Hindawi
Complexity
Volume 2022, Article ID 3999039, 14 pages
https://doi.org/10.1155/2022/3999039

Review Article

Intrusion Detection Systems in Cloud Computing Paradigm: Analysis and Overview

Pooja Rana,[1] Isha Batra,[1] Arun Malik,[1] Agbotiname Lucky Imoize ,[2,3] Yongsung Kim ,[4] Subhendu Kumar Pani,[5] Nitin Goyal,[6] Arun Kumar,[7] and Seungmin Rho [8]

[1] *Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India*
[2] *Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Lagos, Akoka, Lagos 100213, Nigeria*
[3] *Department of Electrical Engineering and Information Technology, Institute of Digital Communication, Ruhr University, Bochum 44801, Germany*
[4] *Department of Technology Education, Chungnam National University, Daejeon 34134, Republic of Korea*
[5] *Krupajal Engineering College, BPUT, Rourkela 751002, Odisha, India*
[6] *Computer Science Engineering Department, Shri Vishwakarma Skill University, Palwal 121102, Haryana, India*
[7] *Panipat Institute of Engineering and Technology, Panipat, Haryana, India*
[8] *Department of Industrial Security, Chung-Ang University, Seoul 06974, Republic of Korea*

Correspondence should be addressed to Yongsung Kim; kys1001@cnu.ac.kr

Received 22 March 2022; Revised 10 May 2022; Accepted 19 May 2022; Published 20 June 2022

Academic Editor: Peican Zhu

Cloud computing paradigm is growing rapidly, and it allows users to get services via the Internet as pay-per-use and it is convenient for developing, deploying, and accessing mobile applications. Currently, security is a requisite concern owning to the open and distributed nature of the cloud. Copious amounts of data are responsible for alluring hackers. Thus, developing efficacious IDS is an imperative task. This article analyzed four intrusion detection systems for the detection of attacks. Two standard benchmark datasets, namely, NSL-KDD and UNSW-NB15, were used for the simulations. Additionally, this study highlights the proliferating challenges for the security of sensitive user data and gives useful recommendations to address the identified issues. Finally, the projected results show that the hybridization method with support vector machine classifier outperforms the existing techniques in the case of the datasets investigated.

1. Introduction

Cloud computing is defined as an Internet-based computing platform in which virtually shared servers provide software, platform, infrastructure, policies, and other functions [1]. It is visualized as a demand from its users to reduce overall cost and complexities. It is gaining popularity due to various advantages of on-demand service provision, flexible resource allocation, higher fault tolerance, and higher scalability. Various cloud service providers (CSPs), including Google, Amazon, and Microsoft, use virtualization technologies with self-service capabilities. Virtualization is the first need of cloud computing [2]. A huge increase in IT technologies leads to daily data increases [3]. Attackers have taken benefit of cloud computing as copious amounts of data are produced by it greater than 665 Gb/s [4]. Huge data generated by the cloud have become its biggest problem as it has come on the target of attackers [5]. Hackers are alluring towards the cloud due to its open and distributed nature and the amount of traffic produced [6]. Attackers can interrupt the services of the users, misuse the sensitive information, and misuse the services and resources given by the CSP. An intrusion can be an attack that can misuse the private or sensitive information of the users, or it can consume the resources such as CPU, bandwidth, and storage. Traditional methods for providing security like firewalls are not sufficient. But there is a need for a proper system that can provide security to the users. An intrusion detection system (IDS) can detect or find attacks in the network by analyzing the data of the network. There are mainly two categories of IDS based on the deployment strategies: host-based IDS and network-based IDS [7, 8]. Host-based IDS analyzes attacks by monitoring the host system only, whereas network-based IDS analyzes the whole network. Every node in the cloud has personal IDS and storage in the case of host-based IDS [9].

Host-based IDS is proposed based on statistics and probability theory [10]. SNORT-based detection is performed in Eucalyptus Cloud in Ref. [11]. Network-based IDS proposed in Ref. [12] has intrusion detection system management unit and intrusion detection system sensor. The distributed intrusion detection system is also growing with time as it merges the characteristics of both the abovementioned IDSs [13]. Two more types of IDS are based on the detection mechanism: signature-based IDS and anomaly-based IDS. Signature-based IDS analyzes the attacks in the network by comparing the signatures of attacks stored in the database. Anomaly-based IDS can detect attacks in the network by analyzing the dynamic activities in the network. A profile is created by observing the activities of the users, applications, and users during a particular period in anomaly-based IDS [14, 15].

Numerous researchers have used data mining and machine learning approaches [16]. Zero-day attacks are the biggest concerns for the cloud [17]. Classifiers based on machine learning are usually used to classify attack packets and normal packets [18]. Another emerging technique is the mining rule association technique [19]. Artificial neural networks are mostly used due to their ability to work on the incomplete dataset [20]. Some researchers have found the importance of machine learning algorithms for intrusion detection in the cloud due to the scalability and elasticity features of the cloud computing paradigm [21–24]. Different optimization algorithms such as genetic algorithm [25], particle swarm optimization [26], harmony search [27], and artificial bee colony [28] are also used with various classifiers for categorizing attack packets and normal packets of the network.

The main contributions of the article are given as follows:

(i) Discerned the methodologies followed by different intrusion detection systems related to the cloud computing environment. Also discerned which attacks they have considered for their research work.

(ii) Analogized four existing intrusion detection systems for the detection of attacks.

(iii) Analogized various attacks of two different standard benchmark datasets: NSL-KDD dataset and UNSWB-15 dataset.

(iv) Epitomized the study of various existing intrusion detection systems of the cloud computing environment. Represented our research work and discerned which methodology outperformed our results and comparative analysis.

(v) Exemplified the remaining challenges in cloud security and suggested possible recommendations for addressing the challenges.

The structure of the remaining article is as follows: Section 2 reviews the literature review. Section 3 describes the proposed methodology. Section 4 presents the experiments and comparative analysis. Section 5 represents the future scopes and recommendations for the cloud computing environment. Conclusions are presented in Section 6.

2. Literature Review

The literature review section of the article is reviewing various good journal papers related to the intrusion detection in the cloud computing environment. Literature review is presented in the tabular form. Table 1 is showing the literature review, and also we have suggested the possible future scopes for the reviewed papers.

Additionally, we have compared our survey article with other latest survey papers. Table 2 shows how our survey article differs from other surveys. In table describes the novelty of our survey.

