Computers, Materials & Continua
DOI:10.32604/cmc.2022.019031
Article

Tech Science Press

Fuzzy Based Latent Dirichlet Allocation for Intrusion Detection in Cloud Using ML

S. Ranjithkumar[1,*] and S. Chenthur Pandian[2]

[1]Sri Ramakrishan Engineering College, Coimbatore, 641022, India
[2]SNS College of Technology, Coimbatore, 641035, India
[*]Corresponding Author: S. Ranjithkumar. Email: ranjith.s@srec.ac.in

Received: 30 March 2021; Accepted: 13 July 2021

Abstract: The growth of cloud in modern technology is drastic by provisioning services to various industries where data security is considered to be common issue that influences the intrusion detection system (IDS). IDS are considered as an essential factor to fulfill security requirements. Recently, there are diverse Machine Learning (ML) approaches that are used for modeling effectual IDS. Most IDS are based on ML techniques and categorized as supervised and unsupervised. However, IDS with supervised learning is based on labeled data. This is considered as a common drawback and it fails to identify the attack patterns. Similarly, unsupervised learning fails to provide satisfactory outcomes. Therefore, this work concentrates on semi-supervised learning model known as Fuzzy based semi-supervised approach through Latent Dirichlet Allocation (F-LDA) for intrusion detection in cloud system. This helps to resolve the aforementioned challenges. Initially, LDA gives better generalization ability for training the labeled data. Similarly, to handle the unlabelled data, Fuzzy model has been adopted for analyzing the dataset. Here, preprocessing has been carried out to eliminate data redundancy over network dataset. In order to validate the efficiency of F-LDA towards ID, this model is tested under NSL-KDD cup dataset is a common traffic dataset. Simulation is done in MATLAB environment and gives better accuracy while comparing with benchmark standard dataset. The proposed F-LDA gives better accuracy and promising outcomes than the prevailing approaches.

Keywords: Cloud security; fuzzy model; latent dirichlet allocation; preprocessing; NSL-KDD

1 Introduction

With today's technological development, Network Intrusion Detection System (NIDS) is a software or device that predicts abnormal functionality of network system by analyzing and monitoring network [1]. Denning has introduced NIDS in 80's and anticipated a detection model for Intrusion identification. It is extensively used in security violations in networking [2]. Recently, the incursions of unknown attacks are constantly increasing, the conventional tolls like access control/encryption and firewalls fails to safeguard network from unknown attacks [3]. As an outcome, various investigators paid attention towards establishment of network architectures or complex systems, for instance, cyber-physical systems, quality-aware service access system, multi-dimensional context aware social network framework [4], NIDS and emotion-aware cognitive systems. With these security applications, NIDS gains increased attention and turns as a primary source of cloud systems [5]. This is owing to the cloud requirements towards huge transmission and data interaction [6]. Data transmission is done with diverse data. It is inevitable that security towards data privacy is constantly exposed by intrusion [7]. To validate privacy and security of data, NIDS is essential during construction of cloud systems [8]. Henceforth, this work anticipates a novel NIDS security system for cloud.

Indeed of effectual NIDS functionality, establishment of NIDS is considered to be more challenging [9]. This is due to certain crisis like intrusion detection and data collections. These have to be taken into consideration. Subsequently, certain benchmark traffic datasets like NSL-KDD and KDDCup 99 are constructed and NIDS is modeled to enhance the intrusion detection performance [10]. As intrusion recognition are considered as an essential part of classification, various artificial intelligence (AI) and Machine Learning (ML) approaches are used in NIDS [11]. Usually, ML based approaches are considered as unsupervised (ULA) or supervised (SLA). The functionality of SLA is to perform mapping of feature samples to certain categories with labeled data utilization [12]. Various SLA like Decision Tree (DT), Deep Neural Networks (DNN), and Support Vector Machine (SVM) have been resourcefully used to recognize and to identify intrusions [13]. The SLA for NIDS has been attained with higher accuracy on diverse benchmark datasets. Moreover, certain disadvantages are clear. Initially, processing of labeled data requires costly expertise, and therefore detection updation is costly. Subsequently, when training process is based on detection model, labeled data can completely recognize newer kinds of attacks. Indeed of SLA, the ULA train detection approach devoid of any labeled instances and determines hidden unlabelled data structure. In ULA, various kinds of network events are differentiated using evaluation of unlabelled data distribution. The samples of similar features to hold similar class [14]. Even though, ULA has no necessity towards labeled data; generally outcomes towards prediction model with lesser accuracy and higher false positive rate (FPR). Fig. 1 depicts the generic view of NIDS in cloud systems.

To get rid of various deficiencies, semi-supervised learning approach (SSLA) is also implemented for NIDS. It merges unlabelled and labeled data to determine the detection model [15]. Subsequently, SSLA diminishes the dependency towards labeled data, and therefore it is considered to be healthier than SLA. As lesser amount of labeled data is initiated, SSLA generally carry out better performance in accuracy and FPA than unsupervised learning. Moreover, SSLA shares similar set of disadvantages of both SLA and ULA models. However, SSLA for NIDS needs more sophisticated model to diminish negative influence using both the approaches. This work anticipates a novel SSLA through hybridization of Fuzzy based semi-supervised approach through Latent Dirichlet Allocation (F-LDA) for intrusion detection in cloud system. This approach reduces the variance in classifier outputs. Here, the generalization ability outperforms the functionality of single model system in more appropriate environment. There are various attack types that are unidentified during training data, it is more appropriate to choose hybrid SSLA. For labeled data, a basic classifier is generated as candidates and constructs a hybrid model during classifier ranking. However, to completely analyze unlabelled data distribution, a Fuzzy based model is used. Then, based on results of ULA, a newer hybridized learning system is constructed and extended with Latent Dirichlet Allocation. At last, the anticipated model will be tested over NSL-KDD dataset. The significant contributions with the anticipated model are given below:

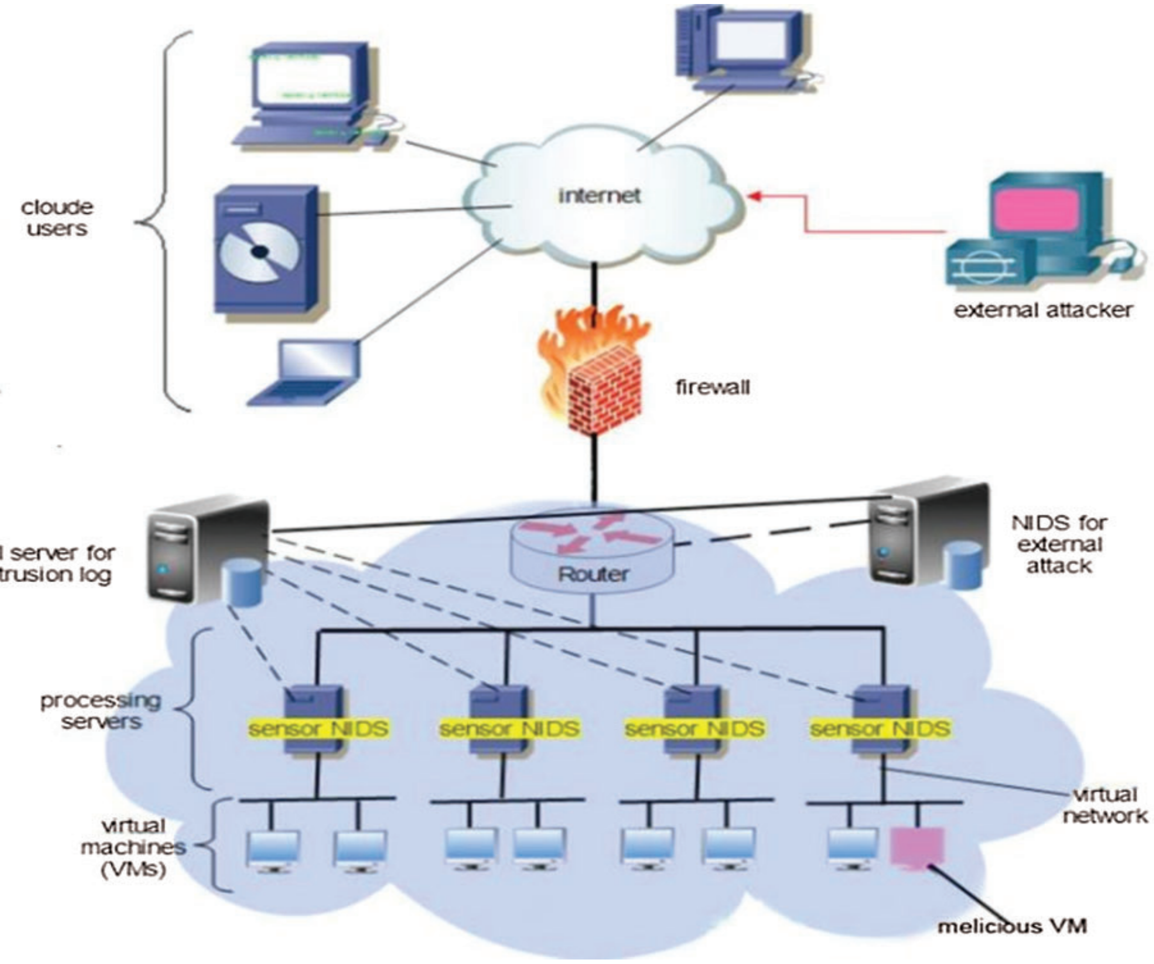

Figure 1: Generic view of NIDS in cloud systems

1) This work presents a novel SSLA for classification purpose. By considering the intrusion detection for non-linear classification approach, this work uses classification as a basic learner with hybridized system. To merge the outcomes of Fuzzy model and LDA is applied to determine weights.

2) This work adopts Fuzzy based model for mining hidden structures of unlabelled data. This method extracts essential information and eliminates redundant data from unlabeled data. This can improves the performance of anticipated model.

3) This work combines both SLA and ULA through hybridization approach. With this, the labeled data has to correct the unlabeled data. This is by means of lacking in labeled data; unlabeled data utilization performs classifier model construction to perform detection procedure for robustness and accuracy.

This is work is structured as trails: Section 2 offers diverse background study based on ML based techniques for NIDS and brief explanation towards SSLA. Then, the anticipated SSLA for cloud system is discussed in Section 3. Experimentation and outcomes are performed in Section 4. Finally, Section 5 draws conclusion of anticipated model.

2 Background Studies

The ultimate objective is to construct an adaptive and an effectual NIDS for predicting malicious functionalities which tries to construct various network services. Various studies are related to the anticipated model and explained in Section 2.1.

2.1 NIDS Approaches

NIDS model is extensively utilized for detecting and monitoring malicious functionalities from network activities [16]. A typical detection model includes four preliminary stages: data source, pre-processing, decision making process and defense mechanisms. Initially, data source comprises of set of network functionalities where every feature is utilized for differentiating suspicious and legitimate observations [17]. Next, pre-processing organizes data by eradicating redundant features to generate pattern set involving suspicious and legitimate properties based activities [18]. Thirdly, detection approach comprises of classification method that recognizes abnormal observations. At last, defense response is considered as decision made by cyber or software administrators for eliminating attack.

NIDS methodologies are categorized as anomaly, misuse and hybrid model. To initiate with this, anomaly based approach builds a normal profile and determines variation from attack profile. As, it can recognizes both zero-day and prevailing attacks, it will not demand any efforts to generate any rules, it is extremely effectual for system mitigation than misuse grounded IDS, when decision making process is effectually modeled [19]. Similarly, misuse grounded NIDS examines network traffic for handling instances over known attack models to be black-listed. Although it generates lower false positive rates and superior detection rates, it may not detect zero-day attacks. Also, it needs considerable attempt to update blacklists with newer attack rules dependent on attacks identified. Various investigators have used ensemble model to enhance NIDS performance [20]. The objective is to merge alerts for diminishing alarms, assisting security administrators to handle alerts effectually. Abaid et al. [21] modeled an ensemble approach for determining diverse attack types using some of features. The author made a final decision for predicting attack based on voting rule based approach. Li et al. [22] constructed an ensemble approach with Classification and Regression Trees (CART) and Bayesian networks (BN) to predict attack samples. As whole, empirical outcomes of these approaches provides the overall performance of ensemble/hybrid approaches that are superior to every individual model. However, it increases computational overhead.

The performance analysis for all NIDS is based on source that comprises of feature set that are classified into various types of classification purposes termed as payload based features, source/destination port, flow and behavioral feature models [23].

