I.J. Wireless and Microwave Technologies, **2021, 5, 9-17**
Published Online October 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2021.05.02

Exploring Deep Learning Techniques in Cloud Computing to Detect Malicious Network Traffic: A Sustainable Computing Approach

Nagesh Shenoy H[1,*], K. R. Anil Kumar[2], Suchitra N Shenoy[3], Abhishek S. Rao[4], Rajgopal K T[1]
[1]Department of Computer Science & Engineering, Canara Engineering College, Benjanapadavu, India
[2]Department of Computer Science, Quality College of Management Studies & Science, Bengaluru, India
[3]Department of Electronics & Communication Engineering, Canara Engineering College, Benjanapadavu, India
[4]Department of Information Science & Engineering, NMAM Institute of Technology, Nitte, India
Email: h.nagesh.shenoy@gmail.com, abhishekrao@nitte.edu.in

Received: 04 July 2021; Revised: 03 August 2021; Accepted: 20 August 2021; Published: 08 October 2021

Abstract: The demand for cloud computing systems has increased tremendously in the IT sector and various business applications due to their high computation and cost-effective solutions to various computing problems. This increased demand has raised several challenges such as load balancing and security in cloud systems. Numerous approaches have been presented for load balancing but providing security and maintaining integrity and privacy remains a less explored research area. Intrusion detection systems have emerged as a promising solution to predict attacks. In this work, we develop a deep learning-based scheme that contains data pre-processing, convolution operations, BiLSTM model, attention layer, and CRF modeling. The current study employs a machine learning-based approach to detect intrusions based on the attackers' historical behavior. Deep learning algorithms were used to extract features from the image and determine the significance of dense packets to generate the salient fine-grained feature that can be used to detect malicious traffic and presents the final classification using fused features.

Index Terms: Cloud Computing, Load Balancing, Intrusion Detection, Convolution Neural Network, Cloud Security.

1. Introduction

Nowadays the demand for high computing and e-commerce related applications has increased drastically. These applications are based on coherent models with interrelated tasks [1]. These tasks are realized in the form of workflow models where a task can be represented in a huge number of smaller tasks. These large-scale task-based applications are generally deployed in cloud computing systems because traditional computing systems require more power and time to accomplish the computations [2]. Cloud computing has emerged as a promising solution to handle high-resource-hungry tasks.

Cloud-based systems are widely adopted in the field of information technology and e-commerce related businesses. The cloud computing-based schemes facilitate on-demand access to a shared pool that contains several service-oriented resources such as computation resources, development platforms, and applications. These resources are offered to end-users as pay-as-per-use models [3]. The cloud computing model is a combination of several high computing interconnected systems where more than one united computing resources are available for specific applications [4]. The current advancements in cloud computing have motivated us to develop inter-connected data centers which are located at diverse geographical locations to enable high-quality query processing by performing huge computational tasks [5]. Cloud computing systems have five main characteristics such as on-demand service, broad network access, resource pooling, rapid elasticity, and measured service [6]. These cloud computing models have four different types of deployment such as private cloud, public cloud, community cloud, and hybrid cloud. The resources are offered in the form of different services such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a service (IaaS) [7].

Cloud computing has several advantages but its frequent use in real-time and offline applications causes several challenging issues such as resource management, green computing, portability, interoperability, security, and computing performance. Moreover, the overloading of a cloud center can lead to a delay in completing the task and consuming more energy. Each cloud model has service level agreements (SLAs) where violations of these agreements lead to service plenty charges which increases the overall cost [8]. Several approaches have been developed to deal with these

issues, but security remains a challenging task for the research community [9]. Because the data is stored in remote locations where maintaining privacy and security becomes a challenging task. Currently, the number of attacks on computer networks has increased rapidly where various hacking tools and intrusion mechanisms have emerged as a security threat to the networks.

To determine these attacks, intrusion detection is a significant mechanism to detect suspicious activities on the data [10]. An intrusion detection system (IDS) is a technique that monitors the activity of the user on the stored data and categorizes these activities as normal or malicious activities. Generally, the intrusion detection systems are classified as host-based intrusion detection [11], network-based intrusion detection [12], and distributed intrusion detection systems [13]. Similarly, based on the deployment schemes in cloud computing schemes the IDS are classified as software, hardware, and VM-based intrusion detection systems [14]. Several techniques have been presented to recognize the intrusions in the cloud datacenters.

