An Intelligent Intrusion Detection System Using Hybrid Deep Learning Approaches in Cloud Environment

Andrea Sharon, Prarthna Mohanraj, Tanya Elizabeth Abraham, Bose Sundan, Anitha Thangasamy

An Intelligent Intrusion Detection System using Hybrid Deep Learning Approaches in Cloud Environment

Andrea Sharon[1], Prarthna Mohanraj[1], Tanya Elizabeth Abraham[1], Bose Sundan[1] and Anitha Thangasamy[2*]

[1, 2*] Department of Computer Science and Engineering,
College of Engineering, Guindy, Anna University, Chennai
andreasharon01@gmail.com
prarthna.ms@gmail.com
tanya.abraham@yahoo.co.in
bosesundan@gmail.com
ani.astt18@gmail.com

Abstract. An Intrusion Detection System (IDS) detects suspicious activities and sends alerts when they are found. Based on these alerts, the issue is investigated, and appropriate actions are taken to remediate the threat. The traffic in a network is examined by a network-based intrusion detection system using various traffic tools that collect and analyse traffic data utilizing detection algorithms. Virtualization is used to construct the cloud infrastructure, which renders the virtual network flow between the virtual machines and it is mostly unidentifiable by typical intrusion detection systems. Previous studies proposed a software-defined network technology to reroute network traffic to a Snort IDS for detection of malicious attacks. However, this is incapable of detecting unknown attacks and adapting to large-scale traffic. Deep learning algorithms are used automatically to extract essential features from raw network data, which can then be fed into a shallow classifier for effective malicious attack detection. The main objective of the proposed system is to utilize a combination of a sparse autoencoder and stacked contractive autoencoder (S-SCAE) along with a Bi-DLDA (Bi-directional LSTM followed by a dense layer, a dropout layer, and a layer with attention mechanism) for detecting intrusions in a cloud environment. Moreover, a cloud intrusion detection system that designed to collect the data traffic from the NSL-KDD dataset and applies the S-SCAE + Bi-DLDA algorithm to determine if the received packet is malicious or non-malicious. To assess the proposed system's detection performance, a variety of measures were used such as precision, recall rate, and accuracy. The proposed model achieves precision, recall rate, and accuracy of 99%, 98%, and over 98 % respectively, according to simulation findings.

Keywords: Attack detection, Bi-Directional LSTM with Dropout and Attention Layer, Cloud Computing, Distributed Denial of Service, Long Short-Term Memory, One-Hot Encoding, Sparse Autoencoder - Stacked Contractive Autoencoder.

1 Introduction

Society has embraced rapid advancements in technology. These advancements have been incorporated into everyday life by people for personal use and by organizations for their internal operations and business solutions. However, this rapid technological growth has increased vulnerabilities, threats, and cyber-crime. Organizations have implemented several mechanisms to protect themselves from these computer system intrusions such as a firewall and an antivirus. A firewall is a type of security device used to monitor inbound and outbound network traffic. Antivirus software examines the data (files, software, web pages, and applications) travelling to a device on which the software is. It also schedules automatic scans and deletes harmful code/software. Although these mechanisms can protect computer systems from malicious attacks, intrusion detection systems are more effective and efficient.

Cloud Computing has arisen a standard platform in the current year for sharing data in a large pool and it also offers several user-friendly characteristics [15]. Moreover, it defined a model, which allows sharing a configurable pool of computing resources like servers, networks, services, applications, and storage that can quickly release with minimum user effort. Profits of making services exist at anywhere at any time and making resources to be added or removed are one the big advantage of the cloud. The majority of cloud computing services are available on a pay-as-you-go basis, with each user assigned an individual collection of devices for the extraction of data. The services of cloud computing classification illustrated in Fig 1. [9] Service Delivery Model depends on the kind of provided cloud service and it is grouped into three categories are expressed in Fig 1.

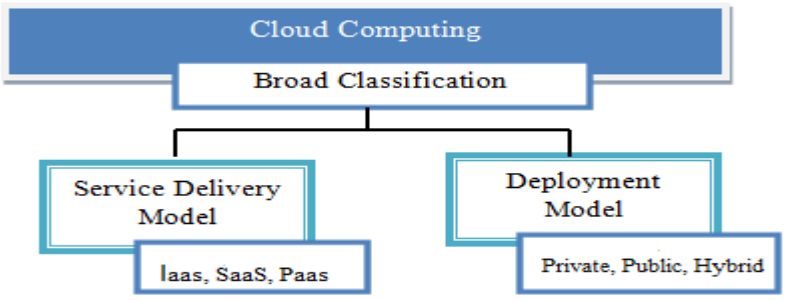

Fig. 1. Classification of Cloud Computing [4]

Whereas the deployment model depends on cloud deployment is sub-divided into the private, hybrid, and public cloud. Data protection from cybercriminals is among the most challenging aspects of a cloud platform because the majority of the data is public. There are several attacks and threats in a cloud environment, which illustrated in Fig 2.

Fig. 2. Types of Threats in Cloud Computing [4]

An Intrusion Detection System (IDS) monitors network traffic and generates notifications whenever suspicious activity is detected. This type of security system collects data and information from various network sources and four systems. The data collected, then analyzed to detect if an activity may constitute an intrusion or attack on the system and helps system administrators and computer systems to prepare and deal with attacks or intrusions aimed at their network(s). In addition, intrusion detection systems used to identify anomalies before hackers can make any or a considerable amount of damage to a network. There are 2 different types of intrusion detection systems: host-based and network-based. This categorization done based on the source of information.

