Efficacy of CNN-Bidirectional LSTM Hybrid Model for Network-Based Anomaly Detection

Toya Acharya[1], Annamalai Annamalai[2], Mohamed F Chouikha[3]
Electrical and Computer Engineering
tacharya@pvamu.edu[1], aaannamalai@pvamu.edu[2], mfchouikha@pvamu.edu[3]
Prairie View A&M University, Prairie View, Texas

Abstract- With the development of the web and the internet, computer networks have become an important tool to transfer information digitally, that increases the system's threats and vulnerability. Cyber attackers can use the internet and tools to compromise the triad of the CIA (confidentiality, integrity, and confidentiality). Network anomaly detection is challenging while detecting anomalous behavior in a network due to the large-scale data, imbalance nature of attacks class, and huge numbers of features in the dataset. Traditional Machine learning methods are not very efficient in solving those problems. Deep learning has proven to be more efficient in detecting network-based anomalies. A Recurrent Neural Network (RNN) model is designed to recognize the sequential data characteristics to predict. We proposed a convolutional neural network with bidirectional long-short memory (CNN Bi-LSTM) model to analyze the hyperparameters, including optimizers (Nadam, Adam, RMSprop, Adamax, SGD, Adagrad, Ftrl), epochs, batch size, learning rate, and neural network model architecture of CNN-BLSTM algorithms. Those analyzed hyperparameters provide the highest anomaly detection accuracy of 98.27% and 99.87% on the NSL-KDD and UNSW-NB15, respectively. Performance assessment regarding the accuracy and F1-score revealed that the proposed CNN Bi-LSTM anomaly detection model exhibited better performance than the other existing anomaly detection methods.

Keywords— Network Intrusion Detection System, Machine Learning, Deep Learning, LSTM, CNN, Bi-LSTM, NSL-KDD, UNSW-NB15

I. Introduction

As technology develops rapidly, the method of transmission of information from source to destination has evolved through the wired, wireless, or guided network. The development of network technology plays a vital role in people's daily activities. Any system is considered secure if the three computer security principles of confidentiality, integrity, and availability (CIA) are properly met. Hence information security is securing information from an unauthorized agent, preventing access, use, disclosure, modification, recording, or data destruction.

A firewall and antivirus software cannot completely protect the traditional network. The antivirus and firewall detect those activities already defined as anomalous and set the rule to block those activities by the expert. Outliers and anomalies are sometimes used interchangeably in anomaly detection. Anomaly detection has abundant applications, including business, network intrusion detection, health monitoring systems, credit card fraud detection, and fault detection in critical information systems. Anomaly detection is important in cyber security for solid protection against cyber adversaries. There must be secure network resources against cyber threats to protect the system.

Anomalies are classified as point, contextual, and collective according to the output from the detection method used [1]. Point anomaly occurs when a certain behavior deviates from the regular pattern. Contextual anomalies are strange patterns in a particular context that always differ from many normal behaviors. The collective anomaly occurs when a group of similar instances acts anomalously competed with the dataset of normal activities.

There are two categories of intrusion detection methods: signature-based intrusion detection systems (SIDS) and anomaly-based intrusion detection systems (AIDS). Anomaly detection systems are classified into two categories based on the sources: network-based and host-based intrusion detection systems. Anomaly detection techniques utilize labels to identify whether the data is normal or anomalous. There are three different anomaly detection techniques such as supervised, unsupervised, and semi-supervised anomaly detection methods. AIDS overcomes the SIDS's drawbacks by modeling normal behaviors using machine learning (ML), statistical-based, or knowledge-based methods. Anomaly-based detection can also produce false results caused by changes in user habits.

Most traditional machine learning algorithms are shallow learning methods emphasizing feature engineering suited for small datasets. Feature engineering requires time and domain expertise to generate the features and remove those irrelevant features from the anomaly detection model. The anomaly detection performance depends on how the feature engineering is implemented and the data preprocessed carried out. The traditional ML methods are simple, have low resource consumption, and perform poorly on computer vision, natural language processing, image translations, etc.