3. Methodology

Our methodology is described in this section of the article. It is implemented in three modules. The modules are preprocessing classification and evaluation. We have used four existing methodologies for the detection of attacks. Out of four methodologies, three methodologies are applied to the cloud computing environment, and the last methodology is applied to general network, which makes our comparison more strong. We have chosen these four methodologies for comparison as they are including the popular classifiers for intrusion detection. We have also chosen one methodology, which is using the optimization concept. So, these four methodologies' comparison will give a good comparison outcome.

3.1. Dataset.

We have used two standard benchmark datasets for the comparative analysis. We have used the NSL-KDD dataset [52] and the UNSW-NB15 dataset [53].

3.1.1. UNSW-NB15 Dataset.

It was created to overcome the drawbacks of the NSL-KDD dataset. This dataset contains low footprint attack characteristics and some traffic schemes, and there is no discrepancy between the distributions of datasets. This dataset contains 49 features. The last two features represent the category and label (0 for normal and 1 for attack records). Figure 1 shows the pie chart of the UNSW-NB15 dataset distribution of various classes.

3.1.2. NSL-KDD Dataset.

It is a publicly available dataset refining the KDD-CUP 1999 dataset. This dataset does not contain redundant records in the training and testing dataset. There is no requirement for creating subsets of the dataset for experimentation purposes. Figure 2 shows the pie chart of NSL-KDD dataset distribution.

Table 1: Literature review.

Ref. No	Year	Research paper/ review paper	Attacks detected	Dataset used	Methodology used	Suggestions
[29]	2012	Review paper	DoS, attacks targeting shared memory, phishing attack, malicious insider attack, cross VM side channel attack	`Dataset generated	Surveyed security threats affecting cloud computing and solutions. A model is developed for detecting threats, generating alerts, and producing information on the type of attack.	In future, try to optimize various classifiers such as naïve Bayes, MLP, decision tree, and PART classifiers. Real-time dataset can be considered for future work and try to reduce the detection time.
[30]	2013	Research paper	Malware	Dataset generated	Bayesian quantum particle swarm optimization is used for the detection of malware.	More types of attacks should be considered in future.
[31]	2014	Research paper	DDoS	DARPA KDD cup 1999	Fuzzy logic is used for the detection of the attacks.	The authors focus on alert generations and prevention of various attacks.
[32]	2014	Research paper	Bots	Dataset generated	Fuzzy recognition pattern is used.	Make intrusion detection system strong by taking more attacks in future.
[33]	2015	Research paper	DoS, U2R, R2L, probe	DARPA KDD cup 1999 dataset	FCM along with BPNN is hybridized for developing a novel intrusion detection system.	The authors should take better datasets as the NSL-KDD dataset is the improved version of DARPA KDD-CUP 1999 dataset.
[34]	2016	Research paper	DoS, U2R, R2L, Probe	DARPA KDD cup 1999 dataset	A novel classifier which is two-stage classifier is made by hybridizing SVM and ANN.	Authors should take better datasets as NSL-KDD dataset is improved version of DARPA KDD-CUP 1999 dataset.
[35]	2018	Research paper	DDoS	Dataset generated	Multi-agent system is used to detect DDoS attacks.	Authors can focus on other latest attacks also.
[36]	2018	Research paper	DDoS	CICIDS dataset	Time-sliding window algorithm is used with the naïve Bayes classifier.	Authors should consider more performance metrics for evaluation purpose.
[37]	2019	Research paper	DoS, U2R, R2L, probe, sql injection, web attack, etc.	CICIDS 2017 dataset, NSL-KDD 2015 dataset, and CIDDS-001 Dataset	Improved genetic algorithm is used with simulated annealing technique.	More complex datasets can be considered in future.
[38]	2019	Research paper	Analysis, backdoor, generic, DDoS, sql injection, Brute-force, etc.	UNSW-NB15 dataset, CICIDS 2017 dataset	Deep neural extended binary bat algorithm is hybridized with random forest classifier.	Realistic environment can be used for experimentation in the future.
[39]	2019	Research paper	DoS, U2R, R2L, probe	NSl-KDD dataset	Logistic regression is used for selecting the optimal feature set and bagging algorithm is used for the classification of the attacks.	More complex datasets can be considered in future.
[40]	2019	Research paper	Various attacks	Dataset generated	Virtualbox is used by authors and LUbuntu15 along with Apache web server is used for generating dataset.	Complex dataset can be taken in future for achieving better accuracy
[41]	2019	Research paper	DoS, U2R, R2L, probe	NSL-KDD dataset	CS is used with PSO. The modification made to PSO is that the updated location of the particles is generated by adding old location to the new location. This increases the exploration phase.	Complex dataset can be taken in future for achieving better accuracy.
[42]	2019	Research paper	DoS	Dataset generated	Features are selected based on the scoring algorithm and ranking algorithm. Classification of attacks is made by using the rule-based algorithm.	Temporal constraints can be used for collecting dynamic information related to attacks. Fuzzy rule concept can also be used for increasing accuracy.

Table 1: Continued.

Ref. No	Year	Research paper/ review paper	Attacks detected	Dataset used	Methodology used	Suggestions
[40]	2019	Research paper	Various attacks	The 36 datasets of real attacks collected from 2014 until 2016 from the network traffic blog	C4.5 algorithm and naïve Bayes classifier are compared and naïve Bayes classifier performs better.	More classifiers like ANN and SVM can be compared in future.
[43]	2020	Research paper	DoS, U2R, R2L, probe	NSL-KDD dataset	FCM is hybridized with SVM for detecting attacks.	Optimization of the classification can produce better results.
[44]	2020	Research paper	DoS, U2R, R2L, probe	NSL-KDD dataset	FCM is hybridized with ANN. The SMO algorithm is used for optimizing the ANN.	The hybrid optimization algorithm can be used for obtaining better accuracy.
[45]	2021	Research paper	Analysis, backdoor, generic, DDoS, sql injection, Brute-force, etc.	UNSW-NB15 dataset, CICIDS 2017 dataset, and CICIDS 2019	Decision Jungle classifier is used for the detection of attacks.	The authors can produce hybrid classification technique by using multiple classifiers.

Table 2: Comparison of our survey article with other survey papers.

