ШИФР «DDOS-NND»

«ЗАСІБ ДЛЯ ВИЯВЛЕННЯ DDOS-АТАК НА ОСНОВІ НЕЙРОМЕРЕЖЕВОГО ПІДХОДУ»

ЗМІСТ

ВСТУП

У наш час інформаційні технології розвиваються величезними темпами, так як і Інтернет. Уявити роботу будь-якої компанії без доступу до Інтернету до локальних і глобальних мереж неможливо. Це дозволяє забезпечити розширення клієнтської бази до світових масштабів. Відмова в роботі інформаційної системи компанії може спричинити катастрофічні наслідки, від крадіжки інсайдерської інформації до втрати клієнтської бази. При розвитку інформаційній технології, також розвиваються і зловмисники, придумуючи все більш різноманітних методі атак.

Сучасні атаки, які завдають великого збитку компаніям, вважаються віддаленні мережеві атаки, серед яких найбільш небезпечною є Distributed Denial of Service (DDoS-атака) [1]. Ця атака популярна своєю простотою, численними відкритими відомостями про її реалізації і нечисленними обчислювальними ресурсами. Головна ідея цієї атаки полягає в тому, що зловмисник намагається здійснити неможливість коректного обслуговування системи. Велика кількість різних DDoS-атак призвела до розробки методів, які використовуються для певної атаки.

Одним з перспективних напрямів забезпечення безпеки мережі є використання методів виявлення побудованих на основі штучних нейронних мереж (ШІНМ), які вже довели свою ефективність у вирішенні складних задач розпізнавання, класифікації, управління і виявлення. Застосування ШНМ дозволить створити ефективну адаптивну систему виявлення мережевих вторгнень і підвищити рівень захисту комп’ютерних систем від зовнішніх атак.

Метою роботи є підвищення точності виявлення DDoS-атак на основі нейромережевого підходу.

Для досягнення поставленої мети потрібно вирішити такі завдання:

- проаналізувати методи виявлення та захисту від DDoS-атак;
- виконати моделювання архітектури нейронної мережі;
- розробити програмний засіб;
- провести тестування розробленого засобу.

Об’єктом дослідження є процес виявлення DDoS-атак.

Предметом дослідження є методи та засоби виявлення DDoS-атак.

Методи дослідження. Для реалізації поставлених задач були використані методи теорії штучного інтелекту для проектування нейронної мережі; статистичні методи для підготовки вхідної та вихідної інформації для моделювання нейронної мережі, методи проектування програмного забезпечення для розробки та верифікації інтелектуальної системи.

Наукова новизна. Запропоновано архітектуру штучної нейронної мережі типу багатошарового персептрона, навчений екземпляр якої дозволяє вирішувати задачу виявлення деяких класів розподілених мережевих атак типу «відмова в обслуговування» на віддалений веб-ресурс, яка відрізняється сукупністю оптимально підібраних параметрів, що дозволяє підвищити точність виявлення до 99,8%.

Практична цінність. Розроблено програмний засіб для виявлення DDoS-атак на основі технологій штучних нейронних мереж, ефект від якої полягає в автоматизації підтримки прийняття рішень системного адміністратора, що базується на використанні інтелектуальних технологій, що дозволяє оперативно та з високою точністю виявляти факти мережевого вторгнення, відповідно, вчасно застосовувати контрзаходи.

За результатами науково-дослідної роботи оформлено та відіслано заявку на отримання авторського свідоцтва на твір «Комп’ютерна програма. Інтелектуальний програмний засіб для виявлення DDoS-атак. «NeuroSoft for DDoS detection».

Результати наукової роботи доповідалися на міжнародній науково-практичній конференції «Фотоніка-ОДС-2018» та опубліковані у збірнику тез. Також подано до друку статтю “DDoS-attack detection using artificial neural networks with Matlab” у співавторстві до видання “Proceedings of SPIE”, що входить до наукометричної бази Scopus.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальна характеристика DDoS-атак

Термін «відмова в обслуговуванні» спочатку був створений компанією Gligor в контексті операційної системи, але оскільки він відображає недоступність сервісу, він використовувався в дослідженні безпеки. На відміну від випадкових збоїв, які, як правило, впливають лише на кілька вузлів одночасно, DoS-атаки призначені для виводу з ладу усіх вузлів, що надають певну послугу. Атака за участю кількох комп'ютерів або декількох мереж для атаки на ціль у скоординованому вигляді називається «розподілена атака на відмову в обслуговуванні» [1].

Популярними жертвами таких атак зазвичай є комерційні та інформаційні сайти. Хакери останнім часом використовують такий вид атаки з метою вимагання грошей за припинення атаки. DDoS-атаки є активним інструментом конкурентної боротьби , а також інформаційних воєн.

Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих) таким чином атаковане устаткування не може відповісти користувачам, або відповідає настільки повільно, що стає фактично недоступним.

Зловмисники з кожним роком знаходять все нові вразливі місця та розробляють техніки реалізації атак. Останнім часом проводяться не прямі, а amplified reflection DDoS-attacks. Так у 2018 році хакери активно використовували вразливі місця в DNS, NTP, SSDP, CLDAP, Chargen та інших протоколах, щоб максимально збільшити масштаб своїх атак [2]. Крім того, спостерігається значне збільшення експлуатації пристроїв IoT для генерації великого потоку пакетів та атак на рівні додатків. Найбільша атака, відбулася в березні 2018 року на GitHub. Обрушилася потужна DDoS-атака в історії, яка становила 1,3 Тбіт/с або 126,9 млн пакетів в секунду [3]. На рис. 1.1 наведено динаміку потужності DDoS-атак [4].

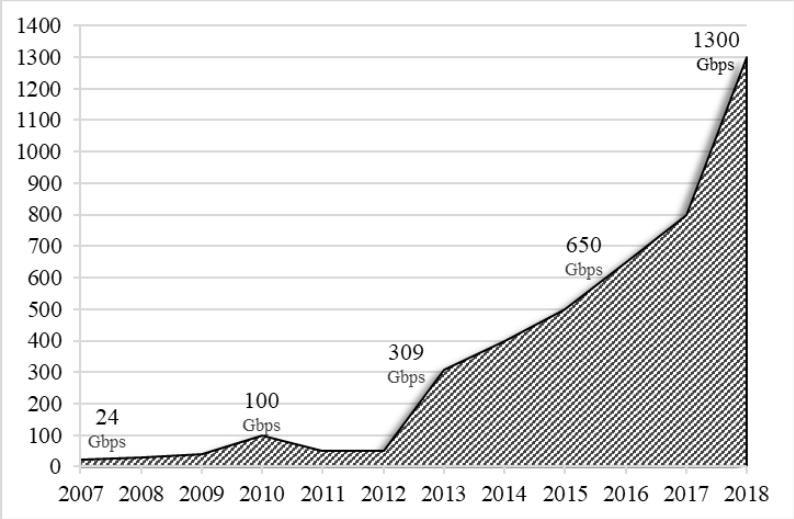

Рисунок 1.1 – Статистика потужності DDoS-атак

Типова DDoS-атака виконується в такі етапи [5]:

– Сканування: атакуючий сканує мережу на наявність вразливих машин (названих агентами чи ботами), які пізніше будуть використовуватися для здійснення нападу на справжню жертву. У минулому цей процес виконувався вручну, але сьогодні для цього може використовуватись декілька інструментів сканування.

– Експлуатація та зараження: машини агенти додаються до ботнету за рахунок використання їх виявлених вразливостей для введення в них шкідливого коду.

– Зв’язок: зловмисник використовує інфраструктуру команд та керування, щоб спілкуватися з ботнетом, визначити, які боти запущені та працюють, або запланувати атаку.

– Атака: зловмисник командує розпочати атаку, боти починають відправляти пакети жертві. Параметри атаки (жертва, тривалість та властивості пакета) зазвичай налаштовуються на цьому етапі (якщо це не було зроблено в попередньому). Незважаючи на те, що підміна IP-адрес не є обов’язковою вимогою для успішної DDoS-атаки, зловмисники часто використовують підробний IP-адрес джерела, щоб приховати справжніх ботів під час атаки.

Класифікація DDoS-атак наведена на рис. 1.2 [6, 7, 8].

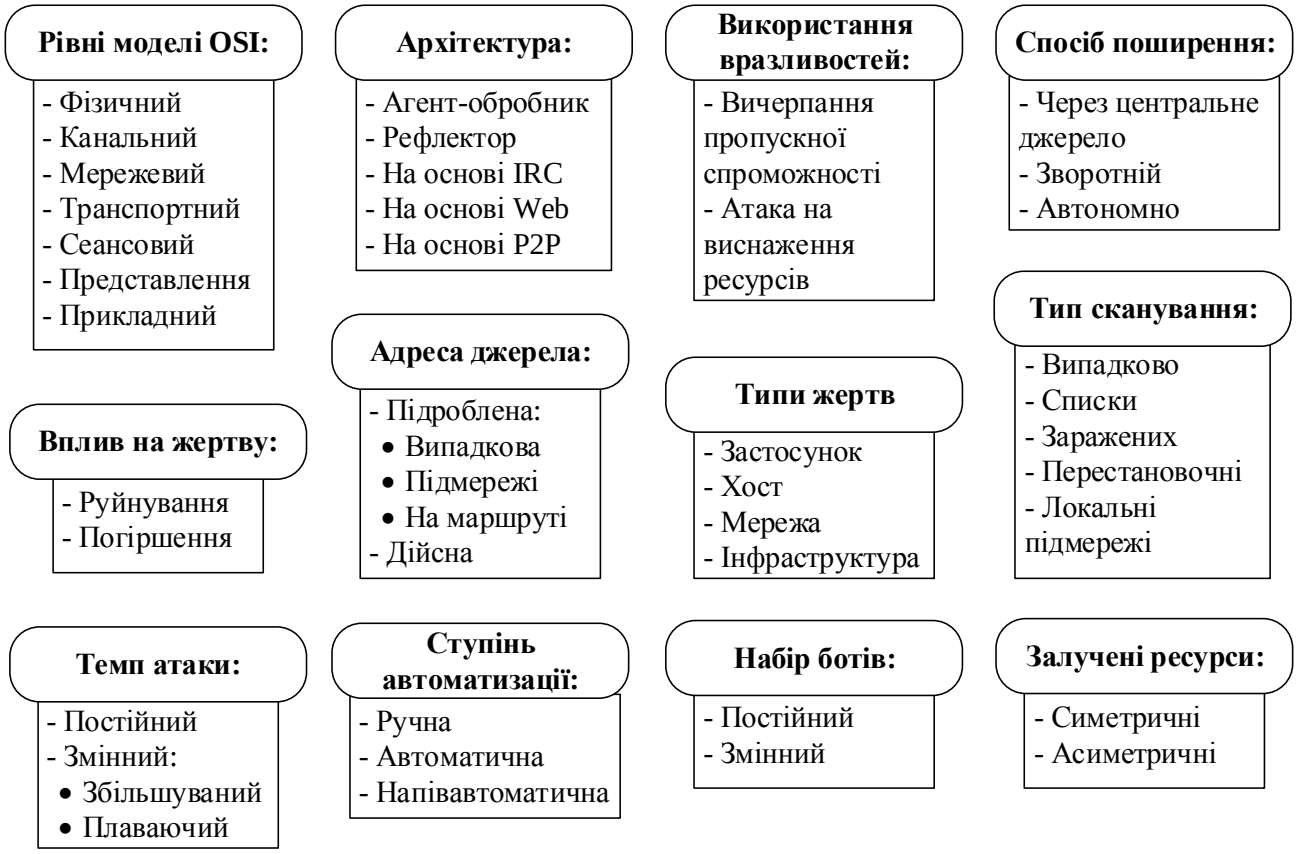

Рисунок 1.2 – Класифікація (таксономія) DDoS-атак

Для ефективного виконання атаки зловмисник може комбінувати різні методи, типи, механізми, що ускладнюють процес виявлення.