CNN is mostly used in image datasets where the lower layer's neurons reduce the network's features, usually identifying important small-scale features, such as boundaries, corners, and intensity differences. Then in higher layers, the network combines the lower-level features to form more complex features such as simple shapes, forms, and partial objects. And on the final layer, the network combines the lower features to produce the output or classification results.

979-8-3503-4731-9/23/$31.00 ©2023 IEEE

An LSTM works differently than a CNN because an LSTM is designed to retain long-range information so that the information is remembered and not lost in a long sequence. Bi-LSTM adds one more LSTM layer, reversing the information flow direction and overcoming the vanishing gradient problems.

The deep learning method overcomes the problems in traditional ML. The performance of the deep learning-based anomaly detection algorithm depends on neural network architecture, number of hidden layers, types of activation function, number of samples (batch size), and epochs during DL model training and testing. Selecting those hyperparameters and architecture of neural networks in deep neural networks is vital in increasing the detection accuracy of network anomaly detection systems.

II. Related Work

Due to the development of information and technology, many end terminals are connected to the internet and network. The most terminal connected to the internet are smart, and they generate a vast amount of data called big data. Machine learning and deep learning algorithms process the data and make predictions from observations and data that generate valuable insights. The volume of big data is growing daily, so the traditional machine learning algorithm cannot be performed well and needs intensive feature engineering tasks. Deep learning greatly improves detection performance. Still, the nature of the dataset, feature engineering, the hyperparameters on deep neural networks, and neural network architecture plays a vital role in detecting the anomaly in network intrusion detection systems.

Traditional ML depends heavily on feature engineering, which is often time-consuming, complex, and impractical during real-time applications. Authors [2] purposed CNN and RNN-based payload classification approach to detect attacks and achieved an accuracy of 99.36% and 99.98%, respectively, on the DARPA98 dataset. Authors [3] proposed the CNN with Gated Recurrent Unit (GRU) model to address the class imbalance problem by adapting a hybrid sampling algorithm combining Adaptive Synthetic Sampling (ADASYN) and Repeated Edited nearest neighbors (RENN). Random forest and Pearson correlation analysis were used to solve the feature redundancy problem. Their CNN-GRU model outperformed with an accuracy of 86.25%, 99.69%, and 99.65% on UNSW_NB15, NSL-KDD, and CIC-IDS2017 datasets, respectively.

Authors [4] proposed that the deep learning-based network intrusion detection model used adaptive synthetic sampling (ADASYN) to balance the dataset. The autoencoder is used to reduce dimensionality on NSL-KDD. The CNN-BLSTM-based deep learning model provided the highest accuracy and F1 score of 90.73% and 89.65%, respectively. Authors [5] federal transfer learning and convolutional neural networks to solve the problem that arises from data imbalance and different data distribution from the different information sources. The model provided average model accuracy of 86.85% on the UNSW-NB15 multiclass network dataset. Authors [6] used a Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection to overcome the data imbalance problem on KDD99, NSL-KDD, and UNSW-NB15 datasets. The model provides the 94.5% true positive rate and 96.2% AUC on the NSL-KDD dataset. In [7], the Authors concluded from the experimental results that the machine learning classifier's performance improved when the number of target classes decreased. Authors examined this concept on traditional machine learning models, including NB, J48, RF, BayesinNet, Bagging, and Adaboost on three NIDS datasets: UNSW-NB15, CIC-IDS2017_Thrusday, and KDD99.