The intrusion detection scheme is considered as a classification task thus several machine learning and artificial intelligence schemes are widely studied in this field. These machine learning-based techniques are classified as supervised and unsupervised learning. The supervised schemes use labeled data to identify the malicious activity whereas unsupervised schemes don't require any labeling for training purposes i.e. it groups the similar data into their corresponding categories. Several supervised schemes are adopted in intrusion detection systems such as neural networks, support vector machines, decision trees, and many more. Currently, deep learning-based schemes have gained attraction from the research community due to their powerful learning and importance in improving accuracy [15]. [16] used a deep convolution neural network for intrusion detection vehicular network. However, fewer works have been carried out in cloud computing to detect intrusions. The major objectives of the study are as follows:

- To detect an intrusion based on the attackers' previous behavior.
- To convert data traffic into an image and convert traffic data into a numerical form.
- To connect the forward and backward LSTMs to extract the feature from each byte of a data packet.
- To determine the significance of dense packets and provide fine-grained features for detecting malicious traffic.
- To design a classifier that will learn the network based on the retrieved features.

Hence, in this work, our main aim is to develop a machine learning-based technique for intrusion detection in cloud computing.

2. Literature Review

In this section, we discuss various techniques of intrusion detection in cloud computing systems. The current research in this field has attracted machine learning and optimization-based techniques. Currently, privacy-preserving is one of the topmost challenges in a cloud computing system. Several techniques have been presented which are based on the authentication and access control policies. However, these techniques are not suitable in the current scenario where cloud computing systems are widely accessed worldwide due to which the cloud computing systems become highly vulnerable. [17] focused on developing an intelligent intrusion detection system (IDS) with the help of a fuzzy neural network and genetic algorithm. This scheme is carried out into four phases which include cluster formation, initial fuzzy rule generation, Fuzzy rule base optimization, and refinement of the system parameters. Here, the genetic algorithm is applied on fuzzy rule-based to obtain the optimized rule base. Further, this rule base is processed through the neural fuzzy system to learn the attributes in a dynamical learning manner which is later used for refining the parameters. These stages generate the final optimal rule base which is used to detect the intrusion by matching the activity patterns in the predefined optimal rule base. [18] adopted a semi-supervised machine learning technique for intrusion detection in cloud-based robotic systems. This scheme uses a combination of fuzzy logic and ensemble learning scheme which is designed with the help of a classification and regression tree (CART) and 3-layers neural network. Further, fuzzy logic is applied to analyze the unlabeled data which extracts useful information based on fuzzy rules. Finally, ensemble learning is used to train the data, and the fuzzy rule-based evaluation category is designed to predict the intrusions. [19] developed a novel IDS approach by combining the multilayer perceptron artificial bee colony and fuzzy clustering approach. The MLP is used to learn the packet patterns as normal and abnormal whereas the weights and biases of MLP are optimized using the ABC algorithm. Moreover, this scheme uses fuzzy clustering to group similar patterns. [15] discussed that conventional cooperative IDS requires the intervention of other IDSs which causes a delay in decision making and is not sustainable for real-time applications. To overcome these issues, machine learning-based schemes have evolved as a promising solution that provides a proactive decision-making mechanism. In this work, the authors used a deep learning-based approach where denoising autoencoder is considered as the primary building block to design the deep learning architecture. The autoencoder helps to generate the decision with partial information of IDs without receiving complete feedback from IDs. [20] reported that accuracy in IDS is a challenging task. To overcome this issue, the authors developed a novel scheme by combining Fuzzy C Means clustering (FCM) and Support Vector Machine (SVM). The complete approach is divided into three stages: first of all, FCM clustering is applied which divides the data into multiple cluster sets, in next phase, the obtained clusters are processed through the learning SVM classifier to

train the dataset, and finally, the fuzzy aggregation module is applied to combine the outcome of different clusters and generates the final decision. Similarly, [21] also presented a combined scheme using FCM clustering and recurrent neural network. The FCM performs grouping of the data and RNN is used to classify these groups. [22] developed a host-based intrusion detection system to protect the virtual machine from various attacks in cloud computing. According to this scheme, logistic regression is used for feature selection and these selected features are processed through regularization. Later, these attacks are classified by using a combination of linear discriminant analysis, neural network, and decision tree with a bagging approach. [23] reported that existing security schemes such as storing the data in encrypted form, security audit, and access control-based schemes fail to provide the proactive decision for intrusions. Moreover, self-samples and non-self-samples have varied attributes which create additional complexity. To overcome this issue, the authors considered two different data as self and non-self-samples to consider the insider and outsider attacks. To analyze these data, an improved dynamic immune algorithm (IDIA) is presented. This scheme considers an improved negative selection algorithm (iNSA) for mutation and an improved dynamic clone selection algorithm (iDCS) for grouping. The iNSA is used to detect the intrusions and iDCS is used to update the intrusions. [24] used a deep learning approach for feature extraction to improve detection accuracy. This unsupervised feature extraction technique is based on the stacked contractive autoencoders (SCAE) technique. The SCAE approach helps to learn attributes efficiently. Later, SVM based model is also used. This scheme combines deep and shallow learning techniques to improve detection performance. [25] introduced a novel approach for intrusion detection systems by incorporating a novel ensemble approach of feature selection and classification. The feature selection method is based on the univariate ensemble feature selection and the ensemble classifier is constructed with the help of the voting technique.