1.2 Методи захисту від DDoS-атак

Небезпека більшості DDoS-атак – в їх абсолютній прозорості і «нормальності». Адже якщо помилка в ПЗ завжди може бути виправлена, то повна витрата ресурсів – явище майже буденне. З ними стикаються багато адміністраторів, коли ресурсів машини (ширини каналу) стає недостатньо, або web-сайт піддається слешдот-ефекту [9]. Якщо обмежувати трафік і ресурси для всіх підряд, то можна врятуватися від DDoS-атаки, у той же час, втративши велику частину клієнтів.

Захист від DDoS-атак розділений на три напрямки: запобігання, виявлення та реакція на атаку. Основною метою запобігання атаці є припинення атак до нанесення нею збитків. Методи виявлення атаки спрямовані на контроль та дослідження системи в якої виникають нетипові

події. Методи реакції на атаку в яких жертва реагує на атаку після виявлення. У цих методах відповідальність на виявлення і боротьбу з нападом лежить на самій жертві.

В таблиці 1.1 показано основні напрямки захисту від DDoS-атак [10-14].

Таблиця 1.1 – Напрями захисту з перевагами та недоліками

Напрям	Типи	Переваги	Недоліки
Запобігання	• Фільтрація вхідного / вихідного трафіка • Фільтрація на рівні маршрутизатора • Протокол перевірки адреси джерела • Фільтрація за рахунком хоп пакетів	• Запобігання підміни IP-адреси джерела • Запобігання підміни IP для статичних маршрутів • Фільтрація трафіку до того як він досягне цілі • Запобігання flood атак	• Атаки з дійсних IP-адреси джерела • Важко розгорнути фільтрацію по всьому світу • Неможливо запобігти атаку з дійсною IP-адресою джерела • Не може запобігти flood атак на пропускну здатності
Виявлення	• Відстеження за активної взаємодії • Вкладені поля • На основі хеша • Сигнатурні • Статистичні	• Інформація в пакеті достатня, без необхідності створювати додаткові пакети • Простота в реалізації • Може відслідковувати шлях атаки навіть для пакетів низького об’єму • Точне визначення адреси джерела або призначення	• Технічно тривіальна • Необхідне широке розгортання, щоб бути ефективним • Велика вартість розгортання • Структура даних, яка контролює частоту пакетів вразливих до атаки виснаження пам’яті • Потрібно компенсувати швидкість обробки та точність виявлення
Реакція	• StopIt • SIFF • TVA	• Запобігання пошкодженню законного трафіку • Немає необхідності співпраці між ISP або кінцевого хоста/ISP • Запобігання атаці flood проти можливостей каналу	• Необхідне ручне налаштування StopIt маршрутизаторів та серверів • Потрібне широке розгортання механізму SIFF для кращої ефективності • Залежить від точності механізмів ідентифікації джерела

За 2017 рік такі методи захисту як: міжмережевий екран, IPS, WAF та списки контролю доступу (ACLs) залишаються найбільш поширеними механізмами захисту від DDoS-атак (рис. 1.3) [4]. Використання міжмережевих екранів, IPS і WAF залишається проблемою, оскільки ці пристрої чутливі до атак із виснаженням ресурсів, на противагу.

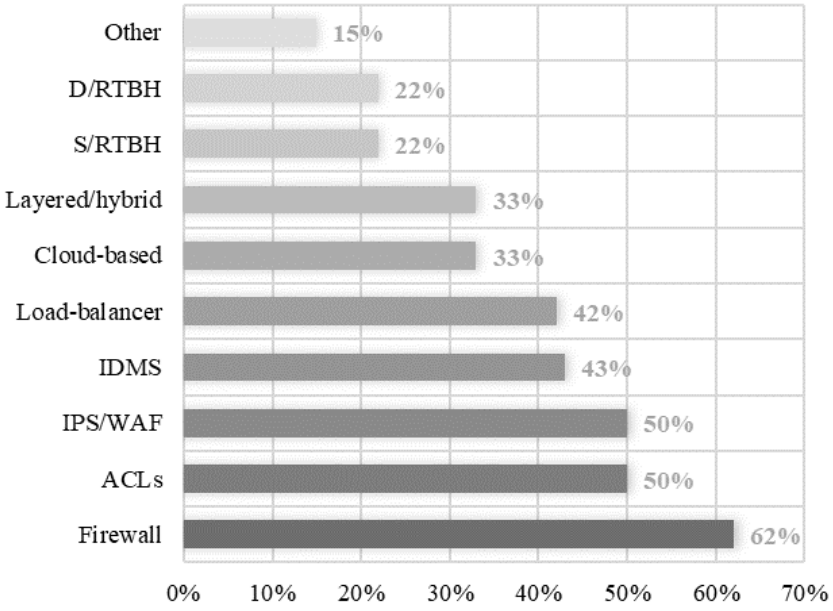

Рисунок 1.3 – Методи захисту від DDoS-атак

Як показав аналіз найбільш ефективні засоби виявлення та попередження DDoS-атак базуються на використання інтелектуальних методів (IDMS), серед яких можна виділити: статистичний, експертні системи (сигнатурний метод, байєсовський класифікатор), на основі soft computing (нейронні мережі, нечітка логіка та генетичні алгоритми), Data Mining та машинне навчання [15, 16]. Статистичні методи спроможні адаптуватись до поведінки суб'єкта, не вимагають знання про можливі атаки, але важко визначити порогові значення відстежуваних характеристик. В експертних системам відсутні хибні тривоги, але необхідно постійно оновлювати сигнатур. Нейромережеві аналізатори та генетичні алгоритми спроможні «вивчати» характеристики атак і ідентифікувати (класифікувати) елементи, але точність виявлення атак залежить від якості навчання нейромережі. При використанні Data Mining та машинного навчання витрачається мало часу для виявлення атаки, але вимагається достатньо великих обчислюваних потужностей.

1.3 Постановка задачі

Одним з перспективних напрямів забезпечення безпеки мережі є використання засобів виявлення атак, побудованих на основі інтелектуальних

технологій, а саме штучних нейронних мереж (НМ). На сьогодні ШНМ дозволяють створити ефективну адаптивну систему з високою точністю виявлення мережевих вторгнень і забезпечити надійний рівень захисту комп'ютерних систем від зовнішніх атак. Ця технологія носить як сигнатурний характер, так і може бути само адаптованою до нових видів атак.

Типовими задачами, що вирішується за допомогою нейронних мереж, є задачі класифікації, ідентифікації та розпізнавання образів [17, 18]. Саме до такого класу задач і відноситься попередження та виявлення DDoS атак. НМ навчають за допомогою набору даних з відомими відповідями. Мережевий трафік в задачі виявлення мережевої атаки можна розділити на два класи: нормальний і аномальний. Де аномальний трафік це мережева атака, яка характеризується певним набором характеристик. При цьому клас аномального трафіку можна розділити на декілька класів по видам атак, що підлягають виявленню.

Перевагами використання нейромережевого підходу при виявленні DDoS-атаки є гнучкість та апроксимаційну універсальність [19]. Нейромережа здатна аналізувати неповні або перекручені дані, одержувані з мережевого трафіку. Здатність обробляти дані від великої кількості джерел є особливо важливою при розгляді DDoS-атак, проведених проти мережі скоординованими численними атакуючими.

Зараз проводиться велика кількість досліджень по застосуванню нейронних мереж до виявлення DDoS-атак. Проте вони не у повній мірі обґрунтовані, недостатньо наповнені деталями отриманих результатів, мають не вискоку точність виявлення. Тому метою роботи є спроба підвищення обґрунтованості методики та точності виявлення DDoS-атак на основі нейромережевого підходу. Для дослідження обрано найбільш популярні типи DDoS-атак по основним рівням стеку TCP/IP:

- прикладний рівень – HTTP-flood;
- транспортний рівень – SYN-flood, UDP-flood;
- мережний рівень – ICMP-flood.

2 ПРОЕКТУВАННЯ НЕЙРОМЕРЕЖІ

2.1 Підготовка навчальної вибірки

Більшість дослідників, пов’язаних із дослідженням DDoS-атак, використовують загальнодоступні реальні зібрані дані, які називаються набором даних (dataset). Набір даних – це колекція однотипних даних, що застосовуються в завданнях обробки даних [20].

Формування начальної вибірки виковувалось в декілька етапів:

1) Збір даних: було використано набори даних KDD [21] та авторів [17].

2) Перетворення даних: перетворення набору даних у формат .xlsx в якому видаляються зайві записи.

3) Попередня обробка: всі записи типу string конвертуються в числа для належного сприйняття даних нейронною мережею.

Результатом буде вхідний вектор нейронної мережі, який складатися з 54 619 рядків, серед яких: 70% виділено на тренування, 15% на перевірку та 15% для тестування. Структура навчальної вибірки по класам діагностування є наступною: нормальний трафік – 48%, HTTP-flood - 2%, ICMP-flood − 25%, SYN-flood 28% − UDP-flood (3%). Рядок матиме наступний вигляд:

[udp; http; 1118; 1500, 0, 40, 40, 112, 40]

В табл. 2.1 наведено вхідний вектор, параметри та ознаки які дозволяють визначити DDoS-атаку.

