A Practical Intrusion Visualization Analyzer based on Self-organizing Map

Jie Wang
College of Information and Communication Engineering
Harbin Engineering University
Harbin, China
heuwangjie@hrbeu.edu.cn

Yun Lin
College of Information and Communication Engineering
Harbin Engineering University
Harbin, China
Corresponding Author: linyun@hrbeu.edu.cn

Lei Chen
Department of Information Technology
College of Engineering and Computing
Georgia Southern University
Georgia, USA
LChen@georgiasouthern.edu

***Abstract*—In the era of big data, devices and services are interconnected through the network. For the purpose of ensuring the security and reliability of network communication and resisting the threat of intrusions, Network Intrusion Detection Systems (NIDS) have become an infrastructure in the Internet ecosystem. It is challenging to develop an effective NIDS which can provide visualization ability due to the high-dimensional characteristic of network traffic. In our work, a practical NIDS based on the Self-Organizing Map (SOM) neuron network is presented. The proposed method can arrange high-dimensional data in a 2D topological graph in an orderly manner, which enables an easy-to-understand insight of the distribution of normal or abnormal data. With the training, similar data is grouped together, based on which intrusions can be identified. In addition, a hybrid preprocessing model that combines two different feature selection methods is adopted to accelerate the detection process. In order to evaluate the performance of the proposed framework in terms of several different metrics, experiments are performed over a benchmark dataset UNSW-NB15 which is widely used in intrusion detection research. Experiment results show that the proposed approach provides a user-friendly visualization and enhances the detection of intrusions.**

***Keywords*—Intrusion Detection System, Self-Organizing Map, Visualization, Feature Selection, Big Data**

I. Introduction

With the popularization of the Internet of Things (IoT) technologies, various devices and applications are interconnected forming a network of things [1]. In such environment, intelligent devices connected with resources typically produce huge amounts of data. Multifunctional devices with sensitive information are subject to vulnerabilities and attacks [2]. Network Intrusion Detection Systems detect intrusions by analyzing traffic and log information, and have become a critical component for the devices connected to the Internet.

Generally, as long as a device has the access to the Internet, the NIDS that is placed in a specific location will extract specific traffic information. It calculates using designated detection algorithm to classify the traffic data into different categories based on predefined rules [3]. According to different detection approaches, there exist two different types of NIDS: signature-based NIDS and anomaly-based NIDS. The signature-based approach analyses the incoming traffic and searches its attack database for a match of signature. By comparing current data with existing features of known attacks, this method is usually expected to accurately detect specific attack types [4]. However, in the current complex and ever-changing network environment, intrusion detection methods based on signature technology suffer from inflexibility, due to the fact that updates of the IDS database are difficult to keep up with the development of attack types and their variants [5]. The other type of NIDS is anomaly-based, which is able to detect unknown intrusions in contrast to signature-based NIDS. Typically, a trustworthy model built upon normal data is obtained by applying machine learning or data mining techniques [6]. The classification of an observed event as normal or abnormal is typically determined by the value of the deviation between the event data and prior normal data. However, low detection accuracy and high false alarm rate are its fatal weaknesses. Therefore, in reality, intrusion detection models are often a hybrid of the two approaches.

For the realization of visualization, high-dimensional data are required to be reduced to 2-D or 3-D. The mainstream dimension reduction approach is the projection method, including Principal Component Analysis (PCA) [7]. Low-dimensional data are obtained according to different projection strategies, for example, PCA based on the variance of the dataset. Upon projection, two components are selected as the representative information and scattered to a plane graph. Unlike the above approaches, the strategy of SOM to reduce features is, instead of through projection transformation, to build a topology-preserving mapping model [8]. SOM can map the sample pattern classes into the output layer in order, thus providing clearer data expression and more accurate discovery of data rules. Every neuron in the output graph has its specific actual class information.

In our work, we propose a practical IDS with visualization ability based on the self-organizing map neural network. In addition, two different feature selection strategies are married to remove noise and unrelated features for accelerating the detection process and improving the overall system performance. Subsequently, the Isolation Forest method is applied to avoid the outliers. The proposed framework is evaluated over a widely-used dataset UNSW-NB15, and the simulation results show that our proposed method has superior performance in detection accuracy, while providing effective and user-friendly visualization.

