Нейромережа

СТРУКТУРА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ ПОПЕРЕДЖЕННЯ ТА ВИЯВЛЕННЯ КІБЕРАТАК ТИПУ *USER TO ROOT* ТА *REMOTE TO LOCAL* НА ПІДПРИЄМСТВА КРИТИЧНОЇ ІНФРАСТРУКТУРИ

ЗМІСТ

ВСТУП

На сьогоднішній день комп’ютерні мережі відіграють важливу роль у повсякденному житті людини, адже вони використовуються майже у всіх сферах її діяльності. Яке б не було підприємство, комерційне, державне, муніципальне чи бюджетне, але у всіх випадках його функціонування так чи інакше забезпечується комп’ютерною мережею. Тож зрозуміло, що забезпечення безпеки є необхідним, оскільки наслідки недостатньої захищеності найрізноманітніші – крадіжка, знищення або поширення конфіденційної інформації (комерційної таємниці), персональних даних, підміна інформації, блокування доступу до неї, обмеження функціональності або повна зупинка діяльності комп’ютерної мережі. А останнє навіть призводить до фактичної зупинки бізнес-процесів, що може завдати великих збитків. Одним із способів вирішення такої проблеми є побудова інтелектуальних систем виявлення вторгнень. Але їх результативність можлива лише за наявності ефективного набору даних.

У цих умовах особливо важливими є питання оцінки ефективності систем безпеки значущих об'єктів критичної інфраструктури. Тому в ході проектування системи безпеки значущого об'єкта критичної інфраструктури з метою тестування рекомендовано її макетування або створення тестового середовища з використанням засобів та методів моделювання виявлення кібер-вторгнень різних класів та типів.

Аналіз сучасних моделей нейронних мереж для попередження та виявлення кібератак

В літературі існує ряд робіт, пов'язаних з виявленням вторгнень [1-8]. Деякі з цих робіт [1,6] стосуються загальної класифікації пакетів на нормальні або атакуючі категорії, в той час як інші стосуються виявлення конкретних категорій атак, таких як атаки типу *Remote to Local User* та *User to Root* [2, 4, 7, 8].

Sağıroğlu та ін. запропонували та розробили інтелектуальну систему виявлення вторгнень (*IDS*) [1]. Автори представили різні модефікації *IDS* на наборі даних *KDD'99* з використанням штучних нейронних мереж (*ANN*). Розроблені моделі *IDS* виконували завдання виявлення атак з високою точністю від 81.93% до 97.92%.

Kumar і Selvakumar представили алгоритм під назвою *NFBoost*, що використовує гібридні нейронечіткі системи для виявлення розподілених атак на відмову в обслуговуванні (*DDoS*) [2]. Автори використовували загальнодоступні набори даних, такі як *KDD*, *UCI* тощо, для навчання та тестування. В алгоритмі використовувались лише звичайні записи з'єднань та записи з'єднань під час атак на відмову в обслуговуванні (*DoS*) з набору даних *KDD*. Розроблена на основі алгоритму система досягла високої точності виявлення атак понад 98% як у навчанні, так і в тестуванні на наборі даних *KDD*.

Wu та Yen представили дослідження для порівняння таких оціночних показників, як точність, частота виявлення та частота хибних тривог на наборі даних *KDD'99*, але оскільки ці дані не є нормалізованими для порівняння в навчанні та тестуванні, то автори використовують різні розподіли даних [3]. Дослідження показало кращі результати, ніж *KDD* "Переможця", особливо в порівнянні з атаками "*User to Root*" (*U2R*) та "*R2L*".

Gupta та ін. запропонували багаторівневий метод виявлення вторгнень з використанням умовних випадкових полів (*CRF*) [4]. Метод використовує набір даних *KDD'99* і застосовує послідовне виявлення атак у багаторівневий спосіб.

Згідно з експериментальними результатами, запропонований метод, працює краще, ніж дерева рішень та наївний байєс. Він забезпечує значне покращення точності виявлення, особливо для атак *U2R* та *R2L*.