Таблиця 2.1 – Список вхідних векторів та ознак атаки

Приклад елементів вхідногой вектор	Параметр	Ознака атаки
udp	Тип протоколу	Аналіз транспортного та мережевого рівня для ідентифікації ознак атаки на ранній стадії
http	Тип обслуговування	Аналіз протоколів прикладного рівня через які здійснюється атака (HTTP-flood). Аномальна кількість запитів GET/POST для надмірного споживання системних ресурсів

Продовження таблиці 2.1

1118	Швидкість передачі	Головні параметри, які характеризують атаку flood. Великі показники які мають на меті за короткий час досягти меж ширини каналу та кількості допустимих підключень чи рівномірні, які поступово виснажують ресурси цілі
1500	Розмір пакету	
0	Якщо IP-адреси джерела і призначення та номера портів рівні – 1, інакше 0	Підміна відправленого пакету свідчить про атаку
40	Кількість підключень до одного і того ж хоста за останні 2 секунди	Статистичні характеристики які дають змогу швидко ідентифікувати атаку. Великий чи аномальний показник найчастіше свідчить про наявність атак типу flood.
40	Кількість підключень до однієї служби (номеру порта) за останні 2 секунди	
112	Кількість з’єднань, що мають однакову IP-адресу хоста призначення	
40	Кількість з’єднань, що мають один і той же номер порта	

Для точного визначення даних нейронною мережею, необхідно конвертувати всі записи, що мають текстову форму (табл. 2.2). Також слід зазначити, що присвоєння відповідям близько діапазону чисел та подальша нормалізація може призвести до погіршення виявлення.

Таблиця 2.2 – Дані про кодування

Тип обслуговування	Код	Тип протоколу	Код
http	11	tcp	1
imap	22	icmp	2
smtp	33	udp	3
other	44	Виходи НМ	Код
echo	55	Normal	1 (10000)
pop 3	66	HTTP-flood	0.2 (01000)
telnet	77	ICMP-flood	0.4 (00100)
finger	88	SYN-flood	0.6 (00010)
		UDP-flood	0.8 (00001)

На рисунку 2.1 зображено фрагмент оброблених даних які подаються нейронній мережі на вхід.

	A	B	C	D	E	F	G	H	I	J
16493	1	11	431	37	0	232	16	255	16	440
16494	1	11	505	59	0	202	1	255	1	440
16495	1	11	536	46	0	272	7	255	7	440
16496	1	11	615	63	0	292	17	255	17	440
16497	3	55	308	55	0	511	511	255	255	330
16498	3	55	34	1543	0	511	511	255	255	330

Рисунок 2.1 – Фрагмент даних для навчання

Причина застосування вибраних параметрів при розробці моделі ґрунтується на попередніх дослідженнях, які проводились в попередньому розділі. Комбінація цих параметрів дозволить ефективно виявляти DDoS-атаки.

2.2 Моделювання архітектури нейромережі

Для моделювання нейронної мережі було обрано середовище розробки MatLab R2017a. MatLab містить інструменти Neural Network Toolbox, які забезпечують алгоритми, попередньо навчені моделі та додатки для створення, навчання, візуалізації та імітації як неглибоких та глибоких нейронних мереж [22].

Перед початком моделювання слід імпортувати дані (див. рис. 2.1), які були збережені в .xlsx файлі. Необхідні дані конвертуються в матрицю та зберігаються в зміні.

Імпортовані дані ділять на дві частини: дані для формування вхідного вектора inputs та для перевірки відповідей нейронної мережі targets.

В дослідженні розглядались декілька типів ШНМ типу багатошарового перцептрону: Free-forward (fitnet), Cascade Free-forward (cascadeforwardnet) та Pattern Recognition (patternnet).

При моделюванні нейронних мереж важливим і складним етапом є оптимальний підбір параметрів, а саме кількість шарів та нейронів у них, функції активації, алгоритм та критерій навчання. Вибір кількості нейронів прихованого шару чи самих шарів, впливає на точність визначення мережі в цілому. Мала кількість нейронів спричинить багато похибок у визначенні, а велика кількість перенавчання. Серед функцій активації, які використовується

для нормалізації вхідних даних, було досліджено: logsig (логарифмічна сигмоїда), яка приймає значення у діапазоні від 0 до 1; tansig (тангенціальна гіперболічна сигмоїда), що приймає діапазон від -1 до 1 та purelin лінійна функція. Алгоритм навчання має важливе значення при проектуванні нейронної мережі, адже за його допомогою модифікуються значення ваг і зміщень для знаходження оптимума відповідно до методу оптимізації. В задачах класифікації та розпізнавання гано себе зарекомендували алгоритм Левенберга-Марквардта (trainlm, вимагає більших ресурсів за рахунок громістких обчислень) та алгоритм масштабованого спряженого градієнта (trainscg, вимагає менше ресурсів, через простоту обчислень). Вибір зазначених параметрів здійснюється зазвичай експериментальним шляхом.

Визначаємо кількість нейронів в прихованому шарі та призначаємо алгоритмом навчання нейронної мережі метод Левенберга-Марквардта:

```
>> hiddenLayerSize = 18;
>> trainFcn = 'trainlm';
```

Створюємо мережу прямого поширення (Free-forward) з виділених параметрів та переглядаємо (рис. 2.2):

```
>> ffnet = fitnet(hiddenLayerSize, trainFcn);
>> view(ffnet);
```

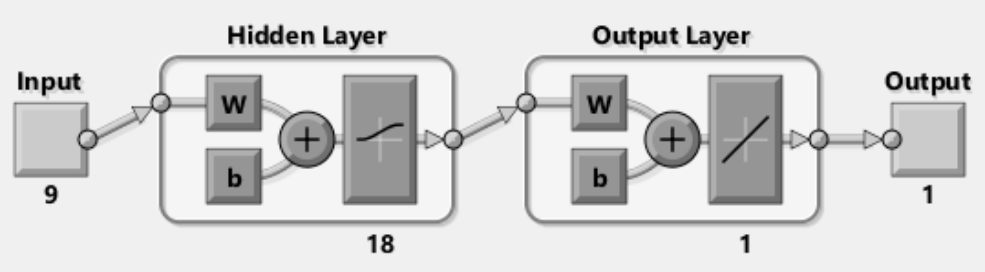

Рисунок 2.2 – Архітектура мережі прямого поширення

Для нормалізації даних використовуються функції removeconstantrows та mapminmax:

```
>>net.input.processFcns = {'removeconstantrows','mapminmax'};
>>net.output.processFcns= {'removeconstantrows','mapminmax'};
```

Вибір даних нейронною мережею – випадковий:

```
>> ffnet.divideFcn = 'dividerand';
```

Розподілення вибірки: 70% на тренування, 15% на перевірку та 15% для тестування:

```
>> ffnet.divideParam.trainRatio = 70/100;
>> ffnet.divideParam.valRatio = 15/100;
>> ffnet.divideParam.testRatio = 15/100;
```

Процес навчання, складається з трьох параметрів: мережа, вхідні дані та правильні відповіді:

```
>> train(ffnet,input,target);
```

Після виконання команди відкриється вікно з процесом навчання яке зображено на рис. 2.3.

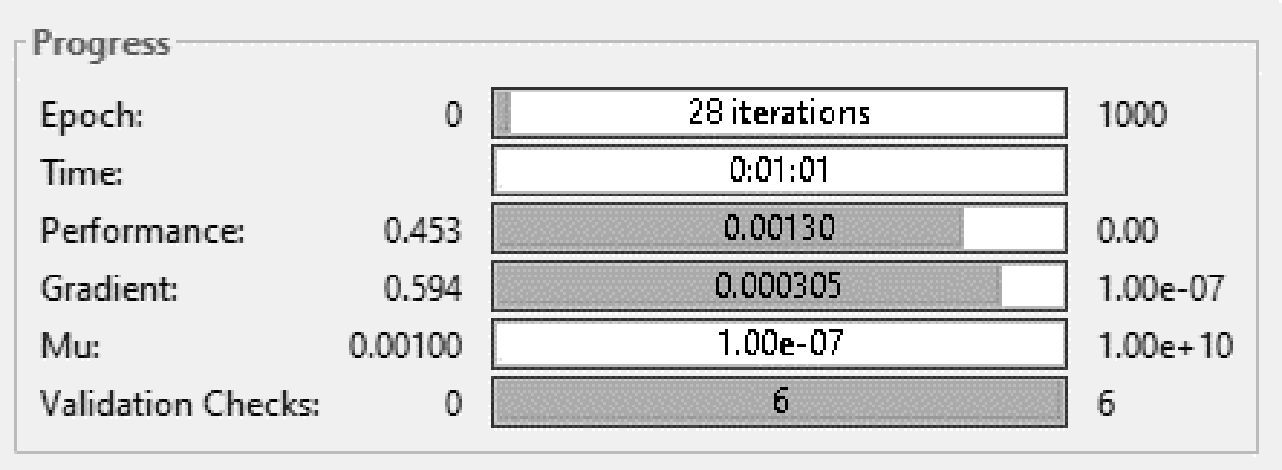

Рисунок 2.3 – Процес навчання мережі

Процес навчання було зупинено, через те, що мережа навчилась до потрібного рівня точності. Навчання зайняло 1 хвилину 37 секунд, пройшовши при цьому 72 ітерації. Отримані результати показали, що коефіцієнт відхилення є малим, при такій простій архітектурі мережі.

Вікно графіка залежності коефіцієнта відхилення від кроків ітерації зображено на рис. 2.4.