The remainder of this paper is organized as follows. Section □ provides the background and reviews the related works of NIDS, especially relevant to the IoT. In Section □, the details of the proposed work are illustrated and data preprocessing techniques are introduced. Section □ describes the dataset and metrics used for the simulations and results are explained and analyzed in the same section. Section □ draws the conclusion of this paper and casts our future work.

II. Background and Related work

Over recent years, intrusion detection has been an intriguing issue in the research of cyber and network security. This section illustrates the common NIDS architecture in the modern environment and reviews the related works of NIDS. In addition, basic theories about SOM will also be reviewed.

A. Network Intrusion Detection System

NIDS is of great importance in the screening and analyzing of any Internet activity. It is a safety line to ensure the normal operation of the network. A typical architecture of modern NIDS is shown in Fig. 1.

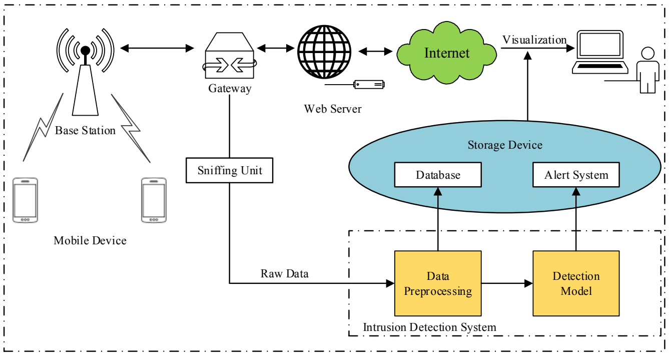

Fig. 1. A typical architecture of modern NIDS.

End-users, including potential intruders, send messages or service requests to destination addresses through the gateways [9], which naturally become the best place for intrusion detection. A sniffing unit is used to capture from the Internet the monitored traffic, which will be passed to the data preprocessing unit where certain techniques, including feature selection, extraction, transformation and normalization, are adopted to obtain useful information. The detection model is already well trained offline using strategies such as machine learning, statistic knowledge, data mining, and expert rules. Additionally, online incremental learning or manual operation will keep the model updated through the actual detection process. The final detection results will be displayed to the network manager or end-users who can observe the dynamic changes of network conditions and make corresponding decisions.

B. Self-organizing Map

Self-organizing Map (SOM) was initially proposed by Kohonen in 1982 as an unsupervised learning algorithm [10]. SOM has evolved into a powerful tool for mapping data from high to low dimensions and providing visualization function, as shown in Figure 2.

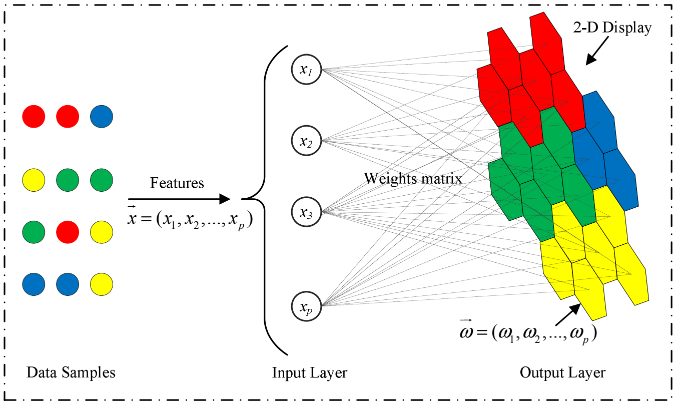

Fig. 2. A simple structure of SOM.

There is no hidden layer in SOM, and each node in the topological graph is connected to every neuron in the input layer. The following four steps are needed to train a SOM model.

- Initialization: this step gives a random value $\omega_{i,j}$ to every connection between neuron i in the input layer and neuron j in the output layer.
- Competition: in this phase, a Best Matching Unit (BMU) is found using distance-based similarity strategies, i.e. the Euclidean distance,

$$\arg\min_{j} d_j(x) = \arg\min_{j} \{\sum_{i=1}^{D} (x_i - \omega_{i,j})^2\} \qquad (1)$$

where $\{x_1, x_2, ..., x_D\}$ is an input data sample set and j is the index of the mapped unit in the output layer.

