International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 1, February 2021, pp. 900~908
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i1.pp900-908

A hybrid method of genetic algorithm and support vector machine for intrusion detection

Mushtaq Talb Tally[1], Haleh Amintoosi[2]
[1]Ministry of Education Directorate of Education in Babil, Iraq
[2]Computer Engineering Department, Faculty of Engineering, Ferdowsi University of Mashhad, Iran

Article Info

Article history:

Received Jan 2, 2020
Revised Aug 3, 2020
Accepted Aug 17, 2020

Keywords:

Feature selection
Genetic algorithm
Intrusion detection
Support vector machine

ABSTRACT

With the development of web applications nowadays, intrusions represent a crucial aspect in terms of violating the security policies. Intrusions can be defined as a specific change in the normal behavior of the network operations that intended to violate the security policies of a particular network and affect its performance. Recently, several researchers have examined the capabilities of machine learning techniques in terms of detecting intrusions. One of the important issues behind using the machine learning techniques lies on employing proper set of features. Since the literature has shown diversity of feature types, there is a vital demand to apply a feature selection approach in order to identify the most appropriate features for intrusion detection. This study aims to propose a hybrid method of genetic algorithm and support vector machine. GA has been as a feature selection in order to select the best features, while SVM has been used as a classification method to categorize the behavior into normal and intrusion based on the selected features from GA. A benchmark dataset of intrusions (NSS-KDD) has been in the experiment. In addition, the proposed method has been compared with the traditional SVM. Results showed that GA has significantly improved the SVM classification by achieving 0.927 of f-measure.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Mushtaq Talb Tally,
Ministry of Education,
Babylon, Iraq.
Email: mushtaqtalb@gmail.com

1. INTRODUCTION

The last two decades have witnessed a dramatic expansion of the internet applications in which different technologies have been emerged regarding the field of computer network [1, 2]. In this manner, both local area network (LAN) and wide area network (WAN) have played an essential role in terms of several domain of interests such as financial, medical, industry and security that made the need of computer network is so imperative for different businesses [3-5]. However, such growth of using computer networks has contributed toward the emergence of new and unseen abuse activities. Regardless of hacking activities, tremendous business networks are significantly vulnerable for malicious acts such as the worms, Trojan and viruses. Regarding to the importance of computer network nowadays and our ever growing dependency on them has made the security manner is so crucial and listed in the top of priorities for many domain of interests [6-8]. One of the major concerns that has been tackled in the information security community recently is the intrusion detection systems (IDS) [9-11].

Intrusion detection is the task of observing, identifying and detecting the operations that would make the network vulnerable to be violated in terms of the security policies [12]. Many researchers have

addressed the problem of detecting cyber-based attacks on computer networks. Denning [13] has claimed that the key characteristic behind any system that intended to detect intrusion lies on the ability to monitor and diagnose the network records searching for abnormal behaviors related to the system usage. Corporations are usually implementing standard authentication metrics that formulate different level of access in which the authorized user is allowed to access particular level. However, this method does not provide an absolute guarantee regarding the prevention of intrusions. Several incidents have been occurred for large corporations such as Yahoo and Amazon would emphasize the insufficient impact of such method. The intruders are usually being attackers who aim to damage or at least obstruct the network traffic and affect its performance using several kinds of attacks. The process of intrusion can be defined as a significant change from the normal behavior to a suspicious behavior occurred in the system intended to affect the security of the network and harm the performance (paper). Such change will significantly impact the confidentiality, integrity and the availability of the network resources [14].

Several approaches have been proposed for the task of intrusion detection, such efforts were mainly relying on machine learning techniques. However, determining the appropriate learning paradigm is a challenging task. Some authors have utilized the supervised learning [15]. Other researchers have used the unsupervised learning [16]. Both learning paradigms have their own advantages and disadvantages. For instance, one of the shortcomings of supervised learning is the need for labelled instances, but it has the advantage to achieve better accuracy to classify similar examples. On the other hand, unsupervised learning techniques deal with the learning tasks with unlabeled or untagged data. Clustering is the most popular unsupervised learning technique. In clustering, the learning algorithm finds similarities among instances to build the clusters (i.e. group of instances). Instances that belong to the same cluster are assumed to having similar characteristics or properties and then are assembled into the same class. The disadvantage of unsupervised learning is the manually assignment of cluster numbers, which results in low accuracy in predictions. However, it has the advantage of detecting new examples better than supervised learning techniques, and considered to be more robust in IDSs.

