УДК 004.056.53:[004.7:004.032.26] DOI https://doi.org/10.35546/kntu2078-4481.2024.3.33

В. М. ПАХОМОВА
кандидат технічних наук, доцент,
доцент кафедри електронних обчислювальних машин
Український державний університет науки і технологій
ORCID: 0000-0002-0022-099X

О. В. ГАЛУШКА
аспірант
Український державний університет науки і технологій
ORCID: 0009-0005-3447-2676

ДОСЛІДЖЕННЯ ДВОРІВНЕВОГО ВИЯВЛЕННЯ PROBE АТАК ЗАСОБАМИ НЕЙРОННИХ МЕРЕЖ

У даній роботі проведено дослідження дворівневого виявлення мережевих атак категорії Probe засобами нейронних мереж.

*Запропоновано використання багатошарового перцептрону конфігурації 31-1-124-5, де 31 – кількість вхідних нейронів; 1 – кількість прихованих шарів; 124 – кількість прихованих нейронів; 5 – кількість результуючих нейронів для виявлення мережевої категорії атаки DoS, U2R, R2L та Probe (на першому рівні) та самоорганізуючої карти Кохонена 10*10 для виявлення мережевих класів атак відповідно до категорії Probe: Ipsweep; Nmap; Portsweep; Satan (на другому рівні). Для виявлення мережевих атак категорії Probe створено з використанням мови Python та бібліотеки PyTorch програмну модель «MLP1-SOM2_Probe», що заснована на реалізації запропонованих конфігурацій багатошарового перцептрону та самоорганізуючої карти Кохонена. Для організації досліджень використані дані із KDDCup99, що пройшли відповідну обробку на підготовчому етапі: очищення даних; вибір ознак; мапінг категоріальних ознак; масштабування та нормалізація; розбиття даних на відповідні вибірки (навчальна, тесту вальна та валідаційна). На створеній моделі «MLP1-SOM2_Probe» визначені оптимальні параметри відповідних нейронних мереж: функція активації, оптимізатор і швидкість навчання для MLP1; ступінь впливу нейрона на сусідні нейрони та швидкість навчання для SOM2. Проведено оцінювання параметрів якості дворівневого виявлення Probe атак на створеній моделі «MLP1-SOM2_Probe». Визначено, що дворівневе виявлення атак на моделі «MLP1-SOM2_Probe» склало в середньому приблизно 98,8 %, що дозволяє досягти більш високої точності в зрівнянні з дворівневим виявленням атак на основі використання моделі «MLP1-MLP2_Probe».*

Ключові слова: *атака, Probe, дворівневе виявлення, категорія, клас, багатошаровий перцептрон, самоорганізуюча карта, вибірка, точність.*

V. M. PAKHOMOVA
Candidate of Technical Sciences, Associate Professor,
Associate Professor at the Department of Electronic Computers
Ukrainian State University of Science and Technology
ORCID: 0000-0002-0022-099X

O. V. HALUSHKA
Postgraduate Student
Ukrainian State University of Science and Technology
ORCID: 0009-0005-3447-2676

TWO-LEVEL DETECTION OF PROBE ATTACKS BY MEANS OF NEURAL NETWORKS

In this paper, a study of two-level detection is carried out network attacks of the Probe category by means of neural networks.

*The use of a multilayer perceptron is proposed configurations 31-1-124-5, where 31 is the number of input neurons; 1 – quantity hidden layers; 124 – the number of hidden neurons; 5 – quantity resulting neurons to detect the network attack category DoS, U2R, R2L and Probe (on the first level) and a self-organizing Kohonen map 10*10 to detect network attack classes according to the Probe category: Ipsweep; Nmap; Portsweep; Satan (on the second level). To detect network attacks categories Probe created using the Python language and the PyTorch «MLP1-SOM2_Probe» software model based on the implementation of proposed multilayer perceptron configurations and of the self-organizing map of Kohonen. For the organization of research, data from KDDCup99 that has been properly processed at the preparatory stage: data cleansing; selection of features; mapping of categorical features; scaling and normalization; splitting the*

data into appropriate samples (training, testing and validation). On the created model «MLP1-SOM2_Probe» defined optimal parameters of the corresponding neural networks: activation function, optimizer and learning speed for MLP1; the degree of influence of the neuron on nearby neurons and learning rate for SOM2. Evaluation conducted quality parameters of two-level detection of Probe attacks on the created model «MLP1-SOM2_Probe». It is determined that two-level detection of attacks on models «MLP1-SOM2_Probe» averaged about 98.8%, which allows achieve higher accuracy compared to two-level attack detection based on the use of the «MLP1-MLP2_Probe» model.

