iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 Issue: IX Month of publication: September 2022

DOI: https://doi.org/10.22214/ijraset.2022.46739

A Cloud Security Attack Detection Using Proposed BGRU and Bi-LSTM Models

Mr. Ashish Soni[1], Mr. Rajneesh Pachouri[2], Mr. Anurag Jain[3]
[1, 2, 3] *Department of Computer Science and Engineering AIST, Sagar (M.P)*

Abstract: In recent years, businesses and organizations have placed significant emphasis on network security. Through the network, the attackers were able to exploit flaws in the network setup. Numerous secure solutions, the most of which are based on attack signatures, have been developed to safeguard communication in such an environment. These systems are often ineffective in detecting every type of attack. Recently, the approach of deep learning has been presented. In this research, we propose a hybrid model that combines the BGRU and Bi-LSTM models for detecting and preventing cloud-based attacks. We have used one hot encoding for data preprocessing and select k best features for feature selection. There have been experiments conducted on the UNSW-NB15 dataset. These experiments have been conducted using the Python simulation tool and the Jupyter notebook environment. The findings indicate that it has a higher rate of network attack & intrusion detection and classification accuracy of above 95 percent.

Keywords: Cloud Computing, Cloud Security, Attack Detection, Deep Learning, Bidirectional-GRU (BGRU), Bidirectional–LSTM (Bi-LSTM).

I. INTRODUCTION

Cloud computing[1] has emerged as a leading and rapidly developing technology in the field of information technology research throughout the course of the last several decades. This technology offers the benefit of on-demand service in addition to the availability of an abundant supply of resources. Even though more people are using cloud computing services, maintaining one's privacy and safety in a cloud setting may be a big challenge [2]. In essence, the cloud environment necessitates the implementation of several preventative actions and safeguards, such as the guaranteeing of data privacy and protection, the availability of data, the location privacy of data, and the transfer of data in a secure manner [3]. Despite this reality, the cloud environment is progressively being negatively impacted by both external and internal threats. As a result, the cloud environment has begun to place a greater emphasis on the use of intrusion detection systems in order to safeguard the data that is stored and processed[4].

Recently, researchers in the fields of network security [5] and cloud security [6] have shown an increasing interest in implementing strategies that use machine learning. In the machine learning-based security system, aberrant and healthy behaviors are classified based on the labelled traces from the training models. These labelled traces come from the models that are used for training. The security of the cloud environment is maintained using machine learning and deep learning-based intrusion detection models [7][8], which work by extracting a variety of feature sets. The security model that is based on machine learning and deep learning may discover vulnerabilities in a substantial way while also having a decreased level of complexity and costing an acceptable amount. In recent years, various studies have been conducted on the creation of machine learning-based intrusion detection models. These models are used to identify potential security breaches. Even if the machine learning approaches defend against the several forms of assaults that may occur in the vast cloud environment, it is not effective against unknown attacks [9]. As a result of this, a number of researchers who are currently active have provided both supervised and unsupervised learning-based security solutions [10][11] in order to safeguard the data from the vulnerabilities. Because the data that is saved and performed in the cloud environment is crucial to cloud users who have malicious intentions, it is important to provide security in the cloud environment with the use of machine learning algorithms [12]. During the process of establishing the cloud security model, it is essential to have a thorough understanding of the security precautions that the cloud service provider is required to take.

Deep learning-based response to these threats is proposed in this paper for a secure CC environment. The following is a summary of the rest of the paper. Deep learning as well as computational modeling for cloud - based security vulnerability scanning are discussed in the following section.... Finally, the research methodology was discussed in section three. For cloud - based security intrusion detection, we announce the findings of implementing this algorithm. Finally, future research is discussed to bring the paper to a close.

II. LITERATURE SURVEY

In deep learning & cloud computing, experts at home and abroad have amassed a wealth of research findings on the detection of intrusions in intelligent power terminals using deep learning and cloud computing. This study [13] presents and compares To a hybrid Piled Contractive Auto - encoders (SCAE) + Svm Classifier IDS model. Deep learning algorithms, such as basic computer vision, may be used to detect and classify attacks simultaneously. Take a look at the detection methods employed by a few current IDSs as well. Using the KDD Cup 99 and NSL-KDD Intrusion Detection datasets, our detection method outperforms existing methods. propose a Deep Learning (DL) model to reduce the FPR which lowers a scalable solution by deploying the model on cloud to increase the responsiveness of the NIDS during high loads, hence increasing the availability. The model which is running on docker containers on the cloud instance can be accessed using REST APIs as it is deployed as microservice. The experimental results showed that Deep Neural Networks (DNN) with have hidden layers achieved the best accuracy of 95.02% and a least accuracy of 88.75% by Long Short-Term Memory (LSTM) with two layers. In traditional ML algorithms, Random Forest approach have achieved 86% accuracy [14]In order to improve NIDS responsiveness under peak loads and thus increase its availability, suggest using deep learning (DL) to lower the FPR. DL is a scalable solution because it can be implemented in the cloud. Since the model is deployed as just a microservice in docker containers just on cloud instance, REST APIs can be used to access it. Ninety-two percent of the time, deep neural networks (DNNs) with four hidden layers were most accurate, while LSTMs with two hidden layers had the worst accuracy of 88.75 percent, according to the results of the tests. The Random Forest technique has obtained an accuracy rate of 86% in traditional ML algorithms. In this work, [15]propose a deep learning strategy for IDSs that is more accurate than existing methods in detecting U2R and R2L assaults. Only basic methods of deep learning, like convolutional neural networks, oversampling, undersampling, and data augmentation, were employed in the suggested approach. In addition, no domain expertise is required for the suggested technique, since the proposed pipeline does not include feature engineering. In this study, they also give the experimental results of a performance evaluation of the suggested technique using the KDD Cup 99 Dataset. The experimental findings demonstrate that the suggested technique can identify U2R or R2L assaults with more precision than earlier research.

This study's [16] objective is to examine the feasibility of utilising machine learning approaches for Application-level detection of SQL injections. Classifiers that have been trained on a variety of harmful and non-harmful payloads will be put to the test. They receive a payload as input and determine whether or not it includes harmful code. The findings demonstrate that these algorithms can discriminate between benign and malicious payloads with a detection rate exceeding 98%. An evaluation of several ML models for detecting SQL injection attacks is also included. Numerous investigations on ML-based Machine Learning IDS use KDD or improved KDD models. This article uses the dataset CSE-CICIDS-2018, which comprises the most recent fundamental system threats.[17] use of an IDS with Machine Learning Based (Random Forest) for CSE-CIC-IDS-2018 results in an excellent score of 99% Accuracy. [18]This project seeks to apply powerful feature representations and excellent detection performance can be achieved using deep learning techniques. We present an effective unsupervised feature extraction approach based on the stacked contractive autoencoder (SCAE). As a result of the SCAE method, better and much more resilient reduced characteristics can be learned from network data. A new cloud IDS is constructed using SCAE and Classifier classification techniques. Using a combination of shallow and machine learning approaches, the SCAE+SVM method drastically reduces the computational burden. Using KDD Cup 99 and NSL-KDD, two widely used datasets for evaluating intrusion detection systems, experiments show that the SCAE+SVM method proposed here outperforms three other leading methods.

In this work, [19]propose an adaptive cloud IDS architecture based on deep reinforcement learning that tackles the aforementioned constraints and provides detection and classification of new and more sophisticated attacks with pinpoint accuracy and fine-grained precision. In comparison to the most advanced IDSs, extensive research with benchmark UNSW-NB15 dataset shows greater accuracy and a lower FPR. In order to begin this research project, the dataset must first be gathered, and since it is openly accessible to the general public, the UNSW-NB15 dataset has been obtained. Prior to overcoming the difficulty of pre-processing process interpretation, there are issues with the dataset itself. At the level of unprocessed data, a technique called preprocessing may resolve potential issues such as missing values and redundant or inconsistent data. In the pre-processing stage, a raw dataset is adapted to the process's requirements. The removal of irrelevant features. It eliminates unnecessary features. The second issue with real-world datasets, particularly categorical data, is assigning labels with numeric labels or classes; hence, one-hot encoding is performed during data preprocessing. An additional challenge with selecting the optimal features from pre-processed data.

Then, perform feature selection using the k-best features that have been identified. The BGRU (Bidirectional Gated Recurrent Unit) and Bi-LSTM (Bidirectional Long Short-Term Memory) deep learning models are then implemented. And determine the accuracy and loss values for this investigation. Deep learning was implemented in this project using Python technology and Jupyter Notebook as a software tool for experimentation. The complete methodological procedure is shown in figure 1.

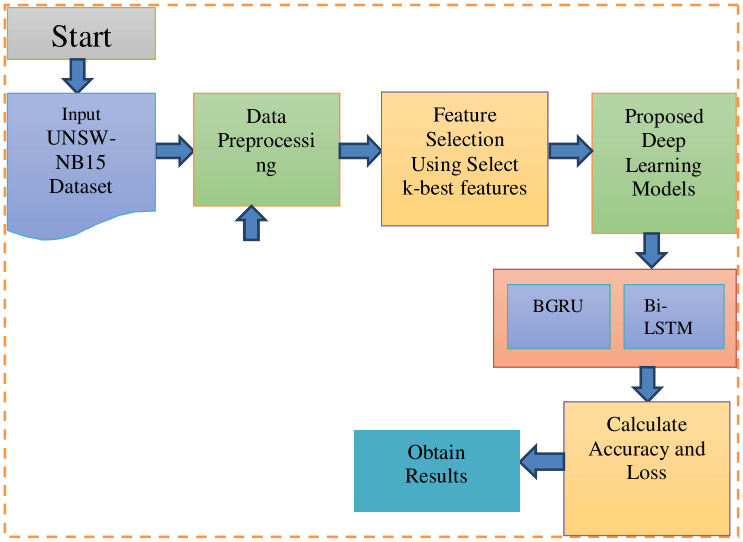

Start
Input UNSW-NB15 Dataset
Data Preprocessing
Feature Selection Using Select k-best features
Proposed Deep Learning Models
BGRU
Bi-LSTM
Calculate Accuracy and Loss
Obtain Results

Fig. 1.Block Architecture of the Proposed Methodology

Here some steps of the proposed methodology are described in detail in the below subsections one by one:

A. *Data Preprocessing*

Pre-processing is a step-in machine learning and deep learning that prepares a dataset. This methodology addresses the handling of missing data & class imbalances. Dealing with missing values is the first step at this time. The phrase "missing values" is used when the expected values of the data are not recorded. As indicated earlier, a number of columns in the dataset are devoid of values. There are many pre-processing processes and numerical one-hot encoding, which are described in further depth in the following section.

- *One hot Encoding:* When categorical variables are encoded, machine learning (ML) methods can use them to make better predictions.

B. *Feature Selection*

A feature is an attribute that has an effect on an issue or is relevant for the problem, and feature selection is the process of selecting the most significant features for a model. Each phase of machine learning is dependent on feature engineering, which consists mostly of two processes: Feature Selection & Feature Extraction. Despite the fact that feature selection and extraction methods may have the same goal, they are quite distinct from one another. Feature selection involves picking a subset of the original collection of features, while feature extraction generates new features. Feature selection is a method for minimising the number of model input variables by utilising only relevant data in order to prevent model overfitting.

- *Select K Best Features:* SelectKBest is a class provided by the Scikit-learn API for extracting the best features from a given dataset. The SelectKBest technique chooses features with the k highest score in mind. By modifying the 'score func' argument, the approach may be used to classification or regression data. Selecting the right features is a crucial step when preparing a big dataset for training. It enables us to discard less significant data and save training time.

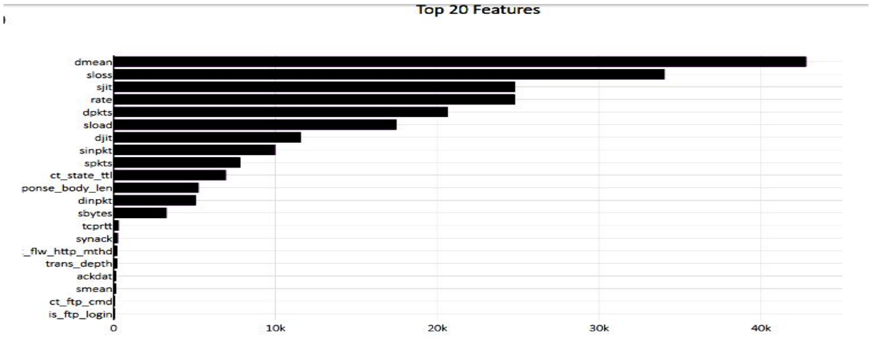


Fig. 2.Graph for Selected Features

C. Proposed Deep Learning Models

There are two deep learning models is given below which is used in this research.

1) *BidirectionalGRU (BiGRU) Model:* A Bidirectional GRU, or BiGRU, is a model for sequence processing that is comprised of two GRUs. One takes the input in the forward direction, while the other takes it in the reverse direction. It is a two-way recurrent neural network with just input and forget gates. As seen in Fig. 2, gated recurrent units (GRUs) are regarded a gating mechanism in artificial recurrent neural networks, comparable to that of LSTM [20]. However, GRUs have been found to perform better with smaller to medium-sized datasets.

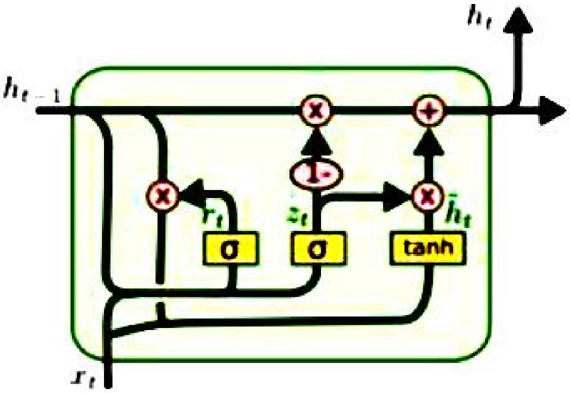


Fig. 3.Fig3. Gated Recurrent Unit (GRU) Model

The word embedding is generated using WordNet, where bi-directional GRU is performed to combine the tweet's words into a single word vector. To ensure that all tweets have the same length, necessary placeholders are provided.4

2) *BidirectionalLSTM (Bi-LSTM) Model:* Bidirectional long-short term memory, or Bi-LSTM, is a term used to describe the development of any neural network capable of storing sequence information in both directions (forwards and backwards) (past to future). Bidirectional input distinguishes a bi-LSTM from the a standard LSTM because it travels in both directions. Input can go forward or backward with a standard LSTM algorithm. The input can flow in both directions in bi-directional communication so that we can preserve both future and past data.[21].

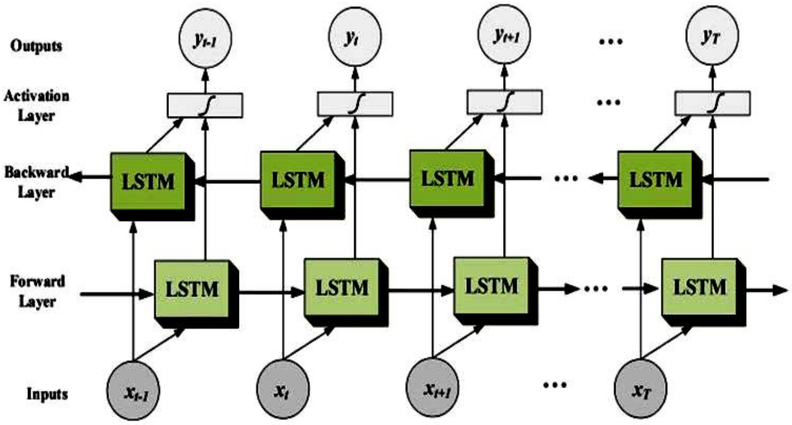


Fig. 4.Bi-LSTM Model

The graphic depicts the information flow between backward and forward layers. Most of the time, the BI-LSTM is used to perform actions from one sequence to another. All of these, as well as text classification and voice recognition, could be aided by a network of this type.

III.PROPOSED ALGORITHM

1) *Input:* UNSW-NB15 Dataset
2) *Output:* Attack Detection

Procedure:
Step 1: Initialize process
Step 2: Firstly, dataset gathering
Step 3: Import Python Libraries
Step 4: Create Pandas data frame
Step 5: Perform Exploratory Data Analysis (EDA) and Statistical Analysis.
Step 6: Perform data preprocessing to preprocess the data.

- Drop irrelevant features
- Apply clamping
- Apply log function to nearly all numeric, since they are all mostly skewed to the right
- Perform one-hot encoding
- Reduce the labels in categorical features

Step 7: Perform feature selection using select k best features.
Step 8: Implement proposed deep learning models.

- BGRU
- Bi-LSTM

Step 9: Evaluating the model performance with accuracy and loss.
Step 10: Final Output to detect the attack.
Step 11: Stop

IV. RESULTS AND DISCUSSION

We need a particular environment in order to deploy and test the suggested deep learning models. For the purposes of this project, the Python programming language is used either as simulation software or, alternatively, the Jupyter notebook is utilized as a simulation environment. For the purpose of determining success, the approach calculates accuracy & loss parameters. In this part, we went into depth about the research that had been done and the tactics that had been developed in order to create a detection mechanism that is accurate, real-time, and has a quick processing capability. Each classifier has access to a completely new database. In addition to that, analyse the findings of the experiment.

A. Dataset Description

To conduct this study, I utilised the UNSW-NB15 Dataset. It is possible to get You can download the UNSW-NB15 pcap files and reports here. IXIA PerfectStorm at the Cyber Range Lab at the University of New South Wales Canberra generated the raw data packets for the UNSW-NB 15 dataset to generate a blend of current real normal activities as well as synthetic contemporary attack behaviour." By using the tcpdump tool, 100 GB of traffic was gathered (e.g., Pcap files). As a whole, this dataset contains nine different types of attacks, including fuzzers and analysis as well as backdoors, denial-of-service attacks as well as exploits. More than forty-nine features with class label can be generated using the Argus as well as Bro-IDS software packages. UNSW-NB15 features.csv is a file containing information on these features.

- Two 2 million but also 540,044 records can be found in the following four CSV files: UNSW-NB15 1.csv, UNSW-NB15 2.csv, and UNSW-NB15 3.csv.
- UNSW-NB15 GT.csv is the file again for ground truth table, while UNSW-NB15 LIST EVENTS.csv is the file for the event list.
- NB15 training-set.csv and NB15 testing-set.csv were created from this dataset and used for training and testing purposes. There are 175,341 records in the training set and 82,331 records in the testing set, which includes attack and normal records.

B. Performance Evaluation Metrics

After implementing deep learning models, we require assessment methods to determine how successfully they completed their assigned tasks. These measurements are known as performance assessment metrics. Studies have established a substantial variety of metrics, each of which evaluates a different element of algorithm performance. These measurements include accuracy & loss.

1) *Accuracy:* Accuracy is less complicated. The accuracy of our model's predictions is determined by comparing the model's predictions to the actual values, expressed as a percentage.

$$Accuracy = \frac{No\ of\ correct\ predictions}{Total\ no.of\ predictions} \quad (1)$$

2) *Loss:* Loss is defined as the difference between your model's anticipated value and the true value. It determines how well (or poorly) our model is doing. If the error rate is high, the loss will also be large, indicating that the model is not performing well. The lower it is, the better our model performs.

C. Results

In this section, we will talk about the results that were acquired using the proposed BGRU model and the Bi-LSTM model that was built by utilising Python programming.

1) *Exploratory Data Analysis (EDA):* Analyzing the data with the use of visual tools is known as Exploratory Data Analysis (EDA). A statistical summary and graphical depiction are used to detect trends, patterns, or to test assumptions. We just utilised one dataset to keep things simple. This was done using the UNSW-NB15 dataset. In addition to pcap and BRO files, there are also argus files, csv files, and reports included. As a visual aid, here are several EDA graphs as shown:

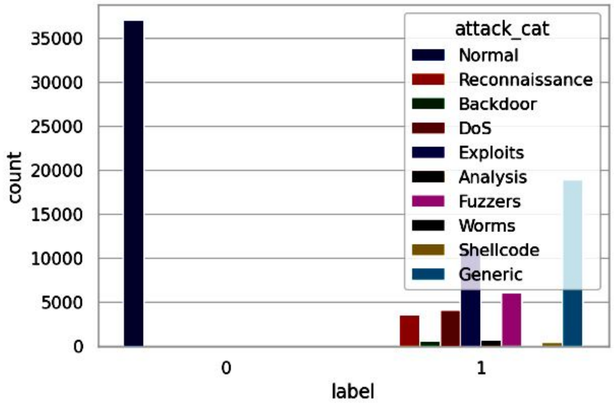
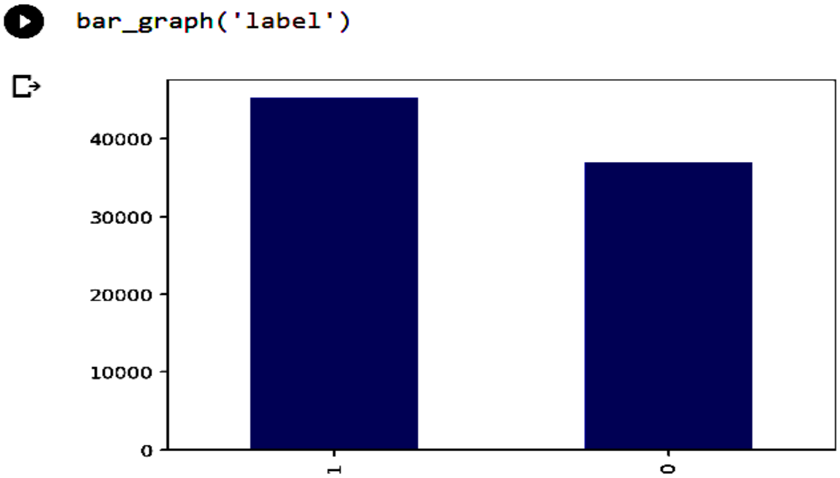


Fig. 5.Count Plot of labels and Attack Category

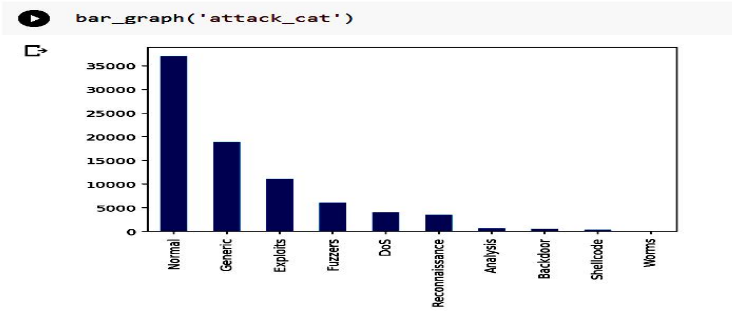
Figure 5 shows a count plot of labels and attack category, where x-axis shows the labels and y-axis shows the number of counts. There are multiple attack categories i.e., This includes everything from normal to backdoors to DOS attacks to exploits to analysis tools to shellcode to fuzzers.



```
bar_graph('label')
```

Fig. 6.Label Graph

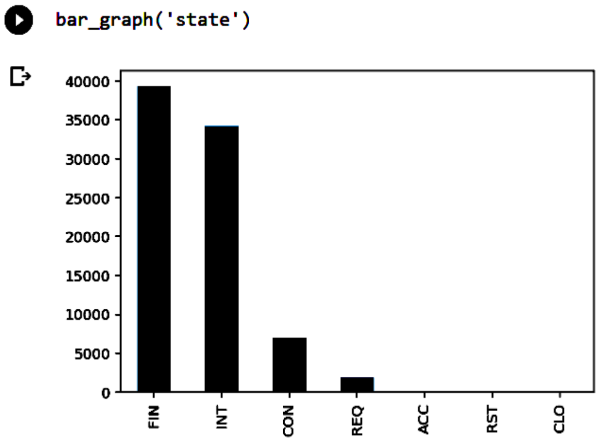
Figure 6 shows a label graph where shows the total number of labels (0 and 1). From this figure see that, above 40000 counts for label 1, and above 30000 counts for label 0is shown in this figure.



```
bar_graph('attack_cat')
```

Fig. 7.Count Plot of Attack Category

There are various attack types represented on the x-axis and the y-axis in the figure above, which depicts a count plot. There are nine distinct attack types included in this dataset, with "normal" denoting the absence of any attacks. As a result, there are far more non-attacks than attacks in the data. Figure 8 depicts a state count plot, in contrast.



```
bar_graph('state')
```

Fig. 8.Count Plot of State

2) Visualization of Experimental Results

This section visualizes the experimental results with their description.

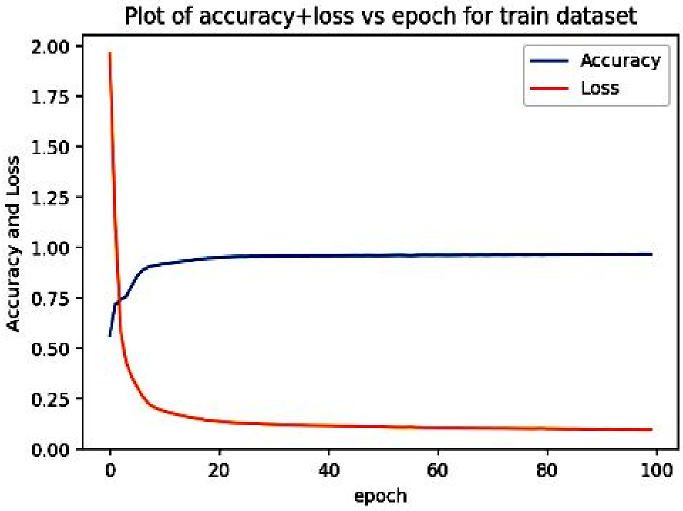


Fig. 9.Accuracy and Loss Graph for BGRU Model

The following figure shows an accuracy and loss line graph for proposed Bidirectional-GRU (BGRU) model with successive in which the blue line depicts accuracy and the orange line depicts loss are plotted against one another. There are epochs on the x-axis, and accuracy and loss are shown on the y-axis in this diagram.As we can see in this figure, accuracy value is low at initial epoch 0, then after few variation, accuracy value increased up to few labels then it constant up to epoch 100, Whereas loss value is very high at starting point then suddenly it decreased at 0.25 value approx. then it constantly decreased up to epoch 100.

The below figure shows the accuracy, and loss line graph for proposed Bidirectional LSTM (Bi-LSTM) model with successive epochs, where blue line shows the accuracy, and orange losses can be clearly seen in the line. On the x-axis, the number of iterations is represented, while the y-axis shows the accuracy. and loss values. As seen in the graph, the accuracy value is low at initial epoch0; however, after a few variations, the accuracy value increased up to a few labels and remained constant up to epoch 100. On the other hand, the loss value is extremely high at epoch0; however, after a sudden decrease to approximately0.2, it remained constant up to epoch 100.

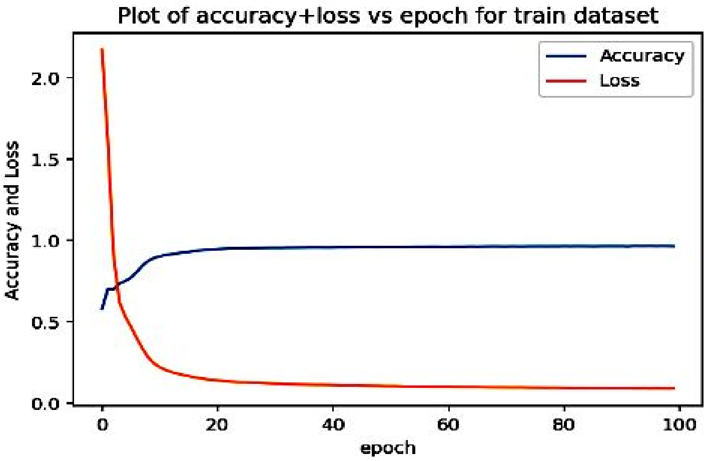


Fig. 10. Accuracy and Loss Graph for Bi-LSTM Model

Table 1 demonstrate the values of accuracy and loss for proposed BGRU and B-LSTM models. The proposed BGRU model has the greatest accuracyvalue on the UNSW-NB15 Dataset than the Bi-LSTM model. Bi-LSTM is also showing the high accuracy value but its less than BGRU model. The accuracy and loss value for proposed BGRU model is 96.23% and 9.68% whereas the accuracy and loss value for proposed Bi-LSTM model is 96.16% and 9.58%, respectively.

TABLE I. Accuracy and Loss Values for Proposed BGRU and Bi-LSTM Model

Deep Learning Model(in%)	*BGRU*	*Bi-LSTM*
Accuracy	96.23	96.16
Loss	9.68	9.58

V. CONCLUSION AND FUTURE WORK

The goal of this study was to present a deep learning-based model for detecting cloud security attacks. Deep learning models able to detect attacks are the primary goal of this research. The UNSW-NB15 dataset was used to demonstrate the model's performance, which is meant to mimic actual network communication techniques and hypothetical or actual attack operations. Various learning percentages as well as hidden layers were used to train the model during this phase. In the testing phase, the trained model to classify the dataset.In this investigation, the gathered dataset was preprocessed, and feature selection was carried out with the use of select K-best features. And put into practice a deep learning model consisting of BGRU and Bi-LSTM. Python is the programming language that is used to carry out this experiment. The findings of the experiments indicate that the proposed BGRU model and the Bi-LSTM model both have an accuracy of 96.23 percent and 96.16 percent, respectively. Since the size of the dataset is not taken into consideration in this work to check the effect that the suggested model will have on performance, future research will be conducted in which we will investigate the influence that the size of the dataset has on the performance of models.

REFERENCES

[1] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," ArXiv, vol. abs/1707.0, 2018.

[2] M. Al Morsy, J. Grundy, and I. Müller, "An Analysis of the Cloud Computing Security Problem Mohamed," Apsec, 2010.

[3] S. Singh, Y. S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," J. Netw. Comput. Appl., 2016, doi: 10.1016/j.jnca.2016.09.002.

[4] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," Journal of Network and Computer Applications. 2017, doi: 10.1016/j.jnca.2016.10.015.

[5] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," Applied Sciences (Switzerland). 2019, doi: 10.3390/app9204396.

[6] R. S. Siva Kumar, A. Wicker, and M. Swann, "Practical machine learning for cloud intrusion detection challenges and the way forward," 2017, doi: 10.1145/3128572.3140445.

[7] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2018, doi: 10.1109/CloudTech.2017.8284731.

[8] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2836950.

[9] M. T. Khorshed, A. B. M. S. Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," 2012, doi: 10.1016/j.future.2012.01.006.

[10] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques," 2019, doi: 10.1109/AICAI.2019.8701238.

[11] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," Computer Science Review. 2019, doi: 10.1016/j.cosrev.2019.100199.

[12] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the Science of Security and Privacy in Machine Learning," Nov. 2016.

[13] P. Jisna, T. Jarin, and P. N. Praveen, "Advanced Intrusion Detection Using Deep Learning-LSTM Network on Cloud Environment," 2021, doi: 10.1109/ICMSS53060.2021.9673607.

[14] C. Archana, H. P. Chaitra, M. Khushi, T. Pradhiksha Nandini, E. Sivaraman, and P. Honnavalli, "Cloud-based network Intrusion detection system using deep learning," 2021, doi: 10.1145/3485557.3485562.

[15] A. Takeda and D. Nagasawa, "A Simple Deep Learning Approach for Intrusion Detection System," 2021, doi: 10.23919/ICMU50196.2021.9638850.

[16] D. Tripathy, R. Gohil, and T. Halabi, "Detecting SQL Injection Attacks in Cloud SaaS using Machine Learning," 2020, doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00035.

[17] M. P. Bharati and S. Tamane, "NIDS-Network Intrusion Detection System Based on Deep and Machine Learning Frameworks with CICIDS2018 using Cloud Computing," 2020, doi: 10.1109/ICSIDEMPC49020.2020.9299584.

[18] W. Wang, X. Du, D. Shan, R. Qin, and N. Wang, "Cloud Intrusion Detection Method Based on Stacked Contractive Auto-Encoder and Support Vector Machine," IEEE Trans. Cloud Comput., 2020, doi: 10.1109/tcc.2020.3001017.

[19] K. Sethi, R. Kumar, N. Prajapati, and P. Bera, "Deep Reinforcement Learning based Intrusion Detection System for Cloud Infrastructure," 2020, doi: 10.1109/COMSNETS48256.2020.9027452.

[20] A. Bhuvaneswari, J. T. Jones Thomas, and P. Kesavan, "Embedded Bi-directional GRU and LSTMLearning Models to Predict Disasterson Twitter Data," 2019, doi: 10.1016/j.procs.2020.01.020

[21] Yugesh Verma, "Complete Guide To Bidirectional LSTM (With Python Codes)," DEVELOPERS CORNER, 2021. https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=Bidirectional long-short term memory(bi-lstm) is,different from the regular LSTM