Cluster Computing (2020) 23:3221–3231
https://doi.org/10.1007/s10586-020-03082-6

FCM–SVM based intrusion detection system for cloud computing environment

Aws Naser Jaber[1] · Shafiq Ul Rehman[2]

Received: 7 July 2019 / Revised: 7 January 2020 / Accepted: 29 February 2020 / Published online: 25 March 2020

Abstract
Cloud computing offer various services over the Internet based on pay-per-use concept. Therefore, many organizations have already adopted this system to attract the users with its desirable features. However, due to its design, makes it vulnerable to malicious attacks. This demands an Intrusion Detection System that can detect such attacks with high detection accuracy in cloud environment. This paper proposes a novel intrusion detection system that combines a fuzzy c means clustering (FCM) algorithm with support vector machine (SVM) to improve the accuracy of the detection system in cloud computing environment. The proposed system is implemented and compared with existing mechanisms. The NSL-KDD dataset is used for experiments. Based on performance evaluation and comparative analysis, the results obtained using this new hybrid mechanism (FCM–SVM) show that the proposed system can detect the anomalies with high detection accuracy and low false alarm rates over the existing techniques.

Keywords Intrusion detection system · Cloud computing · DDoS attacks · Machine learning classifiers · Cyber security · FCM · SVM

1 Introduction

The cloud refers to a distinct information technology (IT) environment designed for remotely provisioning scalable and measured IT resources. Cloud computing (CC) can provide platform, software, infrastructure like service models based on customer requirements and usage [1]. The virtualization of storage resources and their applications is a major requirement of cloud computing [2]. In a cloud system, a hypervisor inspector monitors the shared hardware platform and the overall work of the operating systems.

However, cloud is becoming increasingly attractive to hackers given its open nature and the amount of traffic that the cloud generates [3]. For instance, distributed denial of service (DDoS) attacks are the most prevalent cybercrime after information theft. TCP and/or UDP flood attacks can exhaust the clouds resources, consume most of its bandwidth, and damage an entire cloud project within a short period of time. These security risks require developing and implementing an effective intrusion system that will defend the cloud from newly emerging attacks a.k.a. zero-day attacks [4]. The most common challenges faced by traditional methods are that IDS generate false alarms and do not use proper standards or parameters to evaluate threats. This can lead to the misuse problem.

While Cloud Computing (CC) can adopt anomaly detection techniques to detect and mitigate unknown attacks [5]. A critical issue in implementing these techniques is the large number of events that occur during regular cloud operation, thereby rendering the process of monitoring and detecting attacks burdensome. The capability of soft computing to address true and uncertain data makes them suitable for detecting intrusions. Various soft computing techniques, including artificial neural network

✉ Shafiq Ul Rehman
shafiq_rehman@sutd.edu.sg

Aws Naser Jaber
naserjaber.a@gmail.com

[1] Faculty of Computer Systems and Software Engineering (FSKKP), Universiti Malaysia, Pahang, Malaysia

[2] ST Engineering Electronics - SUTD Cyber Security Laboratory, Singapore University of Technology and Design (SUTD), 8 Somapah Road, 487372 Singapore, Singapore

(ANN), support vector machine (SVM), fuzzy logic (FL), genetic algorithm (GA) and association rule mining, are currently used [6].

These techniques enhance the accuracy and effectiveness of anomaly detection or signature-based IDS. While, ANN intrusion systems are widely adopted given their capability to use incomplete data and classify these data as either intrusive or normal [7]. IDS use different types of ANN. The most commonly used ANN types are multilayer preceptor, back propagation and multilayer feed forward neural networks. The use of an ANN-based IDS is an effective solution for unstructured network data. An IDS generally requires considerable time and numerous training samples to be highly effective; this condition is a major limitation of these systems. Cloud infrastructure requires a rapid mechanism for detecting intrusions and using ANN-based IDS as the only defense mechanism is an ineffective solution.

Nevertheless, several methods have tried to address the ANN issues over network-based IDS (N-IDS) cloud computing based on optimization algorithms which are inspired by biological evolution, such as Genetic Algorithms (GA) [8], particle swarm optimization (PSO) [9], harmony search [10], and artificial bee colony [11].

These methods are global heuristic search techniques that select network features or determine the optimal parameters for use in other techniques, thus improving the performance of the IDS. For instance, some researchers [12] presented a hybrid feature selection and multiclass classification algorithm to detect attacks in VMs. The authors proposed a security mechanism integrating GA with discrete PSO to select the best features from the NSL-KDD dataset. According to authors, their hybrid algorithms can achieve an accuracy rate of greater than 95%. Although, the classier can achieve an acceptable rate of DDoS classification but it still needs to improve and reduce false alarms.

FL is primarily used to address intrusions with uncertain problems and inexact descriptions. FL can be applied in IDS when some features are considered as fuzzy variables. An IDS based on fuzzy logic can handle network intrusions, such as UDP SYN floods, e-mail bombs, ping of death, FTP/Telnet port scanning and password guessing. The problems of ANN, that is, considerable time requirement and extensive variety of training scenarios, can be addressed by evolving fuzzy neural networks which can be used to detect previously unknown attacks.

SVM is being used as a classifier to render the possible solutions for IDS. SVM uses limited sample data to detect intrusions, wherein accuracy is unaffected by data dimensions. Study [13] has confirmed that the SVM exhibits better results than ANN in terms of rating false positives.

Therefore, in this paper we propose an enhanced Fuzzy C-Means Clustering Algorithm with Support Vector Machine (FCM–SVM) as a hypervisor inspector to achieve high accuracy and low false alarm rates for all types of IDS. The proposed mechanism is trained and tested with NSL-KDD dataset, and later, compared with existing mechanisms. Rest of the paper is organized as follows: state-of-the-art techniques are discussed in Sect. 2. Proposed mechanism is described in Sect. 3. Implementation and results are presented in Sect. 4. Finally, the conclusion of this paper is outlined in Sect. 5.