3. Materials and Methods

This section presents the proposed solution for intrusion detection in cloud computing systems. Generally, cloud data centers are in diverse geographical locations and store a huge amount of data. This data is vulnerable to several security threats hence maintain security, privacy, and data integrity are challenging tasks for cloud computing [26]. In this work, we adopt a machine learning-based scheme to detect the intrusion based on the previous behavior of attackers. The proposed approach is based on the deep learning scheme which consists of five components such as input layer, convolution layer, BiLSTM layer, attention unit, and output layer [27, 28]. Each layer has assigned specific tasks. The Input layer considers the attack data and processes it in an n dimensional sequential vector. In the next phase, the traffic data is transformed into a numerical form where we perform data normalization. This architecture has several convolution layers which are used to convert the data traffic into images. The convolution layer helps to obtain the features from the image data representation. Further, the BiLSTM layer is used to establish the connection between forward and backward LSTM which extracts the feature from each byte of a data packet. The BiLSTM. In attention units, the attention mechanism helps to identify the importance of dense packets and generates the salient fine-grained feature which is useful in detecting malicious traffic. The output layers consider the input from the attention layer where these features are imported to a fully connected layer. The fully connected layer fuses these features. Finally, these fused features are used by the classifier to learn the network and present the final classification.

3.1. Data Pre-processing

In this phase, we first arrange the data into a categorical form as a one-hot encoder mechanism. This representation helps to represent the symbolic features into numerical features which are easier to learn [29]. In this dataset, we have network-related protocols parameters such as UDP, TCP and ICMP which can be represented as [0, 1, 0], [1, 0, 0] and [0, 0, 1], respectively. Further, we normalize this data into a specific range by using statistical normalization. This normalizing process removes redundant data and improves the training process by minimizing the overfitting. The statistical normalization process is given in Eq. (1,2).

$$x_i = \frac{v_i - \mu_i}{\sigma} \tag{1}$$

Where μ denotes the mean of given attributes as $= \frac{1}{n}\sum_{i=1}^{n} v_i$ v_i denotes the actual value of an attribute and σ denotes the standard deviation which is computed as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_i - \mu)} \tag{2}$$

This stage provides the normalized attributes which are used for further process.

3.2 Convolution Layers

The pre-processed data is considered as input to this layer. This layer apprehends the local attributes from the data. The convolution layer is the important module of CNN which performs multiple convolution operations on the feature

maps to generate a high resolution or more informative feature map. Mainly, the convolution operations are categorized as shallow and deep convolutions. The deep convolution layers are useful in obtaining the global information whereas the shallow convolution is used to extract the local information. Thus, increasing the feature numbers makes features coarser. In this work, we design the convolution layer with the help of a tensor whose size is $H \times W \times 1$ where H and W denote the height and width of data which is obtained through data pre-processing. Let us consider that we have N number layer followed by a convolution layer. In this convolution layer, we apply m width filter w then the output of a convolution is $(N - m + 1)$. The convolution computation process is given in Eq. (3):

$$x_{i,k}^{l,j} = f\left(\sum_{a=1}^{m} w_{a,k}^{j}\ x_{i+(k-1)\times s+1-1}^{l-1,j} + b_j\right) \tag{3}$$

Where $x_{i,k}^{l,j}$ denotes the i^{th} unit of j feature map of k^{th} section in the l^{th} layer, s denotes the section range, f is a non-linear mapping function that uses the hyperbolic tangent function $\tanh(.)$.

3.3 BiLSTM Layer

The network traffic data is formulated by using traffic bytes and represented in the form of time series. Therefore, we use the BiLSTM model to obtain the context information for efficient feature learning. The traffic data is given as input to this layer where time-series features are extracted, and a packet vector is obtained. The BiLSTM model generates the coarse-grained features with the help of forward and backward LSTM models. The LSTM has the input gate i, forget gate f, and output gate o which is used to control the information in the memory cell C. Let us consider that input sequence is given as $x = (x_0, x_1, \ldots, x_t)$ at the time t and the hidden states at a time t are denoted as $h = (h_0, h_1, \ldots, h_t)$.