- Cooperation: similar to the interaction between excitatory neurons and nearby neurons in biology, the winner node in SOM can affect its neighbors. Furthermore, the degree of influence of node j on k decreases as the distance between the two increases,

$$T_{j,k} = \exp(\frac{-d_{j,k}^2}{2\sigma^2}) \qquad (2)$$

- Adaptation: update the weight matrix so that the distribution of the data mapping can be adjusted,

$$\omega_{k,i} = \eta(t)T_{j,k}(x_i - \omega_{k,i}(t)) \tag{3}$$

The research on intrusion detection theory and related methods has been of great interest in recent years in both academia and industry. Significant amounts of in-depth research have been conducted from several different perspectives, including machine learning [11], expert systems [[12], data mining [13], and statistical knowledge [14] and have been applied to specific network systems. In [15], the authors proposed a framework with five different feature-selection methods and two classifiers were adopted with these five approaches. In [16], Muna built a Deep Feed Forward Neural Network for intrusion detection, where the deep auto-encoder was used for weight initialization. Simulations presented in the paper showed satisfactory performance in detection accuracy and time. Some other research investigates IDS with visualization capabilities. Most of these works used projection techniques, i.e. PCA. The pioneer work of PCA-based visualization approach, transforming high-dimensional features into low-dimensional data, was presented in [17], with an focus on the detection of Denial-of-Service attacks and Network Probe attacks. Bi-plots were adopted to show the detection statistics in a graphical form. In [18], the authors used an approach combining a SOM network with k-means to detect anomaly over the KDDCUP'99 dataset. SOM is capable of obtaining a preliminary clustering result, based on which k-means can used for secondary clustering to further enhance the performance. However, the default visualization in this approach is not effective, resulting in sub-optimized detection performance. In this paper, we pay attention to data preprocessing and emphasize the advantages of SOM to maximize its potential in data representation and visualization.

III. Proposed Model for Intrusion Detection

In this section, a practical model with enhanced visualization for intrusion detection is proposed. As illustrated in Algorithm I, it contains 4 steps, which are further discussed in this section.

ALGORITHM □ THE PROPOSED IDS WITH ENHANCED VISUALIZATION

```
Input: Training dataset and testing dataset
Output: A topological graph and detection results
  Step 0: Initialization
    $T$; The maximum number of iteration in SOM
    $N$; the number of input neurons in SOM
    $L$; The size of output lattice in SOM
    $R$; The ratio of outlier removal
  Step 1: Feature Selection
    Run the GR and ARM algorithms to obtain subset1 and subset2
    Obtain the union of selected two subsets
  Step 2: Outlier Removal
    Run the Isolation Forest algorithm with removal ratio $R$ and keep the
    rest data as training data
  Step 3: Train the model based on the training data
    for (iter = 1 to T; iter++) do:
        run the SOM over selected data samples
    end for
    return a topological graph with label
  Step 4: Detection based on the testing data
    while new data $x$ do:
    if the coordinate $Pos(x)$ of winner($x$) in graph:
        if $Pos(x)$ is attack:
            label $x$ as attack
        else if $Pos(x)$ is normal:
            label $x$ as normal
    else:
            label $x$ as anomaly
    end while
    return predicted label
```

A. Feature Selection

The curse of dimensionality has been considered a roadblocker for the development of IDS. It is pragmatic to remove redundancy even wrong data to facilitate the detection process and provide high detection accuracy [19]. Therefore, feature selection is particularly a vital pre-processing stage.

In this paper, we adopt a hybrid feature selection approach by marrying the Gain Ratio (GR) and Association Rule Mining (ARM) methods. GR evaluates the features with labels and produces a ranking in descending order [20]. On the other hand, the ARM technique generates correlation rules within the features [21]. In this research, the feature selection process, based on what is described in [19], takes the top 10 features of GR method and the most closely related 10 features of ARM method, the union of which forms the final feature subset.

B. Outliers Removal

Outliers are low-frequent samples far from the main distribution of data and they will cause a negative deviation from the exact position of the data center. To tackle this, Isolation Forest approach is adopted for removing the outliers [22].