Ref.	Compared existing intrusion detection systems practically	Datasets taken for comparing existing intrusion detection systems practically	The article focuses on multiple classifiers	Comparison made for every attack of dataset	Comparison made for the whole dataset	Performance metrics used for comparison done practically	Future scopes	Possible recommendations for future scopes
[46]	No	No	Yes	No	No	Not applicable	Yes	No
[47]	No	No	Yes	No	No	Not applicable	Yes	No
[48]	No	No	Yes	No	No	Not applicable	Yes	Yes
[49]	No	No	Focus on bio-inspired techniques only	No	No	Not applicable	Yes	No
[50]	Three existing machine learning techniques are compared practically	KDD-CUP 1999 dataset	Focus on machine learning techniques only	No	Yes	Accuracy	Yes	No
[51]	No	No	Focus on machine learning techniques and deep learning techniques	No	No	Not applicable	Yes	No
This paper	Four existing intrusion detection systems are compared practically	UNSW-NB15 dataset, NSL-KDD dataset	Yes	Yes	Yes	Accuracy, precision, detection rate, F-measure, false alarm rate	Yes	Yes

3.2. Preprocessing. Rough or raw datasets can lead to high false alarms [54]. Datasets used for classification include various attributes, which can be numeric or non-numeric. Symbolic or non-numeric should be converted to the numeric form that easily interprets the classifiers. We have preprocessed the raw datasets and converted the dataset into one form, which is numeric. Like in the NSL-KDD dataset, attribute 41 has no use for classifying the dataset. Hence, we

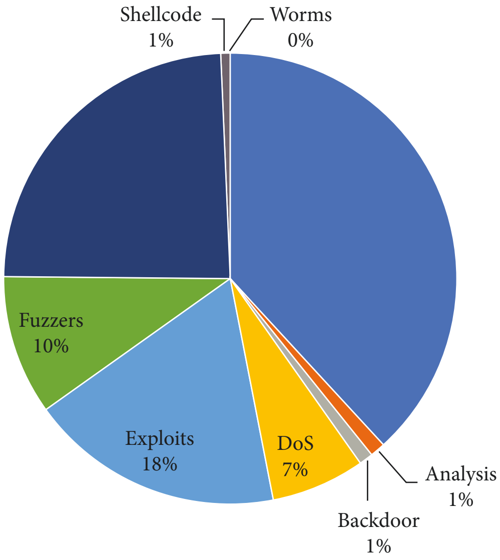

Figure 1: Pie chart of UNSW-NB15 dataset distribution.

have not considered that attribute for the classification of the dataset. Attributes that have no importance for the classification increase the computation time and are excluded from the dataset.

3.3. Classification. Classification of the dataset into normal and attack packets plays an important role in providing security to the cloud computing environment. Classification can be a binary classification or multiclass classification. Binary classification results in two classes. Multiclass classification results in more than two classes. We have performed multiclass classification. For the classification, we have implemented four existing intrusion detection methodologies. The four methodologies are described next.

3.3.1. FCM-ANN. This methodology is implemented in four modules [33]. The flowchart of the methodology is shown in Figure 3.

(1) Preprocessing Module. The raw dataset is preprocessed, and the dataset is converted into a form that is easily analyzed by the classifier.

(2) FCM Module. This module is used for making clusters of the dataset. The membership function used for creating the clusters is represented [33] by the following equation:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} u_{ij}^{m} x_i - c_j^2. \quad (1)$$

where N is the number of elements, K is the number of clusters, M is a real number and, $1 \leq \mathrm{m} \leq \infty$, and U_{ij} is the degree of membership functions of x_i data in the j^{th} cluster.

The output of this module results in creating homogeneity between the cluster and heterogeneity among various clusters.

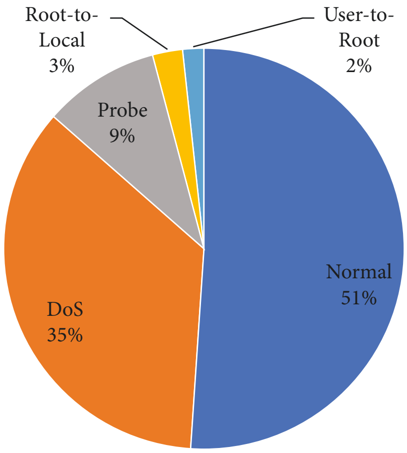

Figure 2: Pie chart of NSL-KDD dataset distribution.

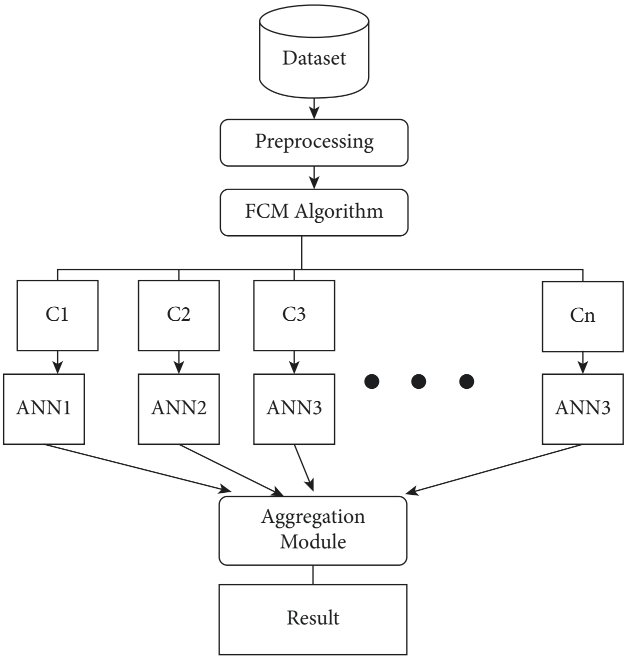

Figure 3: Flowchart of FCM-ANN methodology.

(3) ANN Module. This module is used for classifying the clusters generated by the fuzzy c-means algorithm. Backpropagation algorithm is commonly used for training neural network [55]. In this module, the cluster pattern is learned, and the back propagation algorithm is used to train the feed-forward neural network. A feed-forward neural network has an input layer, an output layer, and numerous hidden layers. The input given to k node (belongs to hidden layer) is ln (k), and it is given [33] by

$$\ln(k) = \theta_k + \sum_{i=1}^{n} x_i w_{ik}. \quad (2)$$

where ln (k) is the input given to k node, k node is belonging to the hidden layer, θ_k is the bias of the hidden layer, x_i is the input given to the i node, i node is belonging to the input layer, and w_{ik} is the weight value between the input layer and hidden layer.

The activation function is the sigmoid function, and it is used for processing the ln (k). It is given [33] by the following equation:

$$f(x) = \frac{1}{1 - e^{x}} - 1. \quad (3)$$

The result of the activation function is $f(\ln(k))$, which is sent to all the neurons of the output layer. It is given [33] by the following equation:

$$y_j = \theta_j + \sum_{k=1}^{m} w_{kj} f(In(k)). \quad (4)$$

where y_j is the output sent to all the neurons j, j node is belonging to the output layer, θ_j is the bias of the output layer, w_{kj} is the weight value between the hidden layer and output layer, and $f(\ln(k))$ is the activation function.