A host-based intrusion detection system [3] has been used in a single host system and it can only serve the system in which it is installed. Host-based agents, sensors, installed on a machine that found to be susceptible to possible attacks on the system. A separate sensor is required for each machine. Sensors collect data on the events occurring, which are being monitored. This type of IDS assesses the host system's security performance and analyzes the log information. Generally, Logs are typically basic text files in which a few lines are written at a time while events and processes take place. Host-based intrusion detection systems are advantageous for many reasons. Host-based systems can monitor access to information ("who accessed what"). In simple terms, these systems can trace harmful or malicious activities to a specific user. This aspect helps identity whether a person within an organization is responsible for the improper use of resources. Host-based systems are very versatile since they can operate in encrypted environments and over a switched network topology. This type of intrusion detection system can also distribute the load across the available hosts on large networks, cutting the deployment cost. When the network traffic becomes too large, this feature of host-based systems provides a benefit by spreading the load evenly over a network. Host-based systems also present several disadvantages. Host-based sensor systems are not portable. Since the sensors are host-based, they must be compatible with the platform, which they are running over. Setting up this type of system can be very costly. The management and deployment costs for this type of intrusion detection system are more costly because each host requires its sensor. An additional disadvantage of a host-based intrusion detection system is that it can't "see" network activity and relies significantly on the host operating system. The integrity of host-

based sensors can be weakened by vulnerabilities.

On the other hand, a network-based intrusion detection system (NIDS) [8] verifies network traffic using traffic tools that capture and analyse traffic data using detection algorithms. Unlike a host-based intrusion detection system, information gathered from a whole network rather than from each separate host. NIDS are deployed with one or many prominent points throughout a network to monitor traffic on a network from together with all of the network's devices. An analysis performed on the network traffic to look for abnormal behaviour and patterns. The content and header information, of all the packets moving through the network, inspected to look for signs of an attack. Network sensors contain attack signatures, rules on what will be considered an attack. Sensors compare the attack signatures to those of the ones captured from the network traffic and then identify the hostile traffic. Network-based intrusion detection systems are portable and are independent of the operating system in which they installed. Similarly, it can be introduced into an existing network, or any part of one, efficiently with minute disruptions if any. However, there are disadvantages to this type of intrusion detection system. Network sensors identify attacks based on their attack signatures, which based on data, collected from previous and known attacks. Even though attacks with recognized signatures can be prevented, ones without predefined signatures have the potential to create a great amount of damage. Another major issue with a network-based intrusion detection system is scalability. Every packet that passes through the segment on which it is placed is inspected by network monitors. This type of IDS has difficulty keeping up with a 100 Mbps environment. High-speed networks are becoming more and more common; attackers will target to exploit this weakness. Encryption is also a problem with these IDS. If network traffic is encrypted, then agents will not be able to scan the contents of those plackets.

Cloud infrastructure built with virtualization renders virtual network traffic across VMs undetectable and unmanageable by standard intrusion detection systems. Previous studies have reported that network traffic be redirected to a Snort IDS for detection of malicious attacks using software-defined network (SDN) technologies [13]. Snort is a network intrusion prevention and detection system that is free and open source that detects any unlawful behaviour using a versatile rule-based language comprised of signature, protocol, and inspection approaches. Snort records the packet in a human-readable form. It has the capability of detecting worms and it exploits the port, which scans and detects suspicious activity through protocol analysis, pre-processors, and content searching. Snort, on the other hand, is incapable of identifying unknown attacks or adjusting to massive amounts of traffic.

Anomaly-based, misuse-based, and hybrid-based intrusion detection systems can also be distinguished by the detection process. An anomaly is an outlier or something, which deviates from the expected. This type of intrusion detection system discovers patterns that are far from normal and labelled as intrusion. Anomaly detections can be characterized by static and dynamic detectors. Anomaly-based IDS been well built to differentiate anomalous traffic from normal traffic and the detection of unknown attacks. However, because some intrusion might simulate normal activities, it is also associated with a high false alarm rate. A misuse-based IDS creates a database that stores several attacks. It is also known as signature-based detection because it employs a set of recognised patterns stored in a database. The patterns, used in a misuse-based intrusion detection system, are a collection of sequences of activities that have a possibility of being harmful. The time taken to match with a pattern from the database is

minimal. The benefit of this type of system is that the patterns can be easily understood if the network behaviour is familiar or recognized. Misuse-based intrusion detection systems are especially suited for intrusions where the attack patterns have already been saved in the database, but they might handle attacks with human interference and perhaps self-modifying behavioural characteristics. Although misuse-based IDS is not suited for detecting novel attacks, it does enable very accurate detection of previously identified anomalous activities. Lastly, the hybrid method combines both anomaly-based and misuse-based methods.

The main contribution of the work is as follows:

- To utilize a combination of Sparse Autoencoder and Stacked Contractive Autoencoder (S-SCAE) along with a Bi-DLDA (Bi-directional LSTM followed by a dense layer, a dropout layer, and a layer with attention mechanism) for detecting intrusions in a cloud environment.
- To design a cloud IDS, which collects the data, traffic from the NSL-KDD dataset and applies the S-SCAE + Bi-DLDA algorithm to determine if the received packet is malicious or non-malicious.

The remaining section of this article, titled Section II, discusses the detection of DDoS attacks using various approaches like Sparse Autoencoder - Stacked Contractive Autoencoder, Bi-Directional LSTM with Dropout and Attention Layer, etc. The proposed system architecture for detecting DDoS attacks is mainly explained in Section III with the Sp-SCAE and the Bi-DLDA method. In Section IV, the findings of the experimental results and the comparison analysis are briefly presented and finally in Section V, the conclusion and future work has been discussed.