Kuang та ін. запропонували новий підхід, заснований на опорних векторах (*SVM*), який поєднує аналіз головних компонент ядра (*KPCA*) та генетичний алгоритм (*GA*) для виявлення вторгнень [5]. Модель реалізує багатошаровий *SVM*-класифікатор для прогнозування того, чи є дія атакою. В експериментах запропонований *SVM*-класифікатор досягає кращої точності та продуктивності при виявленні вторгнень.

Mukhopadhyay та ін. представили новий метод системи виявлення вторгнень (*IDS*) з використанням нейронної мережі зі зворотним поширенням (*BPN*) [6]. Автори проводили експерименти на наборі даних *KDD'99*, і метод показав 95,6% та 73,9% успішності для двох рівнів тестування відповідно.

Subbulakshmi та Afroze запропонували багаторівневий підхід до виявлення атак з використанням кореляційного вибору ознак (*CFS*) на наборі даних *KDD'99* [7]. Запропонований метод дозволив досягти коефіцієнтів класифікації для *R2L*-атак від 91% до 99%.

Sharma та Mukherjee в своїй роботі представили багаторівневий підхід, що поєднує методи наївного байєсівського класифікатора (*NBC*) та наївного байєсівського дерева (*NBTree*), зокрема, для підвищення точності та швидкості виявлення атак типу *U2R* та *R2L*, не знижуючи при цьому ефективність виявлення інших атак [8]. Автори використовували записи з'єднань з набору даних *KDD'99* для навчання і тестування, який містить 24 різних типи атак.

У роботі M. H. Kamarudin, C. Maple, T. Watson та H. Sofian була проведена розробка і аналіз алгоритму для виявлення атак на мережевий трафік з використанням пакетних заголовків[12]. Автори використали LbAD модель, яка виявилася дуже перспективною для використання в системах виявлення аномалій. Ця модель використовує статистичний аналіз значень полів пакетних заголовків на трьох рівнях OSI 7 шарів (рівень з'єднання даних, мережевий рівень, транспортний рівень). Був використаний алгоритм бінарної логістичної

регресії для виявлення кореляції між різними шарами пакетних заголовків. Це допомогло виявити, які саме шари є найбільш важливими для виявлення двох типів атак - R2L (remote-to-local) та U2R (user-to-root). Виявлено, що для атак типу R2L найбільш важливими є шари 3 та 4, тоді як для атак типу U2R - шари 2 та 4. Це дозволило авторам розробити модель, яка враховує лише найбільш важливі шари, що скорочує час обробки. Проведені експерименти показали високу ефективність запропонованого підходу. Виявлено, що модель досягла дуже високого рівня виявлення обох типів атак: 84% для R2L та 93.5% для U2R. Порівняно з існуючими алгоритмами, цей підхід показав значні покращення виявлення атак.

В роботі Shanmugavadivu експериментується система виявлення мережевих вторгнень з використанням нечіткої логіки[13]. Ця методика використовує набір нечітких правил, які отримані з чітких правил з використанням частих елементів. Точність класифікації цього підходу перевищує 90% для всіх типів атак.

P.Giffy Jeya, M Ravichandran та C.S. Ravichandran [14] представили новий метод класифікації аномалій, використовуючи кореляційний аналіз для зменшення кількості ознак та застосовуючи лінійний дискримінантний аналіз Фішера (FLDA) для виявлення даних про вторгнення. В експериментах використовувався набір даних KDD Cup'99, і результати експериментів показали, що запропонований метод покращив точність класифікації U2R і R2L атак порівняно з іншими дослідженнями.

У статті Ployphan Sornsuwit використовувався алгоритм Adaboost для створення ансамблю класифікаторів дерева рішень, наївного Байєса, SVM та MLP для виявлення U2R та R2L атак, які важко виявити[15]. Крім того, для зменшення надлишкових ознак використовується кореляційний метод. Ефективність ансамблевих класифікаторів оцінюється за допомогою наборів даних KDD CUP'99 в репозиторії даних UCI Repository, а результати порівнюються з одиночними класифікаторами. Експерименти проведено на наборі даних з усіма ознаками та деякими вибраними ознаками. Експерименти

показують, що ансамблеві класифікатори на основі Adaboost можуть покращити продуктивність усіх класифікаторів. Крім того, ансамбль Naïve Bayes та MLP мають найвищу чутливість, а ансамбль Naïve Bayes має найвищу специфічність, коли вони тестуються на даних з деякими вибраними ознаками. Однак дерево рішень показує найнижчі результати, оскільки обидва типи атак містять невеликий обсяг даних і мають тривалий період часу. Атаки U2R та R2L більш схожі на поведінку звичайних користувачів, а кількість даних дуже мала.

