Novel Approach for Intrusion Detection Using Simulated Annealing Algorithm Combined with Hopfield Neural Network

Atef Ahmed Obeidat[1]

[1]Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Salt, Jordan

Abstract: With the continued increase in Internet usage, the risk of encountering online threats remains high. This study proposes a new approach for intrusion detection to produce better outcomes than similar approaches with high accuracy rates. The proposed approach uses Simulated Annealing algorithms [1] combined with Hopfield Neural network [2] for supervised learning to improve performance by increasing the correctness of true detection and reducing the error rates as a result of false detection. The proposed approach is evaluated on an intrusion detection data set called KDD99[3]. Experimental tests demonstrate the potential of the proposed approach to rapidly detect high precision and efficiency intrusion behaviors. The proposed approach offers a 99.16% accuracy rate and a 0.3% false-positive rate.

Keywords**:** Intrusion Detection, Simulating Annealing, Hopfield, Neural Network**.**

1. Introduction

Intrusion is considered a major and continuing threat to online security. Different organizations such as governments, businesses, and even individuals work to improve the reliability of their networks by increasing online security measures.

Information infrastructure is highly essential and vital to sustain crucial activities in large networks, such as banking and telecommunications. Thus, informational system attacks present significant threat to societies and compromise the security of a data framework in different ways.

These attacks have different classifications, among which four types are related to attacker behaviors [4]:

- DoS (Denial of Service): The attacker attempts to stop legitimate users from using the system.
- Probe: The attacker attempts to discover information about the victim.
- U2R (User to Root): The attacker creates a local account on the target server and attempts to access the admin rights.
- R2L (Remote to Local): The attacker does not have, and attempts to obtain, a local host account.

To detect any of the abovementioned attacks requires the use of one or more Intrusion Detection System (IDS) tools and approaches. However, a consensus on the concept of Intrusion and IDS is likewise necessary.

Commonly, IDS works as a two-step procedure. The first step includes checking of system configuration files to reveal incorrect settings, passwords to find weak ones, and other factors to verify policy violations. These procedures are host-based and passive.

The second step includes monitoring familiar techniques of attack and enrolling system responses. These techniques are network-based and are considered active[5].

IDS is important in protecting networks from various attacks and plays a large role in online security. Firewall offers certain protection and systems still require help and support from intrusion detection[6]. Any IDS also presents fundamental problems due to the lack of human intervention in certain response scenarios. Thus, no IDS is optimal. However, in this study, we attempt to build a New Network Intrusion Detection System (NNIDS) to obtain better outcomes than current approaches, that is, higher accuracy and lower false positive rates.

With its capacity to effect considerable system damage, intrusion and its detection is one of the most crucial issues in all systems and organizations. Several intrusions can even be considered catastrophic. All intrusions apart from host or authorized networks have negative effects on systems, in which the most valuable part that may be infected is information. Therefore, all systems administrators strive to protect their systems and organizations from intrusions [6, 7].

Intrusion detection needs procedures to manage its different types and resources. These procedures cost organizations considerable effort, time, and finances. Intrusion adjustments are the Confidentiality, Integrity, and Availability (CIA) of resources, especially information[4, 7]. Intrusion effect occurs as a deliberate, unwanted attempt to access or control information and to make a device unstable or unusable. Therefore, determining different aspects of intrusion depends on its nature, but in general, such intrusions are described as a risk, vulnerability, attack, or penetration[8].

Intrusion detection aims to avoid, or at the least track, anomalies. Numerous techniques are currently available to protect online systems from malicious attacks. Most of these techniques focus on process-based rather than user-based intrusion detection, but all aim to detect any intrusion and to adapt to new types of attacks using artificial intelligence (AI) mechanisms by detecting the feature change[9].

Intrusion detection is considered an information problem, a classification that can be specifically used as a main phase for intrusion detection methods. The present intrusion detection method classifications are as follows:

1) Statistical Analysis

In the statistical analysis method [10], the state of general computer behavior and frequency of operations are recorded. Then the system determines if the user actions are legal on the basis of recorded statistical information.