The goal of Isolation Forest is to build a forest where the normal data is move away from the root of the tree while the outliers is close to the root. It is a fast-running method based on Ensemble. The characteristics of linear time complexity and high precision ensure its adoption in the big data environment.

C. Model train and detection

The inputs of the model can only be numeric, so the nominal attribute values should be converted into continuous numerical values, that is feature transformation. For example, the feature *state* has four different nominal values; therefore they can be replaced with 1, 2, 3, 4 respectively. Next, standard format attributes are obtained using a Min-Max technique to normalize the data, as denoted in (15)

$$x^* = \frac{x - \min}{\max - \min} \tag{4}$$

The preprocessed data is then selected to train the model, with parameters such as the number of iterations and the size of the lattice. During the training phase, the quantitative error curve will need to be monitored and testing dataset is used to evaluate the model.

IV. Simulation

In this section, the details of the simulation will be presented and the analysis of simulation results will also be explained. We start with the introduction of the dataset.

A. *UNSW-NB15 Dataset*

As we all know, dataset is of great importance to machine learning problems. Significant amount of work has been done on the dataset for network intrusion detection. KDD'99 and its updated version NSL-KDD are considered classic and exemplary [23]. However, neither of them can easily adapt to the networks in the recent years, mainly due to redundancy and data imbalance as well as newer types of attacks.

A complex and balanced data set is important for intrusion detection. UNSW-NB15 is a hybrid dataset from realistic modern normal activities and synthetic contemporary attack behaviors created by Moustafa using an IXIA PerfectStorm tool [24]. This dataset introduces an increased number of attack types and there are ten different classes, each of which has 42 features, as shown in table I.

TABLE I. UNSW-NB15 DATASET

Number	Description	Number	Description
1	dur	23	dwin
2	proto	24	tcprtt
3	service	25	synack
4	state	26	ackdat
5	spkts	27	smean
6	dpkts	28	dmean
7	sbytes	29	trans_depth
8	dbytes	30	response_body_len
9	rate	31	ct_srv_src
10	sttl	32	ct_state_ttl
11	dttl	33	ct_dst_ltm
12	sload	34	ct_src_dport_ltm
13	dload	35	ct_dst_sport_ltm
14	sloss	36	ct_dst_src_ltm
15	dloss	37	is_ftp_login
16	sinpkt	38	ct_ftp_cmd
17	dinpkt	39	ct_flw_http_mthd
18	sjit	40	ct_src_ltm
19	djit	41	ct_srv_dst
20	swin	42	is_sm_ips_ports
21	stcpb	43	attack_cat
22	dtcpb	44	label

The distribution of normal activities and attacks is shown in Fig. 3. There is no redundant record among the training and testing set.

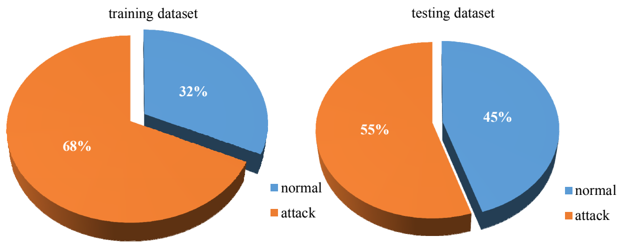

Fig. 3. Distribution of normal activities and attacks in training and testing dataset.

B. *Metrics*

To evaluate the performance of a machine learning model, we adopt several metrics such as the *Accuracy*, *Recall*, *Precision*, and *F1-Score* as further defined in formula (5) ~ (8). *TN* is true negative while *FN* means false negative and *TP/FP* stands for true positive and false positive, respectively.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

C. *Simulation results*

To begin the simulation, representative features need to be selected using GR and ARM. GR ranks the features so that the first 10 features can be selected. ARM implements similar process using a different rule. The results of individual and union subsets are shown in Table II. The 17 hybrid features <*2, 3, 6, 7, 8, 9, 10, 11, 12, 19, 22, 24, 31, 32, 34, 35, 40*> are adopted for the simulation. The training dataset is about 25% of the original training instances, and 10% of the test set data was used to verify the validity of the model.