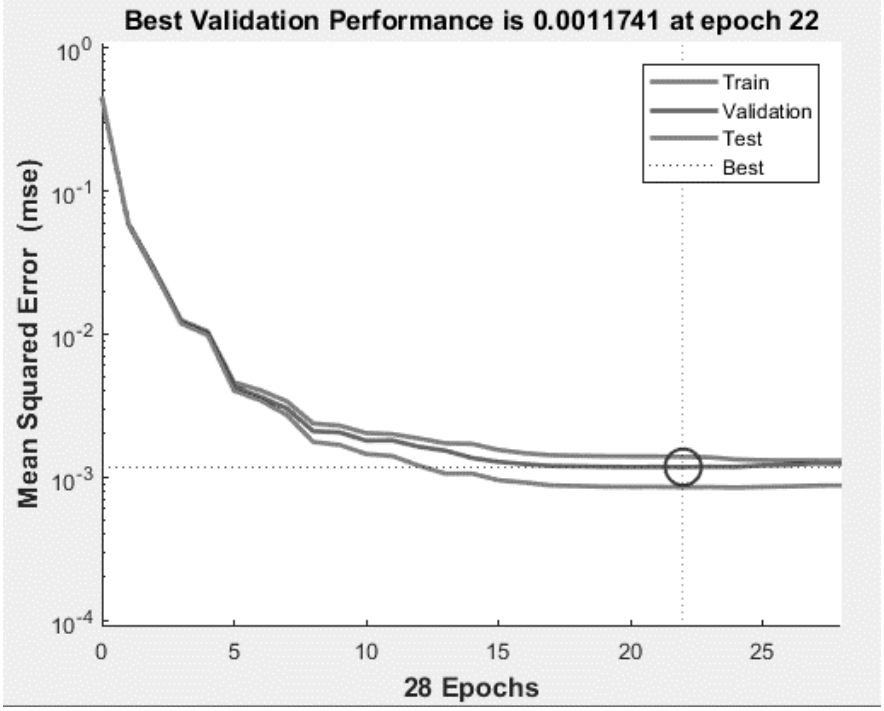

Рисунок 2.4 – Графік коефіцієнта відхилень мережі під час навчання

Аналогічне моделювання відбувається з каскадної архітектурою нейронної мережі (Cascade Free-forward). Головною відмінністю такої нейронної мережі, від мережі прямого поширення в тому, що шар з вхідними даними з’єднаний не тільки з наступним прихованим шаром, але і з вихідним шаром. Створена нейронна мережа зображена на рис. 2.5.

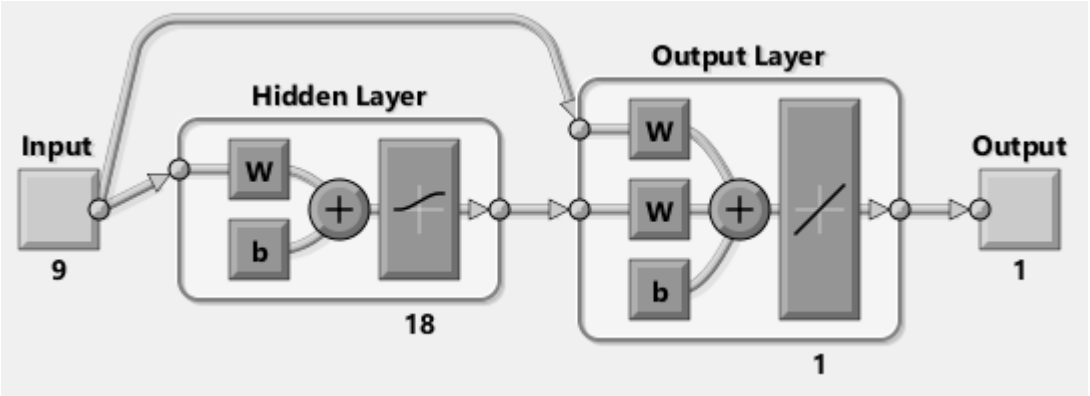

Рисунок 2.5 – Архітектура каскадної нейронної мережі

Процес навчання збільшився, якщо порівнювати з попередньою нейронною мережею, зайнявши 4 хвилини 50 секунд та було виконано 85 ітерацій (рис. 2.6).

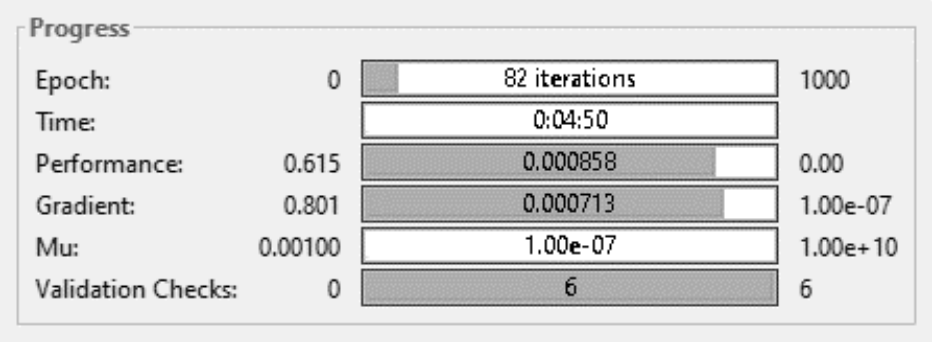

Рисунок 2.6 – Процес навчання мережі

Результати тестування каскадної нейронної мережі прямого поширення показали кращі результати, але процес навчання збільшився в декілька разів порівняно з попередньою мережею. (рис. 2.7).

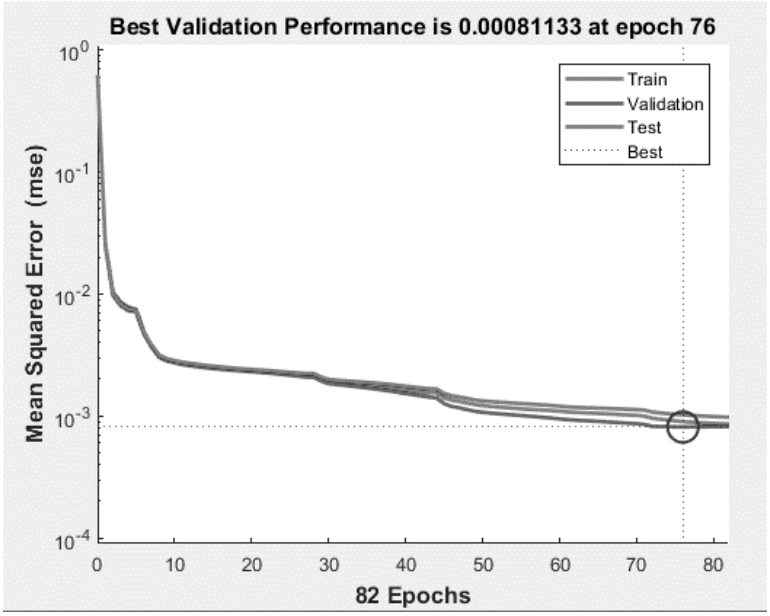

Рисунок 2.7 – Графік коефіцієнта відхилень мережі під час навчання

Останньою моделювалась мережа, яка відповідає за розпізнавання образів та класифікацію (Pattern Recognition). Вона є схожою до попередніх, але на вихідному шарі містить softmax функцію, що «стискує» K-вимірний вектор z із довільним значеннями компонент до K-вимірного вектора σ (z) з дійсними значеннями компонентів в області від 0 до 1.

Виходи softmax роблять її придатними для ймовірнісної інтерпретації, що дуже корисно при машинному вивченні. Зокрема, у завданнях багатокласової

класифікації, коли часто необхідно присвоїти ймовірності того, що вхід належить одному з наборів класів виводу.

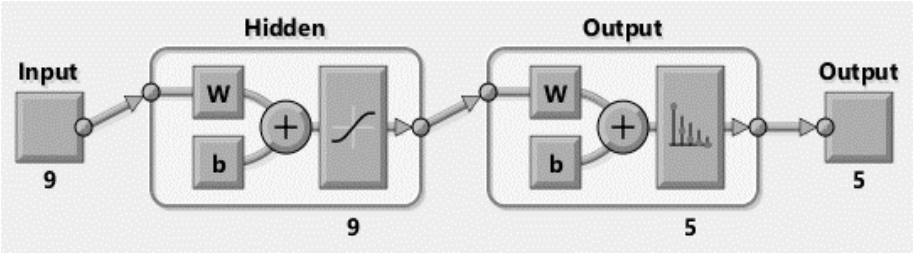

Рисунок 2.8 – Архітектура мережі Pattern Recognition

Отриманні результати показали наближені результати до двох попередніх змодельованих мереж.

Для обґрунтування вибору необхідно провести моделювання нейромереж з різними параметрами.

2.3 Аналіз результатів проектування

Для оцінки точності потрібно визначити помилки, які допускала нейронна мережа до яких відносять:

– помилки першого роду, коли нейронна мережа розпізнає нормальний трафік як DDoS-атаку;

– помилки другого роду, коли DDoS-атака розпізнається як нормальний трафік.

Для визначення точності та помилок використовується функція plotconfusion в якої параметри входу є результат мережі та правильні відповіді. Результати порівняння записані в таблицю 2.3.

Таблиця 2.3 – Порівняльні характеристики нейронних мереж

Тип мережі	Кількість нейронів	Функції активації/метод навчання	Час навчання	Помилки 1-го роду	Помилка 2-го роду	Точність
Free-forward	9	tansig-logsig/trainlm	8:01	1,4%	5,8%	92,8%
	18	logsig-perelin/trainscg	1:29	2,0%	0,4%	97,6%

Продовження таблиці 2.3

Cascade Free-forward	9	tansig-logsig/trainlm	17:04	3,1%	6,5%	90,4%
	18	logsig-perelin/trainscg	1:50	1,3%	2,2%	96,5%
Pattern Recognition	9	tansig-softmax/trainlm	1:23	1,4%	0,6%	98,0%
	18	tansig-softmax/trainscg	0:54	0,1%	0,1%	99,8%

Точність обраховувалась як відношення суми класифікованих даних порівнюючи з округленими даними нейронної мережі до загальної кількості правильних відповідей.

Результати кращої архітектури зображено в матриці помилок на рис. 2.9. Матриця помилок являє собою таблицю, що показує відповідність між результатом класифікації та еталоном. Кожен рядок матриці являє собою екземпляри у передбачуваному класі, тоді як кожен стовпець відображає екземпляри у еталонному класі (або навпаки).