Debojit Boro, Bernard Nongpoh та Dhruba K. Bhattacharyya [16] представили комбінацію декількох моделей для покращення продуктивності систем виявлення вторгнень. Вони використовували чотири шари для об'єднання відповідних класифікаторів за допомогою декількох слабких учнів. Експериментальні результати показали, що запропонований ними метод дозволив покращити швидкість класифікації та частоту помилкових спрацьовувань порівняно з одним класифікатором.

Дослідження Beghdad аналізує продуктивність деякої нейронної мережі коли для навчання використовується весь набір даних KDD з метою класифікації та виявлення атак [18]. П'ять типів нейронних мереж, що досліджуються нейронних мереж, які досліджуються: Багатошарове сприйняття (MLP), самоорганізуюча карта ознак (SOM), нейронні мережі Джордана/Елмана (Jordan/Elman), рекурентні нейронні мережі (RNN) нейронні мережі Джордана/Елмана, рекурентні нейронні мережі та нейронної мережі RBF. Їх результати показали, що відсоток правильної класифікації (PCC) становить 99.16%, 98.28%, 98.36%, 98.44% та 79,23% відповідно.

У статті Kumar і Yadav запропоновано систему виявлення вторгнень на основі штучної нейронної мережі, яка використовує для навчання алгоритм градієнтного спуску з імпульсним поширенням[17]. Хоча для навчання обрано випадкові патерни, запропонована нейронна мережа протестована на повному "тестовому" наборі даних KDD cup 99. Результати показують, що точність запропонованої IDS на основі нейронної мережі для бінарної класифікації

(атака або нормальний стан) є високою, а рівень виявлення атак зондування, R2L та U2R є високим у порівнянні з іншими методами.

У роботі Mehmet Ali було розроблено та реалізовано систему виявлення R2L-атак на основі штучного інтелекту у вигляді бінарного класифікатора, який вирішує, чи є запис про з'єднання частиною R2L-атаки, чи ні. Система використовує набір даних KDD'99, зібраний для оцінки моделей IDS з точки зору частоти виявлення та частоти хибних спрацьовувань [19]. Модуль обробки даних системи видаляє надлишкові записи та налаштовує розподіл записів у навчальному наборі даних, який зазвичай є незбалансованим. Експериментальні результати показують, що MLPNN моделі з добре налаштованими параметрами можуть досягти дуже високого рівня виявлення R2L-атак вище %96 за рахунок усунення надлишкових записів та збалансування розподілу даних на навчальних наборах даних KDD'99.

Вибір структури нейронної мережі

При виборі структури нейронної мережі перевагу отримав такий клас нейронних мереж, як багатошарові мережі прямого поширення, які, зазвичай, називають багатошаровими персептронами і які є окремим випадком персептрона Розенблата.

Нейронні мережі прямого поширення мають вхідний сигнал, який передається від прошарку до прошарку (від одних нейронів до інших). Саме такою мережею і є багатошаровий персептрон, який складається з вхідного прошарку, прихованих обчислювальних прошарків всередині системи і вихідного прошарку нейронів. Багатошаровий персептрон – це односпрямована мережа сигмоїдального типу [10].

Багатошарові персептрони часто застосовуються для контрольованих завдань навчання: вони навчаються на безлічі пар вхідних та вихідних даних і моделюють кореляцію (або залежності) між цими входами і виходами. Навчання включає в себе налаштування параметрів або ваг і зсувів моделі з метою мінімізації помилок.