Key words: *attack, Probe, two-level detection, category, class, multilayer perceptron, self-organizing map, sampling, accuracy.*

Постановка проблеми

Наявність і постійний ріст загроз мережевих атак у режимі реального часу створюють необхідність розробки ефективної системи виявлення таких атак. Існуючі методи не завжди здатні виявити нові, раніше невідомі атаки, що створює ризик для безпеки комп'ютерних мереж. Перспективним напрямком у створенні систем виявлення мережевих атак, які повинні ґрунтуватися на адаптивних алгоритмах здатних до самонавчання, є застосування нейронних мереж, а іноді навіть комплексу нейронних мереж.

Аналіз останніх досліджень і публікацій

На сучасному етапі для виявлення мережевих атак найчастіше використовуються нейронні мережі (НМ): багатошаровий персептрон (Multi Layer Perceptron, MLP); радіально-базисна мережа (Radial Basis Function Network, RBF), самоорганізуюча карта Кохонена (Self Organizing Map, SOM), а також нейронечітка мережа (Adaptive Network Based Fuzzy Inference System, ANFIS). Ефективність машинного навчання НМ залежить від правильно підготовлених даних, що потребує виконання відповідного аналізу бази даних (KDDCup99, NSL-KDD та UNSW-NB15). Відомо, що різні НМ можуть неоднаково виявляти різноманітні мережеві атаки таких категорій як: DoS, U2R, R2L та Probe. З одного боку, для виявлення мережевих атак категорії Probe авторами були використані ANFIS [1], SOM [2], MLP [3]. З іншого боку, авторами досліджувалась можливість дворівневого виявлення мережевих атак на комплексі нейронних мереж [4], де на першому рівні виявляється мережева категорія, а на другому рівні виявляється мережевий клас відповідно до визначеної категорії, але разом з тим важливим недоліком таких методик є відсутність універсальності їх застосування.

Формулювання мети дослідження

Проведені дослідження ставили за мету подальший розвиток методики дворівневого виявлення мережевих атак категорії Probe. Для досягнення поставленої мети вирішувалися наступні задачі: виявлення мережевих категорій атак засобами багатошарового перцептрону на першому рівні; виявлення мережевих класів атак відповідно до категорії засобами самоорганізуючої карти Кохонена на другому рівні; при виконанні машинного навчанні визначення оптимальних параметрів відповідних нейронних мереж, що забезпечить достатньо високий рівень достовірності виявлення вторгнень в комп'ютерну мережу; оцінювання параметрів якості виявлення мережевих атак категорії Probe на створеній програмній моделі.

Викладення основного матеріалу дослідження

Категорія Probe мережевих атак полягає у скануванні портів з метою отримання конфіденційної інформації. До категорії Probe надходять наступні мережеві класи атак [5]: Ipsweep – сканування IP-адрес в мережі для виявлення активних хостів; Nmap – для сканування портів та виявлення відкритих сервісів; Portsweep – сканування портів на одному або декількох хостах для виявлення відкритих сервісів; Satan – для автоматичного сканування мережі та виявлення вразливостей.

Аналіз бази KDDCup99 та підготовка даних. Ефективність машинного навчання НМ залежить від правильно підготовлених даних та включає наступні етапи: 1) очищення даних (видалення дублікатів, заповнення пропущених значень за допомогою середніх значень або за іншими методами); 2) вибір ознак (позбутися неінформативних та сильно корелюючих ознак, що в свою чергу допоможе спростити нейронну модель); 3) мапінг категоріальних ознак (всі категоріальні ознаки повинні бути трансформовані в числа в той чи інший спосіб) [6]; 4) масштабування та нормалізація (всі ознаки приведені до інтервалу [0; 1] з використанням MinMaxScaler [7] та Scikit-Learn [8]); крім того, забезпечено баланс між наявними прикладами для кожного з мережевих класів); 5) формування вибірок (навчальної тесту вальної та валідаційної), при цьому важливо підібрати правильне співвідношення розміру між ними. Таким чином, з початкового набору в 41 ознаку (параметри мережевого трафіку) [5] видалено 2 ознаки за однаковими значеннями для всіх записів та 8 сильно корелюючих ознак; у результаті маємо вхідні вектори з 31 ознакою.