The output of the hidden layer h_{t-1} at the previous time moment is given input to the forget gate, the input x_t at time t forgets the cell state C_t, this can be represented as shown in Eq. (4):

$$f_t = sigmoid\left(W_{hf}h_{t-1} + W_{xf}x_t + b_f\right) \tag{4}$$

Here, the new incoming information is handled by the input gate where $tanh$ function cooperates with the input gate. The $tanh$ function generates the new candidate vector. The input gate generates a value for new information for each item in $\tilde{C}_t$ which controls the adding of new information. This can be expressed as shown in Eq. (5):

$$\begin{gathered} C_t = sigmoid\left(f_t.C_{t-1} + i_t.\tilde{C}_t\right) \\ i_t = sigmoid\ (W_{xi}x_t + W_{hi}h_{t-1} + b_t) \\ \tilde{C}_t = \tanh(W_c x_t + W_c h_{t-1} + b_c) \end{gathered} \tag{5}$$

Now the next phase considers the output gate which is used to control the current state until the state is filtered out which can be expressed as shown in Eq. (6):

$$o_t = sigmoid(W_{x0}x_t + W_{h0}h_{t-1} + b_0) \tag{6}$$

Further, for the time stamp t, the hidden states of a packet for forward and backward BiLSTM model can be expressed as shown in Eq. (7):

$$\begin{gathered} h_t = \overleftarrow{h}_t + \vec{h}_t \\ \vec{h}_t = \tanh\left(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}\right) \\ \overleftarrow{h}_t = \tanh\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}\right) \end{gathered} \tag{7}$$

Where W denotes the connection weights and b denotes the bias vectors. To further improve the performance of prediction for the sequence model, we incorporate a conditional random field. Let us consider that X denotes the observed values and Y denotes the random values which need to be predicted by the model. Here, the CRF defines a conditional probability of y belonging to a class, $y \in Y$. This probability can be given as shown in Eq. (8):

$$p(y|x) = \frac{1}{Z_x}\prod_{s\in S(y,x)}\phi_s\ (y_s, x_s) \tag{8}$$

Where Z_x denotes the normalization factor, S denotes the cliques of undirected graphs G, $\phi_s\ (y_s, x_s)$ denotes the potential on clique s.

3.4 Attention layer

The BiLSTM layer helps to generate the relationship between attribute vectors which is used for learning using the attention layer. Here, the output of the hidden layer at a time t is obtained using a non-linear transformation which can be given as shown in Eq. (9):

$$u_t = \tanh(W_w h_t + b_w) \tag{9}$$

This similarity representation helps to identify the importance of features based on similarity representation u_t with a vector u_w and the importance of weight coefficient α_t. The weight coefficient for these features can be expressed as shown in Eq. (10):

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum \exp(u_t^T u_w)} \tag{10}$$

Finally, these fine-grained features s can be obtained by a weighted sum of h_t based on α_t which can be expressed as shown in Eq. (11):

$$s = \sum \alpha_t h_t \tag{11}$$

Further, a SoftMax function is applied to obtain the initial prediction which is given as shown in Eq. (12):

$$y = softmax(W_h s + b_h) \tag{12}$$

On this data, we apply the CRF model to predict the sequence which is given as shown in Eq. (13):

$$y^* = argmax_{y \in Y}\, \sigma(X, y) \tag{13}$$

where $\sigma(X, y)$ denotes the score function computed as shown in Eq. (14):

$$\sigma(X, y) = \sum_{i=0}^{n} A_{y_i, y_i+1} + \sum_{i=1}^{n} P_{i, y_i} \tag{14}$$

where A is the matrix of scores, P is the score output by the BiLSTM network.

4. Results and Discussion

In this section, we present the outcome of the proposed deep learning-based approach for intrusion detection in a cloud computing system. This approach is tested by using publicly available network intrusion detection datasets which are KDDCup '99' and NSLKDD datasets. The obtained performance is compared with existing techniques. The proposed method is implemented on the Linux Platform with Python 3.7. The processor features 16 GB of RAM, an NVIDIA graphics card with 8 GB of memory, and an Intel i5 processor.

4.1 Datasets

Currently, available datasets are private which are not publicly available due to privacy and security-related issues. Moreover, these datasets are anonymized and suffer from various challenging issues. In this field, the KDDCup 99 is the most commonly available and publicly used dataset.