У нашому випадку вхідний прошарок складається з 41 нейрона (параметри мережевого з'єднання), три прихованих прошарки з експериментально підібраною кількістю нейронів та трьома вихідними прошарками які виявляють факт атаки типів U2R, *L2R* або нормальне мережеве з'єднання.

Запропонована структура представляє собою нейронну мережу з трьома прихованими прошарками та функцією активації *ReLU* (*Rectified Linear Unit* - функція активації, яка широко використовується в нейронних мережах. Для будь-якого вхідного значення *x*, якщо воно менше або дорівнює нулю, вихід буде нульовим. Якщо ж вхід більше нуля, вихід буде рівний вхідному значенню) на кожному з них, а також вихідним прошарком з активацією *Sigmoid*.

Розглянемо детальний опис структури нейромережі.

- *Вхідний прошарок*: Вхідний прошарок має розмірність *dataset_inputs*, що визначається кількістю функцій (ознак) у вхідних даних. У цьому випадку використовуються дані *train_X* та *test_X*.
- *Прихований прошарок №1*: Перший прихований прошарок має 46 нейронів. Активація цього прошарку виконується за допомогою функції активації *ReLU*.
- *Прихований прошарок №2*: Другий прихований прошарок має 32 нейрони. Активація також виконується за допомогою *ReLU*.
- *Прихований прошарок №3*: Третій прихований прошарок має 12 нейронів, активація - *ReLU*.
- *Вихідний прошарок*: Вихідний прошарок має розмірність *dataset_outputs*, який визначається кількістю класів (або кількістю вихідних значень). У цьому випадку використовується активація *Sigmoid*, яка призначена для задач класифікації бінарних даних.

Запропонована структура нейромережі навчається за допомогою оптимізатора *Adam* з коефіцієнтом навчання *lr*=0.00456789. Для оцінки втрат використовується середньоквадратична помилка (*MSE*). Навчання проводиться

протягом 500 циклів з пакетами розміру *batch_size*=300. На кожному циклі вимірюється середньоквадратична помилка на тестовому наборі. Якщо поточне значення *MSE* на тесті є кращим, ніж попереднє краще значення *MSE*, то модель оновлюється. У цьому випадку краща модель зберігається.

Вибір мови програмування

Для розробки програмного забезпечення використовувалась мова програмування *Python 3*.

Python – об'єктно-орієнтована мова програмування, яка набула надзвичайної популярності в сфері машинного навчання та штучного інтелекту завдяки великій системі бібліотек і фреймворків та можливості легкої інтеграції між наявними компонентами. Перевагами *Python* перед іншими мовами програмування для роботи зі штучними інтелектом і, зокрема, нейронними мережами можна вважати:

– широкий вибір бібліотек та фреймворків;

– кросплатформеність;

– адаптивність та гнучкість у стилях використання;

– простий та зрозумілий синтаксис;

– вбудовані засоби візуалізації даних

Використані бібліотеки

Для розробки програмного забезпечення використовувались такі бібліотеки мови програмування *Python* [11]:

- ***NumPy* (*numpy*)**: Це бібліотека для наукових обчислень, яка надає підтримку для масивів та матриць, разом із великою кількістю функцій для роботи з ними.
- ***Pandas* (*pandas*)**: Це бібліотека для обробки та аналізу даних, яка забезпечує структури даних, такі як *DataFrame*, що спрощують роботу з даними.