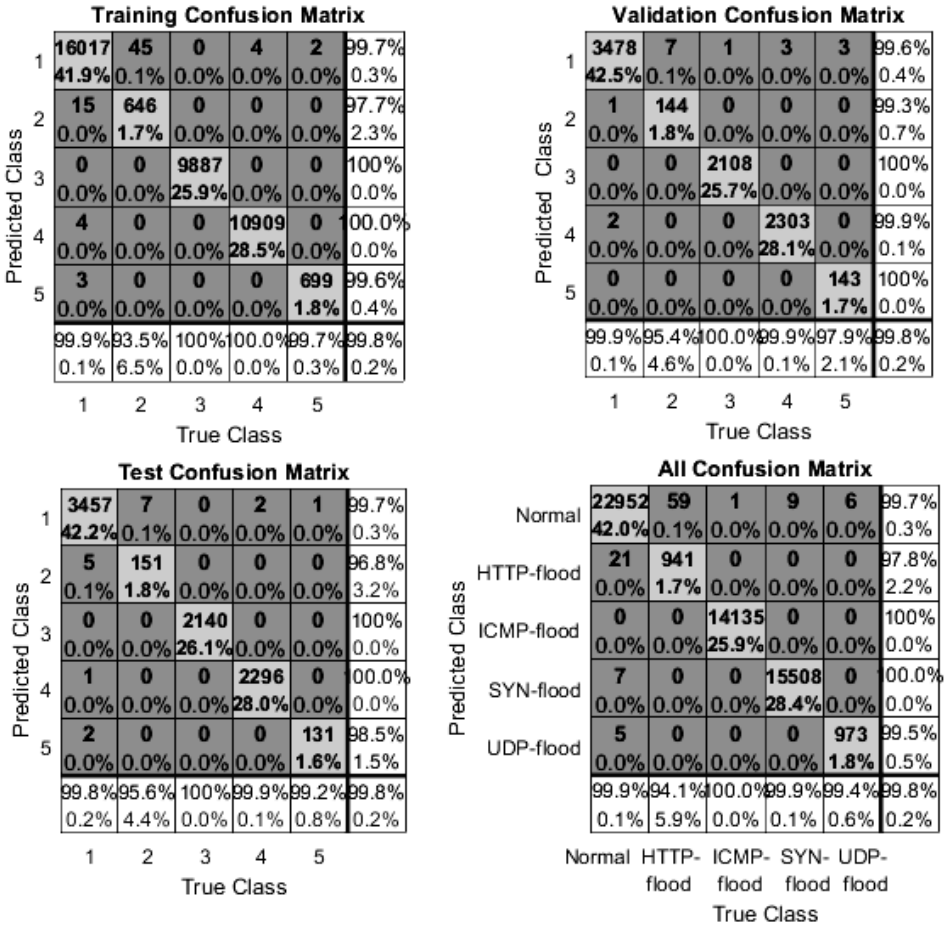

Рисунок 2.9 – Матриця похибок нейронної мережі Pattern recognition

Найкращу точність показала мережа Pattern recognition з функцією активації tansig в якої мінімальна кількість похибок.

Вцілому точність виявлення нейронною мережею досягла 99,8 %, відповідно результати похибки першого та другого роду є дуже малими. В таблиці 2.4 наведено порівняння запропонованої навченої НМ із іншими реалізаціями: Feedforward Neural Network(MLP), Probabilistic Neural Network (PNN), Back-Propagation (BP), статистичний підхід Chi-squar та Support Vector Machines (SVM).

Таблиця 2.4 – Аналоги методів виявлення DDoS-атак

Тип НМ	Точність	Атаки
MLP [15]	98,6%	Smurf, UDP Flood, HTTP Flood, SIDDOS
PNN [23]	97%	TCP, UDP
BP [24]	98%	ICMP, TCP, UDP
Chi-squar [25]	94%	TCP Flood, UDP Flood, ICMP Flood
SVM [18]	98,4%	ICMP Flood, Smurf, TCP Flood, UDP Flood
Pattern recognition	99,8%	HTTP-flood, ICMP-flood, SYN-flood, UDP-flood

Таким чином змодельована нейрона мережа показали досить високу точність виявлення DDoS-атак порівняно з іншими аналогами.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБУ

3.1 Розробка структури та модулів програмного засобу

Програмний засіб складається з двох модулів: модуль роботи з файлом та модуль аналізу даних за допомогою нейронної мережі. Загальний алгоритм роботи зображено на рис. 3.1.

Рисунок 3.1 – Загальний алгоритм роботи програми

За допомогою модуля роботи з файлом виконується завантаження даних з excel файлу та представлення його у вигляді таблиці. Далі дані надсилаються на вхід модуля аналізу даних в нейронній мережі, де результати обчислень зберігається в той самий excel файл. Результати обраховуються та зображуються в списку.

Модуль аналізу даних за допомогою нейронної мережі виконує основі операції обчислення, в результаті якого отримується результати класифікації. Алгоритм роботи модуля зображено на рис. 3.2.

Рисунок 3.2 – Модуль аналізу даних за допомогою нейронної мережі

Вхідні дані представляються у вигляді матриці. Відбувається процес перевірки даних: на вхідному шарі виконується нормалізація даних матриці, обмеження максимальних та мінімальних значень в проміжку від -1 до 1; в

прихованому та вихідному шарах відбувається обрахування даних з урахуванням ваг та функції активації. Виходом нейронної мережі є матриця класифікованих даних.

За допомогою функції genFunction(patnetTR, NeuralNetwork) створюється повна автономна функція Matlab для імітації нейронної мережі, що включає всі параметри, значення ваг та зміщення, функції модулів та розрахунки в одному файлі NeuralNetwork.m (рис. 3.3) [Додаток А].

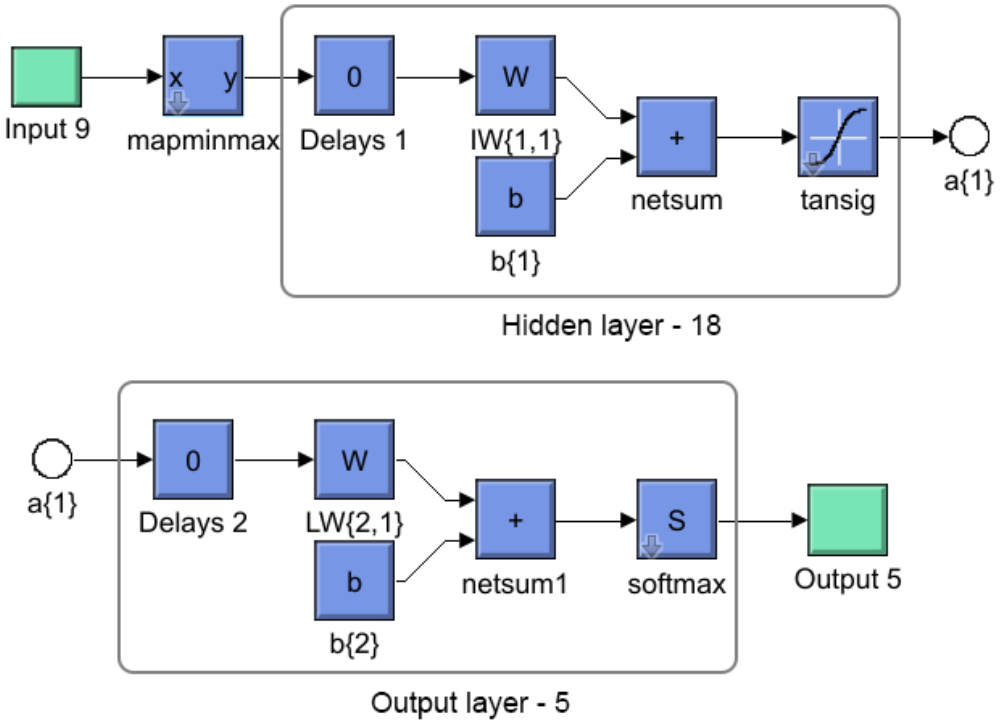

Рисунок 3.3 – Схема складових функції нейронної мережі

Matlab Compiler дозволяє конвертувати Matlab m-файли автоматично в C і C++ для створення незалежних програмних продуктів. Створення виконуваного файлу програми відбувається в три етапи: створення бінарного файлу, пакування у виконуваний файл та встановлення файлу (рис. 3.4).

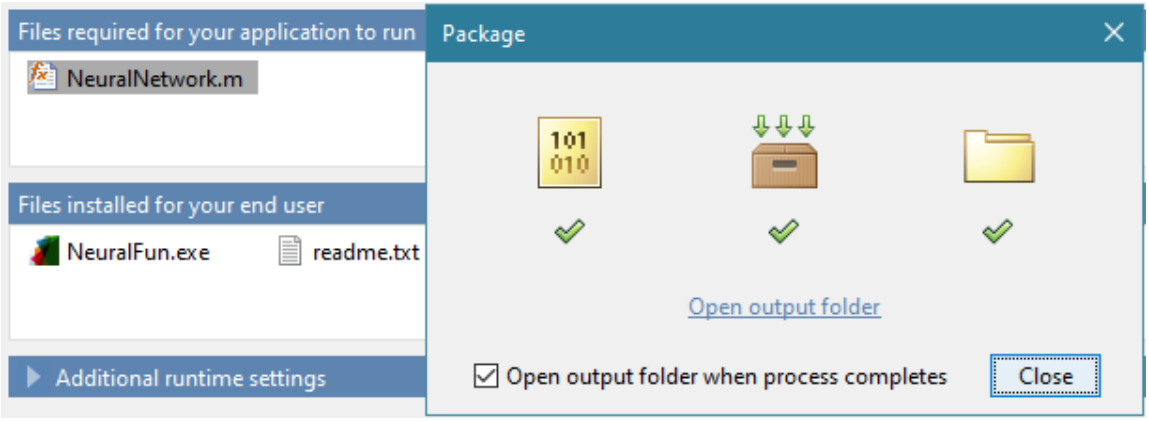

Рисунок 3.4 – Процес створення виконуваного файлу

Для розробки інтерфейсу програмного засобу обрано мову C#, яка є потужною об’єктною-орієнтованою мовою з можливостями спадкування й універсалізації. Велика бібліотека каркасів підтримує зручність побудови різних типів додатків на C#, дозволяючи легко будувати необхідний додаток, інші види компонентів, досить просто зберігати й одержувати інформацію з бази даних й інших сховищ даних.

Середовищем розробки було обрано Microsoft Visual Studio 2015, яке надає досить легкий підхід і широкі можливості в розробці Windows Form додатків та має готові бібліотеки для роботи з C#.

Програма складається з декількох компонентів:

– Вхідні дані - dataGridView, таблиця експортованих даних файлу.

– Статус - lable, показує теперішній стан програми.

– Трафік – ListView, список трафіку та його кількості.

На рис. 3.5 зображено інтерфейс програмного засобу у конструкторі Microsoft Visual Studio.