Therefore, some authors have tended to utilize the semi-supervised learning paradigm in order to combine the advantages of the two learning paradigms [12]. From one hand, semi-supervised has the ability to deal with unlabeled data, and on the other hand, it can simulate the advantage of supervised learning by achieving better accuracy regarding the process of classifying similar examples. Nevertheless, there is still a drawback regarding the features extracted in the process of classification. In fact, features play a crucial role in terms of improving the classification accuracy. In particular, the domain of classifying intrusions yields tremendous amount of features that have been used in the literature. These features such as duration, protocol type, service type, source size, destination size and others. This would lead to high dimensionality of feature space.

- Problem formulation

In order to illustrate the problem mathematically, let the data that contains the network traffic connections represented as $C = \{c_1, c_2, c_3, \dots, c_n\}$. In this manner, each connection would have multiple associated features as $f = \{f_1, f_2, f_3, \dots, f_m\}$. Here, it is necessary to determine the most appropriate features that would be correspond specific class label which belongs to $C = \{C_1, C_2\}$ where C_1 is the legitimate connection and C_2 is the intrusion, the following formula would be be considered:

$$\sum_{i=0}^{n} C_i \, f_i \tag{1}$$

Apparently, the problem tends to be an optimization problem in which the number of possibilities for the solutions is relatively high. Therefore, the need to use the meta-heuristic approach becomes imperative in order to identify the best solutions.

2. RELATED WORK

Generally, there are tremendous approaches have been proposed for the problem of detecting intrusions. The earliest research efforts in terms of intrusion detection were used specification-based method. For instance, Tseng *et al.* [17] have presented a specification-based method for detecting intrusion and attacks within the ad-hoc on demand distance vector (AODV) routing protocol. In their work, the behaviors of the AODV requests and replays were being analyzed and compared with the correct behavior of critical objects. This is due to the fact that the intrusions are commonly leading the object to act incorrectly. Therefore, there will be no need for knowledge-based information to describe the intrusions. The proposed method has the ability to effectively detect most of the serious AODV routing attacks.

Recently, there are many researchers who examined the capabilities of machine learning techniques regarding the intrusion detection. Some of those authors have used supervised machine learning, other

researchers used the semi-supervised and the rest of them used the unsupervised learning. For instance, Peddabachigari *et al*. [18] have proposed a supervised machine learning approach for identifying intrusions using different algorithms. The authors firstly used both of decision tree (DT) and support vector machine (SVM) classifiers using the benchmark dataset of KDDCUP'99. Consequentially, the authors have used an ensemble approach as a hybrid of DT and SVM. The proposed hybrid method has outperformed both DT and SVM in terms of the classification accuracy.

Shreya and Jigyasu [19] have proposed a supervised machine learning technique method in order to classify the intrusions. The authors have used the naïve bayes (NB) and k-nearest neighbor (KNN) to do such purpose. The authors have considered different metrics for the evaluation. First, they used the benchmark dataset of KDDCUP'99 in order to compare with other works that have used the same dataset. Based on the accuracy of classification, time consumption and memory consumption, the proposed method has demonstrated superior performance. Although the time consumed was relatively similar to the related work however, the memory consumption and the accuracy showed remarkable enhancement.

Lin *et al*., [16] have proposed a hybrid method of supervised and unsupervised learning approaches for detecting intrusions. The unsupervised approach aims to utilize a cluster center approach in order to categorize the data into similar groups. This can be performed by initializing centroids and calculate the distance between every data point with the centroids. Then, each data point will be merged to its corresponding centroid in order to form clusters. Consequentially, this new and one-dimensional distance based feature is used to represent each data sample for intrusion detection by the supervised approach of k-nearest neighbor (KNN) classifier. The authors have used the KDDCUP'99 benchmark to evaluate their proposed method. Results showed that the proposed approach is outperforming the conventional KNN.