- ***Scikit-learn* (*sklearn*)**: Це бібліотека машинного навчання для *Python*, яка надає інструменти для класифікації, регресії, кластеризації та інших видів аналізу даних.
- ***Matplotlib* (*matplotlib.pyplot*)**: Це бібліотека для візуалізації даних в *Python*, яка дозволяє будувати різноманітні графіки, діаграми, гістограми тощо.
- ***LabelEncoder* (*sklearn.preprocessing*)**: Це клас для кодування міток (*labels*) у числові подання. Використовується для конвертації категоріальних даних у числові.
- ***PyTorch* (*torch*)**: *PyTorch* - це бібліотека для машинного навчання та обчислювальних графів, яка широко використовується для розробки та навчання нейронних мереж. Вона надає інструменти для створення, навчання та валідації нейронних мереж, а також для обробки даних.
- ***torch.nn* (*torch.nn*)**: Модуль *nn* в *PyTorch* містить різноманітні вбудовані функції та класи, які допомагають збудувати та навчити нейронні мережі.
- ***torch.optim* (*torch.optim*)**: Цей модуль містить різноманітні оптимізатори, які використовуються для навчання нейронних мереж в *PyTorch*. Оптимізатори визначають методи оптимізації, такі як стохастичний градієнтний спуск (*SGD*), *Adam*, *RMSprop* тощо.
- ***copy* (*copy*)**: Модуль *copy* в *Python* містить функції для копіювання об'єктів. Використовують його для створення глибоких або поверхневих копій об'єктів, щоб уникнути проблем з посиланнями.
- ***tqdm* (*tqdm*)**: Це бібліотека для створення прогрес-барів в *Python*, яка дозволяє вам візуалізувати прогрес виконання ітераційних процесів, таких як цикли або завантаження даних.
- ***train_test_split* (*sklearn.model_selection*)**: Цей модуль надає функцію для розділення набору даних на випадкові тренувальні та тестові підмножини. Використовується для оцінки точності та узагальнення моделі. Зазвичай дані розділяються на тренувальні дані, які використовуються для навчання моделі, і тестові дані, які використовуються для оцінки її продуктивності на невідомих даних.

Використання датасету *KDD99* для проведення експериментальних досліджень розробленого програмного забезпечення виявлення атак типів *U2R* та *R2L*

Для навчання нейронної мережі використовувалась база даних атак *KDD99*, яка містить стандартний набір даних, який включає в себе широкий спектр вторгнень.

Загалом така база містить близько 5000000 записів про мережеві з'єднання. Кожний запис представляє собою образ мережевого з'єднання та включає 41 параметр мережевого трафіка (табл. 1) і позначається як "атака" або "не атака" [9].

Таблиця 1 Параметри мережевого трафіка

№ з/п	Параметр	Опис
1	duration	Тривалість (у секундах) з'єднання
2	protocol_type	Тип протоколу (TCP, UDP, etc.)
3	service	Атакований сервіс
4	src_bytes	Кількість байтів від джерела до призначення
5	dst_bytes	Кількість байтів відповіді клієнту
6	flag	Прапорці з'єднання
7	land	1, якщо з'єднання від/до того самого хоста/порта
8	wrong_fragment	Кількість „хибних" фрагментів
9	urgent	Кількість термінових пакетів
10	hot	Кількість „гарячих" індикаторів
11	num_failed_logins	Кількість невдалих спроб реєстрації
12	logged_in	1, якщо успішний вхід в систему; 0 неуспішне
13	num_compromised	Кількість „компроментуючих" умов
14	root_shell	1, якщо root shell отриманий; інакше 0

15	su_attempted	1, якщо виконувалась „su root” ; інакше 0
16	num_root	Кількість „root” доступів
17	num_file_creations	Кількість операцій створення файлів
18	num_shells	Кількість запитів на надання оболонки
19	num_access_files	Кількість операцій на доступ до контролю файлів
20	num_outbound_cmds	Кількість вихідних команд для FTP сесії
21	is_hot_login	1, якщо логін належав до „гарячого” списку
22	is_guest_login	1, якщо „гостьовий” вхід
23	count	Кількість з'єднань на хост в поточній сесії за останні 2 с
24	serror_rate	% з'єднань що мали „SYN” помилки
25	rerror_rate	% з'єднань що мали „REJ” помилки
26	same_srv_rate	% з'єднань що мали однаковий сервіс
27	diff_srv_rate	% з'єднань на різні сервіси
28	srv_count	Кількість з'єднань на такий самий сервіс за останні 2 с
29	srv_serror_rate	% з'єднання з помилкою в „SYN” пакеті
30	srv_rerror_rate	% з'єднання, що мають „REJ” помилки
31	srv_diff_host_rate	% з'єднання від інших хостів
32	dst_host_count	Кількість з`єднань до локального хоста, встановлених віддаленою стороною
33	dst_host_srv_count	Кількість з`єднань до локального хоста, встановлених віддаленою стороною та використовуючих одну службу
34	dst_host_same_srv_rate	% з`єднань до локального хоста, встановлених віддаленою стороною та використовуючих одну службу
35	dst_host_diff_srv_rate	% з`єднань до локального хоста,