2) Neural Network

A neural network model can create an IDS [11, 12]. Thus, when the supervised or unsupervised learning neural network is used, the challenges in this method are how to choose a

good network model and its features [13, 14].

3) Rule-Based Analysis

Security experts created the rules of safe or unsafe computer operations [15]. In this method, the IDS judges the user profile using the rule base.

4) Bayesian Network

The IDS judges the user behaviors using the Bayesian network [5] to detect different probabilities of events and behaviors [10].

5) Finite State Machine

System operations can be expressed as a state by finite state machines [5]. The method bottleneck is how to set the transfer condition [10].

6) Data Mining

Intrusion detection can use the association rules of mining methods [5, 16]. The system extracts different features of messages to construct association rules, such as source and target nodes, IP address, and the called or calling functions.

Today, the number of private networks or Internet usage that broadly utilize various e-government systems continues to increase. As such, no ideal solution can prevent network intrusion, and thus no computer system is absolutely protected from network intrusion. The main means to avoid potential harm is the early detection of intrusion.

Intrusion detection consists of two stages, namely, to extract user characteristics and then decide whether the features are from an authorized user or an attacker. The feature set consists of users [17, 18] logging in a computer network. IDS then determines whether the set belongs to one of two types – licensed user or attacker.

The main problem can be abstracted as developing an efficient IDS, or one that is at least better than all others. This problem can be solved by achieving the following related tasks:

- Distinguish between data patterns dependent on their styles (DoS, Probe, U2R, R2L, Normal).
- Search for any Intrusion Detection Model that can discriminate between labeled and unlabeled patterns before processing.
- Generate a new set to expand the set of comparable data patterns (vectors) that represents the main 5-classes by generating a new set of each.
- Try to maximize accuracy rate (AR), maximize true positive rate (TPR), and minimize the false positive rate (FPR).

2. Related Works

Many research studies on intrusion detection are carried out using different methods, all of which inform us the nature of the issue and attempt to solve it in various ways. Details are given in the IDS survey[16] and the study in [19], which provide an IDS description and a quick overview of its architecture. Some of the neural network approaches to which the proposed approach belongs are summarized as follows:

The study in[15], tried to developed an application for the improvement of the knowledge domain using Machine Learning (ML) to generate Expert System (ES) rules. Thus, people managing the information should be less than its previous numbers. The expected work includes a military subnet that creates AI rules generation[20] with the classic ES and the Network Exploitation Detection Analyst Assistant (NEDAA). NEDAA contains rules generating modules using two software packages of AI: genetic algorithm (GA) for generating rules and decision-making tree (DT) for the deterministic alternative. Both techniques are used to automatically generate network classification rules. DT has advantages over GA in the final rule.

Neural networks [20] are suggested to be used as an IDS application tool. Several neural networks architectures (Backpropagation, Radial Back Feature, and Self-Organization Map) are explored and evaluated to decide which IDS implementations ideally fit in a system that can be used to identify new attacks. The generalization features allow the use of imprecise and incomplete data, making neural networks a good solution to detect a known, aggressor-modified firewall attack. Results of the architecture test using the entire KDD corrected data set and an updated False Alarm Rate (FAR) and detection rate (DR) data sets indicate that the Multi-Layer Perceptron (MLP) network is most successful. Rising feedback behavior also tends to further boost DR.

Expectation–Maximization (EM) [21] suggests a modern hybrid clustering paradigm focused on an enhanced unit range (IUR), key feature analytics (PCA), and unregulated learning algorithms to incorporate related alerts to minimize warnings. The recent combination of IUR, PCA, and EM algorithms is IPCA–EM, which is the innovation of this study. Checking other unattended learning algorithms validates the pattern. Experimental results of the model on the DARPA2000 dataset are clustering accuracy and processing time, which helps capture and delete unwanted notifications from multi-sensor violation reports. Compared with other unregulated learning algorithms, the present findings are promising in terms of grouping rate and processing time.