Similarly, Tahir *et al*., [20] have proposed similar hybrid method of supervised and unsupervised learning for improving the classification accuracy of intrusion detection. The authors have used the k-means clustering technique in order to group the data into similar clusters. Then, the support vector machine classifier has been used in order to classify the intrusions and attacks. The data used in this study is NSL-KDD benchmark dataset.

Puri and Sharma [15] have proposed a hybrid method of support vector machine (SVM) classifier and regression tree (RT) algorithm in order to detect intrusions. The authors have used the benchmark dataset of KDDCUP'99 in which the regression tree algorithm is designed for generating tree rules which will be used for classifying the attacks using SVM. Ashfaq *et al*., [12] have addressed the problem of acquiring a labeled samples of intrusion behaviors. By exploiting the capabilities of the semi-supervised learning technique, in which the labeled data is not compulsory, the authors have utilized a semi-supervised method of single hidden layer feed-forward neural network to train it on the intrusion behaviors. Using labeled dataset such as KDDCUP'99 and NSL-KDD, the authors have demonstrated the efficacy of the proposed neural network in terms of classifying new and unlabeled data.

3. PROPOSED METHOD

This section aims to describe the application of proposed hybrid GA and SVM that intended to detect the intrusions. This requires identifying a benchmark dataset that contains intrusions and normal behavior in order to enable the process of feature extraction and selection with the classification process as shown in Figure 1.

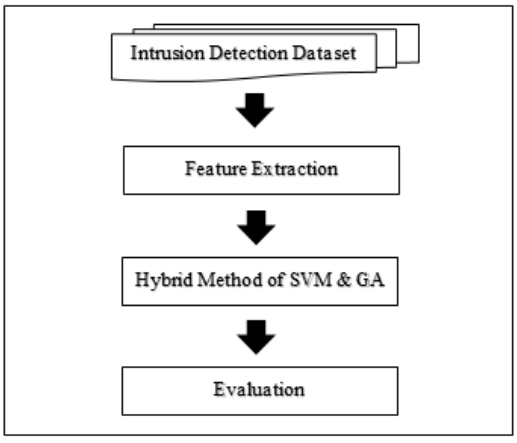

Figure 1. Framework of the proposed method

As shown in Figure 1, the proposed method begins with preparing the dataset that contains the intrusions. Consequentially, a feature extraction process will take a place in order to utilize different type of features. Then, GA will be used to search for the best solutions or in other word identifying the most appropriate features. Then, SVM will accommodate the classification task in which the behaviors will be categorized into normal and intrusion. However, the next sub-sections will tackle each phase separately.

3.1. IDS dataset

With the exponential expansion of computer networks usage and the numerous applications that are mainly depending on it, securing such networks has become a crucial task. Different threats and security vulnerabilities that are facing the networks have been addressed by several research studies. In the literature, the risks behind these threats have been discussed in which it may cause privacy violation and cost consumption. For this purpose, the intrusion detection systems has caught many researchers' attentions in which the behavior of the network operations are being analyzed in order to identify the anomalies. In fact, IDS monitor both the anomaly and normal behavior in order to generate a module that would characterize t he features of both behavior. However, some authors have argue that analyzing the anomaly behaviors rather than the normal behavior would improve the detection performance [21]. In this manner, another problem has been arisen which lies on providing a data for the anomaly behavior. This is due to the new trend in the field of IDS which can be represented by the utilization of machine learning technique.

Supervised learning paradigm works by train a model using a predefined set of examples which is called training data [16]. Such examples contain the network features with a label such as Suspicious and Normal. The model here can generate statistical rules in order to discriminate the situations that occurred with intrusions. These rules will be used to help the model for classifying new or testing data [19]. However, sometimes it is difficult to acquire a labelled example due to the challenging issue of benchmark availability. Meanwhile, the manual labelling for a huge amount of the data seems to be tedious and time consuming [12].