Багатошаровий перцептрон та визначення оптимальних параметрів. Відповідно до теореми Арнольда-Колмогорова та Хехт-Нільсена кількість прихованих нейронів склала $124 \le K_{пр} \le 691$; для дослідження взято MLP1 конфігурації 31-1-124-5 (рис. 1), де 31 – кількість вхідних нейронів (параметри мережевого трафіку [5]); 1 – прихований шар; 124 – кількість прихованих нейронів; 5 – кількість результуючих нейронів, що на першому рівні визначають нормальний стан (відсутність атаки, Normal) або наявність мережевої атаки категорії DoS, Probe, R2L, U2R.

Відповідні вибірки вміщають в собі дані, що представлені всіма наявними категоріями атак та типами атак. Розмірність вхідного вектору – 31, розмірність вихідного вектору – 5, кількість векторів в патчі – 64, кількість навчальних записів – 8099, кількість записів для тестування – 4049, кількість записів для валідації – 1351.

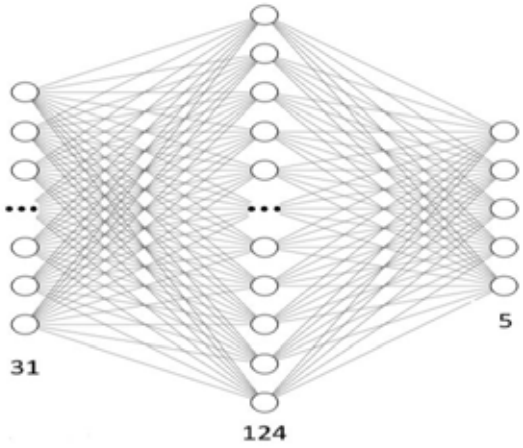

Рис. 1. MLP1 конфігурації 31-1-124-5

Для програмної реалізації НМ прийнято рішення щодо використання мови Python та бібліотеки PyTorch з широким інструментарієм по їх створенню та дослідженню. Загальна структура створеної програмної моделі «MLP1-SOM2_Probe» представлена на рис. 2.