Work [2] primarily uses Hopfield networks to classify binary pattern vector issues. Hopfield networks are built by supplying data or pattern vectors for each class. Patterns are called class patterns with n binary components in n-dimensional data space, where each class pattern corresponds to a cube corner. The network then categorizes skewed trends. When the network displays a deformed pattern, a separate pattern is related.

The Kohonen network [22] does not extend to the competitive learning network and has several different applications. Output units in the Kohonen network are arranged, often in a two-dimensional array, although this depends on the application. The order selected by user1 determines the output neurons. Unattended learning forms a type of artificial neural network to produce an unchecked depiction of the input space of training samples, known as a map, of low dimensions (typically two-dimensional). Self-organizing maps vary in using the neighbor function to maintain input space principal components from all other artificial neural networks. The weight of the output units is adapted to preserve the order in the input space during the presentation of the network learning patterns, which are close to each other (the distance measure used to find the winning unit determines ‘nearness’).

Novel Hybrid NN Machine Learning Mode (HNNMLM) [23] applies neural net classification for new attacks but

expert-based method for known attacks. The model achieves a 97.2% detection rate for DoS and Probing intrusions, and a false negative rate of less than 0.04%.

In work [24], Recurrent Neural Network (RNN) used to detect DDoS attacks with tuned parameters. Learning parameters are adjusted based on features and then implemented to predict the attack. The study concluded the proposed model is a promising approach to detecting intrusion in software-defined networking environments.

The proposed model [25] uses multi-layered neural networks optimized for Fog computing protection that is very similar to end-users and IoT devices. Where experiment results and simulations demonstrated the reliability and robustness of the proposed model for several performance metrics.

An effective IDS was proposed in this work [26] using hybrid data optimization consisting of two parts: data sampling and feature selection called DO IDS. The Isolation Forest (iForest) is used in data sampling to eradicate outliers, genetic algorithm (GA) to maximize the sampling ratio, and the Random Forest (RF) classifier as evaluation criterion to achieve the optimal training dataset. In feature selection, GA and RF are again used to achieve the desired function subset. Finally, an RF-based intrusion detection system is designed using the optimal training dataset obtained by data sampling and features selected by feature selection

3. Methodology

In this study, the proposed NNIDS consists of three phases. The model for intrusion detection is designed using Simulated Annealing (SA) for data clustering and classifying, HNN and SA for training, and a support vector machine (SVM) for learning and detecting. The model clarifies our proposed approach of an improved IDS to enhance performance efficiency. Collected data are used in the training and testing processes in a special unit called the “environment”, which are presented in two tables in our database.

The environment unit represents 10% of the KDD99 [3] data that are randomly selected from the entire set. The unit is split into two subsets, training (normal and abnormal data, 7%) and testing (labeled and unlabeled data, 3%) subsets. This unit supplies all codification and SVM units using the vectors of features (data vectors or patterns).

The proposed approach comprises three phases: Preliminary, Training, and Detection (i.e., determining attack type).

3.1. Preliminary phase

This phase represents our model's first stage, it is considered a initial step towards achieving our model's main purpose by arranging data and transforming it into appropriate input format for next phases. After capturing the network data, the Codification module codes features into a suitable style while Clustering extracts input data features to eliminate space features dimensions. The SA module is used to cluster the main attack types. Figure 1 explains Preliminary phase structure. This phase consists of the following modules:

1. Capturing: the model captures the network data from the environment.
2. Codification: codes features into a specific style.
3. Feature extraction: transforms input data into a set of features that are then stored in the database.
4. Clustering: the dataset is clustered into 22 groups according to the different working features into the main five groups based on intrusion form (DoS, Probe, R2L, U2R, and Normal) using the SA algorithm.