For this purpose, a benchmark dataset has been created which is called KDDCUP'99 [22]. KDD'99 has been widely used for the sake of machine learning in terms of identifying anomaly activities. Such data has been created using the DARPA'98 IDS evaluation program [23]. DARPA'98 is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labelled as either normal or an attack, with exactly one specific attack type. The types of attack included in the KDD'98 dataset can be listed as follows:

- Denial of service (DoS)
- User to root (U2R)
- Remote to local (R2L)
- Probing

However, Tavallaee *et al*., [24] have criticized such data and claimed that it suffers of multiple drawbacks. First, the KDD dataset suffers of the high degree of duplicated records in which nearly 75% of its records are being duplicated. In this manner, such redundant would significantly contribute toward making the learning paradigm rely on the frequent patterns in which the rare patterns would be ignored. Obviously, this will negatively affect the performance of detection. Second issue lies on the KDD dataset is the relatively small number of test set instances in which any classifier would correctly classify the test instances and having a minimum classification rate of 86%. Such results reveals the difficulty of comparing multiple IDSs due to the similar performances that would be resulted. Therefore, Tavallaee *et al*., [24] have proposed a new dataset called NSS-KDD in order to solve the two latter problems by providing more cases for the anomaly in the testing portion, as well as, providing a reasonable number of record in both training and testing sets. Table 1 depicts the details of the new dataset NSS-KDD.

Table 1. NSS-KDD Dataset details

Training set			
	Number of instances	Number of unique instances	Reduction percentage
Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%
Testing			
	Number of instances	Number of unique instances	Reduction percentage
Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

3.2. Feature extraction

Features represents an imperative role in the context of supervised machine learning where the important features would significantly enhance the effectiveness of the classification, vice versa; the less-important features would negatively impact the effectiveness of the classification. The features contained in the KDD-NSS can be represented into three main features; Basic features, Traffic features and Content features. These features would be tackled as follows:

Basic Features: Basic features concentrate on the attributes that are related to the TCP/IP connection in which the features such as the protocol, type of service and type of flag are being considered.

Traffic Features: This type of features concentrates on the window interval of the connections. There are multiple features that are being included in this category which can be illustrated as follows:

a. Time-based Features: This type of features aim at analyzing the number of connection in respect to a time duration and it contains two aspects including same-host-features and same-service Features. In the first aspect, the number of connections that have the same destination host will be computed based on duration (i.e. 2 seconds). In the second aspect, the number of connections that have the same service will be computed based on the duration.
b. Connection-based Features: In the context of time-based traffic features, there are many slow probing attacks that scan the hosts (or ports) using a much larger time interval than 2 seconds, for instance, one in every minute. Hence, such attacks do not generate intrusion patterns with a time window of 2 seconds. To overcome this issue, the connection-based features have been introduced by the NSS-KDD dataset in which the number of connection would be computed based on a window of 100 connections. For the same-host-features, the number of connections that have the same destination host will be computed based on a window of 100 connections. Whereas for the same-service-features, the number of connections that have the same service will be computed based on a window of 100 connections.

Content-based features: The previous type of feature (i.e. traffic) can fit some attacks such as DoS and Probing in which a tremendous amount of connections within short time are being produced. However, for both R2L and U2R attacks this would not be the case where these attacks are not producing intrusion patterns with a time window. Therefore, the new version NSS-KDD has considered such problem by adding a new type of feature that has the ability to analyze the R2L and U2R. This type of feature is called content feature and it concentrates on the login features such as the number of failed logins.

3.3. Hybrid of SVM and GA

Basically, SVM is one of the supervised machine learning techniques which aims to turn the data space into a vector space based on the features' values [15]. Then, a separation task will be performed using a hyper-plane which is a separator that aims to divide the data into multiple portions based on the class labels. Figure 2 depicts the separation task by the hyper-plane.