2 Related Work

In this related work section will be discussing on a comprehensive examination of most advanced intrusion detection technologies, which intended to lessen the risk of DDoS attacks. Self-taught learning (STL)-IDS, a powerful deep learning approach based on the STL framework, has been presented. The sparse autoencoder technique was used to generate the model, which is an efficient unsupervised method of learning for reconstructing novel feature representations. The proposed method used to learn features and reduce dimensionality. The SAE-SVM algorithm, used to [1], employs a 1-n encoding scheme to convert non-numerical attributes to numbers before the application of STL. Because most of the NSL-KDD dataset's features have very huge ranges between the minimum and maximum values, those values of features are contradictory and inappropriate for computation. Those features are then normalized to the range [0,1]. The proposed method significantly lowers the amount of time spent on training and testing while effectively improving the predictive performance of SVM about attacks. The advantage is that this model uses a sparse autoencoder, which is more robust to noise because of which the important features from the data extracted easily but the disadvantage of using this proposed model in this paper has a lower accuracy when compared to the other existing models.
[2] developed new self-organizing map algorithms that successfully updated neighbourhood laws and learning rates to govern basic SOM weight vectors are randomly assigned and have a static architecture as well as the initial data size of the weight

vector. Performance metrics such as detection rate and false alarm rate were pre-owned to evaluate the novel technique.

The developing classifier for IDS using the DL method [5] has been contributed in this paper. The most appropriate optimizer is chosen among 6 optimizers for LSTM RNN is utilized to predict intrusion. Experimental result shows LSTM RNN with Nadam optimizer gives better result than previous work.

The proposed TLS-BLSTM (Transfer Learning-depended on Stacked-BLSTM) [6] network detect the quality of air for newly emerged stations which have to lack data. The proposed technique combines a transfer-learning plan and deep learning methods to use concepts from the current air quality station to the new station for boosting the forecast. Experimental results depicts that the proposed technique lowest RMSE for a sample of 3 pollutants in the new station. [7] presents a solution for predicting botnet activity inside network and consumer IOT devices. The deep learning method utilized to create a prediction model depending on Bidirectional (BLSTM-RNN). Word embedding utilized to recognize text and to convert attacked packed to the format of tokenized integer. Experiments show that the proposed method proves a better model over some time.

D-Sign is a deep learning-powered solution [10] for identifying hybrid intrusions and developing signatures for unknown web vulnerabilities. D-Sign is divided into three tiers: the Misuse Detection Engine, Signature Generation Engine and the Anomaly Detection Engine. The detection engine analyzes the traffic captured by the honey pot servers, decoy servers that are set up to attract the attacker and monitor/log activities, and detects attacks. The misuse detection engine uses a rule-based approach and deployed after the honey pot servers to filter the suspicious traffic
collection for known threats. The anomaly detection engine builds a profile of normal behaviour. A normal profile consists of patterns or descriptive statistic from the network traffic's non-harmful community. The signature generation engine continues to generate content-based attack signatures from a malicious collection of packets. With a high degree of sensitivity, precision, and specificity, D-Sign can successfully detect and generate attack signatures. Here the web-based attacks must be detected and signatures to be generated. The proposed model is a strong defense methodology that detects new threats within a short amount of time from its launch and with a minimal amount of damage to information. The advantage of using the multilayer LSTM model overcomes long-term dependency, the vanishing gradient problem, and the drawback of this LSTM model works at a slow speed when compared to some other models.

The scalable outlier decoder was [11] introduced, which is a combination of LSTM and hierarchical clustering (HC). Where HC give scalability to the outlier detectors by computing correlated sensors. LSTM is combined with M-estimator, robust statistics to accurately predict outliers in time-series data. The simulation result expresses that the proposed method has higher accuracy for various attacks.

[12] proposes a BAT model to address the issues of less accuracy and model evaluation in Bi-LSTM intrusion detection and an attention mechanism are combined with the BAT model. The network flow vector, which mainly composed of packet vectors created by the Bi-LSTM model and may acquire critical components for traffic flow detection, was tested using the attention mechanism. The proposed model consists of an input layer, multiple convolution layers, a BLSTM layer, an attention layer, and an output layer. In the input layer, the input that has been transformed into numerical

features using the one-hot encoding method. Then, a standard scalar is used to normalize the data into the range [0,1]. After pre-processing occurs on the input data, the convolution layer captures the local features of the traffic data. This layer creates different feature maps. The BLSTM layer, an enhanced version of the LSTM layer, connects a forward and backward LSTM to extract coarse-grained characteristics. The time series feature of a data packet acquired via the BLSTM layer. From the BLSTM to the attention layer, forward propagation is carried out. Following that, the attention layer will know the relationships between the packet vectors. The BAT-MC model eliminates the issue of conventional design features. The advantage of this model is that the Bi-LSTM algorithm is faster at learning and better at remembering new features and forgetting older features as compared to other existing algorithms. The attention mechanism is useful in obtaining accurate and reasonable features from the output vectors of the Bi-LSTM algorithm. The main disadvantage is that the model proposed in this paper does not account for over-fitting of data.

[14] proposed novel DL techniques for the prediction of real-time threats in IOT systems with the use of BLSTM RNN. The proposed method had been employed over python programming language and Google Tensor Flow framework. The experiment conducted on UNSW-NB15 datasets and the results shows in terms of intrusion prediction, the suggested strategy outperforms previous methods.

[16] suggested LSTM network depended on numerous-feature layers. Primarily, layer stage feature is presented where historical data is saved and computed to discover variable duration's various stages numerous stage attacks. Then layer time-series feature is utilized to relate autonomous attack stages to see if current data fits under the attack interval. The experiment represents the proposed scheme had the lowest positive rate compared to the existing scheme.