Authors [8] proposed the method to achieve a successful classification with low computational cost by grouping attributes according to the conditions on which they are collected and creating the cluster attributes for each group with K-means with an accuracy of 98.84% on the KDD99 dataset. The detection accuracy for U2R is very low, 21.92%, which reduces the overall model performance. The authors [9] implemented the hybrid approach combining the CNN and LSTM to improve the anomaly classification accuracy of 98.1% and 96.7% on NSL-KDD and CICIDS2017 datasets, respectively. Authors [10] proposed the hybrid model combining CNN and LSTM to improve the intrusion detection capabilities of advanced metering infrastructure (AMI) utilizing the cross-layer features fusion. The model produced the highest accuracy of 99.95% on KDD Cup99 and 99.79% on the NSL-KDD dataset, having low U2R detection capabilities. Authors [11] implemented the hybrid network of CNN and LSTM to improve intrusion detection to extra network traffic data's spatial and temporal features.

Authors [12] in this paper implemented the method based on the mean control of the CNN-BLSTM algorithm to overcome the traditional data preprocessing and unbalanced numerical distribution on the NSL-KDD dataset, providing the highest accuracy of 99.10%. Still, accuracy for the fewer data class shows poorly. Authors [13] proposed a DL model combining with CNN and Bidirectional LSTM to incorporate the learning of spatial and temporal features of the data on the accuracy of 93.84% and 99.30% and binary class UNSW-NB15 and NSL-KDD datasets, respectively. Authors [14] used CNN Bi-LSTM algorithms on multiclass NSL-KDD dataset and obtained an accuracy of 96.3% where one-hot encoding and min-max normalization are used during data preprocessing. Authors [15] implemented the CNN Bi-LSTM algorithm on preprocessed and obtained an accuracy of 95.4% on the NSL-KDD dataset. The C5.0 decision tree model is combined with the CNN Bi-LSTM model to skip the design feature selection and directly learn the model to represent features of high dimensional data. The Authors [16] implemented the deep learning model based on Bi-directional LSTM on KDDCUP-99 and UNSW-NB15 datasets with outstanding results with 99% accuracy for both KDDCUP-99 and UNSW-NB15 datasets. Most existing models cannot efficiently detect rare attack types, especially User-to-Root (U2R) and Remote-to-Local (R2L) attacks. These two attacks often have lower detection accuracy than other kinds of attacks. Authors in [17] proposed a Bi-LSTM-based intrusion detection system to handle the aforementioned

979-8-3503-4731-9/23/$31.00 ©2023 IEEE

challenges on the NSL-KDD dataset. This Bi-LSTM model provided an accuracy of 94.26% for binary classification. The authors [18] proposed a Bi-directional GAN-based approach to the NSL-KDD and CIC-DDoS2019 datasets. The bidirectional GAN model works perfectly on the imbalance NSL-KDD dataset resulting in an accuracy of 91.12% and an f1 score of 92.68%.

The deep learning-based model in [2], [3] overcome traditional ML problems to detect the anomaly. Data imbalance problems are addressed [4], [5],[6], and [7] . Feature engineering is the most important factor in improving the accuracy of the ML/DL model. Huge numbers of research have been done related to feature engineering, grouping attributes in [8], [9], [10], [11]. A Bi-LSTM combines two separate LSTMs to permit running input in two directions from the past to the future and from the future to the past to improve the traditional LSTM. Bi-LSTM was implemented in [12], [13], [14], [15], [16], [17], [18] to improve the model anomaly detection accuracy.

Most of the above research works focus on increasing the accuracy of traditional or deep machine learning models, working for feature engineering and data imbalance. The research on selecting the hyperparameters in deep learning-based models, training testing data ratio, and architecture of deep neural networks are not focused on. Some researchers do not mention how those values are adopted in their research works. Hence, our research focused on improving those limitations on network anomaly detection systems by experimenting with the NSL-KDD and UNSW-NB15 datasets.

The main contributions of this research work are:

1) Investigating the effect of CNN Bi-LSTM architecture Vs. performance of CNN Bi-LSTM.
2) Investigating model performance Vs. Hyperparameters on both NIDS datasets, i.e., NSL-KDD and UNSW-NB15.
3) Investing the number of layers and memory elements to improve the CNN Bi-LSTM.
4) This research presents the development and implementation of network anomaly detection using a CNN Bi-LSTM model that can detect anomalies with high accuracy of 98.27 % and 99.87% on NSL-KDD and UNSW-NB15, respectively.

