогнегасник.

План евакуації наведено на рисунку 5.2

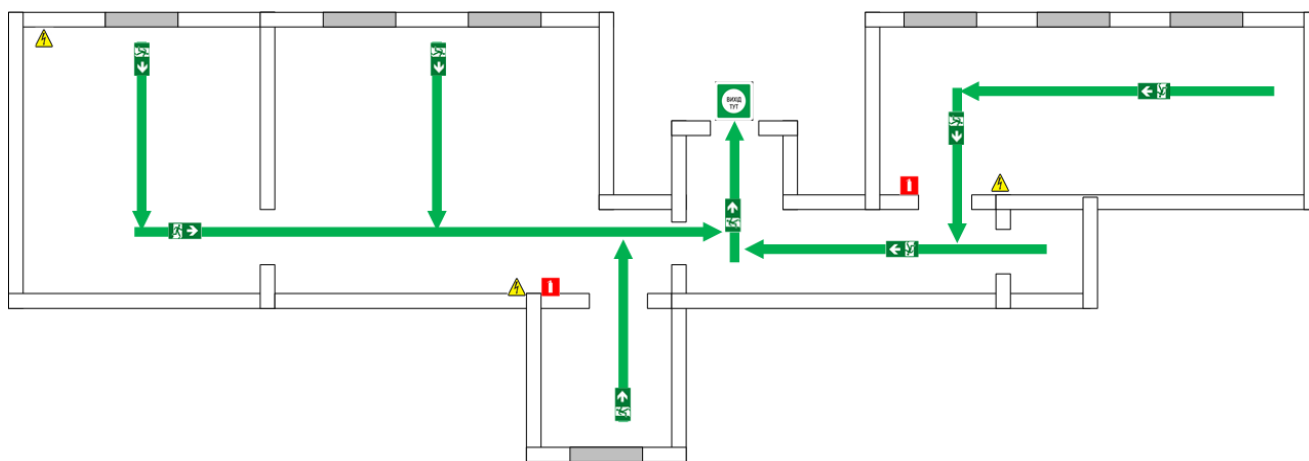


Рисунок 5.2 – План евакуації

Згідно з [40] загальними правилами надання до медичної допомоги є наступними:

- Перш за все оглянути місце пригоди і впевнитись в особистій безпеці і безпеці постраждалого;
- Провести первинний огляд постраждалого;
- Викликати швидку медичну допомогу;
- Провести вторинний огляд постраждалого, з метою виявлення інших проблем (пошкоджень), які потребують надання домедичної допомоги (розпочніть із загального огляду всього тіла, починаючи з голови).

Надання першої домедичної допомоги постраждалим при ураженні електричним струмом:

1) Як можливо швидко відокремити потерпілого від джерела струму.

2) Викликайте швидку, якщо це необхідно.

3) Покладіть та/або зігрійте людину.

4) Закрийте опіки - якщо у потерпілого є опіки, їх треба накрити стерильною марлею (якщо є під рукою) або чистою гладкою тканиною. Звичайно, тільки в тому випадку, якщо стан людини дозволяє зняти або розрізати одяг на обпалених місцях.

5) Якщо з'являються ознаки шоку - блювання, слабкість, сильна блідість, - трохи підніміть ноги, підклавши під ступні валик з речей.

6) Якщо потерпілий погано дихає або не дихає зовсім, негайно починайте робити штучне дихання рот в рот.

7) Якщо у людини немає пульсу і відсутній серцебиття, крім штучного дихання, необхідний непрямий масаж серця.

Надання першої домедичної допомоги постраждалим при пожежі:

1) Якщо горить одяг, його слід скинути або погасити полум'я, щільно накривши людини ковдрою або будь-яким шматком тканини. Обпалені ділянки одягу акуратно розрізати і скидати по частинах, у запобігання подальшої травматизації шкіри.

2) Якщо закрита рана необхідно охолоджувати водою уражену ділянку протягом 10 хвилин.

3) На поверхню рани слід накласти стерильну пов'язку.

4) Забезпечити потерпілому спокій.

5) Дати випити велику кількість рідини (чай, вода і тому подібне).

6) Негайно викликати бригаду невідкладної допомоги.

7) При можливості знеболити потерпілого, дати прийняти таблетку анальгін.

ВИСНОВКИ

1. Для визначення мережових атак на комп'ютерну мережу створений програмний комплекс, в основу якого покладені наступні моделі: «SOM_Clon», що створена в C++ з використанням алгоритму клональної селекції для визначення категорії мережової атаки (на першому етапі); «MLP», що створена в Python для визначення класу мережової атаки відповідно до категорії (на другому етапі).