KDDCup '99' Dataset

Initially, the Defence Advanced Research Projects Agency (DARPA) started the collection of network intrusion datasets in the airforce base local area network, MIT Lincoln Laboratory, and Air Force Research Laboratory. This dataset is named DARPA. To collect this data, this group has installed a huge number of UNIX machines in the MIT Lincon laboratory. Several users are provided access to these machines. The simulation was performed for 9 weeks to acquire the raw TCP data, 7 weeks of data is training and2 weeks of data for testing. In an attack scenario, Windows NT and UNIX machines are used in the MIT laboratory from the external location of the air force LAN. In this experiment, they conducted 39 attacks (32 attacks and 7 attacks) to collect the data. This data has several attack categories where the main aim is to classify the data corresponding to their category as 'dos', 'r2l', 'probe', and 'u2r'. Table 1 illustrates the configuration of the KDDCup '99' data set.

Table 1. KDD 99 dataset (10% instances)

Attack Type	Train	Test
Normal	97278	60593
DOS	391458	229853
Probe	4107	4166
R2L	1126	16189
U2R	52	228
Total	494021	311029

4.2 Performance Measure Parameters

To measure the performance of the proposed approach we use the following terms to determine various parameters:

False Positive (FP): it denotes the total number of instances that are classified into the wrong category i.e. the normal connections are classified as attack category.

True Positive (TP): it denotes the total number of instances that are classified as their normal class i.e. the connection records are classified to their corresponding normal class.

False Negative (FN): it denotes the total count of instances that are wrongly classified to the normal class i.e. attack connection records are classified as normal connection records.

True Negative (TN): it is the count of a total number of instances that are correctly classified to its attack category.

Based on these parameters, we consider accuracy, precision, recall (true positive rate), F1-score, and false-positive rate as the performance metrics.

4.3 Comparative Analysis

Based on the parameters, we measure the performance of the proposed approach and compared it with existing techniques as mentioned in [30]. First, we compare the performance by detecting the instances as attack connection records and normal connection records. For this scenario, the comparative performance is shown in Table 2.

Table.2. Binary classification performance for KDDCup'99' data

Algorithm	Accuracy	Precision	Recall	F-Score
LR	0.811	0.994	0.769	0.867
NB	0.877	0.994	0.852	0.918
KNN	0.925	0.998	0.909	0.952
DT	0.929	0.997	0.915	0.954
AB	0.925	0.996	0.910	0.951
RF	0.927	0.999	0.911	0.953
SVM-rbf	0.877	0.994	0.852	0.918
DNN 1 layer	0.929	0.999	0.914	0.954
DNN 2 layer	0.929	0.998	0.914	0.954
DNN 3 layer	0.930	0.999	0.914	0.95
DNN 4 layer	0.930	0998	0.914	0.95
DNN 5 layer	0.927	0.994	0.9145	0.953
Proposed Model	0.992	0.999	0.989	0.988

The comparative analysis shows that the classification performance of the proposed approach is improved by 22.3181%, 13.1129%, 7.24324%, 6.78149%, 7.24324%, 7.01187%, 13.1129%, 6.78149%, 6.66667%, 6.66667%, and 7.01187% when compared with LR, NB, KNN, DT, AB, RF, SVM-RBF, DNN 1 layer, DNN 2 layer, DNN 3 layer, DNN 4 layer, and DNN 5 layer, respectively. Similarly, we measured the performance for the NSL-KDD dataset. The obtained performance is compared with existing techniques are shown in Table 3.

Table.3. Comparative Analysis for NSL-KDD dataset

Algorithm	Accuracy	Precision	Recall	F-Score
LR	0.826	0.915	0.744	0.820
NB	0.829	0.865	0.805	0.834
KNN	0.910	0.926	0.905	0.915
DT	0.930	0.928	0.43	0.935
AB	0.934	0.961	0.914	0.937
RF	0.929	0.946	0.919	0.933
SVM-rbf	0.837	0.769	0.993	0.867
DNN 1 layer	0.801	0.692	0.969	0.807
DNN 2 layer	0.794	0.685	0.965	0.801
DNN 3 layer	0.793	0.684	0.967	0.801
DNN 4 layer	0.794	0.684	0.967	0.802
DNN 5 layer	0.789	0.680	0.963	0.797
Proposed Model	0.979	0.986	0.999	0.989

According to this experiment, the accuracy performance of proposed approach is improved by 18.523%, 18.0941%, 7.58242%, 5.26882%, 4.81799%, 5.38213%, 16.9654%, 22.2222%, 23.2997%, 23.4552%, 23.2997% and 24.0811% when compared with techniques as mentioned in the previous experiment. Further, we extend the comparative analysis and measured the classification performance for each attack category. The performance is measured by using a multiclass classifier scheme. In this case, we have considered 5 normal classes, DoS, probe, R2L, and U2R. The obtained performance for each classifier is presented in Table 4.