Some source/destination port number features are extracted with Net flow and Coral-Reef tools [24]. Some features are ineffectual as it needs basic information from packet headers are extremely unreliable in contemporary network to recognize attacks with faster and dynamical variation towards present network model [25]. Payload based feature model acquires considerable signatures of diverse applications. Some features assist in predicting malicious functionalities offering superior accuracy. Also, there are some autonomous ports that are utilized and facilitate the provision for prediction during the use of non-standardized ports. Also, some features need enormous effort to update signatures regularly for predicting attacks, with complexity of acquiring higher network traffic rates [26].

Behavioral features draw the attention among hosts based on ports, destinations and behavior of host patterns, At last, flow based feature gains the preliminary flow identifiers (protocols, ports, source/destination IP) and statistical features like packet sizes and inter-arrival times. The two kinds to attain higher accuracy; when used precisely as recommended with anticipating aggregation more than an attribute in extractor module to identify botnet strategies occurrence by flooding of enormous flows towards compromised hosts.

Various investigators perform data model sources from TCP/IP protocols for valuating performance of NIDS. Specifically, it concentrates on recognizing diverse kinds of exploitation methods and botnets like DDoS, DoS and phishing model. For instance, author in [27], modeled a distributed and scalable IDS through set of instances from HTTP, DNS, honeypot and IP-data flow. Also, it will not offer features utilized for execution. In addition, simulated data was gathered from diverse systems devoid of providing configuration environment for computing newer IDS performance.

Moustafa et al. [28] recommends bot detection approach through DNS flow analysis dependent on features constructing with DNS queries. This work was constructed by extracting statistical data from DNS queries like query domain name and source IP address. Also, these may be altered or hidden via virtual private network utilization with no further statistical network flow aggregation analysis that include potential features of aggregated flooding threats like DDoS attacks. Kim et al. [29] anticipates a domain name generation model for identifying botnets via analysis of individual DNS queries. These investigations uses prevailing flow tools and approaches and analysis the outcomes via ML approach utilization. The author cannot use aggregated flow that effectually recognizes botnet threats for decision making process [30]. It cannot find effectual features of protocols as recommended in the anticipated model. From the above discussions, it is known that IDS system can analyze the suspicious activities where the violations are related to the event management and security information where the original cause of threats relies over the benign traffic abnormalities or false alarm rate. Moreover, it takes longer time to differentiate the threat model. It leads to more damage. IDS are provided for monitoring the network traffic and the surface threats. However, some traffic paths are left unmonitored in some cases where the threats are considered to be deeper and avoid network visibility. Some IDS fails to give security to the VM, IoT devices, and container environments. There are no proper alert mechanisms to make the functionality faster for better decision making. The proposed research attempts to finds the attack by violating the data redundancy and formulating efficient rules for analyzing the network traffic flow.

3 Methodology

This section explains the novelty based on LDA and semi-supervised approach and explained in detail. This model is more effectual and can be applied for predicting intrusion. The redundant data is removed and given for training. For unlabeled data, fuzzy modeling is used to compute correntropy among the features. The flow diagram of the anticipated model is depicted as in Fig. 2:

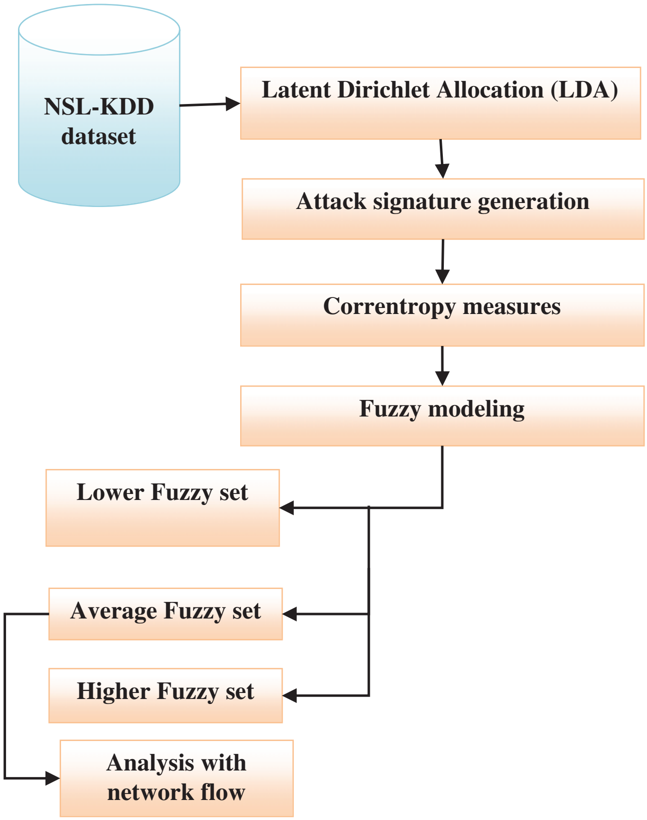

Figure 2: Flow diagram of proposed model

3.1 Latent Dirichlet Allocation (LDA)

LDA gives an insight towards statistical modeling of identify the corpus framework. This is so popular statistical model to identify network traffic and to predict the attack signatures that has not been explored. It is a probabilistic model, as assumes that document was generated using weighted mixture of unknown model. The ultimate objective of LDA is to automatically recognize the set of documents. To perform this, an inferential problem has to be solved using corpus generated by LDA generation process. This problem provides probability distribution function for generation of set of documents. The primary objective is to evaluate posterior distribution of more hidden variables given by the dataset as in Eq. (1):

$$p\left(\theta, \varnothing | D, \alpha, \beta\right) = \frac{p(\theta, \varnothing, D|\alpha, \beta)}{p(D|\alpha, \beta)} \tag{1}$$

The use of Gibbs sampling is used for evaluating LDA. Consider, w and z as vectors of all incoming data that is allocated under traffic “T”. The multi-modal distribution of the anticipated model is given as below in Eq. (2):

$$p\left(z_t = k|z \rightharpoondown t, w\right) = \frac{n_{k,\rightharpoondown t}^{(w)} + \beta}{\left[\sum_{v=1}^{V} n_k^v + \beta\right] - 1} \frac{n_{k,\rightharpoondown t}^{(k)} + \alpha}{\left[\sum_{j=1}^{k} n_i^j + \alpha\right] - 1} \tag{2}$$