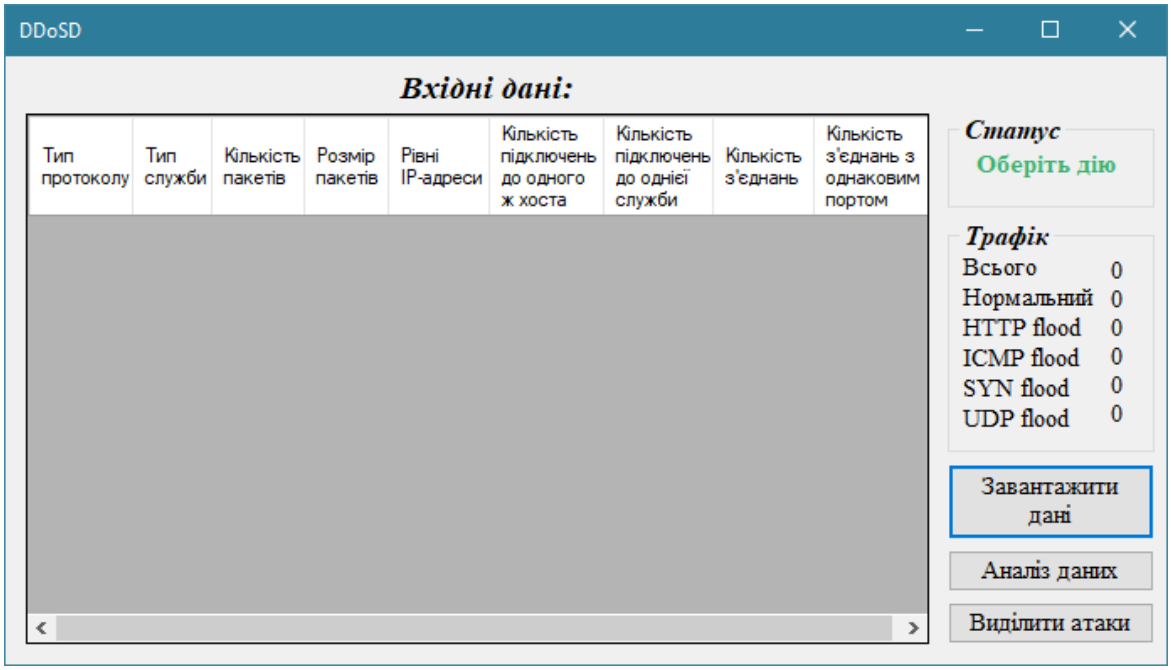

Рисунок 3.5 – Інтерфейс програмного засобу

При натисканні кнопки «Завантажити дані» виконується модуль роботи з файлом. Дані завантажуються з excel файлу з обраним шляхом, форматом, дозволом на читання та вибору аркуша. Дані додаються поступово за допомогою цикла в DataGridView.

```
ExcelWorkBook = ExcelApp.Workbooks.Open(path,0,true,5,"","",
true,Excel.XlPlatform.xlWindows,"\t",false,false,0,true,1,0);
ExcelWorkSheet =(Excel.Worksheet )ExcelWorkBook.Worksheets.get_
Item(1);
ExcelRange = ExcelWorkSheet.UsedRange;
for(rCnt = 1; rCnt <= ExcelRange.Rows.Count; rCnt++){
dataGridView1.Rows.Add(1);
```

При натисканні кнопки «Аналіз даних» виконується модуль аналізу даних за допомогою нейронної мережі. Вхідні дані надходять з excel файлу, обраховуюся мережею, округлюються та знову записуються у файл.

```
input=xlsread('input.xlsx','inputs');
output=NeuralNetwork(input');
output = round(output);
xlswrite('input.xlsx',output','outputs');
res = sum(output,2);
xlswrite('input.xlsx',res,'result');
```

Результати обрахування зображуються в listBox, який містить тип та кількість ідентифікованого трафіку.

3.2 Тестування програмного засобу

Тестування проводилось на основі input.xlsx файлу, який містить в собі 159 векторів вхідних даних різних типи DDoS-атак. Кожен з 9 стовпчиків файлу відповідає з певну ознаку, що допомагає ідентифікувати атаку. При подані на вхід не правильного сформованого файлу програма проінформує про помилку.

При вході в програму буде запропоновано три дії: «Завантажити дані», «Аналіз даних» та «Виділити атаки».

Перша дає змогу побачити вхідні дані, але при великій кількості може викликати довге завантаження (рис. 3.6).

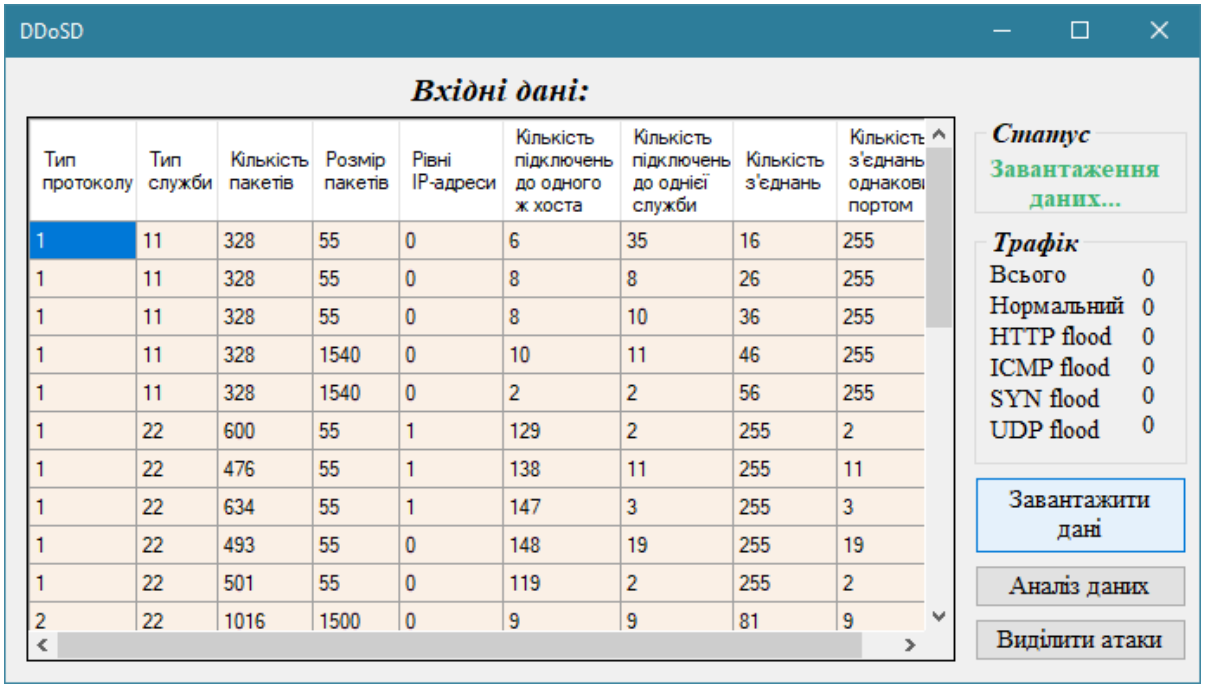

Рисунок 3.6 – Вікно програми з завантаженими даними

Друга обраховує дані та показує кількість визначених даних в колонці «Трафік» яке зображено на рис. 3.7.

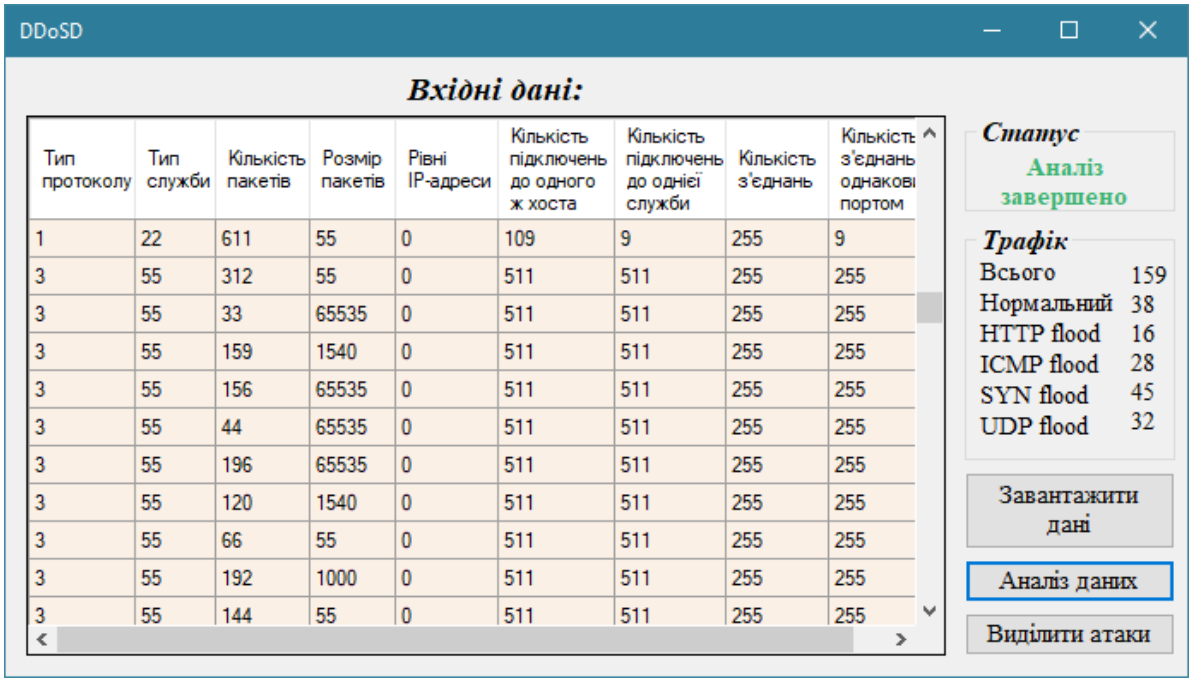

Рисунок 3.7 – Вікно програми з проаналізованим даними

Використання кнопки «Аналіз даних», при старті програми, дає змогу отримати результати без завантаження даних, що дає змогу зекономити час та ресурси системи.

Третя кнопка дасть змогу виділити атаки по кольорам, за умови завантажених вхідних даних (рис. 3.8).