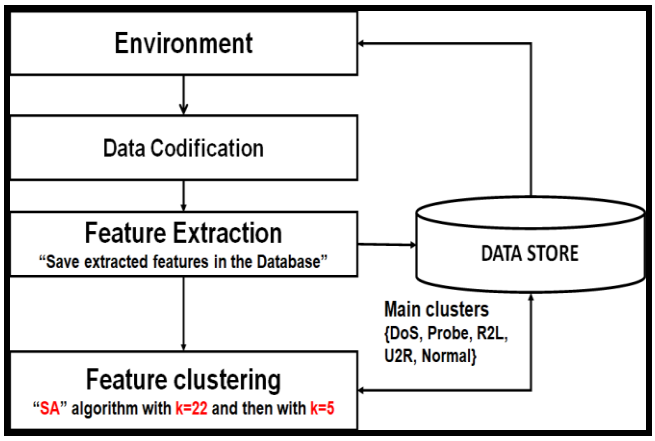

Figure 1. Construction phase

3.2. Training

This second phase represents the training unit, built from HNN as supervised learning with fixed weight [27] and SA algorithm to improve its performance (see Figure 2). HNN works as a classifier, but only for behaviors marked with attack types (DoS, Probe, R2L, U2R, Normal). SA can also serve as a classifier, but only for behaviors that are not yet classified as attacks or not, and if yes, then determines the attack type (DoS, Probe, R2L, U2R). Training works as follows, as shown in Figure 2.

1. Features of Extract Data Pattern
2. Determine whether this data pattern is labeled.
3. Determine if the marked data pattern checks as normal or anomalous.
4. Send the normal data patterns to the Hopfield neural network (HNN) unit and anomalous ones to the SA unit.
5. If the data pattern is labeled anomalous, the SA unit generates new vectors.
6. If the data pattern remains unmarked, set it to be anomalous.
7. After clustering and labeling by SA unit, send the data pattern to the HNN unit.

The output of the training phase is always collected to expand the set of vectors of intrusion compared in Phase 3. This helps in further intrusion detection.

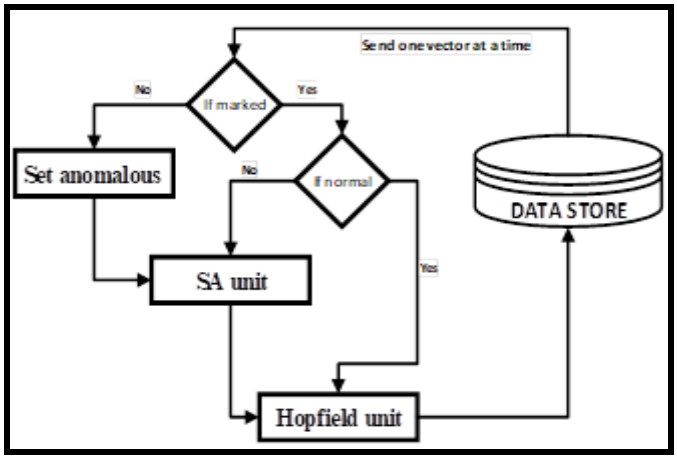

Figure 2. Training phase

3.2.1. Hopfield neural network

First introduced in 1982 by physicist John Hopfield Network. He introduced an asynchronous neural network model that would instantly impact the AI community. This is a particular situation of a Bidirectional Associative Memory (BAM) but it was developed chronologically before the BAM. It refers to artificial neural types known as thermodynamic models[28]. This is an example of an automated or "content-addressed" input memory. An associative or text-addressable memory is a memory device that will inform you of a memory pattern depending on the quality of an object. The memory is auto associative when the pattern is remembered from the memory when a sufficiently similar pattern is shown.

We can also say that Hopfield networks are like symmetric recurring networks. Symmetric networks also converge to strong attractors of nodes. A symmetric network therefore cannot create, know or store a temporary pattern sequence [29].

HNN is the quickest and easiest neural network. A fully integrated one-layer auto-associative network, HNN has a single layer attached to each neuron without hidden layers. Typically, HNNs use binary vector classification problems. Providing data or pattern vectors corresponding to the different classes creates the Hopfield network. These vectors are thus class patterns. Figure 3 displays the key steps of the algorithm [2].