Here, '*t*' specifies iteration counter, $n_{k,\rightharpoondown t}^{(w)}$ is number of incoming data packets to network except current traffic validation. $\left[\sum_{v=1}^{V} n_k^v + \beta\right] - 1$ is total number of traffic allocated to network except present network analysis, $n_i^j + \alpha$ is number of incoming packets to the network. After successive processing, the matrices are computed with Eqs. (3) & (4):

$$\theta_{i,k} = \frac{n_i^{(k)} + \alpha}{\sum_{j=1}^{K} n_i^{(j)} + \alpha} \tag{3}$$

$$\varnothing_{k,w} = \frac{n_k^{(w)} + \beta}{\sum_{v=1}^{V} n_k^{(v)} + \beta} \tag{4}$$

3.2 Attack Signature Generation with LDA

LDA has been extensively used in various applications; however it is used as a tool for validating network traffic classification and diverse attack signatures. The intrinsic application of network attack validates the traffic and resembles text in various aspects.

For example, network flow is composed of diverse ASCII strings, a flow is determined with strings and binary word flow. This flow is determined as corpus flow. The network flow is observed as mixture of data based on intrusion detection applications. The correlation among variables is analyzed using the network packets. In general, network problems are disseminated in E-mail or HTTP flow and the web flow comprises of strings like WWW, HTTP and GET. When network is comprised in web flow, latent flows are included in HTTP flow and corresponding attacks. Thus, when LDA carry out appropriate network traffic and generates cluster flow appropriately. Some are included in cluster specification of latent factors related to flows. This automatically extracts signature for IDS to predict the flows. The functionality of IDS rule signature is based false negatives and positives. There are two metrics that are constructed with LDA. The first metric is to define the superiority of flow associated with the network and signature does not need real NIDS. Time consumption based rule verification is used. It can be achieved by LDA training model. Another metric is associated with false positive of all signatures. The attack signature are chosen based on integrated IDS and validates the rules over real network traffic. The network administrator has to validate the operating overload. Certain rules are needed for attack prediction with higher overload in IDS.

With respect to document classification of data from network traffic, words are depicted as consecutive bytes partitioned by delimiters like punctuations and spaces. However, it will not be applied for all network traffic as traffic payload comprises of both HEX and ASCII strings, generally it does not hold any separators; the objective is to overcome the complexities associated with key content of strings generated over the application. Some examples are 'HTTP/2.0' or 'GET/.file' for all web flows. From these observations, words are defined as a sequence of characters. LDA extracts words from network flow and deals with the set of content strings. The values are empirically analyzed with signature generation and attain acceptable performance.

3.3 Correntropy Measures

It is utilized for computing the association among the feature vectors, where it predicts the similarity and differences between the attack and normal instances. When there is dissimilarity among the samples, then statistical analysis measures the feature significance for predicting malicious functionalities. It is performed with non-linear similarity and second-order statistics for projecting the interactions of various feature observations. This is considered to be a lesser sensitive towards outliers. Consider a two random variables r_1 and r_2; where correntropy is given as in Eq. (5):

$$V_\sigma\left(r_1, r_2\right) = E\left[K_\sigma\left(r_1 - r_2\right)\right] \tag{5}$$

Here, $E[.]$ is mathematical expectation, $K_\sigma(.)$ is Gaussian kernel function and σ is kernel size. It is depicted as in Eq. (6):

$$K_\sigma\left(.\right) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(.)}{2\sigma^2}\right) \tag{6}$$

The joint probability density function is generally unidentified when finite number of observations $\left\{r_i, r_j\right\}_{(i,j)=1}^{2}$ is attained. The correntropy is measured as in Eq. (7):

$$\hat{V}_{M,\sigma}\left(A, B\right) = \frac{1}{M}\sum_{i,j=1}^{M} K_\sigma\left(r_i - r_j\right) \tag{7}$$

When using correntropy measure towards multi-variant network data as in Eq. (7), it is computed for both abnormal and normal feature vectors as in Eq. (8):

$$I_{1:N} = \begin{bmatrix} f_{11} & f_{12} & \dots \\ f_{21} & f_{22} & f_{ij} \end{bmatrix}; \quad Y_{1,N} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{8}$$

From the observations, 'I' is observation towards network data, 'Y' is class label for all observations toward class labels, 'N' is number of observations, 'F' is feature set. When the difference among normal and malicious vector values, then it is proven that it shows the significance towards the features. Correntropy of every vector is evaluated for attack and normal instances. The differences among the instances are revealed. Feature selection an essential role in predicting NIDS for choosing essential features and eliminating unnecessary values and helps in distinguishing malicious and normal instances and enhances NIDS performance. The ultimate objective of selecting features is to diminish computational cost, eliminate data redundancy, improves NIDS accuracy and helps in examining data normality. Here, simple feature selection approach is utilized and correlation coefficient measures the degree of strength among certain features. The least ranked features are chosen as the most essential part of fuzzy modeling for predicting abnormal functionalities of instances. Correlation coefficient of features is measured as in Eq. (9):

$$CC\left(r_1, r_2\right) = \frac{cov(r_1, r_2)}{\delta r_1 . \delta r_2} \tag{9}$$

From the equation mentioned above, δ is standard deviation of features, $cov()$ is feature covariance. The mean value of r_1 and r_2 are performed with $Mr_1 = \frac{\sum_i^N a_i}{N}$ and $Mr_2 = \frac{\sum_i^N b_i}{N}$ respectively. The CC outcomes are changed and ranges from $[+1, -1]$. When values are nearer to $+1$ and -1; then it specifies correlation among two features r_1 and r_2. When values are closer to 0, then no correlation among features are known. Some positive factors determine the features in similar direction, when negative value specifies features in opposite direction.