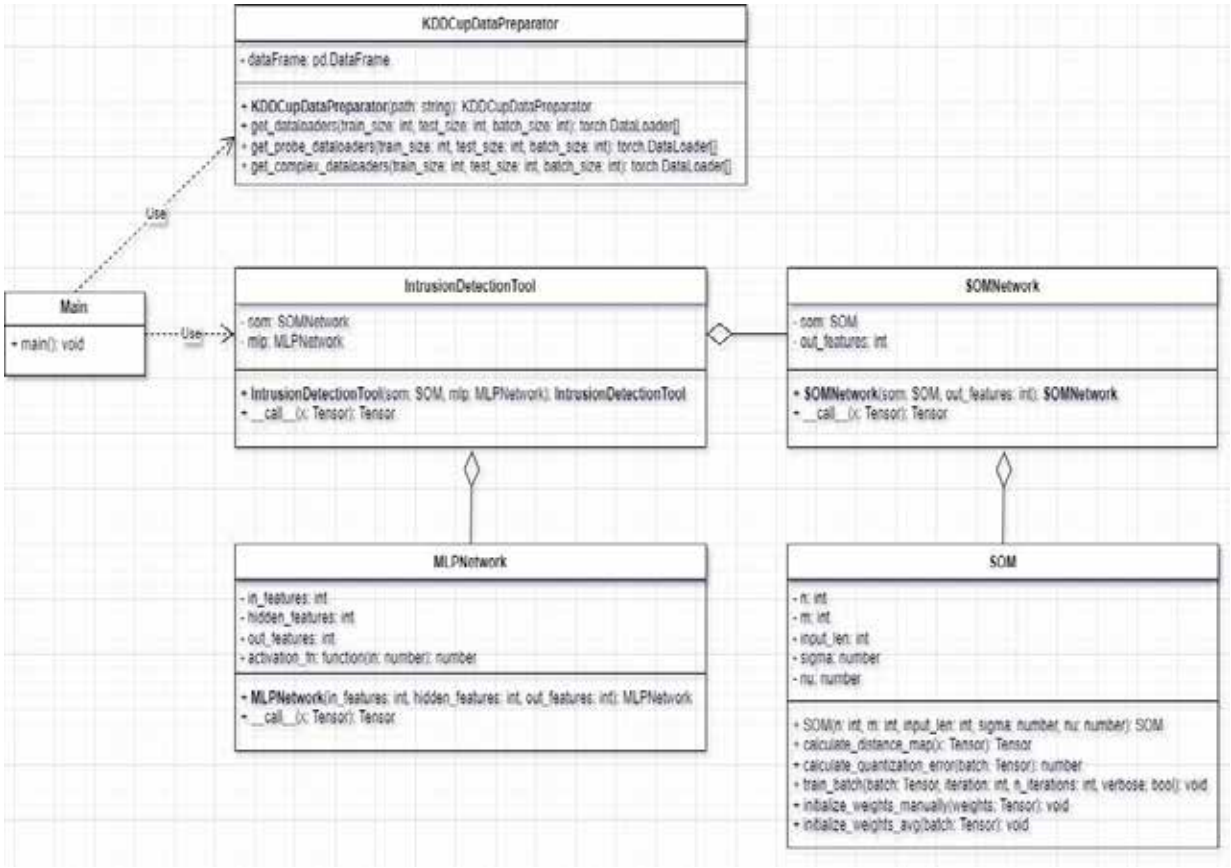

Рис. 2. Структура створеної програмної моделі «MLP1-SOM2_Probe»

Для MLP1 конфігурації 31-1-124-5 на створеній програмній моделі проведені дослідження Loss: за різними функціями активації та оптимізаторами навчання (рис. 3).

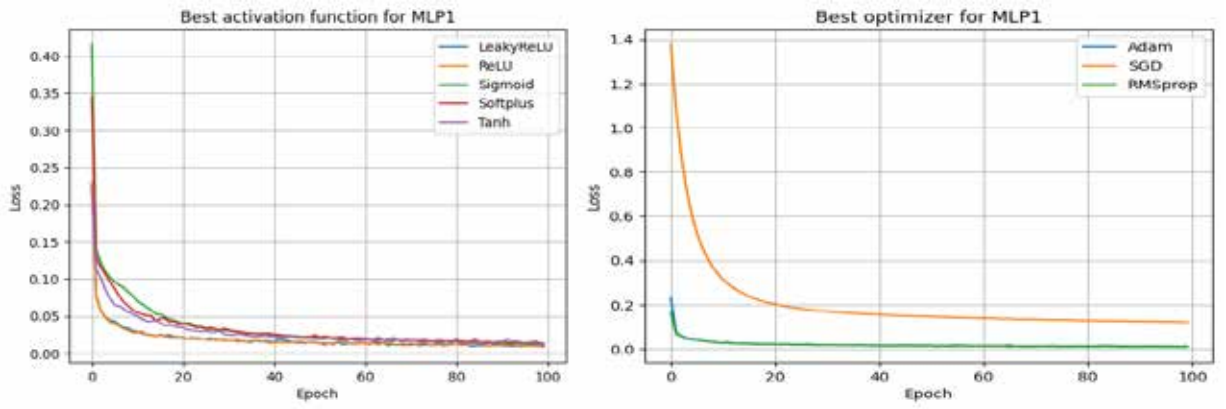

Рис. 3. Дослідження MLP1 конфігурації 31-1-124-5

Результати дослідження MLP1 конфігурації 31-1-124-5 при визначених оптимальних параметрах (функція активації – LeakyReLU; оптимізатор – Adam; швидкість навчання – 0,01) зведені до табл. 1.

Таблиця 1

Результати дослідження MLP1 конфігурації 31-1-124-5

Normal	DoS	Probe	R2L	U2R
0,99	1,00	0,99	0,98	0,53

Самоорганізуюча карта Кохонена та визначення оптимальних параметрів. На другому рівні виявляємо мережевий клас атаки (ipsweep, nmap, portsweep, satan) відповідно до категорії Probe з використанням SOM2(10*10), що представлений на рис. 4.