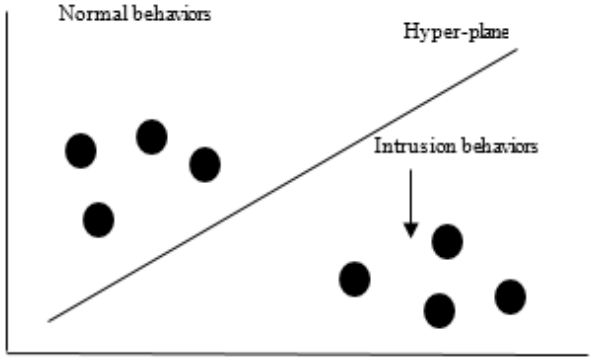

Figure 2. Separating the data space using the hyper-plane

As shown in Figure 2, the data space consists of the network traffic behaviors, while the hyper-plane is dividing the data into two groups or class labels including 'Normal behaviors' and 'Intrusion behaviors'. In fact, identifying the most robust hyper-plane, that has the ability to divide the data accurately, is a challenging task [25]. This is due to the less-accurate adjustment of the hyper-plane would lead to divide the data incorrectly where some normal behavior would be classified as intrusions or vice versa.

The task of identifying the most appropriate hyper-plane is mainly depends on the features used to establish the vectorization of the data space. Therefore, this study utilizes a meta-heuristic approach (i.e. genetic algorithm) in order to determine the best set of features that will lead to the most robust hyper-plane. GA is one of the evolutionary algorithms that have been widely used for optimization problems where the feasible solution is required to be attained [26]. It works by generating an initial population of features, then assessing such population based on the fitness function. The best features from the initial population will be selected [27]. Consequentially, a reproduction operator is being performed to combine the best features. This study utilizes the crossover operation to conduct such task. Figure 3 depicts the workflow of the hybrid method of GA and SVM.

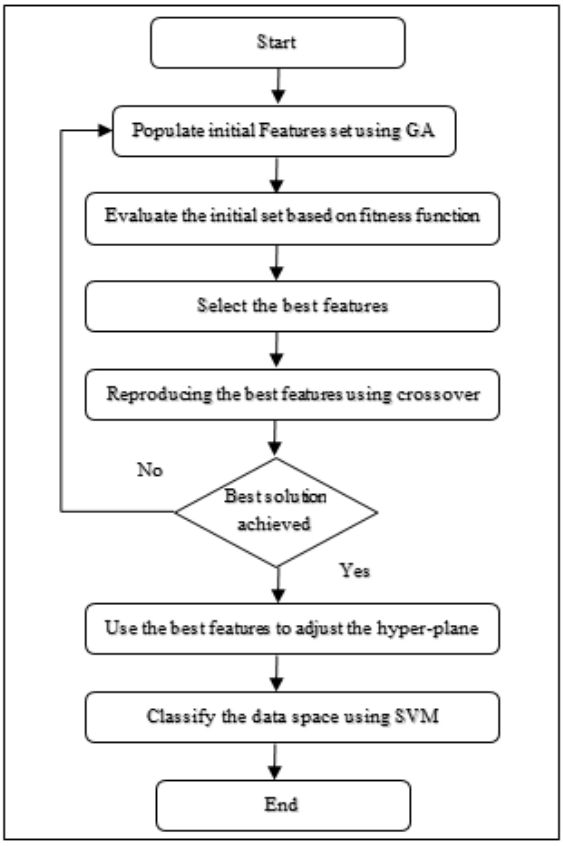

Figure 3. Workflow of the hybrid method

3.4. Evaluation

The evaluation of the proposed hybrid method will be based on the common information retrieval metrics precision, recall and f-measure [28-30]. Such measures can be computed using the contingency table as shown in Table 2.

Table 2. Contingency table

Actual \ Predicted	Legitimate connection	Intrusion
Legitimate connection	*True Negative (TN)*	*False Positive (FP)*
Intrusion	*False Negative (FN)*	*True Positive (TP)*

False Negative (FN) : is the number of correctly un-predicted connections.
False Positive (FP) : is the number of incorrectly predicted connections.
True Negative (TN) : is the number of actual intrusion connections that have not been predicted.
True Positive (TP) : is the number of correctly predicted connections.