The remainder of the paper is as follows. Section II describes the system model of our proposed CNN Bi-LSTM approach. Section III illustrates the results and discussion, while Section IV concludes this research work.

III. System Model

The overall proposed model encompasses the following steps.

Step-1 Data Collection
Step-2 Data Pre-processing
Step-3 Prepare the training and testing dataset
Step-4 Train and Test CNN Bi-LSTM model
Step-5 Model Evaluation and anomaly detection
Step-6 Model Compare and Decision

The overall implementation schematic of the CNN bidirectional LSTM-based model is shown in Fig 1. A detailed discussion of the above-stated methods is provided in the subsequent sections. In figure 2, the detailed architecture of neural networks and CNN and Bi-LSTM layers components are clearly shown.

1. Data Collection and Modelling

This research used two datasets, NSL-KDD KDDTrain+ [19] and UNSW-NB15, where The KDDTrain+ dataset contains the full NSL-KDD train set, including attack-type labels and difficulty level. It has 41 features with five distinct attack classes, Normal, DoS, Probe, R2L, and U2R. Typically, these features are classified into various groups, such as basic, content, and time-based features. NSL-KDD is an improved version of the KDD99 network intrusion dataset, does not include redundant records in the train set, and has no duplicate records in the test sets. The KDDTrain+ dataset contains 125973 records and 41 features. This dataset is balanced because 53.46% of records are normal, and 46.54% are abnormal.

The Australian Centre for Cyber Security (ACCS) cybersecurity research team created the UNSW-NB15 dataset [20] to solve issues with the KDD99 dataset. The data used in this research comprises 42 features. This dataset consists of various attacks, including Analysis, Backdoor, DoS, Exploit, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms counts of 2677, 2329, 16353, 44525, 24246, 58871, 13987, 1511, and 174, respectively. The normal traffic of 93000 data makes the total data 257673.

2. Data Pre-processing

During the KDDTrain+ data preprocessing, the class label is assigned 1 for normal and 0 for abnormal records; hence the dataset becomes the binary class dataset. Then, three categorical features: 'protocol_type,' 'service,' and 'flag,' are converted into numeric features using dummy one hot encoding. The standard scalar method is used to normalize the dataset. For the feature reduction, attributes with more than 0.5 correlation with encoded attack label attributes are only preserved, resulting in 93 features on the final dataset.

UNSW-NB15 data sets consist of test and training separate files. Both contain 45 features, including attack categories and labels. The same methods are used to preprocess both test and training files. Dummy one hot encoding is used for categorical features (proto, service, state), and the standard scalar method is used to normalize the numerical features before combining them. The empty columns are inserted in the location where the features are missed after one hot encoding. All attack categories are grouped into a single attack category to create the binary dataset. After preprocessing, the training and test data sizes become (82332, 199) and (175341. 199), respectively.