(4) Aggregation Module. The last module is the aggregation module that combines the results of all artificial neural networks and creates a single module. This module combines the intermediate results and generates the final result.

3.3.2. SVM-ANN Methodology. In this methodology [34], the SVM classifier uses the anomaly detection technique, and the ANN classifier uses the misuse detection technique. The whole methodology is implemented in three modules. The modules are preprocessing module, SVM module, and ANN module. The flowchart of the SVM-ANN methodology is shown in Figure 4.

(1) Preprocessing Module. Preprocessing module is a very important part of the classification methodology, and this module makes the dataset ready for classification. The raw dataset has redundant and useless data, and the preprocessing makes them free from redundant and useless data.

(2) SVM Module. The preprocessed dataset is given as input to the support vector machine classifier, and this classifier performs the binary classification and results into two classes: normal and attack. The normal packet is labelled as normal, whereas the attack packet is labelled as attack. Support vector machine (SVM) classifier usually increases the dimensionality of the data, which makes it easy for separating or classify the data into different categories or classes. A hyperplane can be expressed as [56] H in R^n in the following equation:

$$H = \{x: x^a = b\}. \quad (5)$$

where x is an element in R^n and b is an element in R.

Some studies state that SVM is implemented successfully in regression and classification [52, 53, 57–59].

(3) ANN Module. The attack packets are the input for the artificial neural network classifier. Backpropagation algorithm with feed-forward neural network is implemented. It is a commonly used algorithm by neural networks [55]. This classifier performs multiclass classification. It outputs the attack packets with their types.

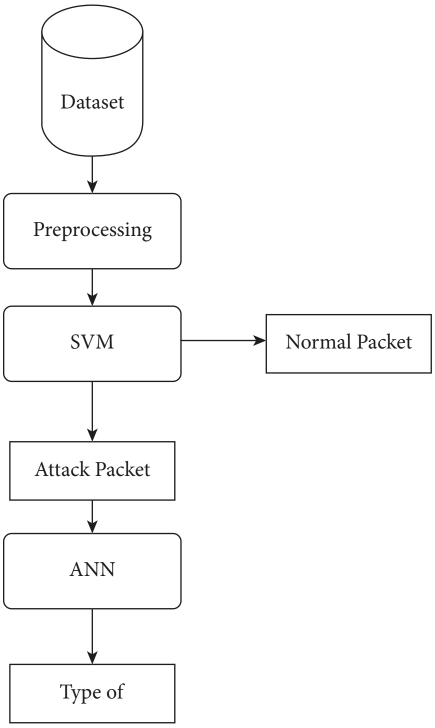

Figure 4: Flowchart of SVM-ANN methodology.

3.3.3. FCM-SVM Methodology. In this methodology [44], the hybrid approach combines FCM with the SVM classifier. The methodology comprises three modules. Figure 5 shows the flowchart of FCM-SVM methodology.

(1) Preprocessing Module. The first module is used for converting the dataset in a form easily understood by the classifier. The preprocessed dataset saves time and resources as unwanted data are removed in this module.

(2) FCM Module. This module makes various groups of the dataset, and the groups are made based on membership functions. The equations related to the FCM algorithm are discussed earlier in this study.

(3) SVM Module. This module classifies various clusters using support vector machine classifiers. SVM classifiers are performing the multiclass classification.

(4) Aggregation Module. The outputs of all the SVM classifiers are combined, and the aggregation module generates the final output.

3.3.4. SMO-ANN Methodology. This is based on a fuzzy C-means clustering algorithm optimized with the Spider monkey optimization algorithm (SMO) [45]. Figure 6 shows the flowchart of SMO-ANN methodology. The methodology

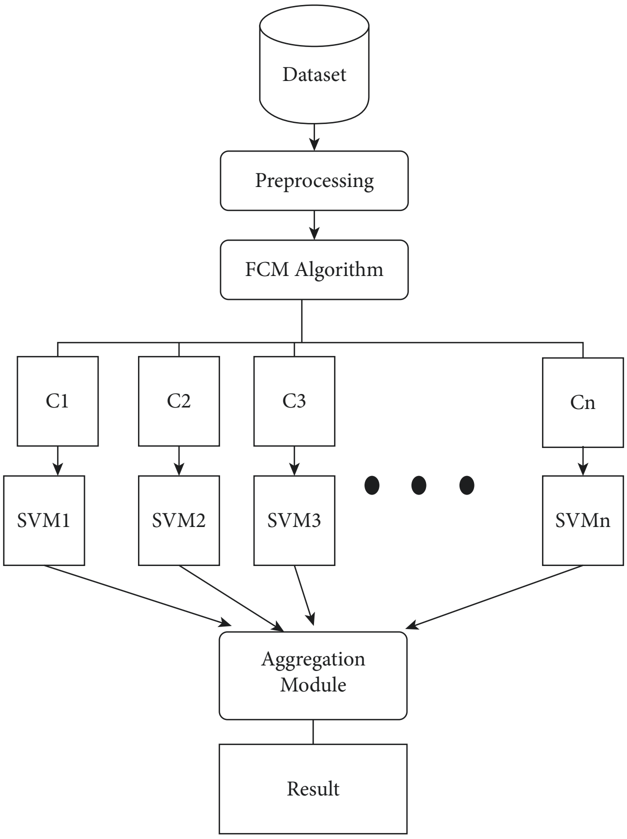

Figure 5: Flowchart of FCM-SVM methodology.

is divided into three modules. The modules are described next.

(1) Preprocessing Module. Preprocessing is carried out to obtain the preprocessed dataset from the raw dataset. The preprocessed dataset is not containing useless data.

(2) FCM-SMO Module. The whole dataset is divided into various clusters in this module. SMO is applied to the clusters to reduce the dataset further and obtain an optimized dataset.

(3) ANN Module. In this module, an artificial neural network (ANN) is applied to classify the dataset into attack packets and normal packets. Attack packets are further classified into their types.

3.4. Evaluation. Performance metrics are vital for comparing different intrusion detection systems, and they also tell which intrusion detection system is performing better than others.

(1) *Accuracy*: Accuracy describes the percentage of true intrusion detection system predictions. Accuracy is represented by the following equation:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \quad (6)$$

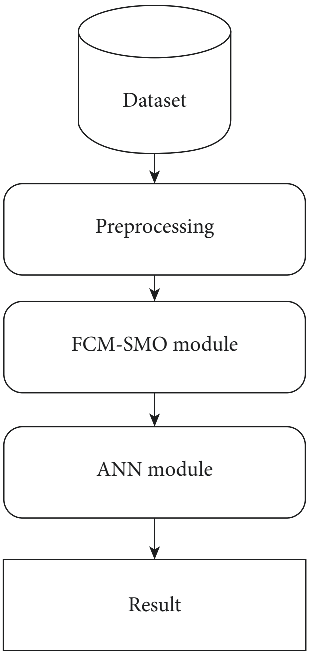

Figure 6: Flowchart of SMO-ANN methodology.