3.4 Fuzzy Modeling

LDA deals with unlabelled data that comes into the network and moves out of the network. To deal with this unstructured or unlabelled data, fuzzy modeling is used. It is measured to be an uncertainty type where the value ranges from 0 to 1. It is used in various applications like classification. Here, fuzzy modeling is adopted to estimate the significance of every sample. Thereby, it eradicates the irrelevant words or corpus generated from the network. Also, it is adopted to strengthen generalization ability. Therefore, it improves the detection competency for newly introduced malicious events. The unlabelled data is defined as $S^a = \{x_1^a, x_2^a, \ldots, x_n^a\}$ with n_a. Here, correntropy modeling is used for extracting features and to train the fuzzy model towards intrusion detection. With this, samples S^a is provided with class labels. The unlabelled samples are re-written with prediction label as $S^{asl} = \left\{x_1^a, \bar{\chi}(x_1^a), \ldots, \left(x_{n_a}^a, \bar{\chi}(x_{n_a}^a)\right)\right\}$ which is a self-labeled sample. Here, information entropy is used to compute fuzzy model of classifier output. It is given as in Eq. (10):

$$F(x) = -\frac{1}{k}(\bar{\chi}(x) \log_2 \bar{\chi}_i(x) + (1 - \bar{\chi}_i(x)) \log_2 (1 - \bar{\chi}_i(x))) \tag{10}$$

Here, $\bar{\chi}_i(x)$ is specified as i^{th} nodes of output value, i.e., probability of samples are given as i^{th} category. For computing the fuzzy model, self-labeled samples are classified based on ranking values. This model is partitioned as: lower fuzzy set, average fuzzy set and higher fuzzy set. The average fuzzy set gives better performance for enhancing NIDS. Hence, these models consider only the average fuzzy set and eliminate higher and lower fuzzy set. Alike of training process, this model uses bootstrapping approach to deal with samples and to construct a fuzzy classifier. The classifier model is trained with average fuzzy set with predictor labels. To fulfill homogeneity with fuzzy where sampling rate is set and number of samples are similar to $\bar{\chi}$. The anticipated F-LDA is merged with fuzzy model for intrusion detection. This generates SSL model with labeled data. The prediction of unlabeled data is used for computing fuzzy value and groups sample as higher, average and lower fuzzy set. It is achieved using fitting the average fuzzy set. When input to the network is fixed, it is not supportable for extending the classifier model. Hence, fuzzy model is constructed appropriately. The labeled samples are integrated with average fuzzy set as training data. At last, supervised and unsupervised learning outcomes are specified as $\bar{\chi}$. The competency to merge the LDA based network signature with fuzzy model is to attain superior generalization. This specifies stronger competency to identify attack patterns. In case of unsupervised region, fuzzy model assists in exploring the inner functionality of unlabeled data. As an outcome, fuzzy model provides unlabelled data for classification this increases the data utilization. The complete detection approach turns to be more robustand the performance is enhanced.

Algorithm 1:

```
Input: Labeled samples $s'$
Initialize network data flow $d = \{\}$;
Use LDA for feature selection and to eliminate the unnecessary features on $s'$.
for $i = 1, 2, \ldots, n$
    Construct bootstraps $b_i$ with sampling $s'$ with sample rate $\bar{\chi}$.
    Train fuzzy model based classifier $F(x)$ with $b_i$;
    Classify fuzzy model into lower, average and higher using unlabeled samples $S^{asl}$;
    //prediction label;
    Compute $\bar{\chi}$ using Eq. (10);
end for
Select $\bar{\chi}$ model for generating prediction accuracy on $s'$.
Train the labeled data over the network flow with labeled samples $s'$.
Construct semi-supervised network with fuzzy model and LDA, i.e., $\bar{\chi}$.
Output: Predict the network flow for intrusion detection, $\bar{\chi}$.
```

4 Experimental Settings

Here, a network traffic dataset termed as NSL-KDD dataset is introduced in association with the anticipated model. To analyze the functionality of this method, diverse experimental comparisons are performed. This dataset includes both testing and training set. The features chosen determine the dataset description with preliminary statistical and contents information towards network connection. The feature size is given as 41. The dataset label includes five diverse network events like probe, normal, denial of service (DoS), user to root, and remote to local (R2L). Various investigators consider NSL-KDD dataset is a authoritative benchmark standards in intrusion detection. Thus, NSL-KDD dataset is considered in this work for valuating semi-supervised approach. It comprises of various attack patterns that are more appropriate for validating generalization capability. Here, random samples are chosen and remaining samples are utilized as unlabeled data. Here, intrusion detection is considered as multi-class problem. The experimentation is performed in PC. The system configurations are given as: Intel i5 processor, windows 7 OS, 8 GB RAM @ 3.00 GHz.

There are two diverse features known as numerical and symbolic. The anticipated model deals with symbolic features and values are not distributed randomly. It also triggers negative effect over learning process. To get rid of this problem, data normalization and one-hot encoding approach are used before learning process. The feature values are sequence encoded with 0 and 1. Dimensionality change based on distinctive values of symbolic features. Features like ‘protocol type’, ‘service’, and ‘flag’ are encoded when values are higher than 2 (Tab. 1). Symbolic features are treated as Boolean type with 1 or 0.

The data dimensionality increases from 40 to 120. Similarly, normalization is used. The objective is to scale feature values with an interval of [0, 1] as given below in Eq. (11):

$$x_{i,j}^{normal} = \frac{x_{i,j} - \min(x,j)}{\max(x,j) - \min(x,j)} \tag{11}$$

Table 1: Dataset features

Features	Data type	Features	Data type
Duration	N	Guest login	S
Protocol type	S	Count	N
Service	S	Server count	N
Flag	S	Server error rate	N
Source bytes	N	Srv error	N
Destination bytes	N	Rerror rate	N
Land	S	Srv rerror rate	N
Wrong fragment	N	Same srv rate	N
Urgent	N	Different source rate	N
Hot	N	Srv diff host rate	N
Number of failed logins	N	Dst host count	N
Logged in	S	Destination host server count	N
Number of compromised nodes	N	Dst host same srv rate	N
Root shell	N	Dst host diff srv rate	N
Attempt	N	Dst host same src port rate	N
Number of root	N	Dst host srv diff host rate	N
Number of file creations	N	Dst host serror rate	N
Number of shells	N	Dst host reeor rate	N
Number of access files	N	Dst host srv error rate	N
Number of outbound commands	N	Dst host reeor rate	N
Hot login	S		

Here, $x_{i,j}$ is attribute value of x_i, x_j specifies the j^{th} features. To compute the performance of prediction, various metrics are computed and specified as in Eqs. (12) & (13):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$FAR = \frac{FP}{FP + TN} \tag{13}$$