3.3. Detection

This phase consists of the SVM unit acting as a classifier to improve the intrusion detection instead of the HNN. The SVM approach [5, 6] is well used to create a new IDS to maximize the AR rate and reduce FPR. Vector samples can be obtained from the environment and then compared with the collection of vectors from the clustering and training phases. Figure 4 describes the SVM workings, which are divided into the following steps.

1. Compare new vectors from the test data with the group formed by the union of the key cluster vectors and those produced by the Simulated Annealing (SA) Model.
2. Identify the type of current attack by the major (recording) types of attack.

Each of these phases provides the necessary data in suitable form to other phases, all of which work together to achieve the main goal, that is, identify different types of intrusion. Our model is reusable and adaptable, can be easily changed and updated, due to its portioned form. Thus, each phase can be changed or updated independently.

3.4. Datasets

The proposed approach applies the experiments on the data set KDD99, which comprises a wide range of interference and normal operations simulated in a military network environment. Simulation attacks are applied as one of the following main types: DoS, R2L, U2R, and Probing. The extracted features of a connection record consist of each instance in the dataset. Experiment results are carried out using 311,029 records, with approximately 30,000 used in training and the rest in testing.

The KDD99 data set is mainly used in the field of intrusion detection and draws considerable research interest due to its good definition and availability.

1. Weights are initialized and allocated.

$$W_{ij} = \begin{cases} \sum_{S=0}^{M-1} X_i^S X_j^S \mu & i \neq j, X_i^S \text{ can be } -1 \text{ or } +1 \\ 0, & i = j, \end{cases} \qquad 0 \le i, j \le N-1$$

W_{ij} is node connection weight i to node j.

2. Initialize with Unknown Pattern:

$$\mu_i(0) = x_i, \; 0 \le i \le N\text{-}1$$

where μ_i (w) is the output node i at time t and x_i.

3. Repeat until convergence is attained.

$$\mu_i(w+1) = f_\eta \left[\sum_{i=0}^{N-1} w_{ij} \mu_i(w) \right] \quad 0 \le j \le M\text{-}1$$

The *function* f_η is the hard-limiting nonlinearity.

4. Repeat from step 2 to remain unchanged with further iterations.

Note: *The final node outputs reflect the layout pattern that best fits the Unknown input*.

Figure 3. Hopfield Learning Algorithm

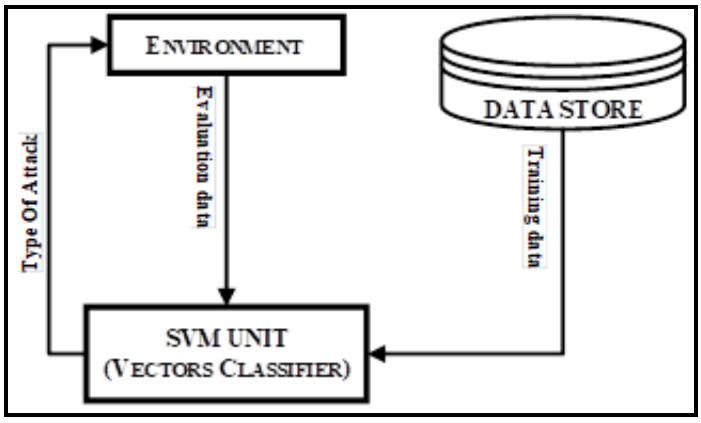

Figure 4. Detection phase

3.5. Evaluation

The experience aims to demonstrate the method effectiveness in the calculations of FPR, AR, and Receiver Operating Characteristic (ROC) curve to trade-off between TPR and FPR at different classification thresholds. Results are evaluated and compared with others.

4. Experiments and Results

Data from the third international technology exploration and data mining applications competition (KDDcup'99) [3] are used to train and test the feasibility of the proposed approach. From the KDD99 dataset, 311,029 samples are used in this study. The training data pattern sets cover five attack types (DoS, Probe, R2L, U2R, Normal). Clustering and classification processes are done by the proposed approach through a set of extracted features. The results show that approximately 74% are DoS threats. Normal patterns are approximately 19%. Other attack patterns for Probe (1.3%), R2L (5%), and U2R (0.0007%) are infrequent.