(2) *Precision*: Precision describes the ratio of the attack packets correctly identified as an intrusion by the intrusion detection system to the total number of attack packets. Precision is represented by

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}. \quad (7)$$

(3) *Detection Rate*: The detection rate describes how many packets are identified correctly. It is represented by

$$\text{Detection Rate} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (8)$$

(4) *F*-measure: *F*-measure is defined as the harmonic composition of recall and precision. It is represented by

$$\text{F} - \text{measure} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (9)$$

(5) *False-Positive Rate*: False alarm rate describes the ROC curve. False-positive rate is represented by

$$\text{False Positive Rate} = \frac{\text{FP}}{(\text{FP} + \text{TP})}. \quad (10)$$

These performance metrics are used for comparing various methodologies by using two standard benchmark datasets.

We are using a multiclass dataset for performance assessment. We will calculate performance metrics for every class of both datasets: the UNSW-NB15 and the NSL-KDD datasets. For example, we will calculate the accuracy of every class of the NSL-KDD dataset. For calculating the overall accuracy for the whole dataset, we will find the average of the accuracies of all the classes. In this way, we will calculate the other performance metrics for both datasets. We have compared every attack of both datasets by calculating the performance metrics for every attack. We have also compared the overall performance metrics of both datasets. We have compared the performance of four existing intrusion detection systems.

4. Experiments and Comparative Analysis

To evaluate the performance of the various existing IDSs, we conducted the experimentation on four existing IDSs using two benchmark datasets: the NSL-KDD dataset and UNSW-NB15 dataset. We have compared four existing methodologies and used two standard benchmark datasets: NSL-KDD dataset and the UNSW-NB15 dataset. We present the analysis of the results by comparison concerning five performance metrics: accuracy, detection rate, precision, F-measure, and false-positive rate. Table 3 shows the hardware and software used in the experiments.

Table 3: The hardware and software used in the experiments.

RAM	8 GB
Processor configuration	Intel Core i3
Operating system	Windows 10
Software	MATLAB 2019
Datasets	UNSW-NB15 dataset and NSL-KDD dataset

In Table 4, the SVM-ANN methodology has the highest precision of 1 and lowest false-positive rate of 0. FCM-SVM methodology has the highest accuracy of 0.99855, highest detection rate of 0.98475, and highest F-measure of 0.98431. In Table 5, the SVM-ANN methodology has the highest precision of 1 and lowest false-positive rate of 0. FCM-SVM methodology has the highest accuracy of 0.99925, highest detection rate of 0.99254, and highest F-measure of 0.99482. In Table 6, the SVM-ANN methodology has the highest precision of 1 and the lowest false-positive rate of 0. FCM-SVM methodology has the highest accuracy of 0.99954 and the highest F-measure of 0.99482. SMO-ANN methodology has the highest detection rate 1. In Table 7, the SVM-ANN methodology has the highest detection rate of 0.98624. FCM-SVM methodology has the highest accuracy of 0.99793 and the highest F-measure of 0.98068. SMO-ANN methodology has the highest precision of 1 and lowest false-positive rate of 0. In Table 8, the SVM-ANN methodology has the highest precision of 0.99926 and lowest false-positive rate of 0.00074. FCM-SVM methodology has the highest accuracy of 0.99838, the highest detection rate of 0.99047, and the highest F-measure of 0.98969. In Table 9, FCM-SVM methodology has the highest accuracy of 0.99983, highest detection rate of 0.99984, and highest F-measure of 0.99934. SMO-ANN methodology has the highest precision of 1 and lowest false-positive rate of 0. In Table 10, the SVM-ANN methodology has the highest precision of 1 and lowest false-positive rate of 0. FCM-SVM methodology has the highest accuracy of 0.99788 and the highest F-measure of 0.97563. SMO-ANN methodology has the highest detection rate 1. In Table 11, SMO-ANN methodology has the highest accuracy of 1, highest detection rate of 1, precision of 1, F-measure of 1, and lowest false-positive rate of 0. In Table 12, SVM-ANN methodology and SMO-ANN methodology have precision of 1 and lowest false-positive rate of 0. SMO-ANN methodology has the highest accuracy of 1, highest detection rate of 1, and highest f-measure of 1. In Table 13, FCM-ANN methodology has the highest accuracy of 0.99862, highest detection rate of 0.98710, highest precision of 0.98710, highest F-measure of 0.98710, and lowest false-positive rate of 0.000658. In Table 14, SVM-ANN methodology has the highest accuracy of 0.99151, highest detection rate of 0.98408, and highest F-measure of 0.98836. FCM-ANN methodology and FCM-SVM methodology have a precision of 1 and the lowest false-positive rate of 0.

In Table 15, SVM-ANN methodology has the highest accuracy of 0.99365 and highest F-measure of 0.96540. FCM-SVM methodology has the highest detection rate of 1. FCM-ANN methodology and SMO-ANN methodology have the highest precision of 1 and the lowest false-positive rate of 0. In Table 16, SVM-ANN methodology has the highest accuracy of 0.99805, highest detection rate of 0.76555, and highest F-measure of 0.86721. All methodologies have precision of 1 and false-positive rate of 0. In Table 17, SVM-ANN methodology has the highest accuracy of 0.99996, highest detection rate of 1, and highest F-measure 0.95652. All methodologies have precision 1 and false-positive rate of 0. In Table 18, SVM-ANN methodology has the highest accuracy of 0.99362, highest detection rate 0.94270, highest precision of 0.96460, highest F-measure of 0.95270, and lowest false-positive rate of 0.00484.

The different attacks of the UNSW-NB15 and NSL-KDD datasets are analyzed to evaluate various intrusion detection systems of cloud computing environments. The above tables are representing the results of our experimentation.