Here, True Positive (TP) specifies number of sample attacks classified appropriately as attack. True Negative (TN) specifies number of sample attacks classified inappropriately as normal. False Positive (FP) specifies number of samples appropriately classified as normal samples. False Negative (FN) specifies number of samples classified inappropriately as attack. Prediction accuracy specifies model competency to perform appropriate prediction. FAR is depicted as a ratio of normal traffic predicted as attacks in inappropriate manner. Hence, detection model gives better accuracy and lower FAR. Here, the efficiency of fuzzy model is investigated and analyzed. The anticipated model is compared to prevailing approaches. As unlabeled data are allocated by classifier model, it is not avoidable as unlabeled samples are misclassified. It is clear that fuzzy model plays an essential role in enhancing detection performance in contrary to other models. The fuzzy set gives superior performance while achieving accuracy. It is proven that fuzzy set gives lesser misclassification and helps to identify anomaly. As well, the numbers of 'R2L' and 'U2R' in training samples are lesser than other classes. The accuracy is extremely lesser. Fig. 3 depicts the network attack types.

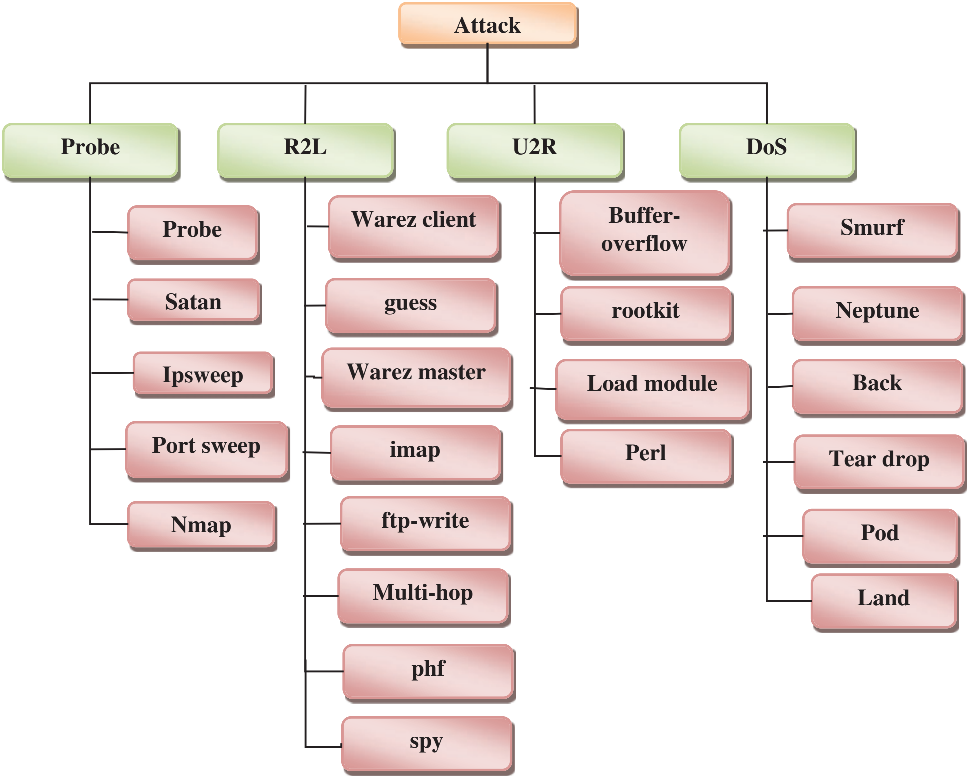

Figure 3: Network attack types

Tab. 2 demonstrates the dataset validation using training and testing model. There are five different attributes like Normal, Probe, DoS, U2R and R2L respectively. The anticipated model works finely and more stable while predicting larger size classes. To validate efficacy of anticipated model, the results are compared to prevailing intrusion detection approaches. As well, the result from conventional ML approaches is given as well. The comparison is done with semi-supervised approach. It is noted that the anticipated model is competitive with various other ML classifier model like ensemble random forest. With conventional ML approaches, tree based techniques outperforms other ML techniques (Multi-layer perceptron and SVM). The result specifies the motivation to select SSLA with NN structure. The anticipated model gives superior performance and seems to be more effectual. As a whole, the anticipated model offers an effectual way for identifying intrusion detection and outperforms the prevailing NIDS.

Fig. 4 shows the accuracy computation of proposed model with the existing approaches. Fig. 5 depicts the FAR computation of fuzzy model in lower, average and higher fuzzy model. Fig. 6 depicts the network traffic analysis for various threat models. With average fuzzy set, the anticipated model shows better performance when compared to higher and lower fuzzy model. Fig. 7 depicts the accuracy and FAR computation of proposed with other fuzzy approaches. The accuracy is higher for F-LDA and reduced false predictions. This model works optimally and shows better outcomes.

Table 2: Dataset validation

	Dataset validation	
	Training	Testing
Normal	67343	13449
Probe	11656	2289
DoS	45927	9234
U2R	52	11
R2L	995	209
Total	125973	25192