Table.4. Multiclass classifier performance for KDDCup '99' data

Method	**Normal**			**DoS**			**Probe**			**R2L**			**U2R**		
	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR
LR	0.811	0.978	0.229	0.832	0.798	0.058	0.987	0.001	0	1.0	0.014	0	0.974	0.063	0
NB	0.898	0.517	0.01	0.874	0.99	0.503	0.983	0	0.004	0.988	0.514	0.012	0.972	0.001	0
KNN	0.642	0.267	0.267	0.617	0.721	0.72	0.975	0.014	0.012	1	0	0	0.972	0	0
DT	0.646	0.261	0.261	0.617	0.722	0.722	0.974	0.014	0.013	0.998	0	0.002	0.971	0.002	0.002
AB	0.24	0.926	0.925	0.266	0.056	0.055	0.973	0.017	0.014	0.999	0	0.001	0.969	0.004	0.003
RF	0.642	0.268	0.268	0.616	0.72	0.719	0.976	0.012	0.011	1	0	0	0.971	0.02	0.002
SVM-RBF	0.624	0.299	0.298	0.602	0.693	0.692	0.977	0.009	0.009	1	0	0	0.972	0	0
DNN 1 L	0.928	0.995	0.088	0.953	0.939	0.004	0.995	0.732	0.001	1	0.243	0	0.975	0.155	0.002
DNN 2 L	0.928	0.996	0.088	0.955	0.942	0.004	0.995	0.764	0.002	1	0.243	0	0.975	0.089	0
DNN 3 L	0.942	0.982	0.068	0.963	0.962	0.031	0.994	0.706	0.002	1	0	0	0.971	0	0.002
DNN 4 L	0.935	0.946	0.067	0.958	0.961	0.052	0.993	0.765	0.004	1	0	0	0.972	0	0.001
DNN 5 L	0.929	0.991	0.086	0.953	0.939	0.004	0.994	0.752	0.002	1	0.043	0	0.974	0.136	0.003
Proposed Model	0.963	0.996	0.095	0.975	0.967	0.003	0.9945	0.812	0.002	1	0.032	0	0.981	0.121	0.002

This experiment shows that the proposed approach achieves better performance in terms of accuracy, FPT, and TPR for five different attack classes Normal, DoS, Probe, R2L, and U2R. Similarly, we measured the performance of the NSL-KDD dataset for different attack classes. Table 5 shows the classification comparative analysis.

Table 5. Comparative Analysis of NSL-KDD dataset for different attack classes

Method	**Normal**			**DoS**			**Probe**			**R2L**			**U2R**		
	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR	Acc	TPR	FPR
LR	0.682	0.919	0.496	0.77	0.638	0.159	0.899	0	0	0.997	0	0	0.879	0	0
NB	0.534	0.031	0.085	0.371	0.827	0.861	0.886	0	0.008	0.938	0.179	0.069	0.865	0	0.016
KNN	0.769	0.977	0.389	0.889	0.731	0.029	0.929	0.59	0.032	0.997	0.119	0	0.879	0	0
DT	0.799	0.975	0.334	0.917	0.788	0.019	0.93	0.605	0.038	0.998	0.419	0.004	0.882	0.07	0
AB	0.698	0.631	0.250	0.757	0.861	0.295	0.89	0.336	0.043	0.999	0.438	0	0.898	0.168	0
RF	0.768	0.976	0.388	0.908	0.758	0.015	0.97	0.648	0.016	0.997	0.522	0	0.885	0.049	0
SVM-rbf	0.712	0.998	0.507	0.875	0.644	0	0.942	0.512	0	0.997	0	0	0.879	0	0
DNN 1 L	0.829	0.973	0.279	0.907	0.777	0.028	0.936	0.61	0.024	0.998	0.433	0	0.886	0.241	0.026
DNN 2 L	0.841	0.971	0.257	0.907	0.764	0.019	0.926	0.703	0.047	0.997	0.224	0	0.883	0.199	0.024
DNN 3 L	0.838	0.973	0.264	0.909	0.772	0.021	0.927	0.636	0.038	0.997	0.239	0	0.89	0.26	0.024
DNN 4 L	0.832	0.974	0.276	0.91	0.764	0.015	0.915	0.663	0.055	0.998	0.672	0.002	0.906	0.242	0.003
DNN 5 L	0.831	0.974	0.277	0.915	0.795	0.024	0.924	0.634	0.041	0.998	0.269	0	0.903	0.229	0.005
Proposed Mod	0.935	0.982	0.257	0.933	0.891	0.021	0.933	0.641	0.032	0.998	0.190	0	0.951	0.211	0.004

The experimental analysis shows a significant improvement in intrusion detection. Moreover, the proposed approach classifies the intrusion according to their attack category which is useful to prevent the attacks. The multiclass classification scheme shows a noteworthy classification performance for different types of attacks.