[17] describe a model using SCAE and SVM for detecting intrusion attacks. The SCAE used for feature extraction and the extracted features are used for training the SVM classifier. SCAE is comprised of multiple hidden layers for encoding. For decoding, the SCAE consists of a set of symmetrical layers. Here, the output of one layer is the input for the next layer in these layers. On the NSL-KDD dataset, five distinct kinds of SCAE + SVM models were created (SCAE1+SVM, SCAE2+SVM, SCAE3+SVM, SCAE4+SVM, and SCAE5+SVM). Out of all five models, SCAE4+SVM had the better categorization performance. As a result, SCAE can extract the best features and meet the aim of dimensionality reduction. The SCAE approach can be used to continuously train better and more robust low-dimensional features from raw network traffic. The SCAE+SVM classifier detects whether the given online traffic or testing data considered normal or an attack (i.e., DOS, Probe, R2L, U2R). Research with the use of KDD Cup 99 and NSL-KDD, two well-known intrusion detection evaluation datasets, reveal that the proposed SCAE+SVM technique outperforms three other state-of-the-art approaches in terms of detection rate. The advantage of this model produces a high accuracy rate as compared to earlier models. The disadvantages of using this SVM classifier model the new attacks in the testing dataset are not effectively recognised.

This study proposes [18] an IDS model based on AE and LSTM cells. The autoencoder (AE) be a feature compression approaches that usually done by a neural network. The autoencoder's encoding step will reduce the number of features and translate the high-dimensional input into a low-dimensional space. This is possible due to the bottleneck structure present in the autoencoder. The data will be re-constructed during the

decoding process so that it can be given to the neural network again with a loss function for network training. The gradient descent process relies heavily on the loss function. The ability of the autoencoder will enhanced by using a stacked autoencoder, multiple autoencoder is stacked together. A feature extractor, a classifier, and an evaluation block are all part of the intrusion detection model's overall design. The input data pre-processed by feature transformation and normalization before feature extraction. The nominal feature transformed into numerical values using feature transformation. More precise and deeper representation of features permitted by the LSTM and the multi-layer neural network. Lastly, the softmax layer used to classify the extracted features. By using LSTM cells is that it excels in extracting information that is more latent. The main advantage of the LSTM algorithm is better at learning new features as compared to other existing algorithms but the disadvantage of the model proposed in this paper produces a high false alarm rate.

Policy repository methodology proposed [19] to store the policies attached with each file. With the help of the Policy Enforcement Engine, Policy Enforcement Agent starts enforcing the policy on the uploaded file. The Policy Management Module is in charge of creating, modifying, deleting, and enforcing policies for each file or group. For each data piece saved, a correct data retention policy to be handle. The computation made more efficient by using a Hadoop framework operating on the private cloud to check the retention duration of samples mixed to each file.

[20] proposed solution relied on a technique called univariate ensemble feature selection, which is used to choose valuable minimized feature sets from incursion datasets. Ensemble classifiers, on the other hand, are capable of competently fusing to create a robust classifier, combine separate classifiers utilizing the voting process. The preferred solution, which based on an ensemble, accurately distinguishes between normal and malicious network traffic pattern.

3 Proposed Methodology

Data traffic obtained from the NSL-KDD cloud dataset. This dataset, which contains 43 features per record, is the global standard for current internet traffic. Of those 43 features, 41 of them refer to the traffic input themselves. The remaining two features are the classification of the attack and the score (severity of the traffic input). Intrusion Detection Systems classify traffic into 4 categories as intrinsic, content-based, time-based, and host-based based on the information obtained through traffic features. The proposed system uses pre-processing techniques and algorithms for detecting malicious attacks. i.e. identifying whether the cloud network traffic behavior is malicious or normal. Suppose the traffic is Denial of Service (DOS), or User to Root (U2R), or Probe, or Root to Local (R2L), that will be considered malicious. The proposed system Figure.3 uses list of modules as Collection of Data, Data Pre-Processing, Sparse Auto Encoder – Stacked Contractive Auto Encoder (S-SCAE), and Bi-Directional LSTM with Dropout and Attention Layer (Bi-DLDA).

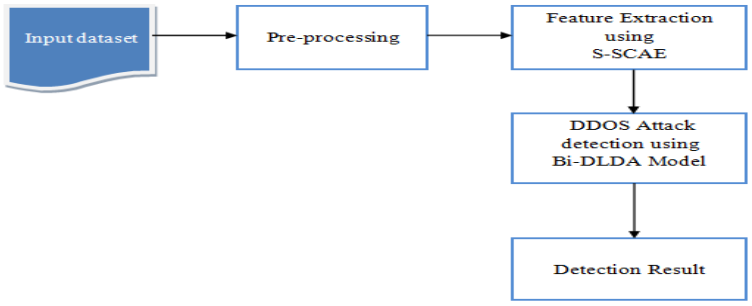

Fig. 3. The architecture of DDoS attack detection based on S-SCAE + Bi-DLDA

3.1 Pre-processing the Dataset

One-Hot Processing

The One-Hot Processing method, used for transforming the symbolic features into numerical features. This necessitates the mapping of category variables to integer values. Except for the integer's index, which is specified with 1, each integer value is shown with vector consisting completely of zero values.

Standardisation

The Standardisation method, used for scaling every feature into a well-balanced range. This method is done to remove the bias in favour of features with higher values along with lower value 0. Standardisation is done using the Z-Score method.

3.2 DDoS Feature extraction using Sparse Auto Encoder – Stacked Contractive Auto Encoder (S-SCAE)

S-SCAE algorithm used for extracting features from the raw network traffic data. It consists of a sparse auto encoder followed by a stacked contractive autoencoder (SCAE). The sparse autoencoder Fig 3. employs sparsity to achieve an information bottleneck. The pre-processed data passed to the input vector of the Sparse Autoencoder. This data encoded using a weight matrix and passed to the inner hidden layer. From here, the data is decoded using a tied weight matrix and is outputted to a reconstruction vector. Fig 4. describes the overall framework of the proposed DDoS attack detection model.

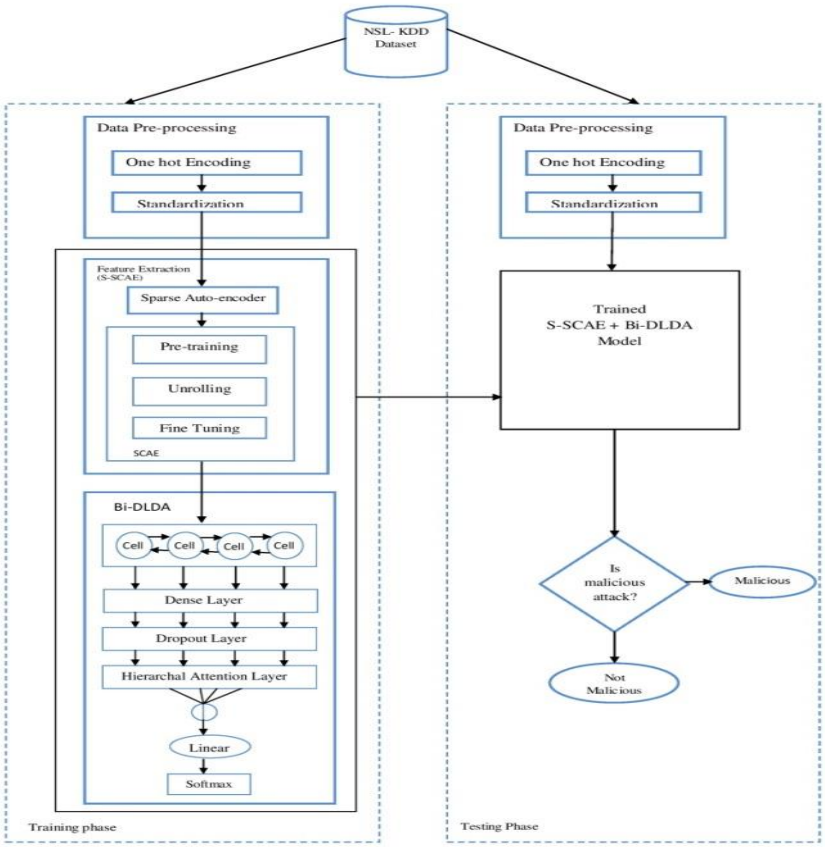

Fig. 4. The overall framework of the proposed DDoS Attacks Detection model

Next, the data from the Sparse Autoencoder passed to the Stacked Contractive Autoencoder Fig 5. Here, the data undergoes pre-training wherein a greedy layer-by-layer strategy is employed for training a series of basic Contractive Autoencoders individually. Each CAE network's output has now become the following CAE network's input.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 117)]             0
_________________________________________________________________
dense (Dense)                (None, 52)                6136
_________________________________________________________________
dense_1 (Dense)              (None, 26)                1378
_________________________________________________________________
dense_2 (Dense)              (None, 26)                702
_________________________________________________________________
dense_3 (Dense)              (None, 117)               3159
=================================================================
Total params: 11,375
Trainable params: 11,375
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5. Structure of the Sparse Autoencoder

The next step is unrolling, in which the hidden layer of each Constructive Autoencoder network is unrolled and stacked to a deep or stacked Constructive Autoencoder network Fig 6. The final step in the process is fine-tuning in which the parameters are adjusted using a multi-class cross-entropy function.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 117)]	0
dense_4 (Dense)	(None, 52)	6136
dense_5 (Dense)	(None, 26)	1378
dense_8 (Dense)	(None, 52)	1404
dense_9 (Dense)	(None, 26)	1378
dense_10 (Dense)	(None, 26)	702
dense_11 (Dense)	(None, 117)	3159

Total params: 14,157
Trainable params: 14,157
Non-trainable params: 0

Fig. 6. Structure of the Stacked Contractive Autoencoder

3.3 Proposed Bi–Directional LSTM with Dropout and Attention Layer (Bi-DLDA) Attack Detection

The Bi-DLDA Classifier trained with the essential features extracted by the S-SCAE algorithm. The structure of the Bi-DLDA Classifier described in Figure 7. Bi-LSTM used for learning the features of each packet and obtaining a characteristic vector related to that packet. The packet vector is passed on to a dense followed by a dropout layer, and then to an attention layer where to extract the detailed features, an attention mechanism is required to carry out feature learning on the packet sequence. The S-SCAE+Bi-DLDA classifier outputs whether the selected packet is malicious or not.
In the context of proposed Bi-DLDA, classification model based IDS learn cloud network connection features from existing data and perform classification tasks on unseen network traffic by categorizing them as normal or malicious. This malicious traffic can be either DoS, probe, R2L, or U2R.

Step-by-Step procedure for Bi-DLDA Learning Method:

Step 1: The extracted features from the S-SCAE passed onto the Bi-DLDA for the classification of the packet.
Step 2: First, data fed into a Bi-LSTM where the input passed in both the forward and backward directions. This employs the use of two activation and candidate values (one

for each direction). Both activation values considered while calculating the output value of each cell.
Step 3: The Bi-LSTM uses a Dropout layer, which randomly sets input units to 0, thereby preventing over fitting.
Step 4: The output passed to the attention mechanism layer. Feature learning is carried out on sequential data made up of data packets in this scenario.
Step 5: From here, we obtain our result stating whether the given packet is malicious or not.

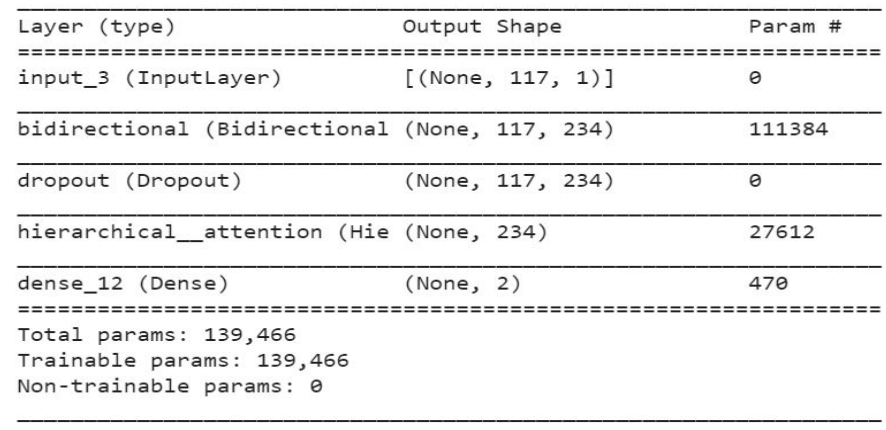

Fig 7. Structure of the Bi-DLDA module

4 Experimental Results

The experiments on S-SCAE + Bi-DLDA were performed by Tensor flow and Keras are used to implement the intelligent intrusion detection system in the cloud environment, which is provided by IBM Watson studio. 80 and 20 percent of the NSL-KDD dataset used to train and test the network respectively. Thus, 18,036 packets used for training the model and 4,509 packets used for testing. The dataset with 43 features used in the experiments. Two characteristics ignored throughout the classification process because they have no bearing on the effectiveness of DDoS attack detection.
In the validation process, we compare the proposed algorithm S-SCAE + Bi-DLDA with other models such as SAE + SVM and SCAE + SVM. To evaluate each approach, we use the 'Precision,' 'Recall,' 'F Measure,' and 'Accuracy' measures, all of which have been used widely in the literature.
The model accuracy and loss score obtained by our system after training with 18,036 packets is 98.32% and 0.0561 respectively while the model accuracy and loss obtained after testing with 4,509 packets is 98.58% and 0.0540 respectively. The model accuracies at each epoch for the training and testing phases are depicted in Figure 8. The model loss scores at each epoch for the training and testing phases are depicted in Figure 9.

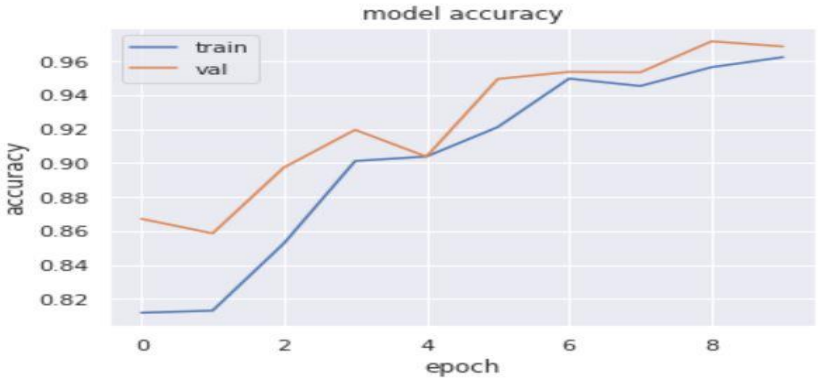

Fig. 8. Graphical Representation of the model accuracy

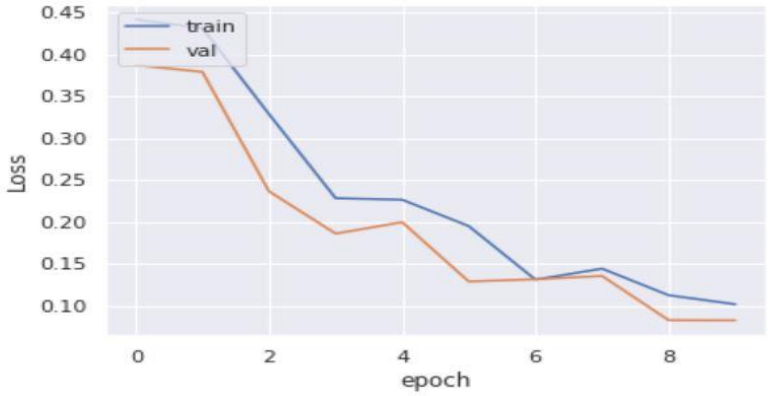

Fig. 9. Graphical Representation of the loss

4.1 Evaluation Metrics

For the comparison of the above-proposed model with the previously proposed models, to evaluate each approach, we widely used measurements such as 'Precision,' 'Recall,' 'F Measure' and 'Accuracy' throughout the literature.
The four basic properties of the confusion matrix, which depict the actual and predicted classes, are used to generate all of these evaluation measures. These confusion matrix elements are:
True Negative (TN): number of correctly predicted instances as normal packets.
False Negative (FN): number of incorrectly predicted instances as normal packets.
False Positive (FP): number of wrongly predicted instances as malicious.
True Positive (TP): number of correctly predicted instances as malicious.

The comparison measures are defined as:
Precision: The proportions of samples that have been correctly categorised to the total number of samples that have been expected to be positive are as in eq. (1).

$$\text{Precision} = \frac{TP}{FP + TP} \quad (1)$$

Recall: The proportion of potential positives to the total number of True Positives and False Negatives is given in eq. (2).

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

F-measure: F1-score, denotes the harmonic mean of precision and recall as follows in eq. (3).

$$\text{F-Measure} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (3)$$

Accuracy: The accuracy of classification is measured as a fraction of correctly identified samples divided by the total number of samples as follows in eqn. (4).

$$\text{Accuracy} = \frac{TP + TN}{TP+TN+FP+FN} \quad (4)$$

The accuracy rate is the proportion of records that are correctly identified. The proposed S-SCAE along with a Bi-DLDA approach provides dimensionality reduction and no over-fitting phenomenon. The result shows that the proposed methodology improves the accuracy of 98.58%. The precision rate indicates the percentage of accurately identified records among all attack records found. The precision rate for non-malicious and malicious packets is 99% and 98% respectively. The recall rate is the proportion of records accurately identified as being related to the original type of attack. The recall rate for non-malicious and malicious packets is 98% and 99% respectively. The harmonic mean of the precision and recall rate is represented by the f-measure. The f-measure for non-malicious and malicious packets is 98% and 99% respectively. Using a confusion matrix M, the above metrics can be obtained. The confusion matrix's leading diagonal members represent the number of records properly predicted. Finally, the ROC - curve displays the true positive rate (TP) and false positive rate (FP) to show classification performance. The higher the TP and the smaller the FP, the greater the area under the ROC curve. In Fig. 10 The confusion matrix for our proposed system is depicted below:

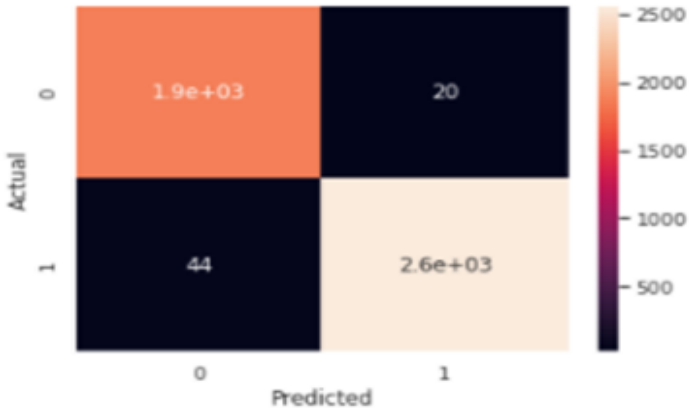

Fig. 10. Confusion Matrix of Proposed System

The true positive rate for this system was found to be 97.7%. The false-positive rate for this system is determined to be 0.7%. The false-negative rate is 2.3%. Lastly, the true negative rate is found to be 99.2%.

Table 1. Comparison of Accuracy, Precision, Recall and F-score in %

	Accuracy	Precision	Recall	F-Score
SAE + SVM [18]	85.09	85.78	85.09	84.83
SCAE+SVM [18]	88.73	89.87	88.73	88.04
S-SCAE+BI-DLDA	**98.58**	**99**	**98**	**99**

The 'Accuracy,' 'Precision,' 'Recall,' and 'F Measure' for each previously suggested approach in the context of intrusion detection such as the SAE+SVM model and the SCAE+SVM model [18] is compared with the S-SCAE + Bi-DLDA model and the various results in % are reported in Tables 1.

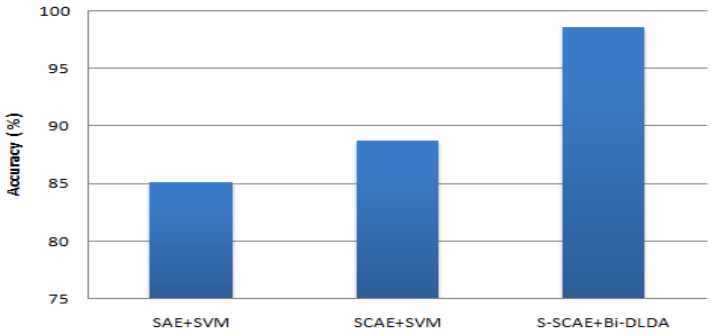

Fig. 11. Accuracy Comparison

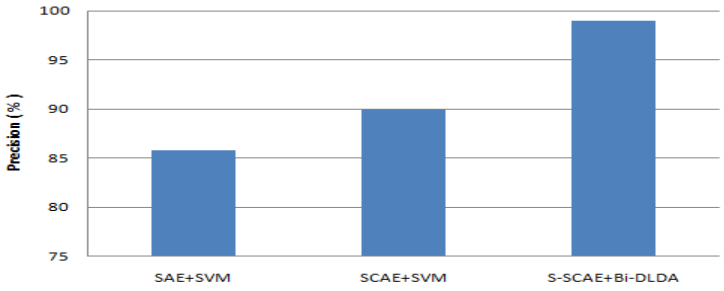

Fig. 12. Precision Comparison

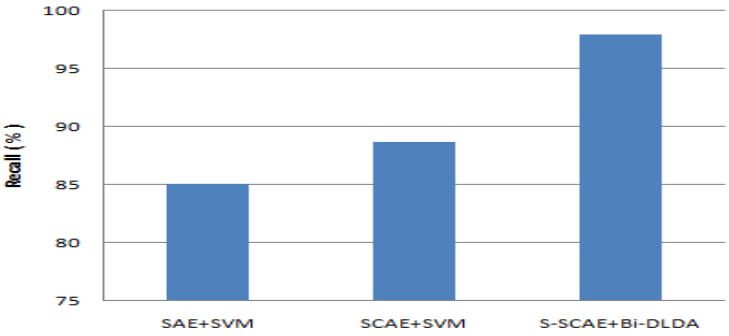

Fig. 13. Recall Comparison

Fig. 11, 12, 13, and 14 show the graphical representation of the comparisons of the 'Accuracy' 'Precision', 'Recall' and 'F-score' for each previously proposed method such as the SAE+SVM model and the SCAE+SVM model with the S-SCAE + Bi-DLDA model respectively.