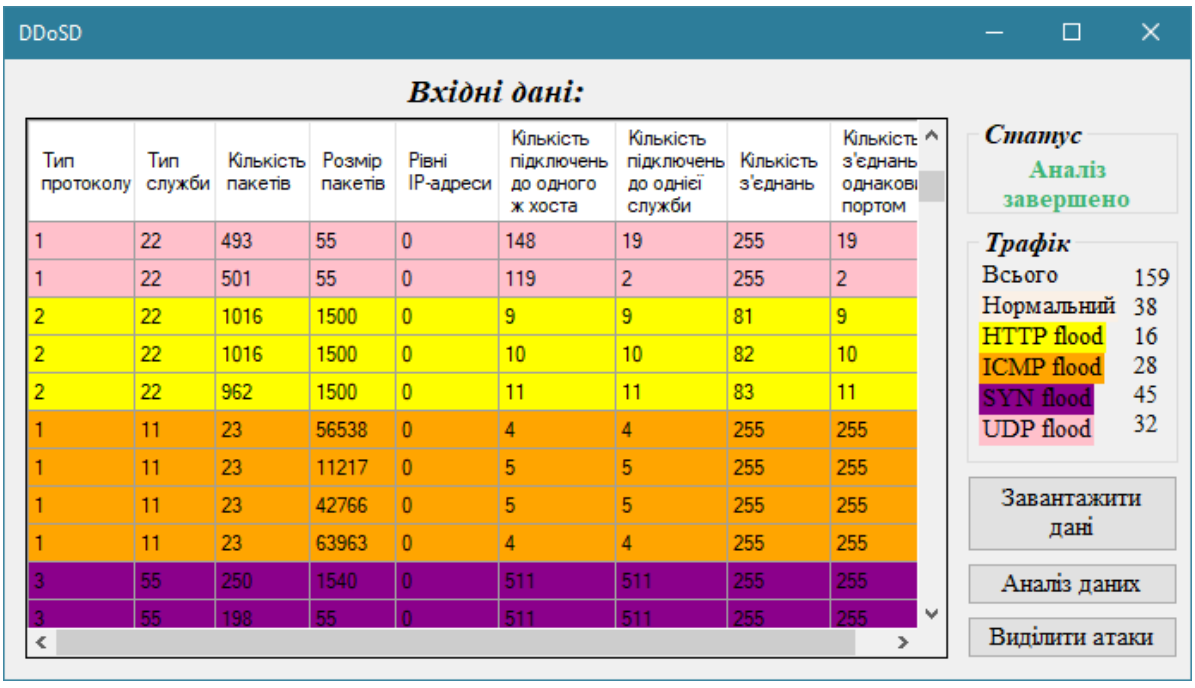

Рисунок 3.8 – Вікно програми з проаналізованим даними одного типу

При спробі завантажити дані була отримана помилка, яка зображена на рис. 3.9. Помилка означає, що файл з вхідними даними відсутній у директорії програми, для її виправлення необхідно перемістити excel файл до виконуваної програми.

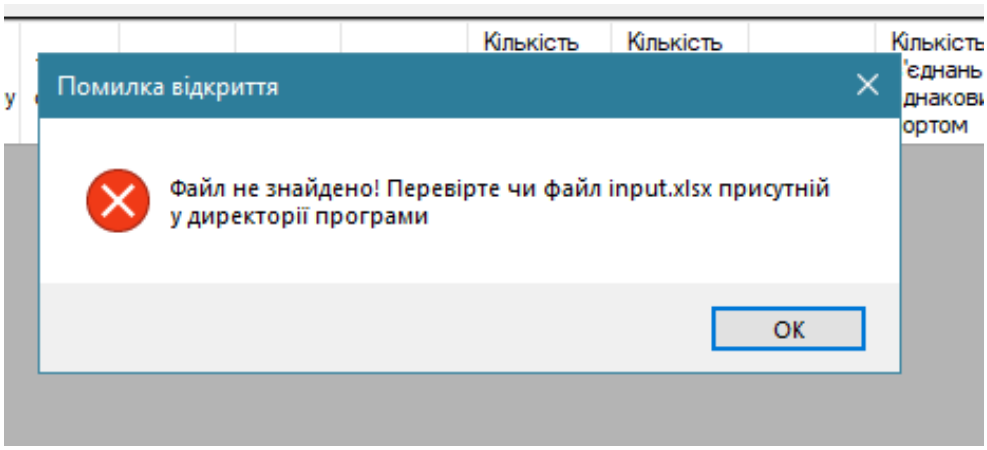

Рисунок 3.9 – Вигляд вікна з помилкою

Проведені тексти показали, що програма працює коректно, виконуючи завдання для яких була розроблена, а саме виявлення кількості DDoS-атак та збереження результатів у файл.

ВИСНОВКИ

Було проведено дослідження DDoS-атак та здійснено аналіз існуючих інструментів захисту. Промодельовано в середовищі MatLab ряд нейронних мереж для вирішення задачі виявлення DDoS-атак. Експериментальні дослідження ще раз довели , що НМ можуть з високою точністю вирішувати задачі класифікації. Точність виявлення атаки склала 99,8%, що є дуже гарним результатом. У перспективах даного дослідження є апробація результатів в реальних або лабораторних умовах. Для цього необхідно реалізувати програмний додаток на основі навченої нейромережі. При цьому MatLab надає усі можливості для цього, а саме дозволяє отримати exe-файл чи dll-файл спроектованої мережі для подальшого використання. А це дає можливість розробити для серверу автономну систему або як модуль системи виявлення вторгнень. Такий засіб буде гарним помічником на службі у системного адміністратора. Також доцільним було б дослідити численні більш вузьконаправленні DDoS-атаки, наприклад , ті, що базуються на udp протоколі, які зараз стали дуже популярними. Також у перспективах є дослідження нейро-нечіткого та само адаптивного підходів до вирішення таких задач.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DDoS-атаки: реальна небезпека віртуального світу [Електронний ресурс]. – Режим доступу: http://zillya.ua/ddos-ataki-realna-nebezpeka-virtualnogo-svitu – Назва з екрану.
2. Shin D. How to defend against amplified reflection ddos attacks, [Електронний ресурс]. – Режим доступу: https://www.a10networks.com /resources/articles/how-defend-against-amplified-reflection-ddos-attacks – Назва з екрану.
3. Memcached-fueled 1.3 tbps attacks, [Електронний ресурс]. – Режим доступу https://blogs.akamai.com/2018/03/memcached-fueled-13-tbps-attacks.html
4. NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report [Електронний ресурс]. – Режим доступу : https://www.netscout.com/report/
5. Принципи організації DDoS-атак [Електроний ресурс] – Режим доступу: http://hi-news.pp.ua/tehnka-tehnologyi/2883-ddos-ataka-yak-zrobiti- programi-dlya-ddos-atak.html – Назва з екрану
6. Voytovych O.P. Investigation of denial-of-service attacks / O.P. Voytovych, E.I. Kolibabchuk, L.M. Kupershtein // Вісник ХНУ. Технічні науки − №3. – 2016. – С. 129−133.
7. A. Asosheh and N. Ramezani, «A comprehensive taxonomy of DDoS attacks and defense mechanism applying in a smart classification» WSEAS Transactions on Computers, vol. 7, no. 4, pp. 281–290, 2008.
8. S. M. Specht and R. B. Lee, «Distributed denial of service: taxonomies of attacks, tools, and countermeasures» in Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems (ISCA PDCS 04), pp. 543–550, San Francisco, Calif, USA, September 2004
9. The Slashdot Effect [Електронний ресурс]. – Режим доступу: https://web.archive.org/web/20081202171653/http://ssadler.phy.bnl.gov/adler/SDE/SlashDotEffect.html – Назва з екрану
10. P. Ferguson D. Senie, Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing, 2000.
11. Park, Kihong, H. Lee. «On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets.» Computer Communication Review. vol. 31. No. 4, pp. 15 – 26, 2001.
12. J. Li, J. Mirkovic, M. Wang, P. Reiher, L. Zhang. «SAVE: Source address validity enforcement protocol.» In proceeding of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), vol. 3, pp. 1557 – 1566, 2002.

13. J, Cheng, H, Wang, K. G. Shin. «Hop-count filtering: an effective defense against spoofed DDoS traffic.» In Proceedings of the 10th ACM conference on Computer and communications security, pp. 30 – 41, 2003.
14. Internet Denial of Service Attacks and Defense Mechanisms [Електронний ресурс]. – Режим доступу: http://www.cs.pitt.edu/~mehmud/docs/abliz11-TR-11-178.pdf – Назва з екрану
15. Slesarchik, Konstantin. (2018). Method for the Detection of Low Intensity Attacks Distributed Denial of Service with a Random Dynamics of Characteristics of Fragmentation and Frequency. Voprosy kiberbezopasnosti. 19-27. 10.21681/2311-3456-2018-1-19-27.
16. Нейронні мережі, сутність мереж [Електронний ресурс]. – Режим доступу: http://opticstoday.com/katalog-statej/stati-na-ukrainskom/nejrom erezhi/ nejronni-merezhi-sutnist-merezh.html
17. M. Alkasassbeh, G. Al-Naymat, A. B. Hassanat, M. Almseidin, «Detecting distributed denial of service attacks using data mining techniques» International Journal of Advanced Computer Science & Applications, vol. 1, no. 7, pp. 436–445, 2017
18. Li, J.; Liu, Y.; Gu, L. «DDoS attack detection based on neural network» 2nd International Symposium on Aware Computing (ISAC), 1-4 Nov. 2010, Tainan, pp. 196 – 199
19. Leu F.; Pai C. «Detecting DoS and DDoS Attacks Using Chi-Square», Fifth International Conference on Information Assurance and Security (IAS-09), 18-20 August 2009, Xian, pp.225-258
20. What is dataset [Електронний ресурс]. – Режим доступу: https://whatis.techtarget.com/definition/data-set – Назва з екрану
21. База даних KDD университету MIT [Електронний ресурс]. – Режим доступу: URL: http://sciencenet.org.ua/bdmit.html – Назва з екрану
22. Neural Network Toolbox - MATLAB - MathWorks [Електронний ресурс]. – Режим доступу: https://www.mathworks.com/products/neural-network.html – Назва з екрану
23. S. T. Zargar, J. Joshi, and D. Tipper, «A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks» IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013.
24. Algorithms to train a neural network [Електронний ресурс]. – Режим доступу:https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_ne ural_network
25. Akilandeswari, V.K.; Shalinie, S.M. «Probabilistic Neural Network based attack traf-fic classification» Fourth International Conference on Advanced Computing (ICoAC), 13-15 Dec. 2012, Chennai, pp. 1-8