5. Conclusion

In this work, we have focused on improving the performance of cloud computing. Cloud computing faces several challenges such as load balancing and maintaining security and privacy. The load-balancing field is widely explored but providing security to cloud service users remains a challenging task. Hence, we focus on the security aspects of the cloud. Intrusion detection is an important mechanism to detect the attack on the data. In this work, we present a deep

learning-based approach for intrusion detection. This approach is based on the BiLSTM and CRF-based model where the outcome of the SoftMax layer is fine-tuned using CRF. The proposed method succeeded in categorizing the intrusions based on the type of attack, which is important for preventing attacks. The multiclass classification system provides impressive classification performance for numerous types of attacks.

References

[1] Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, *91*, 407-415.
[2] Shahidinejad, A., Ghobaei-Arani, M., & Masdari, M. (2020). Resource provisioning using workload clustering in cloud computing environment: a hybrid approach. Cluster Computing, 1-24.
[3] Gochhait, S., Butt, S. A., Jamal, T., & Ali, A. (2020). Cloud enhances agile software development. In *Cloud Computing Applications and Techniques for E-Commerce* (pp. 28-49). IGI Global.
[4] Kumar, M., Sharma, S. C., Goel, A., & Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications*, *143*, 1-33.
[5] Elsedimy, E. I., & Algarni, F. (2021). Toward Enhancing the Energy Efficiency and Minimizing the SLA Violations in Cloud Data Centers. *Applied Computational Intelligence and Soft Computing*, *2021*.
[6] Ali, M. B. (2021). Multi-Perspectives of Cloud Computing Service Adoption Quality and Risks in Higher Education. In Handbook of Research on Modern Educational Technologies, Applications, and Management (pp. 1-19). IGI Global.
[7] Lee, I. (2021). Pricing and Profit Management Models for SaaS Providers and IaaS Providers. *Journal of Theoretical and Applied Electronic Commerce Research*, *16*(4), 859-873.
[8] Kim, M. H., Lee, J. Y., Shah, S. A. R., Kim, T. H., & Noh, S. Y. (2021). Min-max exclusive virtual machine placement in cloud computing for scientific data environment. *Journal of Cloud Computing*, *10*(1), 1-17.
[9] Nagesh Shenoy H, K R Anil Kumar, Rajgopal K T and Abhishek S. Rao (2020). An Audit on Cloud Architectures Addressing Data Privacy and Security Concerns. *International Journal of Advanced Science and Technology*, Vol. 29, No. 6, (2020), pp. 6373 – 6382.
[10] Shyla, S. I., & Sujatha, S. S. (2020). Cloud Security: LKM and Optimal Fuzzy System for Intrusion Detection in Cloud Environment. *Journal of Intelligent Systems*, *29*(1), 1626-1642.
[11] Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of things. *Journal of Cloud Computing*, *9*(1), 1-16.
[12] Bedi, P., Gupta, N., & Jindal, V. (2021). I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems. *Applied Intelligence*, *51*(2), 1133-1151.
[13] Liu, J., Zhang, W., Ma, T., Tang, Z., Xie, Y., Gui, W., & Niyoyita, J. P. (2020). Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection. *Expert Systems with Applications*, *158*, 113578.
[14] Roschke, S., Cheng, F., & Meinel, C. (2009, December). Intrusion detection in the cloud. In *2009 eighth IEEE international conference on dependable, autonomic and secure computing* (pp. 729-734). IEEE.
[15] Abusitta, A., Bellaiche, M., Dagenais, M., & Halabi, T. (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems*, *98*, 308-318.
[16] Song, H. M., Woo, J., & Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. Vehicular Communications, 21, 100198.
[17] Raja, S., & Ramaiah, S. (2017). An efficient fuzzy-based hybrid system to cloud intrusion detection. *International Journal of Fuzzy Systems*, *19*(1), 62-77.
[18] Gao, Y., Liu, Y., Jin, Y., Chen, J., & Wu, H. (2018). A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. *IEEE Access*, *6*, 50927-50938.
[19] Hajimirzaei, B., & Navimipour, N. J. (2019). Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express, 5(1), 56-59.
[20] Jaber, A. N., & Rehman, S. U. (2020). FCM–SVM based intrusion detection system for cloud computing environment. Cluster Computing, 1-11.
[21] Manickam, M., Ramaraj, N., & Chellappan, C. (2019). A combined PFCM and recurrent neural network-based intrusion detection system for cloud environment. *International Journal of Business Intelligence and Data Mining*, *14*(4), 504-527.
[22] Besharati, E., Naderan, M., & Namjoo, E. (2019). LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. Journal of Ambient Intelligence and Humanized Computing, 10(9), 3669-3692.
[23] Wang, W., Ren, L., Chen, L., & Ding, Y. (2019). Intrusion detection and security calculation in industrial cloud storage based on an improved dynamic immune algorithm. Information Sciences, 501, 543-557.
[24] Wang, W., Du, X., Shan, D., Qin, R., & Wang, N. (2020). Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. *IEEE Transactions on Cloud Computing*.
[25] Krishnaveni, S., Sivamohan, S., Sridhar, S. S., & Prabakaran, S. (2021). Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Cluster Computing, 1-19.
[26] Nagesh Shenoy H, K. R. Anil Kumar, Suchitra N Shenoy, & Abhishek S. Rao (2020). "Data Security in Cloud Environment Based on Comparative Performance Evaluation of Cryptographic Algorithms". *International Journal of Advanced Trends in Computer Science and Engineering*, Volume 9, No.4, July – August 2020.
[27] Rao, A. S., Sandhya S, Anusha K, Arpitha, Chandana Nayak, Meghana, Sneha Nayak. (2020). Exploring Deep Learning Techniques for Kannada Handwritten Character Recognition: A Boon for Digitization. *International Journal of Advanced Science and Technology*, 29(5), 11078-11093.