In this manner, the precision, recall and f-measure can be computed based on the following equations.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+TN} \quad (3)$$

$$F-measure = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (4)$$

4. RESULTS & DISCUSSIONS

In order to evaluate the proposed method, SVM has been applied twice; first without GA and second with GA. In addition, a comparison will be made between the above two applications using T-test evaluation method. T-test is a statistical significance indicates whether or not the difference between two groups' averages most likely reflects a "real" difference in the population from which the groups were sampled. Table 3 represents the results of the two applications of SVM.

Table 3. Results of SVM and SVM with GA

No.	Class	SVM (F-measure)	SVM & GA (F-measure)
1.	apache2	0.995984	1
2.	back	1	1
3.	buffer_overflow	0.754557	0.75456
4.	ftp_write	0.60042	0.60042
5.	guess_passwd	0.99447	0.99447
6.	httptunnel	0.996488	0.99649
7.	imap	1	1
8.	ipsweep	0.965311	0.96531
9.	land	1	1
10.	loadmodule	0.666667	1
11.	mailbomb	0.991427	0.99143
12.	mscan	0.982188	0.98219
13.	multihop	0.922994	0.92299
14.	named	0.849914	0.84991
15.	neptune	0.993964	0.99396
16.	nmap	0.759554	0.75955
17.	normal	0.940117	0.94012
18.	perl	1	1
19.	phf	0.80024	0.80024
20.	pod	0.836533	0.83653
21.	portsweep	0.838592	0.92724
22.	processtable	0.9995	1
23.	ps	0.908893	0.90889
24.	rootkit	0.866856	0.86686
25.	saint	0.745988	0.8766
26.	satan	0.778106	0.77811
27.	sendmail	0.777506	0.9605
28.	smurf	0.988878	0.98888
29.	snmpgetattack	0.64556	0.64556
30.	snmpguess	0.574483	0.95013
31.	spy	0.626374	1
32.	sqlattack	1	1
33.	teardrop	0.286525	0.50362
34.	udpstorm	1	1
35.	warezclient	0.996488	1
36.	warezmaster	0.97855	0.97855
37.	worm	1	1
38.	xlock	0.947368	1
39.	xsnoop	1	1
40.	xterm	1	1
	Average	0.875262	0.92716

As shown in Table 3, the proposed genetic algorithm has significantly improved the SVM classification in terms of identifying intrusions. This has been demonstrated via the average f-measure of the proposed SVM with GA which was 0.927 compared to 0.875 the average f-measure of SVM without GA. In addition, a test called T-test has been applied on the f-measure for all the class labels for both applications. The results of such test was less than 0.05 which implied that the GA has significantly improved the classification performance.

On the other hand, comparing the proposed method's results with the state of the art such as Tahir *et al*., [20] who proposed an intrusion detection classification using SVM and obtained a 0.856 of f-measure, it is obvious that the proposed method is outperforming. In addition, a study by Puri and Sharma [15] which also proposed an SVM classifier to detection intrusions, has obtained an f-measure of 0.883. Apparently, the proposed method shows competitive performance against the state of the art.

5. CONCLUSION

This paper has proposed a hybrid method of genetic algorithm and support vector machine for the task of intrusion detection. The proposed method has been assessed using a benchmark dataset NSS-KDD. Moreover, the proposed method has been compared with the conventional SVM. Results showed that the proposed method has outperformed the traditional SVM. This implies the feasibility of using GA in terms of identifying the best features. For future researches, addressing different meta-heuristic approaches such as Particle Swarm Optimization or Ant Colony would be an opportunity to examine the capability of GA.

REFERENCES

[1] P. E. Van Thuan Do, B. Feng, and T. van Do, "Detection of DNS Tunneling in Mobile Networks Using Machine Learning," *International Conference on Information Science and Applications*, vol. 424, pp. 221-230, 2017.

[2] M. Sammour, B. Hussin, M. F. I. Othman, M. Doheir, B. AlShaikhdeeb, and M. S. Talib, "DNS Tunneling: a Review on Features," *International Journal of Engineering and Technology*, vol. 7, no. 20, pp. 1-5, 2018.

[3] A. Riyad, M. Ahmed, and R. Khan, "An adaptive distributed intrusion detection system architecture using multi agents," *International Journal of Electrical & Computer Engineering (IJECE)*, vol. 9, no. 6, pp. 4951-4960, 2019.