		встановлених віддаленою стороною та використовуючих різні служби
36	dst_host_same_src_port_rate	% з`єднань до даного хоста при поточному номері порту джерела
37	dst_host_srv_diff_host_rate	% з`єднань до служби різних хостів
38	dst_host_serror_rate	% з`єднань з помилкою типу SYN для даного хост-приймача
39	dst_host_srv_serror_rate	% з`єднань з помилкою типу SYN для даної служби приймача
40	dst_host_rerror_rate	% з`єднань з помилкою типу REJ для даного хост-приймача
41	dst_host_srv_rerror_rate	% з`єднань з помилкою типу REJ для даної служби приймача

В базі представлені 22 типи атак. При цьому атаки поділяються на 4 основні категорії: *DoS, U2R, R2L* і *Probe*. Серед цих атак найбільш цікавими для розробки програмного забезпечення стали атаки типу *U2R* і *R2L*.

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*.

Зовнішній вигляд бази *KDD99* – текстовий файл у якому у вигляді матриць представлено набір параметрів певного типу атаки або нормального з'єднання (рис. 1).

```
0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1
.00,0.00,0.11,0.00,0.00,0.00,0.00,0.00,normal.
0,tcp,ftp,SF,36,197,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,255,1,0.
00,0.05,0.00,0.00,0.39,0.00,0.05,0.00,warezmaster.
```

Рис. 1. Зовнішній вигляд бази *KDD99*

Результати експериментальних досліджень

Побудована нейронна мережа була навчена з отриманою найкращою середньоквадратичною помилкою *MSE*=0.003573242574930191 (рис. 3, 4).

Середньоквадратична помилка (MSE) обчислюється як квадрат відхилення між прогнозованими значеннями моделі та реальними значеннями цільової змінної, і потім осереднюється за всіма прикладами в тестовому наборі даних.

```
Epoch 499: 100%|██████████| 500/500 [00:36<00:00, 14.12it/s, Best MSE=0.003573242574930191]
Out[3]: <All keys matched successfully>
```

Рис. 3. Скрін-шот результату навчання

В роботі була застосована функція, яка обраховує статистику на тестовому наборі даних. Функція виконує проходження через кожну точку тестового набору, обчислює різницю між фактичними та передбаченими значеннями і виводить статистичні результати у вигляді списку.

Статистика щодо розподілу результатів виконання програми на тестовому наборі даних наступна:

- Загальна кількість даних (записів про нормальні мережеві з'єднання та атаки у датасеті): 11178
- Кількість даних у тренувальному наборі: 8383
- Кількість даних у тестовому наборі: 2795
- Кількість даних, в яких нейромережа впевнена у правильності передбачень (*confident*): 2778
- Кількість сумнівних даних (*doubtful*): 2
- Кількість неправильно передбачених даних (*wrong*): 15

```
MSE: 0.010719727963306619
RMSE: 0.10353611912422939
total: 11178, train: 8383, test: 2795, confident: 2778, doubtful: 2, wrong: 15
```

Рис. 4. Скрін-шот статистичних результатів

На рис. 5 представлені результати перевірки навчання запропонованої структури мережі.

```
Кількість виявлених атак кожного типу та нормальних даних:
normal    10062
r2l        1093
u2r          23
Name: count, dtype: int64
```

Рис.5. Результат перевірки навчання запропонованої структури мережі

З результатів перевірки роботи запропонованої структури нейронної мережі видно, що з 1126 атак типу *R2L* модель виявляє 1093, а з 52 атак типу *U2R* виявляє 23. На рис.6. представлено діаграму виявлених атак обох типів.