TABLE II. FEATURES SELECTION RESULTS

Method	# of Features	Feature number
GR	10	10, 11, 32, 35, 2, 3, 6, 7, 8, 12
ARM	10	3, 10, 24, 19, 22, 31, 32, 40, 9, 34
Hybrid	17	2, 3, 6, 7, 8, 9, 10, 11, 12, 19, 22, 24, 31, 32, 34, 35, 40

When training the model, different lattice sizes of network were tried and performance tests were conducted under different test sets. Fig. 4 is the topological graph organized during the training phase. According to the previous description of the SOM network, it is not difficult to understand that each bubble represents a neuron of the output layer in SOM. The bubbles in the figure can be divided into three types according to the color composition. The orange bubbles represent attacks while the blue ones indicate normal. Mixed-color bubbles indicate that both attack and normal data are mapped to the neuron, possibly leading to false alarms. When traffic sample is mapped to the mixed-color units, the principle of maximum probability is adopted to help determine whether it is normal or abnormal. Areas without bubbles indicate that no traffic data is mapped. This scenario is not uncommon because the similar data is often clustered together or nearby. In addition, the blank places provide a chance to detect new data that never appeared in the database. Therefore, when data appears on these neurons, it brings attention of data analysts who may use network tools to capture the data for analysis. Fig. 4 demonstrates the intuition of the visualization as the distribution of attack samples and normal data can be easily observed, providing excellent convenience and effectiveness for monitoring network status. Online learning approach that fine-tunes the model can help make it significantly more useful and accurate. The dynamic topological graph will help identify the trend of network traffic and better interprets traffic features. This is though beyond the scope of this paper and is regarded as our future work.

The detection process is as follows: a traffic packet can be captured using network tools such as Wireshark and specific information such as *protocol* can be extracted to form a sample. In this manner, 17 features <*2, 3, 6, 7, 8, 9, 10, 11, 12, 19, 22, 24, 31, 32, 34, 35, 40*> are captured and imported into the model. Upon the completion of the calculation, there will appear a shining bubble indicating the sample belonging to the corresponding class, thereby completing the classification.

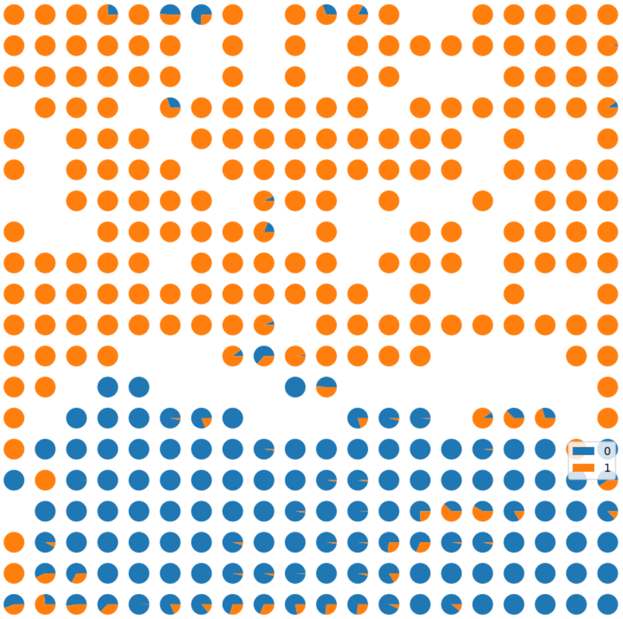

Fig. 4. A 20×20 lattice topological graph for training dataset.

Fig. 5 shows the convergence curve throughout the training process. Mean Square Error is adopted to evaluate the convergence situation. The error changes very slightly when the number of iteration reaches 40 thousand, indicating a convergence from that point.

In addition to the intuitive visualization, this model is capable of detecting intrusions at comparable accuracy and precision without the need of deep learning. The evaluation results of the proposed model using the aforementioned metrics are shown in Table III with the comparison to two approaches presented in the literature within the last two years. The performance of structure 20×20 is better than that of structure 30×30 in all metrics. For the proposed 20×20, the values of its recall and F-1 score are better than those in the comparison papers. The accuracy of 92.78% is superior to that of K-SVCR [25] and Skip-gram [26], and the precision of 91.47% are between the two. Possible reasons would include that the selected features may not be optimal, and that the training set chosen could be limited, and thereby confining the generalization ability for providing superior detection capabilities. In our future work, we will consider using deep learning methods to extract an increased number of separable features over the entire training set for better performance.