To evaluate the proposed approach, its performance is compared with that of two other works, the first HNNMLM [23] using only Hopfield and the second using backpropagation network (BPN) [30]. The same dataset is used to validate the most suitable strategy. Table I shows that

the proposed approach has greater attack classification accuracy than other systems.

The ROC curve also shows classifier accuracy and offers the trade-off between TPR and FPR at different classification levels. TPR = TP/(TP+FN) and FPR = FP/(TN+FP).

TPR is the percentage of correctly expected results. However, FPR is the proportion of correct results incorrectly estimated. Figure 5 demonstrates the ROC contrast of the proposed model with that of BPN.

Table I: General Accuracies

Model	Accuracy of attacks classification					Global accuracy
	Normal	DoS	Probe	R2L	U2R	
NNIDS	99.30	99.49	99.73	99.78	99.92	99.16
BPN [30]	95.97	99.93	99.92	96.13	-	95.94
HNNMLM [23]	-	96.36	96.61	98.13	97.54	97.17

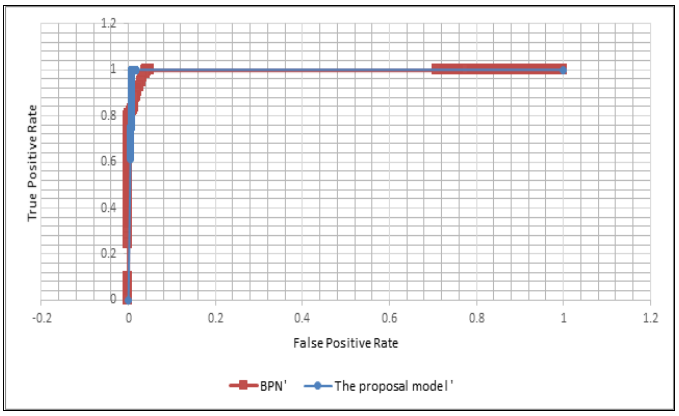

Figure 5. Receiver Operating Characteristic (ROC) curves

5. Conclusion

The proposed method complies with the combined detection of misuse and anomaly systems, and the clustering results in a reduction of dimensionality and identification of attacks by classification, which improves accuracy and increases flexibility. To our knowledge, no previous work analyzes online intrusion in this manner. Most of previous models concentrate on clustering and classification. However, in addition to these two phases, the proposed approach concentrates on expanding the rules by generating a new set of vectors and on reducing training time. Thus, the intrusion detection process results in higher accuracy by employing different approaches in the same model.

The experiment shows that the proposed NNIDS can be successful with improved classification and detection speeds. This model shows improved performance, efficiency, and accuracy, in detecting all intrusion types. In the future, other methods can be used for clustering as an Intelligent Water Drop algorithm and learning as Kohonen to enhance system accuracy and efficiency.

References

[1] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical science*, vol. 8, no. 1, pp. 10-15, 1993.

[2] A. M. A., "Hopfield Neural Networks for Optimal Solutions " *IJCNN*, 1992.

[3] K. C. Data., "http://kdd.ics.uci.edu/databases/ kddcup99/ ".

[4] H. Kozushko, "Intrusion detection: Host-based and network-based intrusion detection systems," *Independent study*, 2003.

[5] R.-C. Chen, K.-F. Cheng, Y.-H. Chen, and C.-F. Hsieh, "Using rough set and support vector machine for network intrusion detection system," in *2009 First Asian Conference on Intelligent Information and Database Systems*, 2009, pp. 465-470: IEEE.

[6] A. Zainal, M. A. Maarof, S. M. Shamsuddin, and Security, "Ensemble classifiers for network intrusion detection system," *Journal of Information Assurance*, vol. 4, no. 3, pp. 217-225, 2009.

[7] S. M. Bridges and R. B. Vaughn, "Intrusion detection via fuzzy data mining," in *12th Annual Canadian Information Technology Security Symposium*, 2000, pp. 109-122: Citeseer.