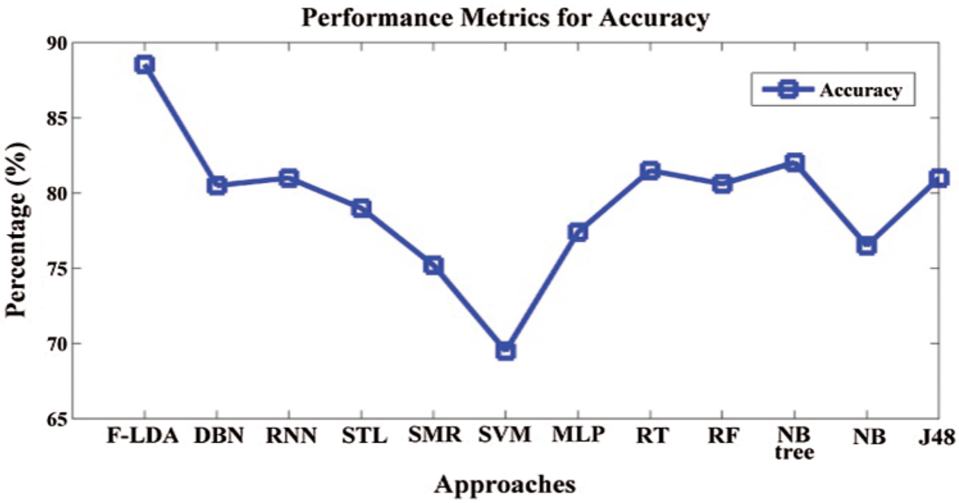

Figure 4: Accuracy computation

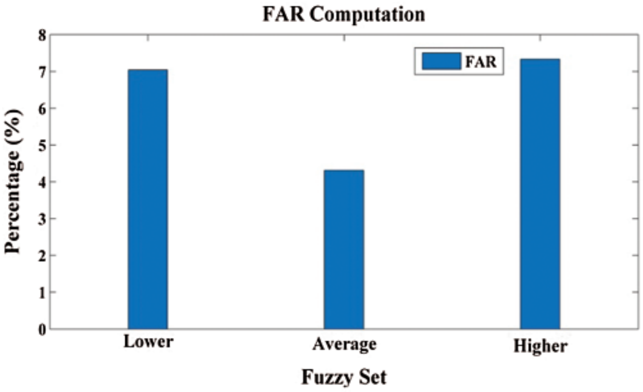

Figure 5: FAR computation

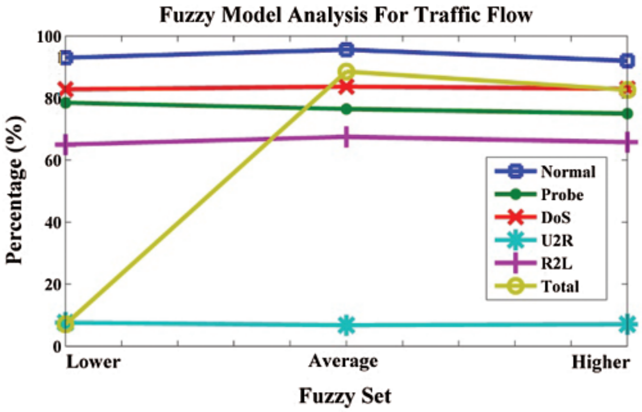

Figure 6: Fuzzy model analysis for network traffic flow

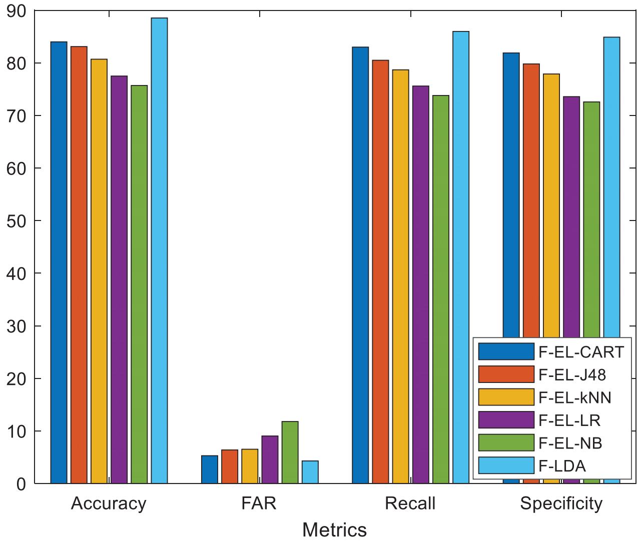

Figure 7: Fuzzy model evaluation

Tab. 3 depicts comparison of F-LDA with other models like DBN, RNN, STL, SMR, SVM, MLP, RT, RF, NB tree, NB, J48 where the accuracy of F-LDA is 88.54% which is 8.04%, 7.54%, 9.54%, 13.34%, 19.04%, 11.14%, 7.04%, 7.94%, 6.51%, 12.04% and 7.54% respectively. Tab. 4 depicts comparison of FAR computation using Fuzzy model. There are three parts: Lower, Average and Higher Fuzzy set. The values are 7.04%, 4.31% and 7.33% respectively. Tab. 5 depicts comparison of various existing Fuzzy models like F-EL-CART, F-EL-J48, F-EL-KNN, F-EL-LR, F-EL-NB and F-LDA. The accuracy of F-LDA is 88.54% which is 4.54%, 5.44%, 7.84%, 11.04% and 12.84% respectively as in Tab. 6. The anticipated model shows 4.31% FAR which is lesser than other models. Similarly, the sensitivity and recall of the proposed F-LDA is superior which shows the significance of the model.

Table 3: Accuracy and FAR computation

Approaches	KDD-test data	
	Accuracy (%)	FAR
F-LDA	88.54	4.31%
DBN	80.5	19%
RNN	81	NA
STL	79	NA
SMR	75.2	NA
SVM	69.5	NA
MLP	77.4	NA
RT	81.5	NA
RF	80.6	NA
NB tree	82.03	NA
NB	76.5	NA
J48	81	NA

Table 4: FAR computation with Fuzzy set

Fuzzy set	FAR (%)
Lower	7.04
Average	4.31
Higher	7.33

Table 5: Fuzzy model analysis for traffic flow

Fuzzy set	KDD test set					
	Accuracy					
	Normal (%)	Probe (%)	DoS (%)	U2R (%)	R2L (%)	Total (%)
Lower	93	78.5	82.8	7.6	65	7.05
Average	95.6	76.5	83.7	6.75	67.5	88.54
Higher	92	75	83	7.06	65.8	82.64

Table 6: Accuracy and FAR computation with various existing Fuzzy model

	NSL-KDD			
	Accuracy (%)	FAR (%)	Recall (%)	Specificity (%)
F-EL-CART	84	5.3	83	81.89
F-EL-J48	83.1	6.40	80.5	79.8
F-EL-kNN	80.7	6.53	78.68	77.89
F-EL-LR	77.5	9.05	75.6	73.58
F-EL-NB	75.7	11.8	73.8	72.58
F-LDA	88.54	4.31	85.98	84.89

5 Summary

Here, a novel SSLA for intrusion detection over cloud has been deliberated. Through continuous process of experimentation, the F-LDA model is validated to be more resourceful in predicting intrusion and enhance the security of cloud model. As a whole, the significant contribution is concluded with three successive factors. Initially, a novel approach offers a robust detection model over intrusion detection. It improves the competency to identify the newer traffic patterns. Next, fuzzy modeling utilizes enormous unlabelled data and improves prediction accuracy and robustness of entire system. Finally, F-LDA is proven as a more appropriate model to resolve intrusion detection problem. While comparing with various prevailing approaches, the anticipated model has proven a promising performance. The experimental outcomes validates that the SSLA is appropriate and can be applied in intrusion detection. In future, it is extremely essential to enhance prediction performance and generalization ability. To reach the goal, various interesting factors are analyzed for providing network security. Some performance measures will be tested in available public benchmarks.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. K. Malladi, T. M. Ravi, M. K. Reddy and K. Raghavendra, “Edge intelligence platform, and internet of things sensor streams system,” US Patent App. 15/250,720, 2017.