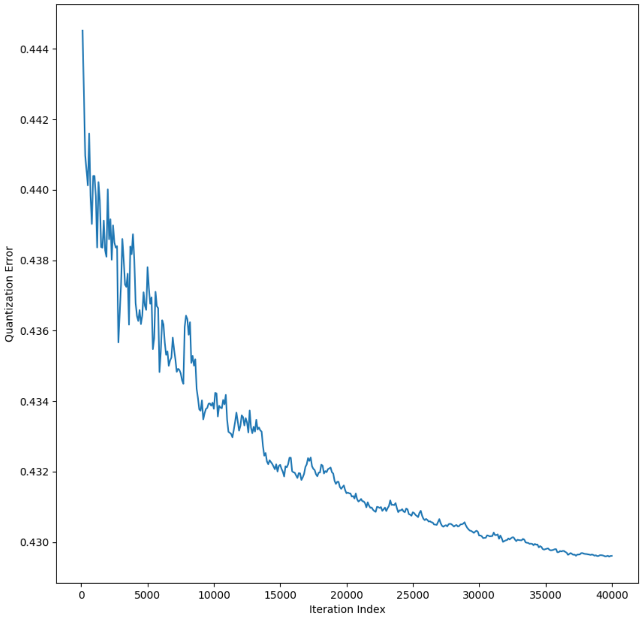

Fig. 5. Quantization error curve during the training phase.

TABLE III. THE COMPARISON OF CLASSIFICATION PERFORMANCE

Method	Accuracy	Precision	Recall	F_1
Proposed 20×20	92.78	91.47	98.58	94.89
Proposed 30×30	90.23	90.01	96.33	93.06
K-SVCR in [25]	84.65	87.42	89.55	88.47
Skip-gram in [26]	91.02	99.20	82.07	89.83

V. CONCLUSION AND FUTURE WORK

The research presented in this paper introduced a practical NIDS which not only detects intrusion effectively but also provides an intuitive visualization based on SOM. To speed up the process of detection and remove redundancy and irrelevant information, a hybrid feature approach composed of GR and ARM is used. 17 features are selected as the input of the neuron network and the widely-used and complex dataset UNSW-NB15 is adopted in the simulation. Several metrics are separately calculated and the results show that the proposed approach can effectively detect intrusions when compared to recent approaches. In addition, the visualization of the proposed model provides an intuitive means to monitor and understand network traffic. While the accuracy is superior to two recent approaches, the precision of the proposed method still has room to improve when compared to Skip-gram. Future work includes using deep learning to enhance performance and implementing multi-class detection.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61771154) and the Fundamental Research Funds for the Central Universities (HEUCFG201830, GK2080260148).

This research is also funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

All the authors declare that there is no conflict of interests regarding the publication of this article. We gratefully appreciate the very helpful feedbacks from reviewers.

References

[1] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, S. C. J. J. o. N. de Alvarenga, and C. Applications, "A survey of intrusion detection in Internet of Things," *Journal of Network & Computer Applications,* vol. 84, pp. 25-37, 2017.

[2] M. H. Bhuyan, D. K. Bhattacharyya, J. K. J. I. c. s. Kalita, and tutorials, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials,* vol. 16, no. 1, pp. 303-336, 2014.

[3] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, S. J. J. o. N. Anwar, and C. Applications, "Intrusion response systems: Foundations, design, and challenges," *Journal of Network & Computer Applications,* vol. 62, pp. 53-74, 2016.

[4] N. Ye, Y. Zhang, and C. M. J. I. T. o. R. Borror, "Robustness of the Markov-chain model for cyber-attack detection," *IEEE Transactions on Reliability,* vol. 53, no. 1, pp. 116-123, 2004.

[5] H.-C. Wu and S.-H. S. J. E. S. w. A. Huang, "Neural networks-based detection of stepping-stone intrusion," Expert Systems with Applications, vol. 37, no. 2, pp. 1431-1437, 2010.