Tables 4 to 18 show the different performance metrics values of different attacks of the UNSW-NB15 dataset. FCM-SVM methodology performs better in detecting every attack of the UNSW-NB15 dataset than other methodologies. Table 12 shows the performance metrics values of a complete UNSW-NB15 dataset. The overall performance of the FCM-SVM methodology for detecting attacks of the UNSW-NB15 dataset is better than other methodologies. Tables 13 to 16 show the different performance metrics values of different attacks of the NSL-KDD dataset. FCM-SVM and SMO-ANN methodologies perform better in detecting every attack of the NSL-KDD dataset than other methodologies. Table 17 shows the performance metrics values of the complete NSL-KDD dataset. The overall performances of the SMO-ANN methodology for detecting attacks of the NSL-KDD dataset are better than other methodologies. The main advantage of the SVM classifier is that it only depends on support vectors. The complete dataset does not influence the SVM function, which is the case in many artificial neural networks (ANNs). Also, SVM deals efficiently with many features because kernel functions have exploitation features. The rate of convergence of the SMO algorithm is low. The premature

Table 4: Comparison of various methodologies based on different performance metrics for analysis attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.92653	0.39036	0.83587	0.53218	0.00919
[34]	SVM-ANN	0.99636	0.72595	1	0.84122	0
[43]	FCM-SVM	0.99855	0.98475	0.98387	0.98431	0.00078
[44]	SMO-ANN	0.97619	0.98077	0.96226	0.97143	0.02703

Table 5: Comparison of various methodologies based on different performance metrics for backdoor attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.93563	0.05993	0.85714	0.11202	0.00073
[34]	SVM-ANN	0.99741	0.76976	1	0.8699	0
[43]	FCM-SVM	0.99925	0.99254	0.99712	0.99482	0.00022
[44]	SMO-ANN	0.97619	0.90909	0.83333	0.86957	0.01739

Table 6: Comparison of various methodologies based on different performance metrics for DoS attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.99205	0.98676	0.95123	0.96867	0.00720
[34]	SVM-ANN	0.97639	0.73752	1	0.84894	0
[43]	FCM-SVM	0.99954	0.99453	0.99543	0.99498	0.00022
[44]	SMO-ANN	0.99206	1	0.92857	0.96296	0.00885

Table 7: Comparison of various methodologies based on different performance metrics for exploits attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.88077	0.96934	0.58007	0.72580	0.13646
[34]	SVM-ANN	0.90387	0.98624	0.727	0.83701	0.27300
[43]	FCM-SVM	0.99793	0.97992	0.98144	0.98068	0.00105
[44]	SMO-ANN	0.99206	0.88889	1	0.94118	0

Table 8: Comparison of various methodologies based on different performance metrics for Fuzzer attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.98101	0.95978	0.89149	0.92437	0.01607
[34]	SVM-ANN	0.96633	0.75577	0.99926	0.86062	0.00074
[43]	FCM-SVM	0.99838	0.99047	0.98890	0.98969	0.00094
[44]	SMO-ANN	0.98413	0.75	0.75	0.75	0.00819

Table 9: Comparison of various methodologies based on different performance metrics for generic attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.95318	0.70241	0.81211	0.75329	0.01841
[34]	SVM-ANN	0.99950	0.99731	0.99880	0.99805	0.00119
[43]	FCM-SVM	0.99983	0.99984	0.99984	0.99984	0.00118
[44]	SMO-ANN	0.98413	0.66667	1	0.8	0

Table 10: Comparison of various methodologies based on different performance metrics for reconnaissance attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.96828	0.67319	0.94376	0.78584	0.00379
[34]	SVM-ANN	0.97790	0.75022	1	0.85729	0
[43]	FCM-SVM	0.99788	0.9761	0.97517	0.97563	0.00113
[44]	SMO-ANN	0.99206	1	0.85714	0.92308	0.00833

Table 11: Comparison of various methodologies based on different performance metrics for shellcode attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.92615	0.01107	0.28358	0.0213	0.00219
[34]	SVM-ANN	0.99718	0.69099	0.99383	0.81519	0.00617
[43]	FCM-SVM	0.99801	0.98539	0.98414	0.98476	0.00111
[44]	SMO-ANN	1	1	1	1	0

Table 12: Comparison of various methodologies based on different performance metrics for worm attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.97153	0.81777	0.76506	0.79054	0.01766
[34]	SVM-ANN	0.99834	0.66923	1	0.80184	0
[43]	FCM-SVM	0.99880	0.98488	0.98768	0.98628	0.00056
[44]	SMO-ANN	1	1	1	1	0

Table 13: Comparison of various methodologies based on different performance metrics for the UNSW-NB15 dataset.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.94340	0.65680	0.73920	0.62550	0.03134
[34]	SVM-ANN	0.97980	0.8056	0.9689	0.8701	0.03114
[43]	FCM-SVM	0.99862	0.9871	0.9871	0.9871	0.00066
[44]	SMO-ANN	0.98889	0.9129	0.9331	0.9184	0.01120

Table 14: Comparison of various methodologies based on different performance metrics for DoS attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	F-measure	False-positive rate
[33]	FCM-ANN	0.90469	0.73998	1	0.85056	0
[34]	SVM-ANN	0.99151	0.98408	0.99268	0.98836	0.007319
[43]	FCM-SVM	0.91069	0.75634	1	0.86127	0
[44]	SMO-ANN	0.90961	0.75471	0.99828	0.85957	0.000752

Table 15: Comparison of various methodologies based on different performance metrics for probe attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.97721	0.74924	1	0.85664	0.258792
[34]	SVM-ANN	0.99365	0.97510	0.95589	0.96540	0.004497
[43]	FCM-SVM	0.88556	1	0.82348	0.90319	0.17652
[44]	SMO-ANN	0.97686	0.74530	1	0.85407	0

Table 16: Comparison of various methodologies based on different performance metrics for root-to-local (R2L) attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.99762	0.71292	1	0.83240	0
[34]	SVM-ANN	0.99805	0.76555	1	0.86721	0
[43]	FCM-SVM	0.99774	0.72727	1	0.84211	0
[44]	SMO-ANN	0.99770	0.72249	1	0.83889	0

Table 17: Comparison of various methodologies based on different performance metrics for user-to-root (U2R) attack.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.99984	0.63636	1	0.77778	0
[34]	SVM-ANN	0.99996	1	0.91667	0.95652	0
[43]	FCM-SVM	0.99980	0.54545	1	0.70588	0
[44]	SMO-ANN	0.99980	0.54545	1	0.70588	0

Table 18: Comparison of various methodologies based on different performance metrics for the NSL-KDD dataset.

Ref.	Methodology	Performance metrics				
		Accuracy	Detection rate	Precision	*F*-measure	False-positive rate
[33]	FCM-ANN	0.95175	0.76770	0.96310	0.84320	0.05176
[34]	SVM-ANN	0.99362	0.94270	0.96460	0.95270	0.00484
[43]	FCM-SVM	0.954224	0.75590	0.96420	0.83400	0.03530
[44]	SMO-ANN	0.95360	0.75340	0.96410	0.83210	0.04956

convergence of the SMO algorithm also affects the performance. SVM hybridization with other classifiers might give an efficient intrusion detection system.