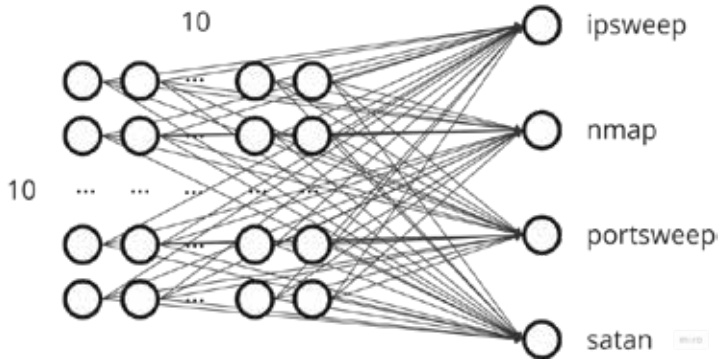

Рис. 4. Архітектура SOM2(10*10) для виявлення Probe атак

Процес навчання SOM2(10*10) починається з обчислення кращого нейрона на карті на основі мінімального значення дистанції між вхідним вектором та ваговим вектором кожного нейрона (під дистанцією розуміється Евклідова відстань) [9]:

$$d_j(x) = \sum_{i=1}^{D} \left(x_i - w_{ij}\right)^2, \tag{1}$$

де D – розмірність вхідного вектору; x_i – i-й компонент вхідного вектора ($i = 1,...,D$); w_{ij} – i-й компонент вагового вектора j-го нейрона ($j = 1,...,N$), N – загальна кількість нейронів.

Після вибору кращого нейрона обчислюється вплив цього нейрона на нейрони, що знаходяться поруч з ним, за наступною формулою:

$$T_{i,I(x)} = exp\left(\frac{-S_{j,I(x)}^2}{2\sigma^2}\right), \tag{2}$$

де $I(x)$ – індекс нейрона-переможця; $S_{j,i(x)}$ – відстань на карті до кращого нейрона; σ – кількість сусідів.

Далі формується матриця зміни вагових векторів, яка потім додається до матриці ваг всіх нейронів, що змінює вектори ваг для всіх нейронів. Кожен компонент обчислюється за формулою:

$$\Delta w_{ij} = \eta(t) T_{j,I(x)}(t), \tag{3}$$

де t – кількість епох; η – гіперпараметр, що визначає швидкість навчання.

Для оцінювання якості кластеризації даних самоорганізуючою картою використана похибка квантизації (метрика, що показує наскільки добре розтягнута нейронна решітка на множині навчальних прикладів), що розраховується за формулою на основі BMU (Best Matching Unit) [10]:

$$Q = \frac{1}{N}\sum_{i=1}^{N} x_i - w_{BM\cup(i)}, \tag{4}$$

де N – кількість вхідних векторів даних для розрахунку похибки; x_i – i-й вхідний вектор; $w_{BM\cup(i)}$ – ваговий вектор найближчого нейрона до x_i.

Вибірки для SOM2(10*10) вміщають в собі дані, що відносяться до категорії Probe. Розмірність вхідного вектору – 31, розмірність вихідного вектору – 4, кількість векторів в патчі – 64, кількість навчальних записів – 2464, кількість записів для тестування – 1232, кількість записів для валідації – 411. Результати дослідження, що отримані на SOM2(10*10), зведено до табл. 2.

Таблиця 2

Помилка квантизації SOM2(10*10)

σ	$\eta = 10^{-1}$	$\eta = 10^{-2}$	$\eta = 10^{-3}$
1	0,0993	0,1664	0,2256
3	0,1340	0,1367	0,1789
5	0,1275	0,1513	0,1744
7	0,1445	0,1479	0,1758
9	0,1711	0,1752	0,1584

Для дослідження обрано SOM2(10*10) з параметрами $\sigma = 5; \eta = 10^{-1}$. Візуалізацію карт активацій показано на рис. 5.