[6] I. S. Thaseen, C. A. J. J. o. K. S. U.-C. Kumar, and I. Sciences, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *Journal of King Saud University - Computer and Information Sciences,* vol. 29, no. 4, pp. 462-472, 2017.

[7] R. Theron, R. Magán-Carrión, J. Camacho, and G. M. Fernndez, "Network-wide intrusion detection supported by multivariate analysis and interactive visualization," in *2017 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1-8, 2017.

[8] E. Corchado and Á. J. A. S. C. Herrero, "Neural visualization of network traffic data for intrusion detection," *Applied Soft Computing Journal,* vol. 11, no. 2, pp. 2042-2056, 2011.

[9] G. Nadiammai and M. J. E. I. J. Hemalatha, "Effective approach toward Intrusion Detection System using data mining techniques," *Egyptian Informatics Journal,* vol. 15, no. 1, pp. 37-50, 2014.

[10] T. J. P. o. t. I. Kohonen, "The self-organizing map," *IEEE Proc Icnn,* vol. 78, no. 9, pp. 1464-1480, 1990.

[11] D. Bhamare, T. Salman, M. Samaka, A. Erbad, and R. Jain, "Feasibility of Supervised Machine Learning for Cloud Security," in *2016 International Conference on Information Science and Security (ICISS)*, pp. 1-5, 2016.

[12] Z. Pan, H. Lian, G. Hu, and G. Ni, "An integrated model of intrusion detection based on neural network and expert system," in *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pp. 2 pp.-672, 2005.

[13] A. L. Buczak, E. J. I. C. S. Guven, and Tutorials, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 2, pp. 1153-1176, 2016.

[14] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Lisa*, 1999, vol. 99, no. 1, pp. 229-238.

[15] H. M. Anwer, M. Farouk, and A. Abdel-Hamid, "A framework for efficient network anomaly intrusion detection with features selection," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 157-162, 2018.

[16] A.-H. Muna, N. Moustafa, E. J. J. o. I. S. Sitnikova, and Applications, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications,* vol. 41, pp. 1-11, 2018.

[17] K. Labib and V. R. Vemuri, "An application of principal component analysis to the detection and visualization of computer network attacks," in *Annales des télécommunications*, vol. 61, no. 1-2, pp. 218-234, 2006.

[18] W. Huai-bin, Y. Hong-liang, X. Zhi-Jian, and Y. Zheng, "A clustering algorithm use SOM and K-means in intrusion detection," in *2010 International Conference on E-Business and E-Government*, pp. 1281-1284, 2010.

[19] I. Manzoor and N. Kumar, "A feature reduced intrusion detection system using ANN classifier," *Expert Systems with Applications,* vol. 88, pp. 249-257, 2017.

[20] A. Mabayoje Modinat, G. Akintola Abimbola, O. Balogun Abdullateef, and A. J. I. J. o. C. A. Opeyemi, "Gain Ratio and Decision Tree Classifier for Intrusion Detection," *International Journal of Computer Applications,* vol. 126, no. 11, pp. 975-8887.

[21] N. Moustafa and J. J. a. p. a. Slay, "A hybrid feature selection for network intrusion detection systems: Central points," 2017.

[22] F. T. Liu, K. M. Ting, and Z.-H. J. A. T. o. K. D. f. D. Zhou, "Isolation-based anomaly detection," *Acm Transactions on Knowledge Discovery from Data,* vol. 6, no. 1, pp. 1-39, 2012.

[23] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1-6, 2009.

[24] N. Moustafa and J. J. I. S. J. A. G. P. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal A Global Perspective,* vol. 25, no. 1-3, pp. 18-31, 2016.

[25] S. M. H. Bamakan, H. Wang, and Y. J. K.-B. S. Shi, "Ramp loss K-support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem," *Knowledge-Based Systems,* vol. 126, pp. 113-126, 2017.

[26] R. S. M. Carrasco, M.-A. J. C. Sicilia, and Security, "Unsupervised intrusion detection through skip-gram models of network behavior," *Computers & Security,* vol. 78, pp. 187-197, 2018.