Додаток А

Програмний код функції навченої нейромережі NeuralNetwork.m в Matlab

```
function [Y,Xf,Af] = NeuralNetwork(X,~,~)
%NEURALNETWORK neural network simulation function.
% Generated by Neural Network Toolbox function genFunction, 10-Jun-2018 00:12:37.
% [Y] = NeuralNetwork(X,~,~) takes these arguments:
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 9xQ matrix, input #1 at timestep ts.
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 5xQ matrix, output #1 at timestep ts.
% where Q is number of samples (or series) and TS is the number of timesteps
%#ok<*RPMT0>
% ===== NEURAL NETWORK CONSTANTS ====
% Input 1
x1_step1.xoffset = [1;11;0;37;0;0;0;0;0];
x1_step1.gain = [1;0.0224719101123596;0.00178890876565295;3.05352835201075e-
05;2;0.00391389432485323;0.00391389432485323;0.00784313725490196;0.00784313725490196];
x1_step1.ymin = -1
% Layer 1
b1 = [1.5267377350267564;1.7789301460875453;2.5808263673739891;-
0.74829139027217584;1.1838076835541644;-0.68817863692449066;-1.2440995924983418;-
1.7041803368119837;-0.60075347298080817;0.34502357858458171;-2.5730575926933064;-
0.16294308620804365;-
0.81118456034279085;2.9814279938889001;4.0505529839102534;2.025024064897226;1.16964513069
96245;-1.9236392254155896];
IW1_1 = [-2.2042102063015365 0.46371969575901972 1.2626971416187112 -0.2509525563315288 -
0.60571491574342418 -0.89426973668736764 -1.8192140735020734 -1.2512688035134323 -
0.1710689277463274;-5.0636828159671463 -0.70976302393544377 1.1180935740544278 -
1.0890197585753516 0.19986341450590073 -0.0086233032150645032 0.26664544817868174
0.1976233020344576 -0.8556790099130952;0.10340492233471837 0.15908214987216918 -
1.0026545231415647 -0.14081774297890154 -2.0593387760781194 0.72215312125767284
2.0175275508137966 -0.36257077596462428 2.6686706625050842;1.218177612327578 -
0.75617740744784268 0.64277498323561577 -0.55712914350505471 0.71388575691160128 -
0.9266714540712957 -0.4365978437311831 -1.016715085809079 -
0.19638729126582513;0.99724548945779568 -1.3976762279339265 -1.9459391151052272
0.16337568560408633 -1.2855655984523511 -0.99160135629879775 -0.60509377446493728 -
0.79672707079301597 1.3967338413696753;2.5164568130307958 2.2096534976757662 -
0.30520617273789563 0.44950530675404743 -1.0679274891062109 -3.0339265757609843 -
1.545284566744024 -1.131111323950583 0.42782630297164592;-0.15581419859619475 -
0.286666903661631 4.1448328761063227 1.3405342453428071 -0.30122684379605769
0.23149528034565078 -0.22941308611015715 -0.21963360772408 -
0.34044128810394469;0.96257716557953044 1.0243332713696629 1.1079462935417244
0.41128849041369087 1.3034749554758651 -2.6537116459843468 -1.7042329895707455 -
0.77818929541839288 -2.7795086728764811;1.4286611157059375 0.0052997771523616635 -
5.6635126593814142 0.82995331246652682 -1.0972637580524363 -1.0698196206714246
1.084046799895324 2.0235804079605728 2.2596015287244207;0.95808613862707759 -
0.82669898646574536 -1.2564523402395753 0.43572782270673283 -0.60403420429393917 -
0.42462172042778124 -0.85753603529048539 0.64470055255391412 -0.14408665977183474;-
1.4251288955795016 -1.2087058663686663 -0.87587626157026144 0.8858834783802737
3.0673658086357674 1.6615849021422948 -4.9970655724749697 -0.22438341507264892
0.28823981127583381;-0.32094885514915278 -1.1721346782659743 -4.145848736362419
1.6392459850228314 1.6684663719102331 1.9724754985889594 -1.4979326502654171
1.842812849836668 -1.7319491987355895;-5.698303266587927 4.5123717743101173
0.17231151695017677 -2.725781028224517 0.40773257303465882 -1.4032782994091257 -
0.25056176700834482 -1.595140009131863 1.3308826459909981;1.9187247155668323
0.33086216601427676 4.9625520032290105 -1.8345575009486579 -1.0471450777456952
2.103956769028775 -0.78514340876284738 0.55748970192399194 0.51851967268906818;-
1.4278075671794741 5.0676158399390046 -0.04654917148489441 -0.68786257653894345 -
1.9627850575139778 3.1446069857998697 0.46309361659085829 0.42927576330009315 -
1.2626086159685379;0.43080203154987851 -2.3805442998960542 -0.66141062089749358
0.43403403563203835 -1.0163030013295917 -0.26759582883304345 -0.16066430191917891
0.77312679984670196 -0.40627727091641236;1.7033747241388821 -1.0098271809353561
0.61275513178440899 0.57360595635081968 -0.36753114853821023 -0.3740791302152025
0.98623852097058451 0.2313387335404955 0.59802899992374836;-0.40637421424936121 -
0.5509395470027727 0.52323337347731547 1.0797930051368592 0.99403023537556745
0.68073998067454278 -0.093122916210772719 -0.58770171439039554 0.093150863197540984];
% Layer 2
```

```
b2 = [0.72210126805987762;-0.11920668728860465;-
0.82162032804191698;0.051646878240351746;-1.6759064950568963];
LW2_1 = [0.73478102243563748 -0.79542710412966278 -2.2022497614552918 0.62208718323577838
-0.6815000476292824 0.34952557746201174 -3.191028500894284 1.5355469708515603
3.2897256292214072 2.1153022653965001 -4.0268144263224777 -3.3129405313462472
5.1184767712724621 -0.98960495493051681 -4.2730880086893563 2.9102924801251064
0.95898198789094702 -1.6330963867642432;-0.42935027835219575 -0.26763119799980373
1.1009816453287162 -0.33227899514168024 1.0873118502275021 1.4943043828880906 -
0.10660868265452123 -0.6120507942476241 0.6083816220290893 -0.25891201068184994
4.2276554749530737 5.2633635623714659 1.1915425255961394 -3.9815162849946195 -
4.719415241210684 -1.1845582806193189 -0.8711526363012152 0.19495917124593071;-
2.8688855113728695 -3.8498322479924227 0.38312592418086799 -1.1608599431998565
0.88316671022233351 -1.9476094501707475 2.1121509813974413 -0.52562403434183436
1.8789458257915235 0.95515721590668012 -0.43179798097511141 0.79129224238687523 -
2.8603564825390988 0.43413930165924719 3.2704891729312746 -0.60657462393378581
0.38794726096253507 0.69929648004848677;0.62945920342217321 0.68030947561501365 -
0.18853983529282686 -0.18418723426577821 -1.1982353197157849 -3.0072470310982098 -
1.0943330812671104 -0.80237954472973116 -4.2609083539589587 0.25799216601059799
2.1806441282318714 -0.8737720637316938 3.5405462934542138 4.8393995697388545
0.639260073753529 -1.3120241596309519 -0.96883652652285335
0.48590921210834243;2.2838270198996242 3.8735756161099371 1.3701118672273047
0.48500030540499389 -0.9918197173144625 1.5347121519749032 2.8804222719541213 -
1.4130867643337184 -1.6287835300540736 -1.2079304579899359 -1.5921074950210534 -
0.18888451217698521 -5.1890053464829489 2.3742757937119405 4.625893007375117 -
0.70666543749070709 -0.035142885798226539 1.2838250192440632];
% ===== SIMULATION ========
% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
  X = {X};
end
% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
  Q = size(X{1},2); % samples/series
else
  Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);

    % Output 1
    Y{1,ts} = a2;
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
  Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
```

```
  y = bsxfun(@minus,x,settings.xoffset);
  y = bsxfun(@times,y,settings.gain);
  y = bsxfun(@plus,y,settings.ymin);
end

% Competitive Soft Transfer Function
function a = softmax_apply(n,~)
  if isa(n,'gpuArray')
    a = iSoftmaxApplyGPU(n);
  else
    a = iSoftmaxApplyCPU(n);
  end
end
function a = iSoftmaxApplyCPU(n)
  nmax = max(n,[],1);
  n = bsxfun(@minus,n,nmax);
  numerator = exp(n);
  denominator = sum(numerator,1);
  denominator(denominator == 0) = 1;
  a = bsxfun(@rdivide,numerator,denominator);
end
function a = iSoftmaxApplyGPU(n)
  nmax = max(n,[],1);
  numerator = arrayfun(@iSoftmaxApplyGPUHelper1,n,nmax);
  denominator = sum(numerator,1);
  a = arrayfun(@iSoftmaxApplyGPUHelper2,numerator,denominator);
end
function numerator = iSoftmaxApplyGPUHelper1(n,nmax)
  numerator = exp(n - nmax);
end
function a = iSoftmaxApplyGPUHelper2(numerator,denominator)
  if (denominator == 0)
    a = numerator;
  else
    a = numerator ./ denominator;
  end
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
  a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

ШИФР «DDOS-NND»

АНОТАЦІЯ

Запобігання мережевих атак – одна з найскладніших завдань в області захисту інформаційних систем. Більшість сучасних систем має розподілену структуру. Забезпечення працездатності таких систем залежить від здатності протистояти зловмисним діям, які спрямовані на порушення роботи як самої мережі, так і інформаційної системи, що функціонує в її рамках. Одним з найбільш небезпечних видів злочинної діяльності в мережі Інтернет є так звані мережеві атаки. Як свідчить статистика, наведена в Інтернет-джерелах, кількість мережевих атак продовжує зростати, методи, якими користуються злочинці, постійно розвиваються і удосконалюються, від одиночних спроб вони переходять до корпоративних розробок. У той же час сучасні системи виявлення вторгнень і атак ще не досконалі і недостатньо ефективні з точки зору безпеки рішень. Тому методи роботи в цьому напрямку необхідні і актуальні.

Метою роботи є підвищення точності виявлення DDoS-атак на основі нейромережевого підходу.

Для досягнення поставленої мети потрібно вирішити такі завдання:

- проаналізувати методи виявлення та захисту від DDoS-атак;
- виконати моделювання архітектури нейронної мережі;
- розробити програмний засіб;
- провести тестування розробленого засобу.

Об’єктом дослідження є процес виявлення DDoS-атак.

Предметом дослідження є методи та засоби виявлення DDoS-атак.

Методи дослідження. Для реалізації поставлених задач були використані методи теорії штучного інтелекту для проектування нейронної мережі; статистичні методи для підготовки вхідної та вихідної інформації для моделювання нейронної мережі, методи проектування програмного забезпечення для розробки та верифікації інтелектуальної системи.

Наукова новизна. Запропоновано архітектуру штучної нейронної мережі типу багатошарового персептрона, навчений екземпляр якої дозволяє вирішувати задачу виявлення деяких класів розподілених мережевих атак типу «відмова в обслуговування» на віддалений веб-ресурс, яка відрізняється сукупністю оптимально підібраних параметрів, що дозволяє підвищити точність виявлення до 99,8%.

Практична цінність. Розроблено програмний засіб для виявлення DDoS-атак на основі технологій штучних нейронних мереж, ефект від якої полягає в автоматизації підтримки прийняття рішень системного адміністратора, що базується на використанні інтелектуальних технологій, що дозволяє оперативно

та з високою точністю виявляти факти мережевого вторгнення, відповідно, вчасно застосовувати контрзаходи.

Наукова робота має таку структуру: розділів - 3, рисунків – 21, таблиць – 5, бібліографій – 25.

Ключові слова: кібербезпека, комп’ютерна мережа, DDoS-атака, нейрона мережа, багатошаровий персептрон.