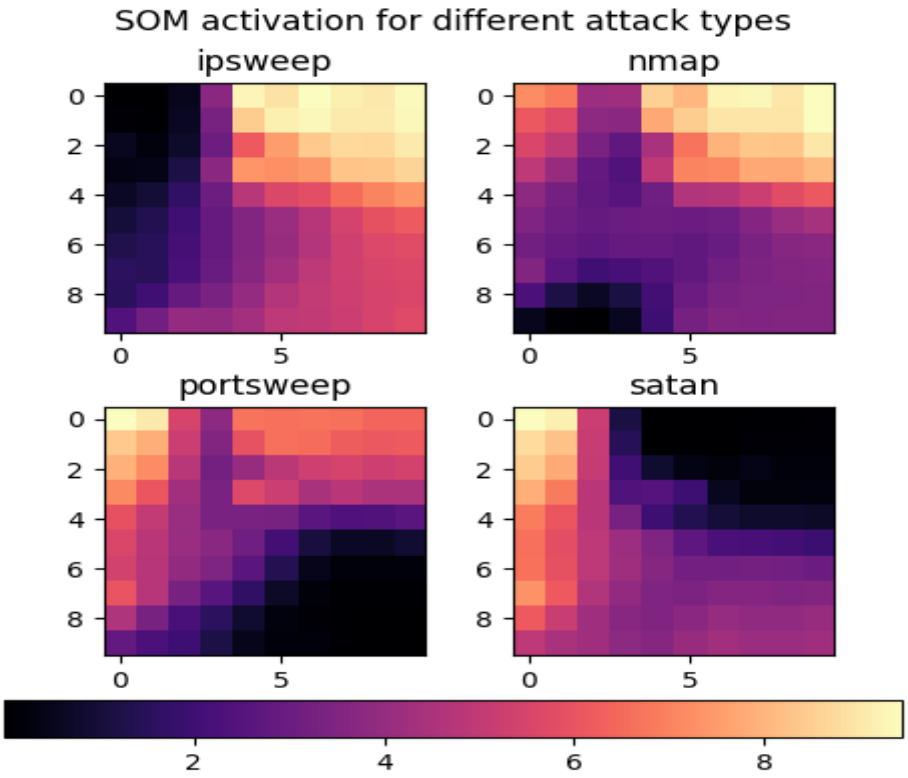

Рис. 5. Візуалізація карт активацій SOM2 (10*10)

Оцінка якості виявлення Probe атак на моделі «MLP1-SOM2_Probe». Отримані результати щодо виявлення Probe атак на моделі «MLP1-SOM2_Probe» зведені до табл. 3, де N – кількість вхідних векторів, $N_{кор}$ – кількість правильно класифікованих векторів, ε – загальна точність моделі, N_α – кількість помилок першого роду, α – доля помилок першого роду, N_β – кількість помилок другого роду, β – доля помилок другого роду.

Таблиця 3

Параметри якості виявлення Probe атак на моделі «MLP1-SOM2_Probe»

Запуски	**N**	$N_{кор}$	ε	N_α	α	N_β	β
1	4049	4012	0,99086	18	0,00444	7	0,00172
2	4049	3990	0,98542	26	0,00642	10	0,00246
3	4049	4007	0,98962	14	0,00345	10	0,00246
4	4049	3993	0,98616	28	0,00691	8	0,00197
5	4049	4006	0,98938	23	0,00568	8	0,00197

Із таблиці видно, що дворівневе виявлення Probe атак дозволяє досягти більш високої точності моделі «MLP1-SOM2_Probe», та склало в середньому приблизно 98,80 % (в зрівнянні з 97,35 %, що отримано в [4], при використанні «MLP1-MLP2_Probe»); при цьому помилки першого та другого роду склали в середньому приблизно 0,54 % та 0,21 % відповідно.

Висновки

Для виявлення Probe атак запропоновано дворівневе виявлення атак, що дозволяє виявляти мережеву категорію засобами MLP1 конфігурації 31-1-124-5 (на першому рівні) та мережевий клас атаки засобами SOM2(10*10) відповідно до категорії (на другому рівні). Для досліджень використані дані із KDDCup99, що пройшли відповідну обробку. З використанням Python та PyTorch створено модель «MLP1-SOM2_Probe», на основі якої визначені оптимальні параметри MLP1 (на першому рівні) і SOM2 (на другому рівні). Проведено оцінювання параметрів якості виявлення Probe атак на створеній програмній моделі «MLP1-SOM2_Probe».

Список використаної літератури

1. Пахомова В. М., Маслак А. В. Визначення атак категорії Probe з використанням бази даних KDDCup99 та нейронечіткої технології. *Вчені записки Таврійського національного університету імені В.І. Вернадського*. *Серія: Технічні науки*. Том 33(72). № 5, 2022. С. 135-140. DOI: https://doi.org/10.32872/2663-5941/2022.5/19.

2. Пахомова В. М., Павленко І. І. Дослідження параметрів якості визначення мережевих атак категорії PROBE з використанням самоорганізуючої карти. *SworldJournal*. 2022. Issue 11. Part 1. pp. 100-104. DOI: 10.30888/2663-5712.2022-11-01-022.