[8] R.-I. Chang, L.-B. Lai, W.-D. Su, J.-C. Wang, and J.-S. Kouh, "Intrusion detection by backpropagation neural networks with sample-query and attribute-query," *International Journal of Computational Intelligence Research*, vol. 3, no. 1, pp. 6-10, 2007.

[9] C. Izbasa, "A Neural Network-Based IDS, Software, www.ieat.ro/researchreports/ids.pdf/download.."

[10] K. Ilgun, R. A. Kemmerer, and P. Porras, "State transition analysis: A rule-based intrusion detection approach," *IEEE transactions on software engineering*, vol. 21, no. 3, pp. 181-199, 1995.

[11] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, no. 2, pp. 222-232, 1987.

[12] R.-C. Chen, K.-F. Cheng, and C.-F. Hsieh, "Using rough set and support vector machine for network intrusion detection," *arXiv preprint arXiv:*. 2010.

[13] P. A. Diaz-Gomez and D. F. Hougen, "Misuse Detection-A Neural Network vs. A Genetic Algorithm Approach," in *ICEIS (2)*, 2007, pp. 459-462.

[14] A. A. Obeidat, "Hybrid Approach for Botnet Detection Using K-Means and K-Medoids with Hopfield Neural Network," *International Journal of Communication Networks and Information Security*, vol. 9, no. 3, pp. 305-313, 2017.

[15] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," in *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, 1999, pp. 371-377: IEEE.

[16] P. Kazienko and P. Dorosz, "Intrusion detection systems (IDS) Part 2-Classification; methods; techniques," *WindowsSecurity. com*, 2004.

[17] J. Bouvrie, "Notes on convolutional neural networks," 2006.

[18] M. Abu-Khalaf, "EE 5322 Neural Networks Notes," *Personal Study* vol. Software, no. arri.uta.edu/acs/abumurad/EE5322/EE5322_NN_notes.pdf 2004.

[19] B. Wahyudi and K. Ramli, "Implementation and analysis of combined machine learning method for intrusion detection system," *International Journal of Communication Networks and Information Security*, vol. 10, no. 2, pp. 295-304, 2018.

[20] P. Kukielka and Z. Kotulski, "Analysis of different architectures of neural networks for application in intrusion detection systems," in *2008 International Multiconference on Computer Science and Information Technology*, 2008, pp. 807-811: IEEE.

[21] M. M. Siraj, M. A. Maarof, and S. Z. Hashim, "Intelligent alert clustering model for network

intrusion analysis," *Int. J. Advance. Soft Comput. Appl*, vol. 1, no. 1, pp. 1-16, 2009.
[22] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.
[23] W. K. AL-Rashdan, R. Naoum, and A. S. Wafa'S, "Novel network intrusion detection system using hybrid neural network (Hopfield and Kohonen SOM with conscience function)," *IJCSNS*, vol. 10, no. 11, p. 10, 2010.
[24] M. A. Albahar, "Recurrent Neural Network Model Based on a New Regularization Technique for Real-Time Intrusion Detection in SDN Environments," *Security Communication Networks*, vol. 2019, 2019.
[25] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice Theory*, vol. 101, p. 102031, 2020.
[26] J. Ren, J. Guo, W. Qian, H. Yuan, X. Hao, and H. Jingjing, "Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms," *Security Communication Networks*, vol. 2019, 2019.
[27] J. J. Hopfield, "Artificial neural networks," *EEE Circuits Devices Magazine*, vol. 4, no. 5, pp. 3-10, 1988.
[28] R. Rojas, "The backpropagation algorithm," in *Neural networks*: Springer, 1996, pp. 149-182.
[29] Q. He, "Neural Network and its Application in IR," *Graduate School of Library Information Science, University of Illinois at Urbana-Champaign Spring: Champaign, IL, USA*, 1999.
[30] R. J. Erb, "Introduction to backpropagation neural network computation," *Pharmaceutical research*, vol. 10, no. 2, pp. 165-170, 1993.