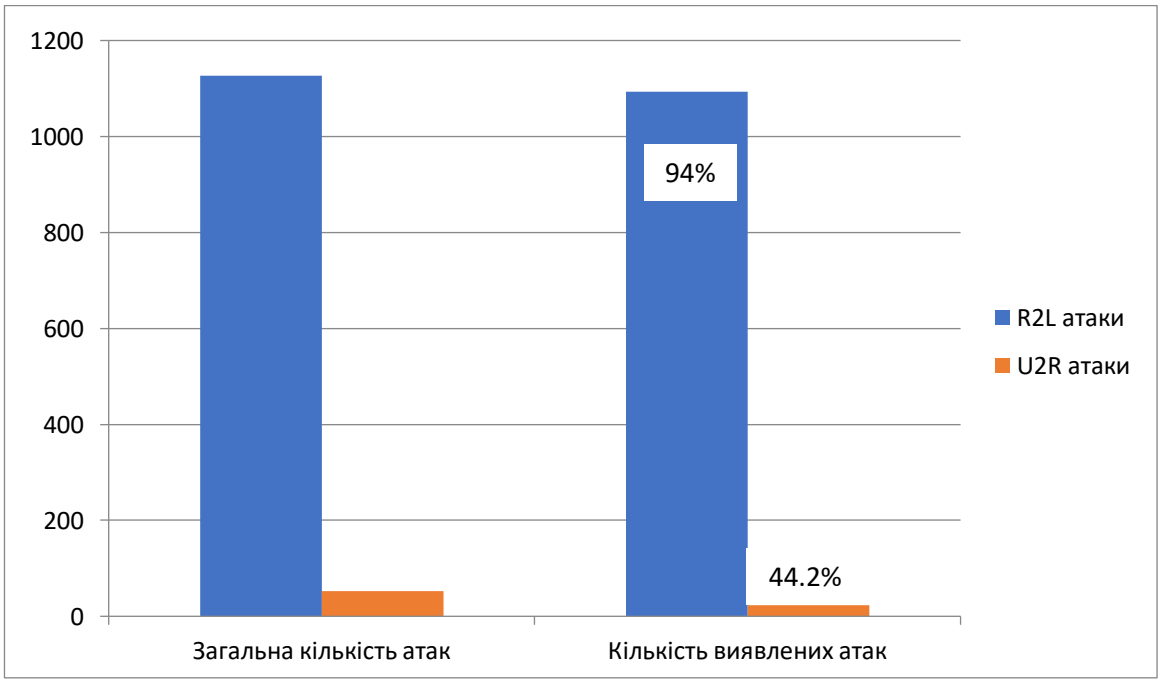

Рис.6. Діаграма виявлених кібератак

ВИСНОВКИ

Важливість систем попередження та виявлення атак на об'єкти критичної інфраструктури не можна недооцінювати в сучасному світі, де залежність від технологій і *Internet*-мереж зростає щодня. Критичні інфраструктури, такі як енергетика, транспорт, фінанси та інші сектори, є ключовими для ефективності існування суспільства і все частіше стають об'єктами кібератак. Це може мати серйозні наслідки для безпеки та економіки не тільки окремих об'єктів критичної інфраструктури, а й держави в цілому.

Системи виявлення вторгнень грають важливу роль у забезпеченні безпеки цих об'єктів, допомагаючи вчасно виявляти, аналізувати та реагувати на потенційні загрози. Вони дозволяють виявляти аномальні дії та спроби несанкціонованого доступу до систем, що допомагає уникнути потенційно серйозних наслідків, таких як втрата конфіденційної інформації, переривання роботи систем або навіть матеріальні збитки та загрози людським життям.

У цій роботі запропоновано використовувати структуру багатошарового персептрона для попередження та виявлення кібератак типів *R2L* і *U2R* на підприємства критичної інфраструктури, а також її реалізація мовою програмування *Python*.

В результаті розробки структури багатошарового персептрона та його навчання, розроблена модель нейронної мережі виявляє 94% атак типу *R2L* та 44.2% атак типу *U2R*. Менший відсоток виявлених кібератак типу *U2R* спричинений значно меншою кількістю цих атак в датасеті *KDD99*.