5. Future Scopes and Recommendations

Intrusion detection systems detect known and unknown attacks. But the copious amounts of data generated and stored on the cloud make the intrusion detection problem more complex. We epitomized the underlying future scopes:

(i) The brisk growing zero-day attacks and their vulnerabilities are the demanding future scope in developing the intrusion detection system for cloud computing.

(ii) Another future scope is developing an adaptive architecture of intrusion detection systems to handle the dynamic computations.

(iii) Researchers can also focus on integrating the intrusion detection system with blockchain technologies.

(iv) The possible recommendations for the above future scopes are as follows.

(v) An adaptive intrusion detection system must be developed that can adapt to change the requirements such as environment configurations, resources of computation, and various locations where intrusion detection systems are deployed.

(vi) It should expand dynamically by adding virtual machines when the cloud network extends.

6. Conclusion

This article reviews various intrusion detection systems related to cloud computing. The article implements various IDSs and compares them. Two standard benchmark datasets were employed and observed that the FCM-SVM methodology outperforms other techniques using the UNSW-NB15 dataset, and the SVM-ANN method outperforms the preliminaries using the NSL-KDD dataset. Hence, SVM is identified as a better classifier than other classifiers. In future work, we will work on zero-day attacks to develop an adaptive intrusion detection system that adapts to changing cloud architecture.

Data Availability

The datasets used in the article are publicly available standard benchmark datasets referred to in Refs. [54, 56, 60].

Consent

Not applicable.

Conflicts of Interest

The authors declare no conflict of interest related to this work.

Authors' Contributions

P.R. and I.B. were responsible for the conceptualization of the topic; article gathering and sorting were carried out by A.M., Y.K., S.K.P., N.G., A.K., S.R., and A.L.I; manuscript writing and original drafting and formal analysis were carried out by P.R., I.B., Y.K., and S.P; and writing of reviews and editing were carried out by N.G., A.K., S.R., and A.L.I. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (Grant no. 2020R1G1A1099559). The work of Agbotiname Lucky Imoize was supported in part by the Nigerian Petroleum Technology Development Fund (PTDF) and in part by the German Academic Exchange Service (DAAD) through the Nigerian-German Postgraduate Program under Grant no. 57473408.

References

[1] S. Singh, K. Saxena, and Z. Khan, "Intrusion detection based on artificial intelligence techniques," in *Proceedings of the International Conference of Advance Research And Innovation (Icari-2014)*, 2014.

[2] S. Prakash, "Role of virtualization techniques in cloud computing environment," in *Advances in Computer Communication and Computational Sciences*Springer, Singapore, 2019.

[3] M. Rana and J. Singla, "A systematic review on data mining rules generation optimizing via genetic algori," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[4] M. Idhammad, K. Afdel, and M. Belouch, "Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest," *Security and Communication Networks*, vol. 2018, Article ID 1263123, 13 pages, 2018.

[5] P. S. Bawa, S. U. Rehman, and S. Manickam, "Enhanced mechanism to detect and mitigate economic denial of sustainability (EDoS) attack in cloud computing environments," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 9, pp. 51–58, 2017.

[6] V. Jyothsna and V. V. Rama Prasad, "FCAAIS: anomaly based network intrusion detection through feature correlation analysis and association impact scale," *ICT Express*, vol. 2, no. 3, pp. 103–116, 2016.

[7] D.-Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognition*, vol. 36, no. 1, pp. 229–243, 2003.

[8] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection techniques in grid and cloud computing environment," *IEEE IT Professional Magazine*, vol. 12, 2010.

[9] Y. Guan and J. A. C. P. Bao, "Intrusion detection strategy on cloud computing," *International Symposium on Web Information Systems and Applications (WISA)*, pp. 84–87, 2009.

[10] C. Mazzariello, R. Bifulco, and R. Canonico, "Integrating a network IDS into an Open source Cloud computing," in *Proceedings of the 6th International Conference on Information Assurance and Security*, pp. 265–270, Atlanta, GA, USA, August 2010.

[11] A. K. Jones and R. S. Sielken, *Computer System Intrusion Detection: A Survey*, 2000.

[12] S. Roschke, F. Cheng, and C. Meinel, "An extensible and virtualization compatible IDS management architecture," in *Proceedings of the 5th International Conference on Information Assurance and Security*, vol. 2, pp. 130–134, Xi'an, China, August 2009.

[13] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, 2010.

[14] S. Kumar, A. Viinikainen, and T. Hamalainen, "Machine learning classification model for network based intrusion detection system," in *Proceedings of the 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 242–249, Barcelona, Spain, December 2016.

[15] A. H. Bhat, S. Patra, and D. Jena, "Machine learning approach for intrusion detection on cloud virtual machines," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 6, pp. 57–66, 2013.

[16] P. Singh, S. Manickam, and S. U. Rehman, "A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architecture," in *Proceedings of the 3rd International Conference on Reliability, Infocom Technologies and Optimization*, pp. 1–4, IEEE, Noida, India, October 2014.

[17] P. Rana and I. Batra, "Detection of attacks in cloud computing environment- A comprehensive review," in *Proceedings of the 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 496–499, IEEE, London, United Kingdom, April 2021.

[18] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," *Applied Soft Computing*, vol. 18, no. 1, pp. 178–184, 2014.

[19] C. Nkikabahizi, W. Cheruiyot, and A. Kibe, "Classification and analysis of techniques applied in intrusion detection systems," *International Journal of Scientific Engineering and Technology*, vol. 6, no. 7, pp. 216–219, 2017.

[20] V. Balamurugan and R. Saravanan, "Enhanced intrusion detection and prevention system on cloud environment using hybrid classification and OTS generation," *Cluster Computing*, vol. 22, no. 6, Article ID 13027, 2019.

[21] C. N. Modi, D. R. Patel, A. Patel, and M. Rajarajan, "Integrating signature apriori based network intrusion detection system (NIDS) in cloud computing," *Procedia Technology*, vol. 6, pp. 905–912, 2012.

[22] S. Gupta, P. Kumar, and A. Abraham, "A profile based network intrusion detection and prevention system for securing cloud environment," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, 2013.

[23] B. Hajimirzaei and N. J. Navimipour, "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm," *ICT Express*, vol. 5, no. 1, pp. 56–59, 2019.