2. На створеному програмному комплексі проведені наступні дослідження: визначення оптимальних параметрів нейронних мереж (MLP-R2L, MLP-U2R, MLP-DOS, MLP-Probe) для визначення класу мережових атак відповідно до категорії (перше дослідження); визначення показників оцінки якості отриманих рішень (друге дослідження). Створений програмний комплекс може бути використаний в навчальному процесі для здобувачів ступеня «магістр» спеціальності «Кібербезпека» з дисципліни «Теорія проектування захищених комп'ютерних мереж».

3. Відповідно до першого дослідження проведена оцінка точності та значення MSLE нейронних мереж від кількості епох навчання за різними функціями активації (relu, sigmoid, softmax, softplus, tanh та ін.) та різною кількістю прихованих нейронів (10, 25, 55 та ін.) при різних алгоритмах навчання (Adam, AdaMax та ін.). Так, наприклад, для визначення класу мережових атак категорії DOS необхідно мати нейронну мережу конфігурації 29-1-25-6 (з логістичною функцією у прихованому шарі та функцією Softmax на результуючому шарі), яка за алгоритмом adadelata (за 25 епох) надає 99,82 точність на основі навчальної вибірки із 849 прикладів.

4. Відповідно до другого дослідження отримані залежності показників якості визначення мережових атак (TPR, FPR, CCR та ICR) від довжини навчальної вибірки. Так, наприклад, при збільшенні вибірки приблизно в 9 разів помилки першого (кількість невірно виявлених атак) та другого роду (кількість пропусків атак) зменшилися приблизно в 1,3 та 0,8 рази відповідно.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cyber Safety Insights Report URL:
https://now.symassets.com/content/dam/norton/campaign/NortonReport/2019/2018_Norton_LifeLock_Cyber_Safety_Insights_Report_US_Media_Deck.pdf (дата звернення: 23.02.2020).
2. Internet Security Threat Report URL:
<https://www.symantec.com/content/dam/symantec/docs/reports/istr-242019-en.pdf> (дата звернення: 23.02.2020).
3. Росенко А. П. Структура нейросетевой динамической экспертной системы защиты информации. Таганрог: «Известия ТРТУ». 2003. С. 216-218
4. Dhangar K., Kulhare D., Khan A. A Proposed Intrusion Detection System. International Journal of Computer Applications. 2013. Vol. 65, N 23. p.p. 46-50.
5. Котов В. Д. Современное состояние проблемы обнаружения сетевых вторжений. Уфа : «Вестник УГАТУ» , 2012. С. 198-204.
6. Лукацкий А. В. Обнаружение атак СПб.: БХВ-Петербург, 2003. 608 с.
7. Саймон Хайкин Нейронные сети. Москва: «Вильямс», 2008. 1104 с.
8. Апанель Е. Н. Нейронауки: достижения и перспективы. «Медицинские новости», 2013. № 10. С. 6-11
9. Пахомова В. М. Теорія проектування комп'ютерних мереж: методичні вказівки до виконання курсового проект. Дніпровск. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна, 2019. 60 с.
10. Пахомова В. М. Можливості розвитку комп'ютерних мереж у автоматизованих системах залізничного транспорту: монографія. Дніпро: Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна, 2015. 207 с.
11. Терейковський І. Вдосконалення алгоритму навчання багатошарового перспетрону, призначеного для розпізнавання мережєвих атак. «Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні», 2012. № 2. 6 с.
12. Частикова В. А., Картамышев Д. А., Власов К. А. Нейросетевой метод защиты информации от ddos-атак. Современные проблемы науки и образования.

2015. №1-1. URL: <http://www.science-education.ru/ru/article/view?id=18343> (дата звернения: 02.04.2020)

13. Катасёв А. С., Катасёва Д. В., Кирпичников А. П. Нейросетевая диагностика аномальной сетевой активности. Вестник Казанского технологического университета, 2015. № 6. 5 с

14. Крыжановский А. В. Применение искусственных нейронных сетей в системах обнаружения атак. «Технические науки», 2008. 2 с.

15. Кохонен Т. Самоорганизующиеся карты 2014. URL: <http://www.studentlibrary.ru/book/ISBN9785996313488.html> (дата обращения: 05.04.2020).

16. Плейфэр Дж., Чейн Б. М. Наглядная иммунология. Москва: ГЭОТАР-Медиа, 2002. 93 с.

17. Лебедев К. А., Понякина И. Д. Иммунология в клинической практике. Москва: «Наука», 1990. 156 с.

18. Пол. У. Иммунология. Москва: «Мир», 1988. 472 с.

19. Коромыслов Н. А. О предварительном анализе параметров для обнаружения инцидентов информационной безопасности в системах со многими параметрами «Решетневские чтения» 2013. URL: <https://cyberleninka.ru/article/n/o-primenenii-iskusstvennyh-immunnyh-sistem-dlya-obnaruzheniya-incidentov-informatsionnoy-bezopasnosti-v-sistemah-so-mnogimi/viewer> (дата звернения: 23.02.2020).