2 State-of-the-art techniques

In this section, we discuss the existing IDS techniques proposed for cloud computing environment.

Reducing false alarms are critical in N-IDS efficiency and usability. When an N-IDS tests for activity on a network to identify and distinguish between real attacks or non-attacks, an alert result [14]. Therefore, developing a more efficient N-IDS is necessary. Snort, an open source system, is one of the important N-IDS development systems that is undergoing constant improvement [15].

Although DDoS attacks have different vectors, all of them aim to overwhelm servers, firewalls, or other perimeter defined devices by sending request packets at high packet rates [16]. Cloud networks becomes overwhelmed to the point where a website is inaccessible [17].

An attacker uses a UDP flood attack. UDP packets are sent to either random or specified ports on a victim's system [18]. Meanwhile, the system identifies the application that sends the request [19]. If no applications are running on the targeted port, the victim's system sends an ICMP packet to indicate that the destination port is unreachable [20]. As with surfing, UDP flooding uses a spoofed IP address when sending the attacking packet [21]. By using this method, return packets are sent to another system with a spoofed address; they are not sent back to the zombie system [22]. That is, these attacks can fill the bandwidth connection around the victim's system, causing the systems to experience connectivity problems.

State-of-the-art techniques are utilized to improve accuracy and successfully defend a system. Each existing technique has its own advantages and limitations which affect the accuracy and efficiency of IDS. Some of the well-known IDS techniques are:

Hota and Shrivas [23] made a comparative study of various hybrid approaches for both binary (normal vs. attack) and multi-class classifications of the NSL-KDD dataset. Each hybrid implementation used the information gain (IG) feature selection and one of five classification algorithms: MLP, C4.5, RF, and REP tree. The authors reported that the best performance was achieved with an IG-RF hybrid classifier.

Pervez and Farid [24] defined a hybrid approach based on feature selection and subsequent classification using the NSL-KDD dataset. Feature selection was implemented following the Leave-One-Out (LOO) method, and, as a classifier, the authors deployed SVMs in a One-against-the-Rest Multi-Class Configuration (OAR–SVM). Their experiment showed that the greatest classification accuracy was achieved by evaluating 14 selected features.

Enache and Patriciu [25] developed a two-stage hybrid approach: (i) feature selection with an IG algorithm and (ii) classification with an SVM method for binary (normal vs attack) IDS classification. In addition, the authors chose to introduce a meta optimization based on swarm intelligence algorithms to find the optimal set of classification parameters for the SVM. Two approaches were used to optimize the SVM classification parameters: PSO and Artificial Bee Colony (ABC). The experimental results for the NSL-KDD dataset indicated that an ABC-SVM approach achieved slightly higher precision than PSO–SVM.

Eid et al. [26] proposed a simple hybrid classifier as a solution to the IDS classification problem. A GA was implemented as a wrapper method for feature selection, in conjunction with an NB classifier. The optimal subset of features was found by minimizing the classification error of the NB classifier trained with a given subset of features. In addition to feature selection, the authors implemented the Entropy Minimization Discretization (EMD) method to discretize the input data. The method was applied to the NSL-KDD dataset, with the whole set used for training, and the effectiveness of the proposed method was evaluated using 10-fold cross-validation.

De la Hoz et al. [27] implemented a two-component hybrid approach, with a feature selection and classification stage. They employed multi-objective feature selection, with the non-dominated Sorting Genetic Algorithm (NSGA) implemented to find the subset of features that maximized the Jaccard coefficient for each class in the dataset. The NSL-KDD dataset was classified by the growing hierarchical self-organizing maps (GHSOM). Similar to Eid et al. [26], the whole NSL-KDD dataset was used in the training phase, and the results were based on 10-fold cross-validation with a reported accuracy of 95.60%.

Rastegari et al. [28] developed an IDS based on GA optimization. Binary classification (normal vs. attack) of the NSL-KDD dataset was performed using a set of IF–THEN rules applied to the selected features. The features for rule construction and condition boundaries were selected by GA optimization, with the goal of minimizing the number of misclassified instances. Additionally, the authors implemented Correlation-Based Feature Selection (CFS), the Consistency Subset Evaluator (CSE), and the selection of only real-valued features. Their results indicate that the developed approach is comparable to other single-stage learning methods.

Alpha profiling was applied to the whole NSL-KDD dataset to combine the protocol and service features into a single alpha feature. To reduce the training time, beta profiling was deployed to remove redundant training pairs from the training set. Feature selection was based on three approaches: Filtered Subset Evaluation (FSE), CFS, and CSE. The authors reported that their Alpha FST Beta OSLEM approach could reduce both the dimensionality and training set size without compromising the classification accuracy.

Kanakarajan and Muniasamy [29] presented an approach based on a greedy randomized adaptive search procedure with annealed randomness (GAR-forest) classifier for both binary (normal vs attack) and multilabel classification of NSL-KDD. The GAR-forest approach uses the meta-heuristic greedy randomized adaptive search procedure (GRASP), which generates a set of randomized adaptive decision trees. Feature selection was implemented through IG, symmetrical uncertainty (SU), and CFS. The authors reported that the GAR-forest classifier outperformed RF, C4.5, NB, and MLP classifiers. Their feature selection method also resulted in improved classification accuracy.

Hassanien et al. [30] presented a multi-layer IDS based on three stages: (i) feature extraction through Principal Component Analysis (PCA), (ii) binary (normal vs anomalous) classification with a GA, and (iii) multi-class categorization of anomalous instances with decision trees. The GA classification was performed as a set of IF–THEN rules, with each observation labelled as either normal network traffic or a network intrusion. The experimental procedure was conducted on the NSL-KDD dataset. An analysis of the developed approach found that two-layer classification offered more reliable classification results than single-stage classifiers.

A similar approach was developed by [5]. As a feature reduction method, they implemented a linear discriminant analysis (LDA) algorithm. The first-tier, binary (normal vs anomalous) classification was performed with an NB classifier, and anomalous data were then classified more precisely in the second tier using kNNCF (kNN with a certainty factor).

The analysis of Hassanien et al. [30] and Pajouh et al. [31] indicates that the latter managed to obtain considerably better classification results.

These results are a direct product of the considerable time and numerous training samples requirements of the ANN to become effective, whereas the SVM is only required to set a few parameters to perform the same classification. In the cloud, the SVM is a more effective solution than the ANN in cases in which sample data are limited, and accuracy is unaffected by data dimensions. In addition, selecting the optimal parameters of the network increases the accuracy of the IDS, which can be combined with other techniques to create a hybrid approach that uses two or more techniques.

The most promising technique that has been successfully used to improve the detection accuracy of IDS is the ANN. Thus, the major drawbacks of the underlying systems should be investigated further. However, the scheme proposed by Pandeeswari and Kumar [32] demonstrates leaks on the limitation of detection, that is, low false alarm rate, remote to local (R2L) and user to root (U2R). Therefore, this scheme should be investigated further to achieve the alarm scenarios of the IDS. A specific problem statement formulated by Pandeeswari and Kumar pertains to the low false alarm rate that requires additional enhancement and instigation. They also used a slandered ANN, which undergoes multiple local minima (considering the gradient-based technique implementation to determine the weights of the neurons).

Ingre and yadav [33] performed an ANN as a classifier for the NSL-KDD dataset. They obtained results for both a binary class and five class classifications (type of attack). The results were analyzed based on measures, and they found better accuracy. However, the ratio of the classifier still needed to be improved. The detection rate that they obtained was 81.2% for intrusion detection and 79.9% for the attack type classification for the NSLKDD dataset.

Bamakan et al. [34] proposed an effective intrusion detection framework by using a new, adaptive, robust, precise optimization method. They used time-varying chaos particle swarm optimization (TVCPSO) to simultaneously set parameters and select features for multiple criteria linear programming (MCLP) and support vector machines (SVM). In the proposed methods, a weighted objective function was provided that considers trade-off between maximizing the detection rate and minimizing the false alarm rate, along with considering the number of features.

Furthermore, to make the particle swarm optimization (PSO) algorithm faster in searching the optimum and to avoid the search being trapped in local optimum, they adopted a chaotic concept in PSO. Time varying inertia weight and time varying acceleration coefficient were also introduced. The performance of the proposed methods was evaluated by conducting experiments with the NSL-KDD dataset, which was derived and modified from well-known KDD cup 99 datasets. Nevertheless, the accuracy needs greater improvement. Bamakan et al. [34] achieved 97.84% rate of accuracy with a high rate of false positives.

However, a trade-off exists with respect to the stability of the ANN architecture and the detection rate for less frequent attacks. Raman et al. [35] presents a new approach based on the Helly property of Hypergraph and Arithmetic Residue-based Probabilistic Neural Network (HG AR-PNN) to address the classification problem in IDS.

The Helly property of Hypergraph was exploited to identify the optimal feature subset. The arithmetic residue of the optimal feature subset was used to train the PNN. The performance of HG AR-PNN was evaluated by using the KDD CUP 1999 intrusion dataset. The experimental results proved the dominance of the HG AR-PNN classifier over the existing classifiers with respect to the stability and improved detection rate for less frequent attacks. It noticed an accuracy of 95.75%.

The advantages of these techniques are utilized to improve accuracy and successfully defend a system. Nevertheless, each existing technique has its own advantages and limitations.

To compare all previous approaches on an equal footing, our examination was restricted to the overall classifications' accuracy based on the same type and size of dataset. Therefore, from the literature review we selected only those studies that applied the full NSL-KDD dataset were used for comparison as shown in Table 1.

3 Proposed mechanism

In this section we describe the design of our proposed mechanism. Our scheme uses three-steps method to fulfill the research objectives. The first step is defined to create a cluster group based on the membership function using the NSL-KDD dataset [37]. Hence, we comprehend the mechanism of fuzzy logic in our proposed scheme. Fuzzy logic was developed by Zadeh in [38]. Fuzzy logic is a concept for all techniques and technologies that use a fuzzy set. A multivalued logic allows defining the intermediate values between conventional equations, such as true/false, white/black and yes/no, and the degrees of truth between 0 and 1.

The point between the centers of two clusters is assumed to have a continued membership in both clusters. The FCM algorithm assigns a specific cluster to each data point in accordance with the degree of membership grade. This procedure indicates that the data point can be a part of several clusters simultaneously. The letter c denotes the number of clusters. All the clusters should have a similar size, and the number of clusters is fixed and known in advance. The FCM algorithm separates k data points that are specified in m-dimensional vectors uk $(k = 1,2,\ldots,$ $k)$ within the c fuzzy clusters. The FCM algorithm finds the center of the cluster for each cluster by minimizing the

Table 1 Comparison of studies that classified NSL-KDD dataset in terms of overall accuracy

Authors	Approach	Accuracy (%)
Pajouh et al. [31]	IG-GAR	82.00
Hassanien et al. [30]	PCA-BFtree	68.28
Kanakarajan and Muniasamy [29]	LDA–NB–kNNCF	78.90
Pervez and Farid [24]	LOO–OAR–SVM	82.68
Tavallaee et al. [36]	MLP	77.41

objective function. This algorithm differs from the hard c-means in implementing a fuzzy separation. The M membership matrix can have elements within the range [0,1].

The second step will use these clusters to train and test the Genetic Programming (GP) and SVM algorithms. The third step will evaluate the performance of the FCM using different soft computing methods, such as ANN, GP and SVM, to compare the errors from each method.

Cloud computing mainly works on a concept of virtualization [2]. Hence, the present research is aimed at detecting abnormalities in a virtual network and in network-based events by analyzing multiple virtual machines. The hypervisor inspector that was trained by the FCM–SVM is developed and trained to observe virtual machines and their operations. Fuzzy linguistic values, that is, fuzzy classes, are defined by a collection of fuzzy sets.

The amount of fuzziness between data points is determined to be within the range of [1, *n*] in accordance with the fuzzification parameters. Fuzzy clustering is a process that clusters the data points to their matching cluster group by assigning a fuzzy membership function. The FCM algorithm assigns a collection of n elements $X = (\times 1, x2, \ldots, xn)$ and creates similar sized clusters in accordance with the membership function. Then, the partition matrix $w = wij\ ^{\text{TM}}$ [0,1] and the list of c cluster centers $c = (c1, c2, \ldots, ck)$ is returned by the algorithm in which:

$$i = 1, 2, \ldots, n \tag{1}$$

$$j = 1, 2, \ldots, n \tag{2}$$

Wij denotes the value of the fuzzy membership function that appoints the degree to which the data point is related. Figure 1 demonstrates the framework of the FCM–SVM which is divided into the following three phases:

Phase 1	Fuzzy clustering is used to separate large datasets into small clusters in accordance with the membership values
Phase 2	SVM is trained in accordance with various clusters
Phase 3	The fuzzy aggregation module is used to combine the results of the hypervisor inspector to limit different SVM errors

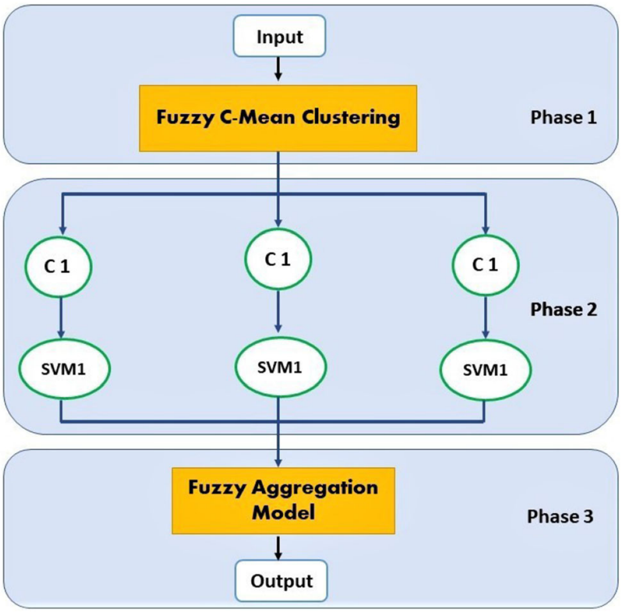

Fig. 1 Proposed FCM–SVM framework

4 Implementation and results

To implement the proposed hybrid mechanism based on FCM–SVM we used Weka simulator [39]. The FCM–SVM algorithm is trained and tested in Weka simulation. FCM is compared with other improved soft computing techniques such as GP and SVM. Factors that influence different attacks on the cloud such as DoS, R2L, normal and U2R, are identified, and a hybrid IDS that comprises FCM and SVM is designed.

A large-scale virtual network was created with multiple switches and a router, and then installed in the server environment. The system used for attack coordination assumed the dual roles of Master client and Handler. The system specifications are presented in Table 2.

Improved results of the FCM–SVM algorithm are obtained from the WEKA simulation. While a binary classifier produces output with two class values or labels, such as Yes/No and 1/0, for given input data. The class of interest is usually denoted as positive and the other as negative.

Table 2 System specifications

System	Specifications
Model	VL HP Proliant dl32e gen8
CPU	Intel Xeon E3-1220Lv2(2.3GHZ/2-core/3 MB/17 W, HT)
RAM	16 Gigabytes 1600 MHz DDR3
Hard Drive	2 terabytes
Network interface	HP Ethernet 1 GB 2-port 330i Adaptor
Operating system	Debian 9.2
Linux Kernel version	4.9 2017

4.1 Performance evaluation

We used NSL-KDD test dataset for performance evaluation. It contains the observed labels for all data instances. These observed labels are used to compare with the predicted labels for performance evaluation after classification. Attacks are very common against networks, as they tend to break into accounts with weak username and password combinations. We used nine algorithms in Weka for the classification task. The test option used in all techniques was 10-fold cross-validation. NSL-KDD is suggested to solve some of the problems in the original KDD99 dataset. One of the most important deficiencies in the KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records.

Data files in NSL-KDD have KDDTrain+.ARFF: The full NSL-KDD train set with binary labels in ARFF format. KDDTest+.ARFF: The full NSL-KDD test set with binary labels in ARFF format. Figure 2 depicts the NSL-KDD dataset classification process.

However, the normal traffic in this dataset were 972,781 records and attacks traffic were 3,925,650. Whereas, the dataset contains 41 features which are listed in Table 3. In each record there are 41 attributes unfolding different features of the flow and a label assigned to each either as an attack type or as normal. The details of the attributes namely the attribute name contains type information of all the 41 attributes available in the NSL-KDD dataset.

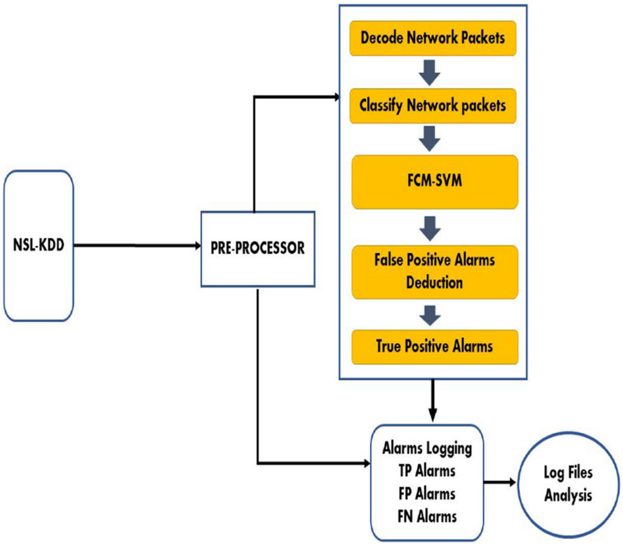

Fig. 2 Data classification process

In addition, these attributes contain data about the various 5 classes of network connection vectors, and they are categorized as one normal class and four attack classes. These attack classes are further grouped as DoS, Probe, R2L (Remote-to-Local), and U2R (User-to-Root).

4.2 Comparative analysis

In this subsection, we present the comparative analysis. Tables 4, 5, and 6 respectively, depicts the results we obtained from the experiments which clearly indicates that FCM–SVM classifier in U2R, R2L, and Probe scenarios has reduced the False Negative rates and has increased the accuracy rates. Moreover, FCM–SVM has achieved better results in U2R, R2L, and Probe classes of attacks in terms of Accuracy, Incorrect classification, True positive (TP), False negative (FN), Precision, Recall, and F-measure. Regarding accuracy, we got 97.37% in U2R, 98.46% in R2L and 98.85% in Probe type attacks, which means our classifier is efficient enough to identify these attacks with higher accuracy rates. As far as TP rates are concerned, we obtained 97.90% in U2R, 99.60% in R2L and 99.80% in Probe attacks which are considerably good for our mechanism. FN results in U2R, R2L, and Probe found are 0.030%, 0.004% and 0.109% respectively, which are substantially lower rates.

Additionally, FCM–SVM has obtained better precision percentages as (96.20% in U2R, 98.90% in R2L and 99.90% in Probe), Furthermore, the recall recorded percentages are (97.90% in U2R, 99.60% in R2L and 99.80% in Probe respectively). It is important to note that MCC usually gives lower values such as 94.70% in U2R, 91.10% in R2L and 92.50% in Probe. Using the FCM–SVM to classify new incoming traffic is very fast, which is the main benefit of this method. However, we obtained best

Table 3 NSL-KDD dataset features

Serial no	Class	Features
1. Basic features	F1–F9	Duration Protocol Type, Service Flag, Source Bytes, Destination Bytes Land Wrong Fragment, Urgent.
2. Content features	F10–F22	Number Failed Logged In,Number Compromised, Root Shell, Su Attempted,Number Root, Number File Creations,Number Shells, Number Access Files,Number Outbound Cmds, Is Host Login.
3. Same host features	F23–F31	Count, Srv Count, Serror Rate, Rerror Rate, Srverror Rate, Same Srv Rate, Diff Srv Rate, Srv Diff Host Rate.
4. Same services features	F32–F41	Dst Host Count, Dst Host Srv Count, Dst Host Same Srv Rate, Dst Host Diff Srv Rate, Same Scr Port Rate, Dst Host Srv Diff Host Rate, Dst Host Serror Rate, Dst Rate, Dst Host Srv Diff Host Rate, Dst Host Serror Rate, Dst Host Srverror Rate Dst Host Rerro Rate, Dst Host Srverror Rate

Table 4 FCM–SVM classifier for U2R over ML metrics

Evolution	U2R (%)	Evolution	U2R (%)
Accuracy	97.37	Accuracy	98.85
Incorrectly classified instances	02.62	Incorrectly classified instances	1.147
TP rate	97.90	TP	99.80
FN rate	0.030	FN rate	0.109
Precision	96.20	Precision	99.90
Recall	97.90	Recall	99.80
F-measure	97.70	F-measure	99.40
MCC	94.70	MCC	92.50
AUC	99.80	AUC	99.90

Table 5 FCM–SVM classifier for R2L over ML metrics

Evolution	U2R (%)
Accuracy	98.46
Incorrectly classified instances	0.17
TP	99.60
FN rate	0.004
Precision	98.90
Recall	99.60
F-measure	99.30
MCC	91.10
AUC	98.00

Table 6 FCM–SVM classifier for Probe over ML metrics

Evolution	U2R (%)
Accuracy	98.85
Incorrectly classified instances	1.147
TP	99.80
FN rate	0.109
Precision	99.90
Recall	99.80
F-measure	99.40
MCC	92.50
AUC	99.90

performances in AUC for testing with percentages (99.80% in U2L, 98.00% in R2L and 99.90% in Probe).

Tables 7, 8, 9, and 10 shows the comparative results of our proposed mechanism with some of the existing IDS mechanisms under various attacks such as DoS, Probe, U2R, and R2L. These Tables show the performance comparisons of various classification methods such as NB, GA-DT, GHSOM, OAR–SVM, NB-KNNCF, GAR-FOREST and FCM–SVM under different evaluation criterion such as Accuracy, Incorrect Classification rate, FN rate, Precision, Recall, and F1 score for various attacks.

Table 7 DoS comparison of several IDS methods under various ML metrics

Authors	Method	Performance metrics (%)						
		Accuracy	IC rate	FN rate	TP rate	Precision	Recall	1 Score
Eid et al. [26]	NB	98.00	2.00	0.008	99.50	98.90	98.25	98.40
Hassanien et al. [30]	GA–DT	81.97	5.00	0.207	81.74	81.96	81.48	81.69
De la Hoz et al. [27]	GHSOM	87.00	3.60	0.009	78.67	77.15	77.93	77.92
Enache and Patriciu [25]	OAR–SVM	82.68	5.18	0.098	82.14	81.99	81.39	81.30
Pajouh, et al. [31]	NB–kNNCF	82.68	5.18	0.098	82.14	81.99	81.39	81.30
Kanakarajan and Muniasamy [29]	GAR-forest	82.39	5.49	0.016	82.99	81.92	82.33	82.28
Proposed mechanism	FCM–SVM	**99.10**	**0.90**	**0.005**	**99.70**	**99.60**	**99.70**	**99.20**

Table 8 Probe comparison of several IDS methods under various ML metrics

Authors	Method	Performance metrics (%)						
		Accuracy	IC rate	FN rate	TP rate	Precision	Recall	F1 Score
Eid et al. [26]	NB	95.90	4.10	2.800	88.10	96.20	95.90	96.00
Hassanien et al. [30]	GA–DT	64.63	35.37	11.30	61.04	65.42	61.64	64.21
De la Hoz et al. [27]	GHSOM	95.60	4.40	0.071	95.18	95.15	95.18	95.59
Enache and Patriciu [25]	PSO–SVM	98.68	1.32	0.037	98.27	98.11	98.78	98.37
Pajouh, et al. [31]	NB–kNNCF	79.76	20.24	0.166	79.20	79.30	79.50	79.11
Kanakaranjan and Muniasamy [29]	GAR-forest	78.33	21.67	2.02	78.11	78.28	78.27	78.57
Proposed mechanism	FCM–SVM	**98.80**	**1.40**	**1.900**	**99.80**	**99.00**	**99.80**	**99.40**

Table 9 R2L comparison of several IDS methods under various ML metrics

Authors	Method	Performance metrics (%)						
		Accuracy	IC rate	FN rate	TP rate	Precision	Recall	F1 score
Eid et al. [26]	NB	94.1	5.90	0.008	97.00	96.10	95.90	95.90
Hassanien et al. [30]	GA–DT	32.90	67.10	2.081	21.07	24.30	18.00	28.94
De la Hoz et al. [27]	GHSOM	94.38	5.62	0.011	94.28	94.11	94.25	94.29
Enache and Patriciu [25]	PSO–SVM	88.10	11.90	0.286	88.01	85.09	84.62	82.62
Pajouh, et al. [31]	NB–kNNCF	84.68	15.32	0.169	84.19	84.12	84.47	84.52
Kanakarajan and Muniasamy [29]	GAR-forest	78.98	27.02	78.15	78.88	78.10	78.22	78.55
Proposed mechanism	FCM–SVM	**98.46**	**1.35**	**0.400**	**99.60**	**99.60**	**99.30**	**91.10**

The Tables 7 and 8 respectively shows the performance comparisons for high frequent attacks such as DoS and Probe. From the Tables 7 and 8, it can be ob served that FCM–SVM obtains better results compared to existing methods. Furthermore, the Tables 9, and 10 presents the comparison results of low frequent attacks such as R2L and U2R respectively. These Tables exhibit that performance of FCM–SVM is far better than existing techniques.

4.2.1 DoS comparison

The Table 7 shows the performance comparison under different evaluation criterion. The performances of FCM–SVM with various models are compared based on DoS attacks detection rates by using Accuracy, Incorrect Classification rate, FN rate, TP rate, Precision, Recall, and F1 score as presented in Table 7. As compared to existing mechanism, our proposed mechanism obtained best results

Table 10 U2L comparison of several IDS methods under various ML metrics

Authors	Method	Performance Metrics (%)						
		Accuracy	IC rate	FN rate	TP rate	Precision	Recall	F1 score
Eid et al. [26]	NB	95.40	4.60	0.005	97.00	95.70	95.40	95.40
Hassanien et al. [30]	GA–DT	68.35	35.65	2.680	66.28	65.33	68.33	68.13
De la Hoz et al. [27]	GHSOM	93.00	07.00	0.009	93.00	89.94	89.66	89.17
Rastegari et al. [28]	GA	79.91	20.09	8.690	79.50	79.58	79.88	79.21
Pajouh, et al. [31]	NB–kNNCF	67.16	23.84	11.018	67.18	67.01	67.08	67.02
Kanakarajan and Muniasamy [29]	GAR-forest	77.56	0.35	12.581	77.08	77.40	77.19	77.16
Proposed mechanism	FCM–SVM	**97.37**	**02.60**	**0.003**	**97.90**	**96.20**	**97.90**	**97.00**

in terms of accuracy with 99.10% and lowest in terms of incorrect classification rate with 0.90%. Furthermore, our mechanism is lowest in FN rate with 0.005%, highest in TPR rate with 99.70%, and highest in precision rate with 99.60%. While the Recall, FCM–SVM mechanism achieved highest percentage rate with 99.70% than existing mechanisms have achieved. Furthermore, F1 score has obtained 99.20% which is also highest than existing mechanisms.

4.2.2 Probe comparison

The Table 8 depicts the performance comparison of our proposed scheme with existing schemes based on Probe attacks. The performance evaluation of existing mechanisms is compared with our proposed FCM–SVM method by using Accuracy, Incorrect Classification rate, FN rate, TP rate, Precision, Recall, and F1 score as shown in Table 8. Upon comparing our classifier with existing methods, we found that our mechanism has obtained highest results in terms of accuracy with 98.80% and 1.40% lower rate in terms of incorrect classification rate. Furthermore, our classifier archived 1.900% lower in FN, 99.80% higher in TPR and 99.00% higher in precision. While the Recall, FCM–SVM classifier obtained 99.80% than existing methods. Additionally, F1 score obtained 99.40% which is highest than any other classifier.

4.2.3 R2L comparison

The Table 9 shows the performance comparison of existing methods with our FCM–SVM on R2L attacks. The performance evaluation of existing mechanisms is compared with FCM–SVM method based on Accuracy, Incorrect Classification rate, FN rate, Precision, Recall, and F1 score as shown in Table 9. Our mechanism is better than existing mechanisms in terms of accuracy. FCM–SVM is 98.46% highest in terms of accuracy and 1.35% lowest in terms of incorrect classification rate. However, our FCM–SVM is 0.400% which is slightly higher in FN. Nevertheless, 99.60% higher in TPR, and 99.60% higher in precision. Recall has achieved 99.30% than existing mechanisms. However, F1 score has achieved 91.10% than related works.

4.2.4 U2L comparison

The Table 10 depicts the performance comparison of existing methods with our FCM–SVM on U2R attacks. The performances evaluation of existing mechanisms with FCM–SVM method are compared based on Accuracy, Incorrect Classification rate, FN rate, Precision, Recall, and F1 score as shown in Table 10. Lastly, FCM–SVM is also better than existing mechanisms in terms of accuracy. As reported, FCM–SVM is 97.37% higher. Meanwhile, our FCM–SVM is 02.60% lower in terms of incorrect classification rate. Furthermore, FCM–SVM is 0.003% lower in terms of FN, 97.90% higher in terms of TPR, and 96.20% higher in terms of precision respectively. Recall has achieved 97.90% than existing mechanism. F1 score has achieved 97.00% which is also higher than existing techniques.

5 Conclusions

In this work, we propose an improved intrusion detection system for cloud environment. This mechanisms establishes an intrusion detection system at the monitoring layer of the virtual machine. The hypervisor inspector is created using a hybrid approach that combines SVM and FCM clustering. The created model is divided into three phases. The first phase introduces the FCM clustering module that is used to separate big datasets into small clusters to enable the SVM to learn effectively in a timely manner. The fuzzy clustering module enhances the performance of the SVM through this process. In the second phase, different SVM modules are trained in accordance with the assigned cluster values. The third module (fuzzy aggregation module) combines the results of the hypervisor inspector. Here, the proposed technique is compared with existing techniques by using the various evaluation metrics such as accuracy, incorrect classification rate, FN rate, TP rate, precision, recall and F-1 score under various attacks. The performance results of FCM–SVM confirm that it outperforms the existing mechanisms. Hence, the proposed mechanism is suitable for detecting various attacks with high detection accuracy and low false alarm rates, thus can be deployed to detect the anomalies in cloud environment.

Acknowledgements This work was supported by the Faculty of Computer Systems and Software Engineering (FSKKP), Univer-siti Malaysia Pahang (UMP), Malaysia. In collaboration with ST Engineering Electronics-SUTD Cyber Security Laboratory, Singapore University of Technology and Design (SUTD), Singapore.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Velte, A., Velte, T.: Cloud Computing: A Practical Approach. McGraw-Hill, Ney York (2019)
2. Prakash, S.: Role of virtualization techniques in cloud computing environment. In: Bhatia, S.K., Tiwari, S., Mishra, K.K., Trivedi, M.C. (eds.) Advances in Computer Communication and Computational Sciences, pp. 439–450. Springer, Singapore (2019)
3. Bawa, P., Rehman, S., Manickam, S.: Enhanced mechanism to detect and mitigate economic denial of sustainability (EDoS) attack in cloud computing environments. Int. J. Adv. Comput. Sci. Appl. **8**(9), 51–58 (2017)
4. Singh, P., Manickam, S., & Rehman, S.: A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architecture. In: Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization. IEEE pp. 1–4, (2014)
5. Osanaiye, O., Choo, K.K., Dlodlo, M.: Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. J. Netw. Comput. Appl. **67**(1), 147–165 (2016)
6. Kuang, F., Xu, W., Zhang, S.: A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl. Soft Comput. **18**(1), 178–184 (2014)
7. Nkikabahizi, C., Cheruiyot, W., Kibe, A.: Classification and analysis of techniques applied in intrusion detection systems. Int. J. Sci. Eng. Technol. **6**(7), 216–219 (2017)
8. Ghamisi, P., Benediktsson, J.: Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geosci. Remote Sens. Lett. **12**(2), 309–313 (2014)
9. Saljoughi, A., Mehrvarz, M., Mirvaziri, H.: Attacks and intrusion detection in cloud computing using neural networks and particle swarm optimization algorithms. Emerg. Sci. J. **1**(4), 179–191 (2017)
10. Costa, K., Pereira, C., Nakamura, R., Pereira, L., Papa, J.: Boosting Optimum-Path Forest clustering through harmony Search and its applications for intrusion detection in computer networks. In: 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), pp.181-185 (2012)
11. Aljawarneh, S., Aldwairi, M., Yassein, M.: Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J. Comput. Sci. **25**(1), 152–160 (2018)
12. Raja, S., Ramaiah, S.: Performance comparison of neuro-fuzzy cloud intrusion detection systems. Int. Arab J. Inf. Technol. **13**(1A), 142–149 (2016)
13. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Discov. **29**(3), 626–688 (2015)
14. AL-Utrakchi, E., AL-Mousa, M.: Analyzing network traffic to enhance the IDS accuracy using intrusion blacklist. Int. J. Comput. Sci. Inform. Secur. **15**(1), 46–47 (2017)
15. Kenkre, P., Pai, A., Colaco, L.: Real time intrusion detection and prevention system. In: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 405–411 (2015)
16. Saied, A., Overill, R., Radzik, T.: Detection of known and unknown DDoS attacks using Artificial Neural Networks. Neurocomputing **172**(1), 385–393 (2016)
17. Freedman, A. T., Pye, I. G., Ellis, D. P., Applegate, I.: Network monitoring, detection, and analysis system. U.S. Patent 9,942,253, issued April 10 (2018)
18. Rosli, A., Taib, A., Ali, W.: Utilizing the enhanced risk assessment equation to determine the apparent risk due to user datagram protocol (UDP) flooding attack. Sains Hum. **9**(1), 1–4 (2017)
19. Kaur, G., Saxena, V., Gupta, J.: Detection of TCP targeted high bandwidth attacks using self-similarity. J. King Saud Univ.-Comput. Inform. Sci. **49**, 105–110 (2017)
20. Kumar, D.: DDoS attacks and their types. In: Network security attacks and countermeasures. IGI, Global (2016). https://doi.org/10.4018/978-1-4666-8761-5.ch007
21. Suhasaria, P., Garg, A., Agarwal, A., Selvakumar, K.: Distributed denial of service attacks: a survey. Imp. J. Interdiscip. Res. **3**(2), 71–80 (2017)
22. Bhushan, K., Gupta, B.: Security challenges in cloud computing: state-of-art. Int. J. Big Data Intell. **4**(2), 81–107 (2017)
23. Hota, H.S., Shrivas, A.K.: Data mining approach for developing various models based on types of attack and feature selection as intrusion detection systems (IDS). In: Mohapatra, D., Patnaik, S. (eds.) Intelligent computing, networking, and informatics. Advances in intelligent systems and computing, vol. 243. Springer, New Delhi (2014). https://doi.org/10.1007/978-81-322-1665-0_85
24. Pervez, M., Farid, D.: Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA). IEEE, pp. 1–6 (2014)
25. Enache, A.C., Patriciu, V.: Intrusions detection based on support vector machine optimized with swarm intelligence. In: 9th international symposium on applied computational intelligence and informatics (SACI). IEEE, pp. 153–58 (2014)
26. Eid, H., Darwish, A., Hassanien, A., Kim, T.H.: Intelligent hybrid anomaly network intrusion detection system. In: International Conference on Future Generation Communication and Networking, pp. 209–218 (2011)
27. De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., Martínez-Álvarez, A.: Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organizing maps. Knowl.-Based Syst. **71**, 322–338 (2014)

28. Rastegari, S., Hingston, P., Lam, C.P.: Evolving statistical rule-sets for network intrusion detection. Appl. Soft Comput. **33**, 348–359 (2015)
29. Kanakarajan, N., Muniasamy, K.: Improving the accuracy of intrusion detection using GAR-Forest with feature selection. In: Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA), pp. 539–547 (2016)
30. Hassanien, A., Kim, T.H., Kacprzyk, J., Awad, A.: Bio-inspiring cyber security and cloud services: trends and innovations. Springer, New York (2014)
31. Pajouh, H., Dastghaibyfard, G., Hashemi, S.: Two-tier network anomaly detection model: a machine learning approach. Jo. Intell. Inform. Syst. **48**(1), 61–74 (2017)
32. Pandeeswari, N., Kumar, G.: Anomaly detection system in cloud environment using fuzzy clustering-based ANN. Mob. Netw. Appl. **21**(3), 494–505 (2016)
33. Ingre, B., & Yadav, A.: Performance analysis of NSL-KDD dataset using ANN. In: International Conference on Signal Processing and Communication Engineering Systems, pp. 92–96 (2015)
34. Bamakan, S., Wang, H., Yingjie, T., Shi, Y.: An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing **199**, 90–102 (2016)
35. Raman, M., Somu, N., Kirthivasan, K., Sriram, V.: A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems. Neural Netw. **92**, 89–97 (2017)
36. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A detailed analysis of the KDD CUP 99 data set. In: 2009 Symposium on Computational Intelligence for Security and Defense Applications. IEEE, pp. 1–6 (2009)
37. Revathi, S., Malathi, A.: A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int. J. Eng. Res. Technol. (IJERT) **2**(12), 1848–1853 (2013)
38. Zadeh, L.: Fuzzy logic: a personal perspective. Fuzzy Sets Syst. **281**, 4–20 (2015)
39. Weka Simulation: Weka 3 Machine Learning Software in Java. University of Waikato. https://www.cs.waikato.ac.nz/ml/weka/ (2019). Accessed 16 Mar 2019

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Aws Naser Jaber received the M.S. degrees in Advanced Computer Networks from the University of Science Malaysia (USM), and PhD in Computer networks and security from Universiti Malaysia Pahang (UMP), Malaysia. His research interests include Network security, Cryptography and Artificial Intelligence and bioinformatics.

Shafiq Ul Rehman has M.Sc. in Network Technology and Management from Amity university, India. He received a Ph.D. degree in Cyber Security from the University of Science Malaysia (USM), Malaysia. He is currently working as a Post-doctoral Research fellow at ST Engineering Electronics-SUTD Cyber Security Laboratory, Singapore University of Technology and Design (SUTD). He has authored and co-authored several conference and journal publications. He is involved with various research projects related to the fields of networking and communication security, Internet and other emerging technologies. His current research interests include Cyber Security, Artificial Intelligence, Internet of Things, Cloud Computing, and IPv6.