[28] Rao, A. S., Kamath, B. S., Ramya, R., Chowdhury, S., Shreya, A., & Pattan, R. K. K. (2020). Use of Artificial Neural Network in Developing a Personality Prediction Model for Career Guidance: A Boon for Career Counselors. International Journal of Control and Automation, 13(4), 391-400.
[29] Rao, A. S., Aruna Kumar, S. V., Jogi, P., Chinthan Bhat, K., Kuladeep Kumar, B., & Gouda, P. Student Placement Prediction Model: A Data Mining Perspective for Outcome-Based Education System. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*, 2497-2507.
[30] Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550.

Authors' Profiles

Nagesh Shenoy H received his B. E degree in Information Science & Engineering from Canara Engineering College, Bantwal, M.Tech degree in Computer Science and Engineering from NMAM Institute of Technology, Nitte, Visvesvaraya Technological University, India and M.B.A degree in Information Technology, SMU. His major research interest is in the fields of Cloud Computing and Cloud Security. He has 4 year of industrial experience and 8 years of teaching experience. He is currently pursuing his Ph.D. in the area of Cloud Computing under Visvesvaraya Technological University and working as an Assistant Professor in the Department of Computer Science & Engineering at Canara Engineering College, Benjanapadavu. He is a member of ISTE.

Dr. K. R. Anil Kumar is working as a Principal at Quality College of Management Studies & Science, Bangalore. In the year 2012, he received his Ph.D. in the area of Cloud Computing. His major research interests are in the area of Cloud Computing and Security.

Suchitra N Shenoy received her B. E degree in Electronics and Communication Engineering from Canara Engineering College, Bantwal, MS degree in VLSI CAD from MIT, Manipal, India Her major research interest is in the fields of Image Processing and Machine Learning. She has 2 years of industrial experience and currently pursuing her Ph.D. in the area of Image Processing at Visvesvaraya Technological University, India.

Abhishek S. Rao received his B. E degree in Information Science & Engineering from Canara Engineering College, Bantwal, M.Tech degree in Computer Science and Engineering from NMAM Institute of Technology, Nitte, Visvesvaraya Technological University, India and M.B.A degree in Operations Management from MIT, Pune. His major research interest is in the fields of Machine Learning, Deep Learning, and Computer Vision. He has 2 years of industrial experience and 9 years of teaching experience. He is currently working as an Assistant Professor in the Department of Information Science & Engineering at NMAM Institute of Technology, Nitte. He is a member of ISTE and IAENG.

Rajgopal K T received his B. E degree in Information Science & Engineering from SDMCET, Dharwad, M.Tech degree in Computer Network Engineering from R V College of Engineering, Bengaluru, Visvesvaraya Technological University, India. His major research interest is in the fields of Cloud Computing. He has 8 years of teaching experience. He is currently pursuing his Ph.D. in the area of Cloud Computing under Visvesvaraya Technological University and working as an Assistant Professor in the Department of Computer Science & Engineering at Canara Engineering College, Benjanapadavu. He is a member of ISTE.

How to cite this paper: Nagesh Shenoy H, K. R. Anil Kumar, Suchitra N Shenoy, Abhishek S. Rao, Rajgopal K T, "Exploring Deep Learning Techniques in Cloud Computing to Detect Malicious Network Traffic: A Sustainable Computing Approach", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.11, No.5, pp. 9-17, 2021.DOI: 10.5815/ijwmt.2021.05.02