[4] C. Kiennert, Z. Ismail, H. Debar, and J. Leneutre, "A survey on game-theoretic approaches for intrusion detection and response optimization," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, 2019.

[5] A. Panigrahi, and M. R. Patra, "Intrusion Detection using Rule Learning based Classifiers," *International Journal of Applied Engineering Research*, vol. 14, no. 17, pp. 3616-3621, 2019.

[6] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained IoT devices," *Mechanical Systems and Signal Processing*, vol. 136, pp. 1-13, 2020.

[7] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, 2020.

[8] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Systems with Applications*, vol. 148, pp. 1-14, 2020.

[9] S. Bhattacharya, R. Kaluri, S. Singh, M. Alazab, and U. Tariq, "A Novel PCA-Firefly based XGBoost classification model for Intrusion Detection in Networks using GPU," *Electronics*, vol. 9, no. 2, pp. 1-16, 2020.

[10] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, and L. Cui, "Robust detection for network intrusion of industrial IoT based on multi-CNN fusion," *Measurement*, vol. 154, pp. 1-10, 2020.

[11] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51-62, 2020.

[12] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484-497, 2017.

[13] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, vol. SE-13 no. 2, pp. 222-232, 1987.

[14] E. Hernández-Pereira, J. A. Suárez-Romero, O. Fontenla-Romero, and A. Alonso-Betanzos, "Conversion methods for symbolic features: A comparison applied to an intrusion detection problem," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10612-10617, 2009.

[15] A. Puri, and N. Sharma, "A novel technique for intrusion detection system for network security using hybrid svm-cart," *International Journal of Engineering Development and Research*, vol. 5, no. 2, pp. 155-161, 2017.

[16] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13-21, 2015.

[17] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, "A specification-based intrusion detection system for AODV," in *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pp. 125-134, 2003.

[18] S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems," *Journal of network and computer applications*, vol. 30, no. 1, pp. 114-132, 2007.

[19] S. Dubey, and J. Dubey, "KBB: A hybrid method for intrusion detection," *2015 International Conference on Computer, Communication and Control (IC4)*, Indore, pp. 1-6, 2015.

[20] H. Mohamad Tahir, W. Hasan, A. Md Said, N. H. Zakaria, N. Katuk, N. F. Kabir, M. H. Omar, O. Ghazali, and N. I. Yahya, "Hybrid machine learning technique for intrusion detection system," *International Conference on Computing and Informatic*s, Istanbul, Turke, pp. 464-472, 2015.
[21] A. Patcha, and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer networks*, vol. 51, no. 12, pp. 3448-3470, 2007.
[22] K. Cup, "Dataset," vol. 72, 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[23] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, and R. K. Cunningham, "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation," *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, Hilton Head, SC, USA, vol. 2, pp. 12-26, 2000.
[24] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Application*s, Ottawa, ON, pp. 1-6, 2009.
[25] L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, "An enhanced support vector machine classification framework by using Euclidean distance function for text document categorization," *Applied Intelligence*, vol. 37, no. 1, pp. 80-99, 2012.
[26] A. Al Malki, M. M. Rizk, M. El-Shorbagy, and A. Mousa, "Hybrid Genetic Algorithm with K-Means for Clustering Problems," *Open Journal of Optimization*, vol. 5, no. 2, pp. 71-83, 2016.
[27] A. S. Desai, and D. Gaikwad, "Real time hybrid intrusion detection system using signature matching algorithm and fuzzy-GA," *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, Pune, pp. 291-294, 2016.
[28] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, 2018.
[29] S. N. Mighan, and M. Kahani, "Deep Learning Based Latent Feature Extraction for Intrusion Detection," *Electrical Engineering (ICEE), Iranian Conference on*, Mashhad, pp. 1511-1516, 2018.
[30] C.-R. Wang, R.-F. Xu, S.-J. Lee, and C.-H. Lee, "Network intrusion detection using equality constrained-optimization-based extreme learning machines," *Knowledge-Based Systems*, vol. 147, pp. 68-80, 2018.