[2] N. Moustafa and J. Slay, “Unsw-nb15: A comprehensive data set for network intrusion detection systems (UNSW-nb15 network data set),” in *Military Communications and Information Systems Conf., 2015*, Canberra, ACT, Australia, IEEE, pp. 1–6, 2015.

[3] T. S. Wang, H. T. Lin, W. T. Cheng and C. Y. Chen, “Dbod: Clustering and detecting DGA-based botnets using dns traffic analysis,” *Computers & Security*, vol. 64, no. 2, pp. 1–15, 2017.

[4] N. Moustafa, G. Creech and J. Slay, “Flow aggregator module for analysing network traffic,” in *Progress in Computing, Analytics and Networking*. Berlin, Germany: Springer, pp. 19–29, 2018.

[5] F. A. Teixeira, G. Machado, P. M. Fonseca, F. M. Q. Pereira, H. C. Wong *et al.,* “Defending internet of things against exploits,” *IEEE Latin America Transactions*, vol. 13, no. 4, pp. 1112–1119, 2015.

[6] X. Hu, X. Li, E. C. H. Ngai, V. C. M. Leung and P. Kruchten, “Multidimensional context-aware social network architecture for mobile crowdsensing,” *IEEE Communications Magazine*, vol. 52, no. 6, pp. 78–87, 2014.

[7] J. Zhang, M. Zulkernine and A. Haque, “Random-forests-based network intrusion detection systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[8] P. Preethi and R. Asokan, “A high secure medical image storing and sharing in cloud environment using hex code cryptography method—secure genius,” *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 7, pp. 1337–1345, 2019.

[9] Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai *et al.,* “An efficient intrusion detection system based on support vector machines and gradually feature removal method,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

[10] M. Ahmed and A. N. Mahmood, “Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection,” *Annals of Data Science*, vol. 2, no. 1, pp. 111–130, 2015.

[11] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, “A deep learning approach to network intrusion detection,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[12] P. Preethi and R. Asokan, “Modelling LSUTE: PKE schemes for safeguarding electronic healthcare records over cloud communication environment,” *Wireless Personal Communications*, vol. 117, no. 4, pp. 2695–2711, 2021.

[13] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, “Deep learning approach for network intrusion detection in software defined networking,” in *Int. Conf. on Wireless Networks and Mobile Communications*, Fez, Morocco, pp. 258–263, 2016.

[14] C. Yin, Y. Zhu, J. Fei and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[15] P. Preethi and R. Asokan, “An attempt to design improved and fool proof safe distribution of personal healthcare records for cloud computing,” *Mobile Networks and Applications*, vol. 24, no. 6, pp. 1755–1762, 2019.

[16] P. Kazienko and P. Dorosz, “Intrusion detection systems (IDS) part I-(network intrusions; attack symptoms; IDS tasks; and IDS architecture),” *Retrieved April*, vol. 20, no. 2009, 2003.

[17] S. Chebrolu, A. Abraham and J. P. Thomas, “Feature deduction and ensemble design of intrusion detection systems,” *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[18] S. Marchal, X. Jiang, R. State and T. Engel, “A big data architecture for large scale security monitoring,” in *Big data (BigData Congress), 2014 IEEE Int. Congress on IEEE*, Anchorage, AK, USA, pp. 56–63, 2014.

[19] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou *et al.,* “Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, 2016.

[20] K. Jee, L. Zhichun, G. Jiang, L. Korts-Parn, Z. Wu *et al.,* “Host level detect mechanism for malicious dns activities,” US Patent App. 15/644,018, 2018.

[21] Z. Abaid, D. Sarkar, M. A. Kaafar and S. Jha, “The early bird gets the botnet: A markov chain based early warning system for botnet attacks,” in *Local Computer Networks, 2016 IEEE 41st Conf. on IEEE*, Dubai, UAE, pp. 61–68, 2016.

[22] W. Li, W. Meng, L.-F. Kwok and H. Horace, “Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model,” *Journal of Network and Computer Applications*, vol. 77, no. 2, pp. 135–145, 2017.

[23] O. AlKadi, N. Moustafa, B. Turnbull and K.-K. R. Choo, “Mixture localization-based outliers models for securing data migration in cloud centers,” *IEEE Access*, vol. 7, pp. 114607–114618, 2019.

[24] N. Moustafa, G. Creech, E. Sitnikova and M. Keshk, “Collaborative anomaly detection framework for handling big data of cloud computing,” in *Proc. 2017 Military Communications and Information Systems Conf.*, Canberra, ACT, Australia, pp. 1–6, 2017.

[25] M. Ahmed, A. N. Mahmood and J. Hu, “A survey of network anomaly detection techniques,” *Journal of Network and Computer Applications*, vol. 60, no. 1, pp. 19–31, 2016.

[26] J. Lopez, J. E. Rubio and C. Alcaraz, “A resilient architecture for the smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3745–3753, 2018.

[27] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in *Pro. 9th EAI Int. Conf. on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (BICT’15)*, Brussels, BEL, pp. 21–26, 2016.

[28] N. Moustafa, G. Creech and J. Slay, “Anomaly detection system using beta mixture models and outlier detection,” in *Proc. Progress in Computing, Analytics and Networking*, Singapore, Springer, pp. 125–135, 2018.

[29] J. Kim, N. Shin, S. Y. Jo and S. H. Kim, “Method of intrusion detection using deep neural network,” in *Proc. 2017 IEEE Int. Conf. on Big Data and Smart Computing*, Hong Kong, China, IEEE, pp. 313–316, 2017.

[30] N. Thillaiarasu and S. ChenthurPandian, “A novel scheme for safeguarding confidentiality in public clouds for service users of cloud computing,” *Cluster Computing*, vol. 22, no. 1, pp. 1179–1188, 2018.