3. Пахомова В. М., Квочка М. Ю. Визначення атак категорії Probe засобами багатошарової нейронної мережі. *Вчені записки Таврійського національного університету імені В.І. Вернадського*. *Серія: Технічні науки*. Том 34(73). № 4, 2023. С. 93-98. https://doi.org/10.32787/2663-5941/2023.4/15.

4. Zhukovyts'kyy I. V., Pakhomova V. M., Ostapets D. O., Tsyhanok O. I. Detection of attacks on a computer network based on the use of neural networks complex. *Science and Progress of Transport.* 2020, no. 5(89), pp. 68–79. doi: https://doi.org/10.15802/stp2020/218318.

5. KDD Cup 1999 Data. *Intrusion detection dataset*. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

6. Géron Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow concepts, tools, and techniques to build intelligent systems. *O'Reilly Media*, *Inc.* 2019. 856 p.

7. Daython M. Scaling your data using scikit-learn scalers. *Medium*, 2020. Retrieved from https://medium.com/@daython3/scaling-your-data-using-scikit-learn-scalers-3d4b584107d7.

8. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V. Scikit-learn developers. Scikit-learn: machine learning in Python. *The Journal of Machine Learning Research*. 2011. Vol. 12. p.p. 2825-2830. Retrieved from https://scikit-learn.org.

9. Alves Gisely. Discovering SOM, an unsupervised neural network. *Medium*, Dec 27, 2018. Retrieved from https://medium.com/neuronio/discovering-som-an-unsupervised-neural-network-12e787f38f9.

10. Vesanto J., Alhoniemi E. Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, 2000. Vol. 11. Iss. 3. p.p. 586-600. DOI: 10.1109/72.846731.

References

1. Pakhomova V. M., Maslak A. V. (2022). Network attack detection using KDDCup99 database and neuron fuzzy technology. *Vceni zapiski tavriysky natsionalnogo university imeni V.I. Vernadskogo*. *Seria: technical nauki*, Vol. 33(72), No. 5, pp. 135-140, DOI: https://doi.org/10.32872/2663-5941/2022.5/19 [in Ukrainian].

2. Pakhomova V. M., Pavlenko I. I. (2022) Research of parameters of quality of definition of network attacks of the PROBE category with use of the Self organizing Map. *SworldJournal*, Vol. 11, Iss. 1, pp. 100-104, DOI: 10.30888/2663-5712.2022-11-01-022 [in Ukrainian].

3. Pakhomova V., Kvochka M. (2023). Definition of network attacks of PROBE category by means of multilayer neural network. *Vceni zapiski tavriysky natsionalnogo university imeni V.I. Vernadskogo*. *Seria: technical nauki*, Vol. 34(73), No, 4. pp. 93-98 [in Ukrainian].

4. Zhukovyts'kyy I. V., Pakhomova V. M., Ostapets D. O., Tsyhanok O. I. (2020). Detection of attacks on a computer network based on the use of neural networks complex. *Science and Progress of Transport,* No. 5(89), pp. 68–79, DOI: https://doi.org/10.15802/stp2020/218318 [in English].

5. KDD Cup 1999 Data. *Intrusion detection dataset*. Retrieved from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [in English].

6. Géron Aurélien (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow concepts, tools, and techniques to build intelligent systems. *O'Reilly Media, Inc,* 856 p. [in English].

7. Daython M. (2020). Scaling your data using scikit-learn scalers. *Medium*. Retrieved from https://medium.com/@daython3/scaling-your-data-using-scikit-learn-scalers-3d4b584107d7 [in English].

8. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V. (2011). Scikit-learn developers. Scikit-learn: machine learning in Python. *The Journal of Machine Learning Research*, Vol. 12, p.p. 2825-2830. Retrieved from https://scikit-learn.org [in English].

9. Alves Gisely. (2018). Discovering SOM, an unsupervised neural network. *Medium*. Retrieved from https://medium.com/neuronio/discovering-som-an-unsupervised-neural-network-12e787f38f9 [in English].

10. Vesanto J., Alhoniemi E. (2000). Clustering of the Self-Organizing Map. *IEEE Transactions on Neural Networks*, Vol. 11, Iss. 3, pp. 586-600, DOI: 10.1109/72.846731 [in English].