20. Браницкий А. А. Обнаружение аномальных сетевых соединений на основе гибридизации методов вычислительного интеллекта: диссертация кандидата Технических наук: 05.13.19. ФГБУН СанктПетербургский институт информатики и автоматизации Российской академии наук, 2018. 305 с.

21. Brownlee J. Clever algorithms: nature-inspired programming recipes. 2011. 441 pp.

22. Литвиненко В. И. Объектно-ориентированная реализация алгоритма клональной селекции. URL: [https://cyberleninka.ru/article/n/primenenie-algoritma-](https://cyberleninka.ru/article/n/primenenie-algoritma)

klonalnogo-otbora-dlya-resheniya-sistem-algebraicheskikh-uravneniy/viewer (дата звернення: 25.03.2020).

23. Бурлаков М. Е. Модель многослойной универсальной системы обнаружения вторжений. URL: <http://old.tusur.ru/filearchive/reports-magazine/2014-32-2/41.pdf> (дата звернення: 25.03.2020).

24. Обзор корпоративных IPS-решений. URL: https://www.anti-malware.ru/IPS_russian_market_review_2013 (дата звернення: 25.03.2020).

25. Котов В. Д., Васильев В. И. Система обнаружения сетевых вторжений на основе механизмов иммунной модели, «Известия Южного федерального университета. Технические науки», 2011. 10 с. URL: <https://cyberleninka.ru/article/n/sistema-obnaruzheniya-setevyh-vtorzheniy-na-osnove-mehanizmov-immunnoy-modeli/viewer> (дата звернення: 25.03.2020).

26. Котов В. Д., Васильев В. И. Современное состояние проблемы обнаружения сетевых вторжений «Вестник Уфимского государственного авиационного технического университета», 2012. 7 с. URL: <https://cyberleninka.ru/article/n/sovremennoe-sostoyanie-problemy-obnaruzheniya-setevyh-vtorzheniy/viewer> (дата звернення: 25.03.2020).

27. Риндич Є. В., Зайцев В. В., Коняши С. В., Усов Я. Ю. Особливості створення мережевої системи виявлення вторгнень у комп'ютерні системи. Математичні машини і системи. 2018. № 3. С. 89-96.

28. Пахомова В. М. Дослідження інформаційно-телекомунікаційної системи залізничного транспорту з використанням штучного інтелекту. Дніпро: Вид-во ПФ «Стандарт - Сервіс», 2018. 220 с.

29. KDD Cup 1999 Data URL: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> (дата звернення: 10/05/2020).

30. Пахомова В. М., Коннов М. С. Дослідження двох підходів до виявлення мережних атак із використанням нейромережної технології. Наука та прогрес транспорту. 2020. № 3(87). С. 81-93. DOI: 10.15802/stp2020/208233

31. Keras URL: <https://keras.io/> (дата звернення: 10.05.2020).

32. TensorFlow URL <https://www.tensorflow.org/> (дата звернення: 10/05/2020).
33. Pandas URL <https://pandas.pydata.org/> (дата звернення: 10.05.2020).
34. Закон України «Про охорону праці» згідно з Постановою Верховної Ради України № 345-VI від 2 вересня 2008 року
35. НПАОП 40.1–1.21–98 «Правила безпечної експлуатації електроустановок споживачів». Затверджено: наказ Держнаглядохоронпраці України № 4 від 9 січня 1998 року
36. НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час робот з екранними пристроями». Затверджено: наказ Міністерства соціальної політики України «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» № 207 від 14 лютого 2018 року.
37. ДСанПІН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Затверджено Постановою Головного державного санітарного лікаря України № 7 від 10 грудня 1998 року.
38. ДБН В.2.5-28:2018 «Природне і штучне освітлення». Затверджено наказом Мінрегіону № 264 від 3 жовтня 2018 року.
39. ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень». Затверджено Постановою Головного державного санітарного лікаря України № 42 від 1 грудня 1999 року.
40. Ненько С. К., Полівода Л. А. Надання першої медичної допомоги при надзвичайних ситуаціях, Херсон: «Навчально-методичний центр цивільного захисту та безпеки життєдіяльності Херсонської області», 2014, 28 с.
41. Биковська Д. Г.; кер.: доц. Пахомова В. М. Використання методів штучного інтелекту для виявлення атак на комп'ютерну мережу. Тези Всеукраїнської конференції студентів та молодих вчених 2020 р. «Інформаційно-управляючі технології і системи на залізничному транспорті».

42. Биковська Д. Г.; кер.: доц. Пахомова В. М. Виявлення мережових атак на створеному програмному комплексі з використанням методів штучного інтелекту. Тези XIV міжнародній науково-практичній конференції 2020р. «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті»