Cloud Attacks Detection System for Cloud Load Balancing

Swathi Sambangi[1†] and Lakshmeeswari Gondi[2††],

1,2Department of Computer Science & Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, INDIA
1 Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, INDIA

Summary
One of the most recent problems of interest in cloud computing is securing the cloud networks by identification and mitigation of cloud network attacks such as the distributed denial of service attacks. By deploying efficient IDS in cloud networks which we term in this paper as the Cloud IDS, we can achieve cloud load balancing. For this, in this paper, we propose a machine learning based IDS for deployment in cloud networks. Our machine learning method has two stages. In the first stage, the traffic dimensionality is reduced by the proposed method. In the second stage, the network traffic classification is carried by the proposed network traffic similarity function. For experimental analysis, we have used CICIDS 2019 traffic data. The results proved that the proposed method has performed substantially better to state-of-art machine learning classifiers.

Key words:
Cloud, DDoS attacks, Classification, Prediction, Intrusion detection, anomaly detection.

1. Introduction

Machine learning and deep learning techniques are applied extensively in various real-world applications related to several application areas. One of the recent applications of machine learning is in securing cloud services and applications through securing cloud networks. Cloud security is an essential requirement to provide continuous availability and accessibility of cloud resources and cloud services to cloud users. If cloud networks are not secured then, network attacks can eventually affect cloud users and cloud service providers. One way to secure cloud networks is via building and deploying efficient cloud intrusion detection systems (CIDS) in cloud networks at appropriate network points. The deployment of Cloud IDS in cloud networks can also help to achieve cloud load balancing by restricting superfluous voluminous network attack traffic through capturing malicious network traffic. For example, distributed DDoS attacks are most challenging network attacks in cloud environments which not only affect cloud services but also cloud service users and providers technically and economically. Thus, restricting DDoS network attacks by detection of these network attacks through Cloud IDS deployment in cloud networks makes cloud resources and services available for its legitimate users by securing cloud networks. In general, one can achieve cloud load balancing via securing cloud networks through detection of network attacks that are targeted by cloud attackers to make cloud services unavailable to legitimate users of cloud. In this paper, we propose a Cloud IDS system that can be deployed in cloud networks for cloud network attack detection to achieve cloud load balancing. By cloud load balancing, we mean restricting sudden rise for voluminous cloud resource requirements due to over flooding of cloud network traffic.

An important step before making any analysis on network traffic data is network traffic feature extraction. In general, feature extraction is an important step in retrieval of interested and relevant network features when the raw network data is used to identify malicious network traffic. Given, raw network data such as pcap files which represent the unstructured network traffic, such a raw network traffic data can be converted into a structured format by carrying extraction of various network traffic features from the raw network traffic data. In our case, this process is called as feature extraction. The structured network traffic data can be in the form of a CSV. By network feature (or network attribute), we mean a dimension or a variable parameter which is related to network traffic instance. For example, a network traffic instance can usually be in the form of packets or flow. Network traffic features may include source ip address, destination ip address, network protocol, timestamps, ports, flags etc. Usually, pcap files may contain information which is not necessary for analysis or is not of interest for network analysts. The pcap format is one of the standard formats for capturing network traffic data. However, network traffic data in pcap format is not suitable for statistical analysis because it contains only raw data. Hence, the focus of feature extraction is to retrieve required traffic feature set from the network traffic flow. For example, in this paper we have considered the CICIDS 2019 dataset for performance analysis. Using CICFlowmeter [31], we have extracted 80 features and the resulting traffic data is stored in CSV format.

1.1 Load Balancing in Cloud Computing

Most of the cloud outages occur due to unsteady load on the data center of a cloud service provider depending on the requests raised by the number of increasing users who

Manuscript received May 5, 2022
Manuscript revised May 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.5.100

utilize the cloud for their application deployment or to make use of any other benefit that could be provided by the cloud. Still, several challenges exist in cloud computing among which load balancing cannot be avoided. Load balancing in the cloud deals with the even distribution of tasks/workload among the resources available. The main benefits of balancing the load among the VMs helps to increase effective resource utilization and reduce energy consumption to reduce the cost of tasks performed. Also, assigning the load evenly among virtual machines in the cloud helps the cloud service provider to improve scalability and reliability which in turn makes the cloud maintain QoS level agreements. It also helps to make the cloud servers highly available by reducing service downtime. Another view to achieve cloud load balancing is via securing cloud networks through detection of network attacks that are targeted by cloud attackers to make cloud services unavailable to legitimate users of cloud. In this paper, our aim is to secure cloud networks to achieve cloud load balancing.

1.2. Related Works

In this section, some of the related works that have been interesting during literature study and have paved way for this research are presented. An important challenge before cloud service providers is to provide uninterrupted service to cloud users and this makes securing cloud networks one of the implicit concerns. Recently, several studies are carried on providing security to cloud networks. The studies suggest various IDS methods, models to secure cloud networks from network attackers by utilizing statistical, data mining, machine learning and deep learning techniques. One common task irrespective of any technique applied for detection of network attacks is the distance computation between network traffic instances. In distance-based computations, a traffic instance is considered same as another traffic instance if the distance is minimal. In case, similarity operation is carried then, the similarity value must be maximum [9]. Jiang et al. [1], has proposed a dimensionality reduction method for text documents. The method is based on text feature reduction by carrying clustering. Text classification is then achieved by using dimensionality reduced text documents. To carry clustering, Jiang et al. [1] has applied the gaussian function to obtain membership value. Text processing is a data mining or machine learning task which requires performing similarity computations. Jiang and Lee et al. [2], have proposed a novel similarity measure to carry text processing. Lin, Jiang, and Lee et al. [3] have then extended their previous work [2] carried for text processing and come out with a similarity measure which considered of text features to carry similarity computations. Many times, text documents can belong to more than one category. A classification approach which considers categorizing text documents into more than one category is important in such situations. For this, a fuzzy method is proposed by Lee et al. [4] to achieve multi-label text document categorization. Regression analysis technique is applied in early studies in various statistical, mathematical, data mining and machine learning applications related to detection of network attacks. However, the studies did not discuss visualization importance. Multiple linear regression concept which is one of the statistical analysis techniques is chosen in the study by Swathi et al. [5][6][7]. In these studies, Swathi et al. throws light on visualization importance in understanding regression outcome for detection of network attacks. In [5,6,7], the research study utilized CICIDS 2017 dataset and CICIDS 2019 dataset for regression analysis. Apart from showing visualization of regression analysis, the complexity of dataset is also studied [7] by utilizing Andrew's curve plot for dataset non-linearity visualization. DDoS attacks are a sort of cyberwarfare wherein cloud users and applications are prevented from utilizing cloud network infrastructure. One of the recent studies that has proposed a mathematical model for DDoS attacks identification is by Kumari et al. [8]. The performance of the model is studied on CAIDA 2007 dataset. In [8], two ML algorithms namely naïve bayes and logistic regression are considered for study of the mathematical model. Although, [8] is one of the recent studies but its limitation is w.r.t dataset. The CAIDA 2007 dataset considered for evaluation of the mathematical model is not a recent dataset that resemble modern network traffic. Swathi et al. [9] proposed a ML model for attack detection and tested their model on CICIDS 2019 dataset. In [10], ML techniques are applied for malicious attack detection in cloud. In [11] Aljawarneh et al. gives a method for feature representation to carry anomaly detection. The anomaly detection method proposed by [11] utilizes a new distance function and applies this distance function on traffic data to identify anomaly traffic. The dataset used in their study is KDD and NSL-KDD dataset. In the literature, many of the existing methods have applied statistical, data mining and machine learning methods for intrusion and anomaly detection on IOT datasets. Hussain et al. [12] proposed to convert the network traffic into image equivalent and then apply the deep learning methods for performance evaluation on the IoT DoS and DDoS dataset. The study by [12] makes use of the deep learning technique to improve the prediction accuracy of network attacks. The dataset utilized in the study [12] is made available at the IEEE Dataport [13] for researchers to evaluate the performance of existing or new methods and algorithms. Usually, in network environments, sometimes it is possible to collect the network data and in such cases some attribute values may be lost or miss. To address the missing values in network traffic data, Vangipuram et al. [14] proposed an approach to impute missing traffic data values. For imputation of

missing traffic attribute value, a similarity measure is proposed to identify nearest traffic record. Pranitha et al. [15] and Swathi et al. [16] provides image datasets which represent network traffic in binary visualization form for Cloud and SDN environments. These datasets may be used to study the performance of various machine learning techniques. In [17], authors propose a temporal distance measure for obtaining the distance between two temporal patterns whose prevalence values are distributed across various time points. Such distance measures can be also applied in other fields and areas wherever similarity and distance calculations are needed by suitably designing and fitting to our requirements.

Cloud service providers are constantly facing important threats such as DDoS attacks. For instance, volumetric DDoS attacks are an important type of DDoS attacks that bring complex challenges to identify and mitigate them. Clement et al. [18] presents an overview of various attacks that have targeted OVHcloud infrastructure in 2021. It is observed that HTTP based services are targeted by attackers using TCP. Similarly, in the case of video games, it is UDP which is utilized by attackers to make attacks. [18] lists various attacks that have targeted OVHcloud infrastructure over one year in 2021. Gopal Singh et al. [19] proposed a hybrid model for detecting DDoS attacks. The model is based on extreme ML and adaptive differential evolution. The model is tested on three datasets (ISCX IDS 2012, NSL-KDD and CICIDS001). The hybrid model [19] showed 91.46% detection accuracy on ISCX IDS 2021, 97.23% detection accuracy on NSL-KDD and 99.28% for CICIDS001 dataset. The sensitivity of the hybrid model is obtained as 82.98%, 96.07% and 100% while the specificity is obtained as 99.97%, 98.5% and 99.96% respectively. In [20], Britto et al. discussed the need for EDoS and DDoS attacks detection in cloud setting. The challenge is the selection of appropriate traffic features for precise classification. The study by Britto et al. suggests utilization of deep learning techniques for obtaining higher detection accuracy. In [21], a novel robust cloud IDS solution is proposed which utilizes deep neural networks concept for anomaly detection (low-rate attacks and application layer attacks). The solution comprises of two deep generative models CDAAE and CDAAE-KNN. In [22], an intelligent IDS model which employs feature reduction is proposed for detection of exploitation attacks and reflection attacks in cloud. The model is tested for its performance on CICIDS 2019 dataset with J48 classifier. After feature reduction, the features are reduced by a minimum of 56% and a maximum of 82.92%. The performance is compared for binary and multiclass classification. Other related studies on similarity measures and cloud attack detection include [23-40].

2. Traffic Feature Similarity Function

The proposed traffic attribute similarity function is given by equation (1).

$$\mathcal{S}\text{wathi}\left(X_{(N_{ai})}, X_{(N_{aj})}\right) = \frac{\frac{\sum_{k=1}^{k=m} \mathcal{S}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)}{\sum_{k=1}^{k=m} \mathcal{T}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)} + \delta}{1 + \delta} \quad (1)$$

Where

$$\mathcal{S}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right) = \begin{cases} \left(1 - \exp^{-\left[\frac{1-\left(\Pr\left(\frac{N_{ai}}{C_q}\right)-\Pr\left(\frac{N_{aj}}{C_q}\right)\right)}{\sigma_g}\right]^2}\right) & ; \Pr\left(\frac{N_{ai}}{C_q}\right) \neq 0 \text{ and } \Pr\left(\frac{N_{aj}}{C_q}\right) \neq 0 \\ -\left(1 - \exp^{-\left[\frac{\Pr\left(\frac{N_{ai}}{C_q}\right)-\Pr\left(\frac{N_{aj}}{C_q}\right)}{\sigma_g}\right]^2}\right) & ; \Pr\left(\frac{N_{ai}}{C_q}\right) = 0 \text{ and } \Pr\left(\frac{N_{aj}}{C_q}\right) \neq 0 \\ -\left(1 - \exp^{-\left[\frac{\Pr\left(\frac{N_{ai}}{C_q}\right)-\Pr\left(\frac{N_{aj}}{C_q}\right)}{\sigma_g}\right]^2}\right) & ; \Pr\left(\frac{N_{ai}}{C_q}\right) \neq 0 \text{ and } \Pr\left(\frac{N_{aj}}{C_q}\right) = 0 \\ 0 & ; \Pr\left(\frac{N_{ai}}{C_q}\right) = 0 \text{ and } \Pr\left(\frac{N_{aj}}{C_q}\right) = 0 \end{cases} \quad (2)$$

$$\mathcal{T}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right) = \begin{cases} 0 & ; \Pr\left(\frac{N_{ai}}{C_q}\right) = 0 \text{ and } \Pr\left(\frac{N_{aj}}{C_q}\right) = 0 \\ 1 & ; \text{else} \end{cases} \quad (3)$$

In eq. (1), $\frac{\sum_{k=1}^{k=m} \mathcal{S}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)}{\sum_{k=1}^{k=m} \mathcal{T}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)}$ symbolizes the normalized similarity between two traffic attribute pattern vectors $\mathbf{X_{(N_{ai})}}$ and $\mathbf{X_{(N_{aj})}}$. By solving analytically, we get $\boldsymbol{\delta}$ value equal to 0.6321. In eq. (2), $\boldsymbol{\mathcal{S}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)}$ denotes the membership similarity between the kth element of the m-dimensional traffic attribute pattern vectors $\mathbf{X_{(N_{ai})}}$ and $\mathbf{X_{(N_{aj})}}$. The notation $\mathbf{Pr}\left(\frac{\mathbf{N_{ai}}}{\mathbf{C_q}}\right)$ denotes likely chance of attribute $\mathbf{N_{ai}}$ to be associated to class $\mathbf{C_q}$. In eq. (3), $\boldsymbol{\mathcal{T}^{k}\left(X_{(N_{ai})}, X_{(N_{aj})}\right)}$ is used to infer whether the kth element of the m-dimensional traffic attribute pattern vectors $\mathbf{X_{(N_{ai})}}$ and $\mathbf{X_{(N_{aj})}}$ is considered to determine membership between traffic attribute pattern vectors $\mathbf{X_{(N_{ai})}}$ and $\mathbf{X_{(N_{aj})}}$.

3. Proposed Research

This section outlines the proposed research from viewpoints of the problem definition, research gaps, research objectives and the proposed ML model which can be considered for cloud load balancing by detection of DDoS and other network attacks.

3.1 Problem Definition

Let 'N' be the total number of network packet traffic instances wherein each traffic packet instance is captured over 'M' number of attributes. Suppose that, set of all these M-dimensional network traffic instances are represented as a network traffic dataset, denoted by NTDS consisting of 'L' class labels such that each traffic instance is categorized into one of the L' class labels. The objective is to propose a machine learning model-based CLOUD IDS which applies proposed similarity measures to detect network attacks in incoming network traffic with better accuracy, precision, and detection rates.

3.2 Research Gap

Some of the gaps in the present research studies are as mentioned below.

(i) Most of the existing studies have not tested the performance of the ML models on network traffic data which near resembles modern network traffic.

(ii) The availability of a proper benchmark dataset in public domain which resembles modern network traffic is an important limitation. Even if such a dataset is available, the availability of a sufficient number of attack traffic and normal traffic is another issue which forms limitation of most of the existing studies.

(iii) Network traffic datasets are not captured with enough traffic features. Hence, most of the existing benchmark datasets do not match real time network traffic characteristics. Hence, the performance of ML models which are built by using these datasets is another issue.

(iv) Existing studies on cloud attack detection did not propose similarity-based network traffic feature reduction methods.

3.3 Research Objectives

The research objectives of the proposed research for cloud load balancing by detection of DDoS and other modern network attacks are outlined below.

(i) Minimization of the network traffic data non-linearity

(ii) Represent network traffic data instances in optimal form by projecting them on to suitable data plane.

(iii) Discovery of network traffic features that are similar w.r.t a chosen similarity constraint. Then, use the similarity information obtained from these network traffic features to carry feature reduction.

(iv) An attack detection system which has a better detection rate than existing ML classifiers even for modern network traffic.

3.4 Proposed ML model for Cloud Load Balancing

The architecture of the proposed machine learning model for cloud load balancing is depicted using fig 2. The working of the ML model is explained below.

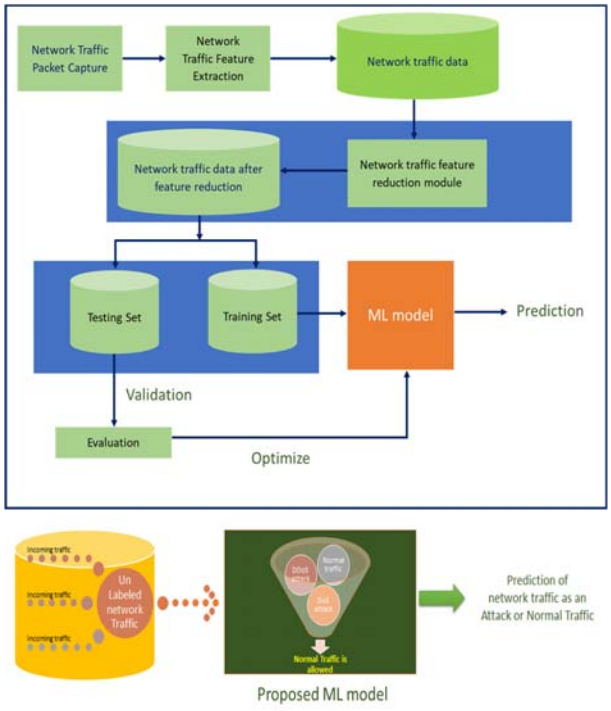

Fig. 1 Proposed Architecture for Cloud Load Balancing

Initially, to build the machine learning model, the network traffic data is captured and stored in pcap format. The traffic data which is stored in pcap format is then utilized to extract various traffic features. The resulting network traffic features which are retrieved from pcap files are then stored as in csv format. So, each traffic instance is now a multi-dimensional vector of network traffic attributes that are obtained after feature extraction. This traffic data is then fed to the feature reduction module which applies the proposed feature reduction technique. The output of the

feature reduction module is the network traffic which is high dimensional network traffic data which is projected on to a low dimensionality space. The low dimensionality network traffic is denoted as network traffic data after feature reduction in Figure 1. The feature reduced network traffic is then split into training traffic set and testing traffic set. The model is trained using training set and validated using testing set. During training, we choose to apply k-fold cross validation. The testing traffic is unseen network traffic during training phase. In case, ML model performance requires further improvement then, the model can be optimized by tuning the model's Hyperparameters. Finally, the ML model so built can be deployed for real time traffic monitoring and validation. However, there is always a scope to iteratively improve model performance as and when we have new real world traffic data captured and new attacks identified. In this paper, we employ the procedure discussed in [9] to evaluate the model on CICIDS2019 dataset by considering the problem as binary classification problem and using balanced dataset.

4. Performance Evaluation Metrics

The performance evaluation of a machine learning classifier can be carried using various performance evaluation metrics such as accuracy, precision, specificity, sensitivity, F-score, area under curve, balanced accuracy which can be obtained from the confusion matrix. Consider the confusion matrix representation depicted in Fig. 2. The confusion matrix consists of four elements namely, True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

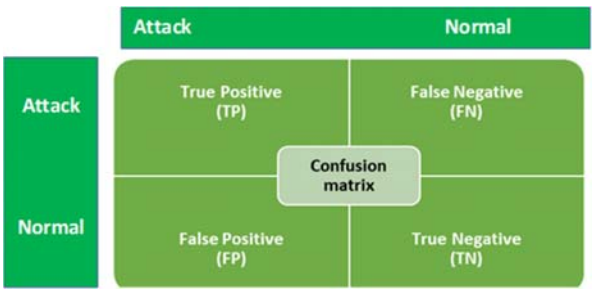

Fig. 2 Classifier Confusion matrix.

When a network traffic instance is an attack instance, and it is predicted as an attack then, it is said to be a true positive (TP). When a network traffic instance is an attack instance, and it is predicted as normal traffic then it is said to be a false negative (FN).

Alternately, when a network traffic instance is a normal traffic instance, and it is predicted as an attack then it is said to be a false positive (FP). When a network traffic instance is a normal traffic instance, and it is predicted as normal traffic then it is said to be a true negative (TN).

4.1 Accuracy

Accuracy is a classifier evaluation metric which reflects the proportion of correctly classified instances output by the classifier to the total number of problem instances.

In terms of confusion matrix, accuracy is defined as ratio of the sum of true positive traffic instances and true negative traffic instances to the total number of traffic instances.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

where

TP represents that attack traffic is predicted as an attack.

TN represents that a normal traffic is predicted as a normal.

FP represents that a normal traffic is predicted as an attack.

FN represents that attack traffic is predicted as a normal.

This is suitable to evaluate the performance of binary and multi-classifier models. However, this metric does not reflect the classifier performance when the dataset is either highly imbalanced or highly skewed in nature.

But most of the real-life datasets are highly imbalanced in nature. For example, it is quite difficult to get balanced medical datasets, network traffic datasets, and financial datasets in real life scenario.

4.2 Precision

Precision refers to the ratio of total number of attack traffic samples that are indeed predicted as attack traffic to the total number of traffic samples that are predicted as attack traffic.

Mathematically, the precision is defined as ratio of true positives to the sum of true positives and false positives.

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

where

TP represents that attack traffic is predicted as an attack.

FP represents that a normal traffic is predicted as an attack.

Thus, precision metric is a function of true positives and false positives, and it neglects false negatives which is evident from equation (2). This metric reflects how better

is the prediction performance of the machine classifier under evaluation w.r.t a given class.

4.3 Sensitivity or Recall

Sensitivity refers to the ratio of the number of attack traffic samples that are indeed predicted as attack traffic to the total attack traffic samples that are predicted as attack traffic and normal traffic.

In terms of confusion matrix, Sensitivity is defined as the ratio of true positives to sum of true positives and false negatives.

$$Sensitivity = \frac{(TP)}{(TP + FN)} \quad (3)$$

where

TP represents that attack traffic is predicted as an attack.

FN represents that attack traffic is predicted as a normal.

Sensitivity is a function of true positives and false negatives, and it neglects false positives.

4.4 F1-score

F1-score refers harmonic mean of precision and recall.

$$F1 - score = \frac{2 * precision * recall}{(precision + recall)} \quad (3)$$

From eq. (4), it reflects that F1-score is a function of precision and recall. Thus, both false positives and false negatives are considered in F1-score.

5. Working Example

In this section, a working example is demonstrated to show how the proposed approach [9] performs feature reduction and the way prediction of network traffic is carried. For this, we consider a sample training dataset with 10 network traffic instances. Each network traffic instance is defined in terms of 9 traffic attributes. Consider the sample network traffic data shown in Table.1. These 10 network traffic instances are categorized into two traffic classes (i) attack traffic and (ii) normal traffic. Initially, the idea is to perform feature reduction of the training data and then use the feature reduction training data to perform prediction of unseen test traffic data. For validation of the proposed approach, a sample test traffic dataset consisting of 5 traffic instances is considered. In the test data considered to validate the proposed approach, all the traffic instances are attack traffic. In the proposed approach, we obtain two types of transformation matrices wherein the first one is known as hard membership matrix and the later one is known as soft membership matrix.

Table.1 Sample network traffic data for binary classification

	A1	A2	A3	A4	A5	A6	A7	A8	A9	Traffic Class
T1	64	32	0	121	93	50	135	53	133	Attack traffic
T2	109	72	52	57	71	57	111	92	121	Attack traffic
T3	111	92	123	93	50	92	135	72	52	Attack traffic
T4	64	36	135	57	80	92	109	50	50	Attack traffic
T5	123	36	0	85	25	25	50	85	50	Attack traffic
T6	11	92	135	32	80	29	64	135	37	Normal traffic
T7	42	50	121	93	71	85	50	29	50	Normal traffic
T8	135	92	50	93	50	68	50	92	50	Normal traffic
T9	50	25	50	25	93	93	0	50	50	Normal traffic
T10	50	128	72	93	93	93	107	85	94	Normal traffic

Table 2 depicts the network traffic attribute pattern vectors computed from Table 1. The network traffic attribute pattern vectors are two dimensional vectors representing their chance to belong to attack traffic and normal traffic respectively.

Table.2 Network Traffic Attribute Pattern Vectors

Network Traffic Attribute Pattern Vectors	
α_1	(0.6205, 0.3794)
α_2	(0.4091, 0.5908)
α_3	(0.4200, 0.5799)
α_4	(0.5514, 0.4485)
α_5	(0.4518, 0.5481)
α_6	(0.4619,0.5380)
α_7	(0.6658, 0.3341)
α_8	(0.4737, 0.5262)
α_9	(0.5909, 0.4090)

Table.3 Mean and Deviation of first cluster

Cluster-1(α_1,α_4,α_7,α_9)	
Mean (α_1,α_4,α_7,α_9)	(0.6071, 0.3928)
Dev (α_1,α_4,α_7,α_9)	(0.5482, 0.5482)

The hyperparameter values chosen are 0.9999 for similarity threshold and 0.3113 for gaussian deviation.

Table.4 Mean and Deviation of second cluster

Cluster2(α_2,α_3,α_5,α_6, α_8)		
Mean (α_2,α_3,α_5,α_6, α_8)	0.44336	0.55664
Dev (α_2,α_3,α_5,α_6, α_8)	0.527639	0.527639

Table.5 Network traffic attribute pattern vector soft membership to two generated clusters

	Soft Membership to cluster 1	Soft Membership to cluster 2
α_1	0.977919	0.970966
α_2	0.961349	0.982684
α_3	0.963166	0.982919
α_4	0.976701	0.978691
α_5	0.967842	0.983096
α_6	0.969141	0.982993
α_7	0.976565	0.963642
α_8	0.970533	0.982776
α_9	0.977884	0.974764

Table.6 Network traffic attribute pattern vector hard membership to two generated clusters

	Hard Membership to cluster 1	Hard Membership to cluster 2
α_1	1	0
α_2	0	1
α_3	0	1
α_4	0	1
α_5	0	1
α_6	0	1
α_7	1	0
α_8	0	1
α_9	1	0

Table 3 depicts the mean of the first cluster and its standard deviation. Similarly, Table 4 depicts the mean of the second cluster and its standard deviation. The first cluster consists of 1st, 7th and 9th traffic attribute pattern vectors and the second cluster consists of 2nd, 3rd, 4th, 5th, 6th and 8th network traffic attribute pattern vectors. The similarity information of these traffic pattern vectors w.r.t generated clusters is represented by Table 5. Thus, Table 5 denotes the network traffic attribute pattern vector soft membership to two generated clusters. Hard membership of the traffic attribute patterns to clusters are represented by Table 6.

Table.7 Network traffic dataset after feature reduction

	D1	D2	Traffic Class
T1	332	349	Attack traffic
T2	341	401	Attack traffic
T3	298	522	Attack traffic
T4	223	450	Attack traffic
T5	223	256	Attack traffic
T6	112	503	Normal traffic
T7	142	449	Normal traffic
T8	235	445	Normal traffic
T9	100	336	Normal traffic
T10	251	564	Normal traffic

Table.8 Sample testing network traffic belonging to attack traffic

	A1	A2	A3	A4	A5	A6	A7	A8	A9	Traffic Class
T11	124	50	50	67	87	129	124	134	50	Attack traffic
T12	50	124	124	124	62	129	124	0	124	Attack traffic
T13	124	0	103	25	50	84	124	124	124	Attack traffic
T14	103	126	3	107	87	84	118	132	104	Attack traffic
T15	42	112	91	62	134	42	41	64	106	Attack traffic

Table.9 Dimensionality reduced sample testing network traffic belonging to attack traffic after feature reduction

	D1	D2	Traffic Class
T11	298	517	Attack traffic
T12	298	563	Attack traffic
T13	372	386	Attack traffic
T14	325	539	Attack traffic
T15	189	505	Attack traffic

The traffic data obtained after carrying feature reduction is denoted by Table 7. This traffic data representation is used by ML model to predict the class of testing traffic instances which are fed during testing phase for validation. Table 8 and Table 9 represents the testing traffic data before and after feature reduction. The traffic data represented by Table 9 is used by ML model to predict whether the network traffic is attack or normal.

For sake of simplicity, let us consider the Euclidean distance function to determine to which category of network traffic the testing traffic are maximally similar. In the present case, the distance between T11 to T3 is equal to 5 which is the minimum distance when compared to all other traffic instances. Hence, the traffic instance T11 is categorized as attack traffic. Similarly, the distance between T12 and T3 is the minimal distance when compared to distance w.r.t remaining training instances. Hence, the traffic instance T12 is categorized as attack traffic. In similar lines, all other testing traffic instances T13, T14 and T15 are categorized as attack traffic.

6. Visualization of Network Traffic Data

This section outlines the Andrews curves visualization to understand the non-linearity in sample training and testing traffic data. Andrews curves visualization plot can be used to analyze the non-linearity degree in network traffic data before and after feature reduction.

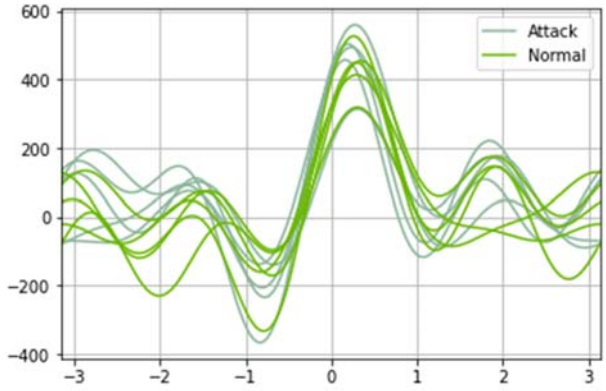

Fig. 3 Visualization of non-linearity of training network traffic data before feature reduction using Andrews Curves

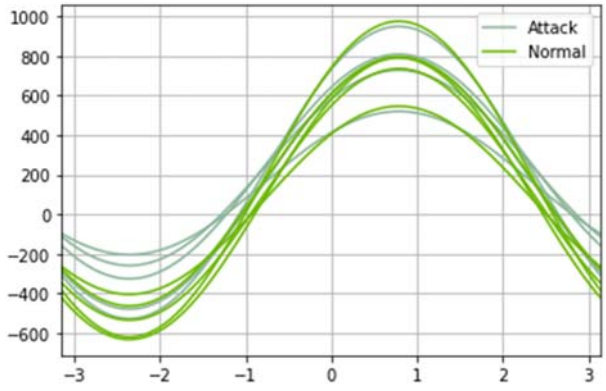

Fig. 4 Visualization of non-linearity of training network traffic data after feature reduction using Andrews Curves

Fig. 3 and Fig. 4 shows the Andrews curves plot which is obtained by considering the training network traffic data before carrying feature reduction and after feature reduction respectively.

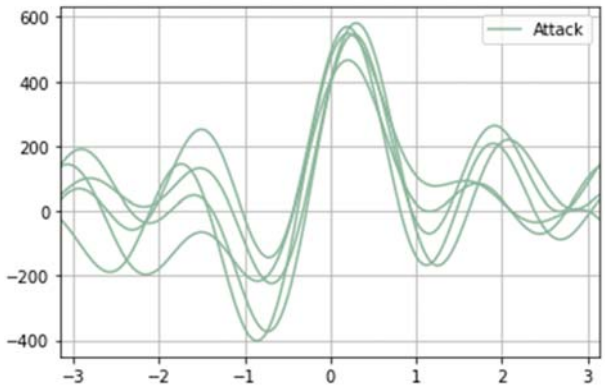

Fig. 5 Visualization of non-linearity of testing network traffic data before feature reduction using Andrews Curves

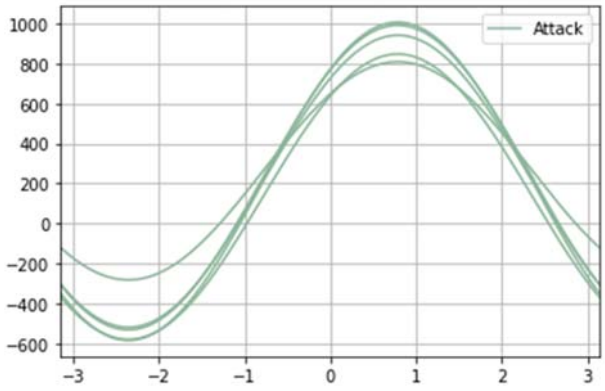

Fig. 6 Visualization of non-linearity of testing network traffic data after feature reduction using Andrews Curves

In similar lines, Fig. 5 and Fig. 6 shows the Andrews curves plot which is obtained by considering the testing network traffic data before carrying feature reduction and after feature reduction respectively.

In both cases, it can be visualized that the high non-linearity that existed in the training traffic data before reduction is reduced after carrying feature reduction. The Andrews curves visualization plot of Fig. 3 and Fig. 4 proves this fact. On comparing Andrews curves visualization plot of Fig. 5 and Fig. 6, we can infer that the non-linearity that existed in the testing traffic data before reduction is reduced after carrying feature reduction using proposed approach.

7. Experiment Results and Discussions

This section outlines the experimental results using proposed approach. Experiments are done on a system with Intel(R) Core (TM) i7-9700F CPU @ 3.00GHz 3.00 GHz Processor with 16GB installed RAM, 64-bit operating system, x64-based processor. The machine learning model is developed using Java and a GUI is built to input parameters.

Initially, the learning model in our previous work [9] is trained on a training dataset which consisted 5000 benign traffic and 5000 DDoS traffic instances. After the model is trained, then the model is tested on a test dataset which consisted 583 benign traffic instances and 583 DDoS attack traffic instances. For this, in the proposed method, the similarity threshold is set to 0.9999 and deviation is set to 0.001. The confusion matrix obtained for test dataset is shown in Table.10. In this case, all 584 normal traffic instances are classified as normal. Out of 584 attack traffic instances 2 traffic instances are predicted as benign traffic and remaining 582 are predicted correctly as attack traffic.

Table.10 Confusion matrix for test data

	Benign	Attack
Benign	584	0
Attack	2	582

Table.11 Prediction Values of ML classifier for test data used

Parameter	Normal traffic	DDoS traffic
Accuracy	99.82%	99.82%
Precision	99.65%	100%
Sensitivity	100%	99.65%
Specificity	99.65%	100%
FPR	0.0034	0
F-score	0.9982	0.9982

The testing results obtained for DDoS attack traffic recorded an accuracy (99.82%), precision (100%), sensitivity or detection rate (99.65%) and specificity (100%), FPR (0), F-score (0.9982). In case of Benign traffic, the values for accuracy, precision, sensitivity, specificity, F-score are obtained as 99.82%, 99.65%, 100%, 99.65% and 0.9982 as shown in Table.11.

In another test scenario, experiment is conducted on unseen test instances consisting of 584 normal and 584 attack traffic instances. So, when experiment is conducted by choosing similarity threshold (0.95) and deviation (0.5), the confusion matrix is obtained as shown in Table 12. In this case, all the 584 normal traffic instances present in the test set are predicted as normal traffic. In the case of attack traffic, out of 584 instances, 29 are predicted as benign and remaining 555 are predicted as attack traffic.

Table.12 Confusion matrix for test data

	Benign	Attack
Benign	584	0
Attack	29	555

On test dataset, for benign traffic, the accuracy is 97.52% and precision is 95.27%. For attack traffic, the accuracy is 97.52% and precision is 100%.

Table.13 Accuracy and Precision values for various similarity thresholds

Similarity threshold	Accuracy (%)	Precision (%)
0.38736	99.91	99.82
0.38908	98.89	97.98
0.39359	99.91	99.82
0.40171	99.91	99.82
0.41452	99.91	99.82
0.43349	99.91	99.82
0.46063	99.91	99.82
0.49881	53.76	100
0.55212	53.76	100
0.62654	53.76	100
0.73054	99.31	100
0.99998	99.82	100

Table.13 depicts the accuracy and precision values for the test traffic for various chosen values of similarity threshold. For thresholds ranging from 0.3873 to 0.4606, it is observed that the accuracy and precision values are better but when the threshold is varied further, for instance consider the threshold values from 0.4999 to 0.6265. It is observed that the accuracy and precision are having high deviation. An optimal value for threshold is observed to be 0.99998 for which the attack class accuracy is obtained as 99.82% and attack class precision is obtained as 100%.

`

Performance Analysis of classifier is also compared to the state-of-the-art classifiers such as naïve bayes, Bayes net, logistic regression, SVM, SOM, Multi-objective Evolutionary fuzzy classifier. Table.14, Table.15, Table.16 and Table.17 shows the accuracy, precision, sensitivity, specificity, false positive rate (FPR) and F-score values for

naïve bayes, bayes net, logistic regression, multi-objective evolutionary fuzzy classifiers respectively.

Table.14 Prediction Values of Naïve Bayes Classifier on test data

Parameter	Normal traffic	DDoS traffic
Accuracy	99.57%	99.57%
Precision	99.65%	99.49%
Sensitivity	99.48%	99.65%
Specificity	99.65%	99.48%
FPR	0.0034	0.0051
F-score	0.9957	0.9957

Table.15 Prediction Values of Bayes Net Classifier on test data

Parameter	Normal traffic	DDoS traffic
Accuracy	99.65%	99.65%
Precision	100%	99.32%
Sensitivity	99.31%	100%
Specificity	100%	99.31%
FPR	0	0.0068
F-score	0.9965	0.9965

Table.16 Prediction Values of Logistic regression Classifier on test data

Parameter	Normal traffic	DDoS traffic
Accuracy	59.67%	59.67%
Precision	55.35%	100%
Sensitivity	100%	19.35%
Specificity	19.35%	100%
FPR	0.8065	0
F-score	0.7126	0.3242

Table.17 Prediction Values of Multi-objective Evolutionary Fuzzy Classifier on test data

Parameter	Normal traffic	DDoS traffic
Accuracy	63.18%	63.18%
Precision	57.59%	100%
Sensitivity	100%	26.37%
Specificity	26.37%	100%
FPR	0.7363	0
F-score	0.7309	0.4173

Experiments are also carried by using SVM and SOM classifiers. For SVM classifier, it is seen that the accuracy is 57.79% for normal traffic and DDoS attack traffic. The precision, sensitivity, specificity in case of normal traffic class is 54.22%, 100% and 15.58%. For DDoS attack traffic class, the precision, sensitivity and specificity are 100%, 15.58%, 100%. For SOM classifier, the accuracy for DDoS attack traffic is 61.98% and detection rate is 23.97%.

The contribution reported in this paper is based on one of our recent works [9] in which we have proposed a ML model SWASTHIKA for high-rate and low-rate cloud network attacks detection. A detailed case study which explains the proposed method for feature reduction is included in this paper and a study is carried to gauge the effect of chosen hyperparameter values on prediction and classification. Also, in this study, we have tried to obtain the appropriate deviation value and similarity threshold for gaussian space from the similarity function and conducted experiments for obtaining better performance of the proposed model. For detailed understanding about the computations, design equations discussed in this paper, our previous work [9] can be referred. The dataset used in the present work is available at [41] for researchers who are interested in testing the performance of the ML or DL models.

8. Conclusions

Modern network environments such as Cloud, IoT, SDN environments are greatly targeted by network attacks whose identification is becoming huge challenge for service providers despite the deployment of several security measures. In a general case, usually deep learning algorithms and techniques use images as input to perform classification. The image representations are free from attribute relations and order. Thus, the idea is to convert network traffic instances into respective images which are either 3-channel or 1-channel to serve as input the ML model. In this paper, we propose a machine learning model which utilizes the network traffic instances that are converted into binary visualization images. These are then used to evaluate the performance of proposed model. For performance analysis, we have considered traffic instances from CICIDS2019 dataset. The experiments are carried by using a balanced dataset consisting of 2500 DDoS attack traffic and 2500 benign traffic instances and a test set consisting of 584 DDoS attack traffic and 584 benign traffic instances. The experiment results proved that the performance of the proposed approach is better than state-

of-art classifiers. In future, the present study can be extended to imbalanced datasets and multi-class classification.

References

[1] J. Jiang, R. Liou and S. Lee, "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 335-349, March 2011, doi: 10.1109/TKDE.2010.122.

[2] J. Jiang, W. Cheng, Y. Chiou and S. Lee, "A similarity measure for text processing," *2011 International Conference on Machine Learning and Cybernetics*, 2011, pp. 1460-1465, doi: 10.1109/ICMLC.2011.6016998.

[3] Y. Lin, J. Jiang and S. Lee, "A Similarity Measure for Text Classification and Clustering," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575-1590, July 2014, doi: 10.1109/TKDE.2013.19.

[4] S. -J. Lee and J. -Y. Jiang, "Multilabel Text Categorization Based on Fuzzy Relevance Clustering," in *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 6, pp. 1457-1471, Dec. 2014, doi: 10.1109/TFUZZ.2013.2294355.

[5] Sambangi S, Gondi L. A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression. *Proceedings*. 2020; 63(1):51. https://doi.org/10.3390/proceedings2020063051

[6] Sambangi, S., Gondi, L. (2021). Multi Linear Regression Model to Detect Distributed Denial of Service Attacks in Cloud Environments. In: Singh, J., Kumar, S., Choudhury, U. (eds) Innovations in Cyber Physical Systems. Lecture Notes in Electrical Engineering, vol 788. Springer, Singapore. https://doi.org/10.1007/978-981-16-4149-7_48

[7] Swathi Sambangi and Lakshmeeswari Gondi. 2021. Multiple Linear Regression Prediction Model for DDOS Attack Detection in Cloud ELB. The 7th International Conference on Engineering & MIS 2021. Association for Computing Machinery, New York, NY, USA, Article 4, 1–9. https://doi.org/10.1145/3492547.3492567

[8] Kumari, K., Mrunalini, M. Detecting Denial of Service attacks using machine learning algorithms. J Big Data 9, 56 (2022). https://doi.org/10.1186/s40537-022-00616-0

[9] Swathi Sambangi, Lakshmeeswari Gondi, Shadi Aljawarneh, A Feature Similarity Machine Learning Model for DDoS Attack Detection in Modern Network Environments for Industry 4.0, Computers and Electrical Engineering, Volume 100, 2022, 107955, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2022.107955.

[10] Arunkumar, M., Ashok Kumar, K. Malicious attack detection approach in cloud computing using machine learning techniques. Soft Comput (2022). https://doi.org/10.1007/s00500-021-06679-0

[11] S.A. Aljawarneh, R. Vangipuram. Garuda: gaussian dissimilarity measure for feature representation and anomaly detection in internet of things. Journal of Supercomputing, 76 (2020), pp. 4376-4413

[12] F. Hussain, S.G. Abbas, M. Husnain, U.U. Fayyaz, F. Shahz ad, G.A. Shah. IoT DoS and DDoS Attack Detection using ResNet. 2020 IEEE 23rd International Multitopic Conference (INMIC) (2020), pp. 1-6, 10.1109/INMIC50486.2020.9318216

[13] Faisal Hussain, Syed Ghazanfar Abbas, Muhammad Husnain, Ubaid U. Fayyaz, Farrukh Shahzad, Ghalib A. Shah. IoT DoS and DDoS Attack Dataset. IEEE Dataport (August 16, 2021), 10.21227/0s0p-s959

[14] R Vangipuram, RK Gunupudi, VK Puligadda, J Vinjamuri. A machine learning approach for imputation and anomaly detection in IoT environment. Expert Systems, 37 (2020), p. e12556, 10.1111/exsy.12556

[15] Laxmi Pranitha Rachamalla, Anusha Akkidasari, Sandhya Madiga, Harshitha Mittapalli, Radhakrishna Vangipuram, November 30, 2021, "CLOUD ATTACK DATASET", IEEE Dataport, doi:https://dx.doi.org/10.21227/05ep-zk84.

[16] Swathi Sambangi, Lakshmeeswari Gondi, Shadi Aljawarneh, Sreenivasa Rao Annaluri, December1, 2021, "SDN DDOS ATTACK IMAGE DATASET", IEEE Dataport, doi:https://dx.doi.org/10.21227/k06q-3t33.

[17] V. Radhakrishna, S.A. Aljawarneh, P.V. Kumar, K.-K.R. Choo. A novel fuzzy gaussian-based dissimilarity measure for discovering similarity temporal association patterns. Soft Computing, 22 (6) (2018), pp. 1903-1919, 10.1007/s00500-016-2445-y

[18] Clément Boin, Xavier Guillaume, Gilles Grimaud, Tristan Groléat, Michaël Hauspie. One Year of DDoS Attacks Against a Cloud Provider: an Overview. *4th International Conference on Advances in Computer Technology, Information Science and Communications*, Apr 2022, Suzhou, China. ⟨hal-03655003v2⟩

[19] Kushwah, G.S., Ranga, V. Detecting DDoS Attacks in Cloud Computing Using Extreme Learning Machine and Adaptive Differential Evolution. Wireless Pers Commun (2022). https://doi.org/10.1007/s11277-022-09481-9

[20] Britto Dennis, J, Shanmuga Priya, M. Deep belief network and support vector machine fusion for distributed denial of service and economical denial of service attack detection in cloud. *Concurrency Computat Pract Exper*. 2022; 34(1):e6543. https://doi.org/10.1002/cpe.6543

[21] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang and E. Dutkiewicz, "Deep Generative Learning Models for Cloud Intrusion Detection Systems," in IEEE Transactions on Cybernetics, doi: 10.1109/TCYB.2022.3163811.

[22] Kshirsagar, D., Kumar, S. A feature reduction based reflected and exploited DDoS attacks detection system. *J Ambient Intell Human Comput* **13,** 393–405 (2022). https://doi.org/10.1007/s12652-021-02907-5

[23] A. Cheruvu, V. Radhakrishna and N. Rajasekhar, "Using normal distribution to retrieve temporal associations by Euclidean distance," 2017 International Conference on Engineering & MIS (ICEMIS), 2017, pp. 1-3, doi: 10.1109/ICEMIS.2017.8273101.

[24] Shadi Aljawarneh, Vangipuram Radhakrishna, and Aravind Cheruvu. 2019. Nirnayam: fusion of iterative rule based decisions to build decision trees for efficient classification. In Proceedings of the 5th International Conference on Engineering and MIS (ICEMIS '19). Association for Computing Machinery, New York, NY, USA, Article 26, 1–7. https://doi.org/10.1145/3330431.3330458

[25] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty and V. Sravan Kiran, "Similarity Based Feature Transformation for Network Anomaly Detection,"

in IEEE Access, vol. 8, pp. 39184-39196, 2020, doi: 10.1109/ACCESS.2020.2975716.
[26] Vangipuram, R., Kumar, P.V., Janaki, V. et al. Krishna Sudarsana—A Z-Space Interest Measure for Mining Similarity Profiled Temporal Association Patterns. Found Sci 25, 1027–1048 (2020). https://doi.org/10.1007/s10699-019-09590-y
[27] Elmasry, Wisam, Akbulut, Akhan and Zaim, Abdul Halim. "A Design of an Integrated Cloud-based Intrusion Detection System with Third Party Cloud Service" Open Computer Science, vol. 11, no. 1, 2021, pp. 365-379. https://doi.org/10.1515/comp-2020-0214
[28] Elmasry W., Akbulut A., Zaim A. H., Comparative evaluation of different classification techniques for masquerade attack detection, International Journal of Information and Computer Security, 2020, 13(2), 187–209.
[29] Elmasry W., Akbulut A., Zaim A. H., Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic, Computer Networks, 2020, 168, 107042.
[30] Elmasry W., Akbulut A., Zaim A. H., Empirical study on multiclass classification-based network intrusion detection, Computational Intelligence, 2019, 35(4), 919–954.
[31] CICFlowMeter, https://www.unb.ca/cic/research/applications.html#CICFlowMeter, 2017.
[32] Wang Y., Wen J., Wang X., Tao B., Zhou W., A cloud service trust evaluation model based on combining weights and gray correlation analysis, Security and Communication Networks, 2019, 2019.
[33] Vieira K., Schulter A., Westphall C., Westphall C., Intrusion detection for grid and cloud computing, It Professional, 2009, 12 (4), 38–43.
[34] Vani R., Towards efficient intrusion detection using deep learning techniques: a review, Int J Adv Res Comput Commun Eng ISO, 2017, 3297, 2007.
[35] Yin Chuanlong, Zhu Yuefei, JinlongFei G., He Xinzheng
[36] A deep learning approach for intrusion detection using recurrent neural networksIEEE Access, 5 (2017), pp. 21954-21961
[37] Stergiou Christos, Psannis Kostas E., Kim Byung-Gyu, Gupta Brij Secure integration of IoT and cloud computing Future Gener. Comput. Syst., 78 (2018), pp. 964-975
[38] Nguyen KhoiKhac, Hoang Dinh Thai, DusitNiyato Kai, Wang Ping, Nguyen Diep, ErykDutkiewicz. Cyberattack detection in mobile cloud computing: A deep learning approach 2018 IEEE Wireless Communications and Networking Conference, WCNC, IEEE (2018), pp. 1-6
[39] Shone Nathan, Ngoc Tran Nguyen, DinhPhai Vu, Shi Qi. A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell., 2 (1) (2018), pp. 41-50
[40] Hajimirzaei Bahram, NimaJafariNavimipour Martine. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm ICT Express, 5 (1) (2019), pp. 56-59
[41] Sambangi, Swathi (2022), "Cloud Attack Dataset For Building Machine Learning and Deep Learning Models", Mendeley Data, V1, doi: 10.17632/5ct875rx9c.1
[42]

Swathi Sambangi is a Research Scholar at Department of CSE, Gitam Institute of Science and Technology, GITAM (Deemed to be University), Visakhapatnam and working as an Assistant Professor in the Department of Information Technology at VNR Vignana Jyothi Institute of Engineering and Technology, Telangana, India since 2015. She is awarded B. Tech in Information Technology from JNTU Kakinada and Master of Technology in Software Engineering from JNTU Kakinada. She has ten years of academic teaching experience and has presented and published several papers at international conferences and international journals. Her areas of research interest are in Algorithm design, Cloud Security and Machine learning.

Lakshmeeswari Gondi is an Associate Professor at GITAM Institute of Technology, GITAM (Deemed to be University), Visakhapatnam, India. She is awarded M. Tech in 2009 and Ph. D in 2013 from GITAM University. Several research scholars are working towards their doctoral degree under her esteemed guidance. She has twenty years of academic teaching and fifteen years research experience. She has to her credit several publications in international journals and conferences. Her areas of interest include Data mining, Cloud Computing, Network Security, Visual Cryptography and IOT.