979-8-3503-4731-9/23/$31.00 ©2023 IEEE

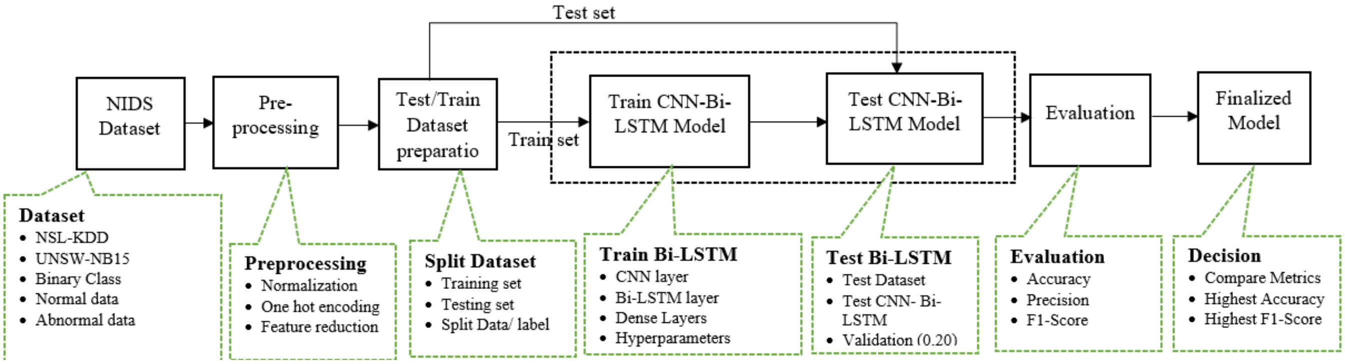

FIGURE 1. BLOCK DIAGRAM OF CNN BI-LSTM MODEL

3. *Prepare the Training and Testing Dataset*

The train-test split approach measures how well machine learning algorithms perform when used to make predictions from data that was not used to train the model. We choose the 70:30 split ratio where our CNN Bi-LSTM model for KDDTrain+ dataset with 70% train and 30% test datasets. There are two separate files chosen in the case of UNSW-NB15, one for training and another for testing the model. The details about the number of training and testing data are explained in the data preprocessing section above.

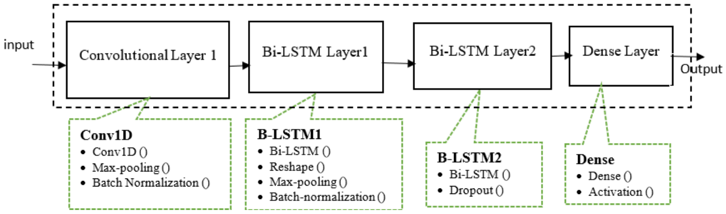

FIGURE 2. CNN BI-LSTM MODEL ARCHITECURE

4. *Bi-LSTM Model*

Convolutional Neural Network (CNN) are deep neural networks that can recognize and classify using the image format. CNN used the convolutional operation to identify the various features of the images then pooling layers extracts the features and a fully connected layer that utilizes the output from the previous layer to classify. Convolutional layers and pooling layers are used for feature extraction whereas the last fully connected dense layer is used for classification purpose.

A recurrent neural network (RNN) consists of feedback loops that process the sequences of data patterns and predict outcomes. RNN consists of memory to store the previous and future state information. RNN has been used to solve machine learning problems such as speech recognition, language processing, and image classification. LSTM addresses the problem of the vanishing gradients of RNN. LSTM architecture consists of the memory block and three multiplicative units- the input, output, and forget gates which are analogous to write, read and reset operations for the cells. The LSTM memory cells can store and access data for extended periods because of the multiplicative gates, which prevents the vanishing gradient problem. A bidirectional RNN often combines two separate RNNs to permit running input in two directions: from the past to the future and from the future to the past. The forward and backward LSTM networks comprise the two LSTM networks that comprise the Bi-LSTM. The goal of the forward LSTM hidden layer is to extract features in the forward direction, and the backward one is to extract features in the backward direction. The bi-directional LSTM predicts or tags the sequence of each element by using finite sequences in the context of previous and subsequent items. This results from two LSTMs processed in series, one from right to left and the other from left to right. The CNN and Bi-LSTM model consists of several layers with hyperparameters. The CNN Bi-LSTM architecture is shown in Figure 2.

5. *Model Evaluation and Anomaly Detection*

Machine learning (ML) or deep learning (DL) model does not provide consistency in performance. Hence the model hyper-parameters need to be examined to obtain better performance. The determination of optimizer, the number of epochs, batch size, dropout, and learning rate are determined by comparing the accuracy and F1-score of the Bi-LSTM model. Finally, the CNN Bi-LSTM model performance parameters are compared with the previously published research results to evaluate our Bi-LSTM model's performance.

6. *Model Comparision and Decision Making*

Different sets of experiments to determine the values of the hyperparameters for the best result. The determination of optimizer, the number of epochs, batch size, and train-test split ratio are determined by comparing the accuracy and F1-score of the Bi-LSTM model. Finally, the Bi-LSTM model performance parameters are compared with the previously published research results to evaluate our Bi-LSTM model's performance. The performance metrics for NSL-KDD and UNSW-NB15 binary NIDS datasets regarding f1-score and accuracy are recorded and compared.

979-8-3503-4731-9/23/$31.00 ©2023 IEEE

IV. Results and Discussions

The experiment was performed on the Anaconda Navigator Jupyter python platform installed on the central processing unit encompassing a 64-bit Windows 10 machine with 16G RAM and an i7-1.99GHz processor. The versions of python, Keras, and TensorFlow used during this research work were 3.7.13, 2.6.0, and 2.9.1, respectively.

The model architecture shown in fig 2 consists of 1 convolution layer with 16 units, max-pooling, and batch normalization, Bi-LSTM layer 1 with 50 memory units, reshape, max-pooling, and batch normalization; the Bi-LSTM layer 2 with 100 memory units and dropout. Finally, the output is taken using a Dense layer with a sigmoid activation function. The model detection accuracy is compared by tuning optimizers, learning rate, epochs, batch size, and dropout rate in the different experiments on NSL-KDD and UNSW-NB15 datasets, which are explained below.

A. Experiment-1 Optimizers Vs. Bi-LSTM Performance

During the training of the CNN Bi-LSTM model, the selection of an optimizer is very important because the helps the ML /DL model to get results faster. Based on the algorithms used by the optimizer, TensorFlow supports nine optimizer classes, including Adadelta, Adagrad, Adam, Adamax, Ftrl, Nadam, RMSprop, SGD, and gradient descent. During the optimizer Vs. Accuracy calculation experiment, the relu activation function, and a 20% dropout rate are used on the model and experimented with seven optimizers, including Nadam, Adam, RMSprop, Adamax, SGD, Adagrad, and Ftrl to find the best optimizer for our model. The performance metrics are recorded in Table 1. the results found that the Nadam optimizer is the winning optimizer for NSL-KDD, and adam optimizer provides the highest accuracy for the UNSW-NB15 dataset. Two optimizers perform differently for both NIDS datasets; even the same model architecture is used.

Table 1. Optimizers Vs. Performance

Epochs = 10, Batch Size = 256					
SN	**Optimizer**	**ACC_NSL**	**F1_NSL**	**ACC_UN**	**F1_UN**
1	**Nadam**	**98.13**	**98.26**	99.11	99.34
2	**Adam**	98.02	98.16	**99.15**	**99.38**
3	RMSprop	97.87	98.01	97.93	98.46
4	Adamax	97.65	97.78	95.33	96.51
5	SGD	97.74	97.91	99.14	99.37
6	Adagrad	96.98	97.21	94.043	95.62
7	Ftrl	53.47	69.68	0.8099	80.99

B. Experiment-2 Learning Rate Vs. Performance

The same model architecture is used to find the learning rate for better model performance where the optimizers are selected from the previous experiment [A]. The learning rate determines how the neural network model weights are updated. The learning rates vary to tune the model accuracy, keeping the other hyperparameters unchanged during this experiment. The learning rate Vs. CNN Bi-LSTM model performance is tabulated in Table 2. The model provides the highest performance at a learning rate of 0.01 on UNSW-NB15 and a learning rate of 0.0002 on the NSL-KDD dataset. The same learning rate provides different model performances.

Table 2. Learning Rate Vs. Performance

epochs = 10, batch Size = 256, KDD (Nadam), UNSW-NB15 (adam)					
SN	**LR**	**ACC_NSL**	**F1_NSL**	**ACC_UN**	**F1_UN**
1	**0.01**	97.49	97.67	**99.67**	**99.76**
2	0.001	98.16	98.29	99.54	99.66
3	0.0001	98.06	98.2	95.81	96.85
4	**0.0002**	**98.18**	**98.3**	97.9	98.44
5	0.0003	98.14	98.27	98.44	98.86
6	0.0004	97.97	98.11	99.13	99.35
7	0.0005	98.11	98.25	99.09	99.32

C. Experiment-3 Drop out Vs. Performance

The dropout rate refers to dropping the neurons during the training model to prevent overfitting. The CNN Bi-LSTM model was trained and tested using epochs of 10 batch size 256 for both datasets. Different values of dropout rate are chosen to study the model performance. The model performs better at a dropout rate of 30% on UNSW-NB15, and a 60% dropout rate performs better on the NSL-KDD dataset. The hyperparameters values, dropout rates, and performance are tabulated in Table 3. The experiment results show the different drop rates for different datasets even though both data sets are similar.

Table 3. Drop Out Vs. Performance

epochs = 10, batch Size = 256, KDD (Nadam), UNSW-NB15 (adam)					
SN	**DropOut**	**ACC_NSL**	**F1_NSL**	**ACC_UN**	**F1_UN**
1	0.1	98.1	98.24	97.44	98.15
2	0.2	98.02	98.16	98.98	99.25
3	**0.3**	98.16	98.29	**99.87**	**99.9**
4	0.4	98.04	98.17	99.27	99.47
5	0.5	97.93	98.09	99.47	99.61
6	**0.6**	**98.21**	**98.33**	99.81	99.86
7	0.7	98.01	98.15	99.58	99.69
8	0.8	98.04	98.18	98.57	98.94

D. *Experiment-4 Batch Size Vs. Performance*

Batch size is the number of samples utilized in a single iteration. The smaller batch size introduces small amounts of data samples and takes longer to train the CNN Bi-LSTM model compared to the larger batch size. The batch size is varied, keeping the other hyperparameters fixed, such as epochs of 5, optimizer's learning rate, and dropout rate values assigned on the model to the respective dataset based on the previous experiment's (Experiment 1-3) finding.

Table 4. Batch Size Vs. Performance

epochs = 5, KDD (Nadam), UNSW-NB15(adam)					
SN	**batch_size**	**ACC_NSL**	**F1_NSL**	**ACC_UN**	**F1_UN**
1	**32**	97.89	98.04	**99.40**	**99.55**
2	64	97.95	98.10	99.35	99.52
3	**128**	**98.06**	**98.20**	99.33	99.50
4	256	97.64	97.79	96.36	97.26
5	512	97.92	98.08	96.90	97.70

979-8-3503-4731-9/23/$31.00 ©2023 IEEE

This experimental result in table [4] shows that the combination of hyperparameters in the neural network provides a different performance. During this experiment, the CNN Bi-LSTM model performed better when batch size is 128 for NSL-KDD and 32 for UNSW-NB15 datasets with epochs of 5.

E. *Experiment-5 Epochs Vs. Performance*

The number of times the learning algorithm will go over the complete training dataset is determined by the hyperparameter known as the epoch which can be any integer value that lies between 1 to infinity. The model takes a long time to train when we choose smaller epoch values and vice versa. The CNN Bi-LSTM model performance for different values of epochs and assigned the other hyperparameters values found from previous experiments are recorded in the table [5].

TABLE 5. EPOCHS VS. PERFORMANCE

Batch size = 256, KDD (Nadam)			
SN	**Epochs**	**ACC_NSL**	**F1_NSL**
1	2	95.48	95.94
2	10	98.13	98.26
3	25	98.21	98.33
4	50	98.20	98.33
5	**75**	**98.27**	**98.39**
6	100	98.26	98.39

V. CONCLUSION

The literature review shows that the NSL-KDD and UNSW-NB15 have average model accuracy of 99%, but the smaller attack class (U2R, R2L, etc.) detection is very low. The enemy is the enemy, and every attack is responsible for destroying network machines equally. Hence compare the result with the existing result of 91.12% [18], and 90.83% [4] accuracy for NSL-KDD and 99.70% [16], 82.08% [13] 82.08% for the UNSW-NB15 dataset. Our experiment improves accuracy, which is 98.27% on NSL-KDD and 99.87% on UNSW-NB15 binary dataset. The values of CNN Bi-LSTM model hyperparameters, including optimizer, epochs, batch size, the learning rate, and dropout for the CNN Bi-LSTM neuron architecture, are investigated for the highest detecting accuracy for binary NSL-KDD and UNSW-NB15 dataset.

ACKNOWLEDGMENT

The work is supported in part by funding from the NSF awards #1910868, #2219611, and the NSF CyberCorps SFS program.

REFERENCES

[1] N. Moustafa, J. Hu and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications,* vol. 128, p. 33–55, 2019.

[2] H. Liu, B. Lang, M. Liu and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowledge-Based Systems,* vol. 163, p. 332–341, 2019.

[3] B. Cao, C. Li, Y. Song, Y. Qin and C. Chen, "Network Intrusion Detection Model Based on CNN and GRU," *Applied Sciences,* vol. 12, p. 4184, 2022.

[4] Y. Fu, Y. Du, Z. Cao, Q. Li and W. Xiang, "A Deep Learning Model for Network Intrusion Detection with Imbalanced Data," *Electronics,* vol. 11, p. 898, 2022.

[5] X. Ji, H. Zhang and X. Ma, "A Novel Method of Intrusion Detection Based on Federated Transfer Learning and Convolutional Neural Network," in *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2022.

[6] T. Acharya, I. Khatri, A. Annamalai and M. F. Chouikha, "Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection," in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, 2021.

[7] T. Acharya, I. Khatri, A. Annamalai and M. F. Chouikha, "Efficacy of Machine Learning-Based Classifiers for Binary and Multi-Class Network Intrusion Detection," in *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, 2021.

[8] M. Xiong, H. Ma, Z. Fang, D. Wang, Q. Wang and X. Wang, "Bi-LSTM: Finding Network Anomaly Based on Feature Grouping Clustering," in *2020 The 3rd International Conference on Machine Learning and Machine Intelligence*, 2020.

[9] S. N. Pakanzad and H. Monkaresi, "Providing a hybrid approach for detecting malicious traffic on the computer networks using convolutional neural networks," in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020.

[10] R. Yao, N. Wang, Z. Liu, P. Chen and X. Sheng, "Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach," *Sensors,* vol. 21, p. 626, 2021.

[11] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Security and communication networks,* vol. 2020, 2020.

[12] L. Zhang, J. Huang, Y. Zhang and G. Zhang, "Intrusion detection model of CNN-BiLSTM algorithm based on mean control," in *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*, 2020.

[13] J. Sinha and M. Manollas, "Efficient deep CNN-BILSTM model for network intrusion detection," in *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*, 2020.

[14] A. Li and S. Yi, "Intelligent Intrusion Detection Method of Industrial Internet of Things Based on CNN-BiLSTM," *Security and Communication Networks,* vol. 2022, 2022.

[15] J. Gao, "Network Intrusion Detection Method Combining CNN and BiLSTM in Cloud Computing Environment," *Computational Intelligence and Neuroscience,* vol. 2022, 2022.

[16] T. S. Pooja and P. Shrinivasacharya, "Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cyber security," *Global Transitions Proceedings,* vol. 2, p. 448–454, 2021.

[17] Y. Imrana, Y. Xiang, L. Ali and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications,* vol. 185, p. 115524, 2021.

[18] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina and J. Kwak, "Improved Bidirectional GAN-Based Approach for Network Intrusion Detection Using One-Class Classifier," *Computers,* vol. 11, p. 85, 2022.

[19] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, 2009.

[20] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, 2015.