В цілому, отримані результати свідчать про ефективність запропонованої структури нейронної мережі для виявлення атак типу *R2L*, а також вказують на потенційні можливості подальшого вдосконалення запропонованої структури, зокрема, використовуючи збільшення обсягу даних для атак типу *U2R*.

ЛІТЕРАТУРА

1. Ş. Sağıroğlu, E.N. Yolaçan, and U. Yavanoğlu, “Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi”, J. Fac. Eng. Arch. Gazi Univ., vol. 26(2), pp. 325-340, 2011.

2. P.A.R. Kumar, S. Selvakumar, “Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems”, Computer Communications, vol. 36, pp. 303-319, 2013.

3. S.Y. Wu, E. Yen, “Data mining-based intrusion detectors”, Expert Systems with Applications, vol. 36, pp. 5605– 5612, 2009.

4. K.K. Gupta, B. Nath, R. Kotagiri, “Layered Approach Using Conditional Random Fields for Intrusion Detection”, IEEE Transactions On Dependable And Secure Computing, vol. 7(1), pp. 35-49, 2010.

5. F. Kuang, W. Xu, S. Zhang, “A novel hybrid KPCA and SVM with GA model for intrusion detection”, Applied Soft Computing, vol. 18, pp. 178–184, 2014.

6. I. Mukhopadhyay, M. Chakraborty, S. Chakrabarti, T. Chatterjee, “Back Propagation Neural Network Approach to Intrusion Detection System”, 2011 International Conference on Recent Trends in Information Systems, Kolkata, pp. 303-308, December 2011.

7. T. Subbulakshmi, A.F. Afroze, “Multiple Learning based Classifiers using Layered Approach and Feature Selection for Attack Detection”, 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN 2013), Tirunelveli, pp. 308- 314, March 2013.

8. N. Sharma, S. Mukherjee, “A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS”, Procedia Technology, vol. 6, pp. 913–921, 2012.

9. KDD Cup 1999 Data [Електронний ресурс] – Режим доступу до ресурсу: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

10. Swingler K. Lecture 4: Multi-Layer Perceptrons URL: http://www.cs.stir.ac.uk/courses/ ITNP4B/lectures/kms/4-MLP.pdf

11. Best Python Libraries for Machine Learning and Deep Learning. URL: https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deeplearning-b0bd40c7e8c

12. M. H. Kamarudin, C. Maple, T. Watson and H. Sofian, "Packet Header Intrusion Detection with Binary Logistic Regression Approach in Detecting R2L and U2R Attacks," 2015 Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec), Jakarta, Indonesia, 2015, pp. 101-106, doi: 10.1109/CyberSec.2015.28.

13. R. Shanmugavadivu and Dr. N. Nagarajan, " Network intrusion detection system using fuzzy logic", in Indian Journal of Computer Science and Engineering, Vol. 2, pp. 101-111.

14. P. G. Jeya, M. Ravichandran, and C. S. Ravichandran, "Efficient Classifier for R 2 L and U 2 R Attacks," International Journal of Computer Applications, vol. 45, no. 21, 2012.

15. P. Sornsuwit and S. Jaiyen, "Intrusion detection model based on ensemble learning for U2R and R2L attacks," 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand, 2015, pp. 354-359, doi: 10.1109/ICITEED.2015.7408971.

16. B. Debojit, N. Bernard, and K. B. Dhruba, "Anomaly based intrusion detection using meta ensemble classifier," in Proceedings of the Fifth International Conference on Security of Information and Networks, Jaipur, India, 2012.

17. S. Kumar and A. Yadav, "Increasing performance Of intrusion detection system using neural network," *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, Ramanathapuram, India, 2014, pp. 546-550, doi: 10.1109/ICACCCT.2014.7019145.

18. Beghdad. R, "Training all the KDD dataset to classify and detect attacks" in International Journal on Neural and Mass – Parallel computing and Information Systems, Vol. 17, March 2007.

19. ATICI, Mehmet Ali, Ibrahim Alper DOGRU, and Seref SAGIROGLU. "Detecting Remote to Local User Attacks with High Ratio."