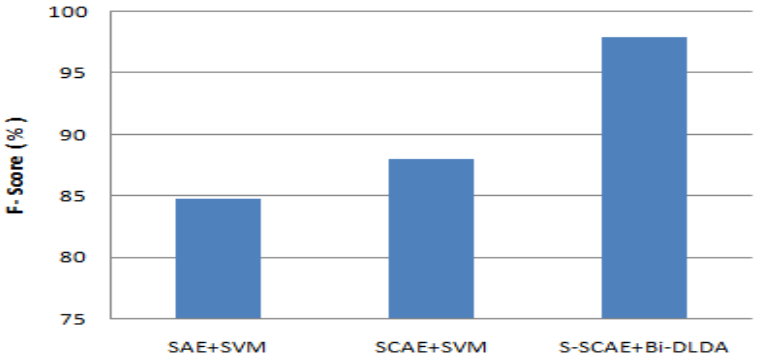

Fig. 14. F-score Comparison

5 Conclusion and Future work

Major losses in the cloud computing environment have occurred due to security concerns. In addition, these concerns have led to the loss of confidence in users on cloud computing. An intrusion detection system is a cost-effective way to safeguard cloud computing environment against unwanted intrusions. We have proposed an intelligent intrusion detection system that uses a sparse auto encoder – stacked contractive auto-encoder (S-SCAE) for feature extraction and a Bi-Directional LSTM with Dropout and Attention Layer (Bi-DLDA) for classification of the packets. By the NSL-KDD dataset, we have proposed highly efficient IDS with accuracy higher than 98%. In the future, a cloud computing context, powerful deep learning techniques can be used to detect assaults.

References

1. Al Qatf, M., Lasheng, Y., Al Habib, M., Al Sabahi. K..: Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection. IEEE Access, vol.6, pp.52843-52856 (2018).

2. Aneetha, A., S., Bose, S.: The combined approach for anomaly detection using neural networks and clustering techniques. Computer Science & Engineering: An International Journal (CSEIJ), vol.2, no.4, pp.37-46 (2012).

3. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity , vol. 2, no. 20, (2019).

4. Dhanapal, A., Nithyanandam, P.: The slow HTTP distributed denial of service attack detection in cloud. Scalable Computing: Practice and Experience, vol. 20,no. 2, pp. 285–297 (2019).

5. Le, T., Kim, J., Kim, H.: An effective intrusion detection classifier using long short-term memory with gradient descent optimization. 2017 International Conference on Platform Technology and Service (PlatCon), pp. 1-6 (2017).

6. Wang, W., Du, X., Wang, N.: Building a Cloud IDS Using an Efficient Feature Selection Method and SVM. IEEE Access, vol.7, pp. 1345-1354 (2019).

7. McDermott, C, D., Majdani, F., Petrovski, A, V.: Botnet detection in the internet of things using deep learning approaches. International joint conference on neural networks (IJCNN) , pp. 1-8 (2018).

8. Md Tarique Prwez., Kakali Chatterjee.: A framework for Network Intrusion Detection in Cloud. IEEE 6th International Conference on Advanced Computing (2016).

9. Osanaiye, O, A.: Short Paper: IP spoofing detection for preventing DDoS attack in Cloud Computing. 2015 18th International Conference on Intelligence in Next Generation Networks, pp. 139-141(2015).

10. Qureshi, A, S., Khan, A, K.., Shamim, N., Durad, M, H.: Intrusion detection using deep sparse auto-encoder and self-taught learning. Neural Computing and Applications, vol.32, pp.3135– 3147 (2020).

11. Shukla, R, M., Sengupta S.: Scalable and Robust Outlier Detector using Hierarchical Clustering and Long Short-Term Memory (LSTM) Neural Network for the Internet of Things. Internet of Things, vol.9, pp. 1-18 (2020).

12. Su, T., Zhu, J., Wang, S. Li, Y.: BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," IEEE Access, vol. 8, pp. 29575-29585 (2020).

13. Sumit Badotra., Surya Narayan Panda.: SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking. Cluster Computing, vol.24, pp.501–513 (2021).

14. Roy, B., Cheung, H.: A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1-6, (2018).

15. Velliangiri, S., Karthikeyan, P., Vinoth Kumar, V.: Detection of distributed denial of service attack in cloud computing using the optimization-based deep networks. Journal of Experimental & Theoretical Artificial Intelligence, vol.33, pp.405-424 (2020).

16. Xu, M., Li, X., Ma, J, f., Zhong, C., Yang, W.: Detection of Multi-Stage Attacks Based on Multi-Layer Long and Short-Term Memory Network. ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1-4 (2019).

17. Yan, Y., Qi, L., Wang, J., Lin, Y., Chen. L.: A Network Intrusion Detection Method Based on Stacked Autoencoder and LSTM. ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1-6 (2020).

18. Wang, W., Du, X., Shan, D., Qin, R.: Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine. IEEE Transactions on Cloud Computing, pp.1-14 (2020).

19. Lino Abraham Varghese., Bose, S.: Efficient Data Storage Model to Overcome the Storage Problems in Industries," Dynamic Systems and Applications, vol. 30, no. 6 , pp.994-1002, (2021).

20. Krishnaveni, S., Sivamohan, S., Sridhar, S, S, Prabakaran, S.: Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing. Cluster Computing, vol.24, no.3, pp.1-19. (2021).