[24] M. A. Hatef, V. Shaker, M. R. Jabbarpour, J. Jung, and H. Zarrabi, "HIDCC: a hybrid intrusion detection approach in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 3, 2018.

[25] P. Ghamisi and J. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm

optimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 309–313, 2014.

[26] A. S. Saljoughi, M. Mehrvarz, and H. Mirvaziri, "Attacks and intrusion detection in cloud computing using neural networks and particle swarm optimization algorithms," *Emerging Science Journal*, vol. 1, no. 4, pp. 179–191, 2017.

[27] K. Costa, C. Pereira, R. Nakamura, L. Pereira, and J. Papa, "Boosting Optimum-Path Forest clustering through harmony Search and its applications for intrusion detection in computer networks," in *Proceedings of the 4th International Conference on Computational Aspects of Social Networks (CASoN)*, pp. 181–185, IEEE, Sao Carlos, Brazil, November 2012.

[28] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[29] M. T. Khorshed, A. B. M. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 833–851, 2012.

[30] Y. Liu and R. Ma, "Network anomaly detection based on BQPSO-BN algorithm," *IETE Journal of Research*, vol. 59, no. 4, pp. 334–342, 2013.

[31] N. Pitropakis, D. Anastasopoulou, A. Pikrakis, and C. Lambrinoudakis, "If you want to know about a hunter, study his prey: detection of network based attacks on KVM based cloud environments," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–10, 2014.

[32] K. Wang, C.-Y. Huang, L.-Y. Tsai, and Y.-D. Lin, "Behavior-based botnet detection in parallel," *Security and Communication Networks*, vol. 7, no. 11, pp. 1849–1859, 2014.

[33] N. Pandeeswari and G. Kumar, "Anomaly detection system in cloud environment using fuzzy clustering based ANN," *Mobile Networks and Applications*, vol. 21, no. 3, pp. 494–505, 2015.

[34] J. Hussain, S. Lalmuanawma, and L. Chhakchhuak, "A two-stage hybrid classification technique for network intrusion detection system," *International Journal of Computational Intelligence Systems*, vol. 9, no. 5, pp. 863–875, 2016.

[35] M. Idhammad, K. Afdel, and M. Belouch, "Distributed intrusion detection system for cloud environments based on data mining techniques," *Procedia Computer Science*, vol. 127, pp. 35–41, 2018.

[36] R. Kesavamoorthy and K. R. Soundar, "Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system," *Cluster Computing*, pp. 1–8, 2018.

[37] Z. Chiba, "Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms," *Computers & Security*, vol. 86, pp. 219–317, 2019.

[38] R. Patil, H. Dudeja, and C. Modi, "Designing an efficient security framework for detecting intrusions in virtual network of cloud computing," *Computers & Security*, vol. 85, pp. 402–422, 2019.

[39] E. Besharati, M. Naderan, and E. Namjoo, "LR-HIDS: logistic regression host-based intrusion detection system for cloud environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, 2019.

[40] M. Jelidi, A. Ghourabi, and K. Gasmi, "A hybrid intrusion detection system for cloud computing environments," in *Proceedings of the International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–6, Sakaka, Saudi Arabia, April 2019.

[41] P. Ghosh, A. Karmakar, J. Sharma, and S. Phadikar, "CS-PSO based intrusion detection system in cloud environment," in *Emerging Technologies in Data Mining and Information Security*, pp. 261–269, Springer, Berlin, Germany, 2019.

[42] R. Rajendran, S. V. N. Santhosh Kumar, Y. Palanichamy, and K. Arputharaj, "Detection of DoS attacks in cloud networks using intelligent rule based classification system," *Cluster Computing*, vol. 22, no. 1, pp. 423–434, 2019.

[43] A. N. Jaber and S. U. Rehman, "FCM–SVM based intrusion detection system for cloud computing," *Cluster Computing*, vol. 23, pp. 1–11, 2020.

[44] J. K. Samriya and N. Kumar, "A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing," *Materials Today*, 2020.

[45] S. Rajagopal and P. P. Kundapur, "Towards effective network intrusion detection: from concept to creation on Azure cloud," *IEEE Access*, vol. 9, Article ID 19723, 2021.

[46] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019.

[47] S. Shamshirband, M. Fathi, A. T. Chronopoulos, A. Montieri, F. Palumbo, and A. Pescapè, "Computational intelligence intrusion detection techniques in mobile cloud computing environments: review, taxonomy, and open research issues," *Journal of Information Security and Applications*, vol. 55, Article ID 102582, 2020.

[48] O. Alkadi, N. Moustafa, and B. Turnbull, "A review of intrusion detection and blockchain applications in the cloud: approaches, challenges and solutions," *IEEE Access*, vol. 8, Article ID 104893, 2020.

[49] M. M. Ahsan, K. D. Gupta, A. K. Nag, S. Poudyal, A. Z. Kouzani, and M. A. P. Mahmud, "Applications and evaluations ofBio-InspiredApproaches in cloud security: a review," *IEEE Access*, vol. 8, Article ID 180799, 2020.

[50] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: a review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[51] M. C. Babu and K. Senthilkumar, "Machine learning based strategies for secure cloud," *Materials Today Proceedings*, 2021.

[52] https://www.unsw.adfa.edu.au/un%20sw-canberra-%20cyber/cybersecurity/%20ADFAN%20B15-%20Datasets/.

[53] N. Kunhare, R. Tiwari, and J. Dhar, "Particle swarm optimization and feature selection for intrusion detection system," *Sādhanā*, vol. 45, no. 1, pp. 1–14, 2020.

[54] X. Luo, J. Sun, L. Wang et al., "Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4963–4971, 2018.

[55] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1–27, 2021.

[56] Y. Zhang, Y. Wang, G. Zhou et al., "Multi-kernel extreme learning machine for EEG classification in brain-computer interfaces," *Expert Systems with Applications*, vol. 96, pp. 302–310, 2018.

[57] X. Luo, C. Jiang, W. Wang, Y. Xu, J.-H. Wang, and W. Zhao, "User behavior prediction in social networks using weighted extreme learning machine with distribution optimization,"

Future Generation Computer Systems, vol. 93, pp. 1023–1035, 2019.

[58] J. Li, X. Shi, Z. H. You et al., "Using weighted extreme learning machine combined with scale-invariant feature transform to predict protein-protein interactions from protein evolutionary information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 5, pp. 1546–1554, 2020.

[59] Unsw, "NSL-KDD | datasets | research | Canadian institute for cybersecurity | UNB," 2021.

[60] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, "A switching delayed PSO optimized extreme learning machine for short-term load forecasting," *Neurocomputing*, vol. 240, pp. 175–182, 2017.