Network Intrusion Detection System Using Deep Learning Models

[1]Amit Patti, [2]Kusuma Shree V, [3]Mayuri D Patil
[1, 2, 3] Department of Computer Science and Engineering, PES University
[1]amit.patti1405@gmail.com, [2]kusumareddy2002@gmail.com, [3]mayuri.dpat@gmail.com

***Abstract*— Deep learning algorithms have shown promise in speech recognition, picture processing, natural language processing, and a wide range of other fields. A Network Intrusion Detection System (NIDS) can be used by system administrators to detect network security breaches in their organizations. Network Intrusion Detection Systems (NIDSs) are important tools for network administrators to use in detecting various security breaches on their company's network. If an intrusion is identified, NIDS monitors and analyses network traffic entering and exiting an organization's network devices, and raises alarms. Our solution helps detect the two types of networks which intrude the system, these are normal and abnormal attacks on a computer. The dataset we have used is the NSL-KDD dataset which contains 43 classes and close to 126000 samples. This paper talks about 5 different models. The final model is built using a deep learning approach which has an detection rate of 99.04%.**

Keywords—data analytics, auto encoder, NIDS, ensemble models, CNN BiLSTM

I. Introduction

Cyber-crime has risen substantially as a result of rapid technological advancements and the global proliferation of internet networks. Because the interchange of digital information across networks has created exploitable vulnerabilities that can harm both persons and businesses, a good network security solution is essential for maintaining confidentiality, integrity, and availability. If there's one takeaway from 2019, it's that no company, no matter how big or little, is immune to a cyber-attack. Cyber-attacks have never been more sophisticated, elusive, and targeted. As a result, new security methods must be developed on a regular basis.

A network intrusion detection system (NIDS) is a critical component of network security since it detects intrusions and notifies the proper authorities. To regulate and monitor network traffic packets at many places for a potential intrusion or abnormality, NIDS may include both hardware (sensors) and software (console). According to detection techniques, Intrusion Detection Systems are classified into two kinds: Anomaly and Misuse Detection. An anomaly Detection System is more suitable for detected unknown attacks.

Our paper focuses on implementing NIDS using both machine learning(ML) and deep learning(DL) techniques. To demonstrate this model, the NSL-KDD dataset is used. We have implemented several models to perform both binary and multi classification on the dataset with good accuracies.

II. Related Works

Most of the papers detecting attacks have used machine learning models to classify as normal or abnormal attacks. There are multiple datasets available for network intrusion detection: NSL-KDD, UNSW-NB15 and KDDCup1999.

We can divide IDS into two categories based on detection techniques: anomaly detection and misuse detection. Unknown attacks are better identified with an anomaly detection system. Direct ways for performing these IDSs include rule-based methodologies, machine learning, and data mining. [1] Because of its relevance and efficiency, the suggested system uses a Deep Neural Network (DNN) to detect network intrusion packets. This algorithm assigns appropriate weights to all features in the input layer, which are then used to make decisions. To prevent data privacy violations, the proposed intrusion detection system employs a deep neural network algorithm in conjunction with anomaly detection techniques to detect attacks.

[2] The proposed intrusion detection system identifies attacks using a deep neural network algorithm and anomaly detection techniques, all without accessing data in the packet payload, maintaining data privacy. With synthetically created attack actions, the model reflects real-world current network communication behavior. The proposed deep learning model uses a CNN with regularized multi-layer perceptron instead of a fully coupled feed-forward neural network (FNN). As the input propagates over successive convolutional layers, we used input padding to control diminishing tensor size. The pooling layer is used to reduce feature dimensions between subsequent convolutional layers.

If an intrusion is discovered, an NIDS monitors and analyzes network traffic entering and exiting an organization's network devices, and raises alarms. [3] Signature (misuse) based NIDS (SNIDS) and anomaly detection based NIDS (ANIDS) NIDS are classified into two groups based on their intrusion detection methodologies (ADNIDS). When ADNIDS detects a divergence from the regular traffic pattern, it classifies network traffic as an intrusion. Because attack signatures are limited, the performance of existing approaches suffers. [4]

[5] Propose a deep learning-based real-time network intrusion detection method that incorporates big data, natural language processing, and deep learning technologies. Small number of labeled samples, NN with

semi-supervised learning LSTM NN regression Robe DoS attacks Make up for the high false alert rate. In the last layer of the GRU Model, Linear SVM was introduced to replace Softmax. Convolutional and feed forward neural networks. The system architecture consists of i) Data Acquisition Layer, ii) Data Collection model and iii) Design of Intrusion Detection Module.

III. Proposed Solution

Network Intrusion Detection using various machine learning models provided good accuracies but as the dataset increases, the accuracies of the models are decreasing. Our solution includes various deep learning models and a couple of machine learning models for comparison of the accuracies.

Currently, all existing solutions [reference related study] dealing with this problem statement deal with it in a unidirectional manner i.e., either less number of samples or considering less attributes to classify an attack.. Our approach is unique as it considers 42 attributes and has a greater dataset than the referenced papers.

A. Dataset Description

The dataset used in this analysis is NSL-KDD dataset, it's a real dataset curated by the Canadian Institute for Cybersecurity. Our dataset has 125973 samples and 43 attributes that describe network attacks. The dataset contains some missing data (NaN) which are being removed.

B. Preprocessing

Attributes such as difficulty level are redundant. Among the remaining relevant attributes, there were multiple attributes which are categorical variables and the rest were numerical variables.

Attributes could be divided into two types of attacks: i) Normal and ii) Abnormal which has been used for binary classification. The abnormal attack can be further divided into 4 categories: i) DoS ii) Probe iii) U2R and iv) R2L. This is used for multi-class classification of the attacks. The various attacks have been one-hot encoded into numeric values (0,1,2,3 and 4).

Three of the 41 features are nominal, four are binary, and 34 are continuous classes. The RandomForest model was used for feature extraction. There are 38 trac classes in the test data, with 21 attack classes from the training data, 16 unique attacks, and one normal class.

C. Descriptive analysis

The binary classification contains 53.46% of normal attacks and 46.54% of abnormal attacks[Fig 1]. The multi-class classification contains 53.46% normal attacks, 36.46% of DoS attacks, 9.25% of R2L attacks, 0.95% of Probe attacks and 0.05% of U2R attacks.[Fig 2]

We then extracted a dataframe that consists of only numeric attributes of the binary class dataset and encoded label attribute. Only few attributes were selected which were found using Pearson correlation coefficient. The same is done for multi-class classification.

The breakdown of the attacks for binary classification is shown below.

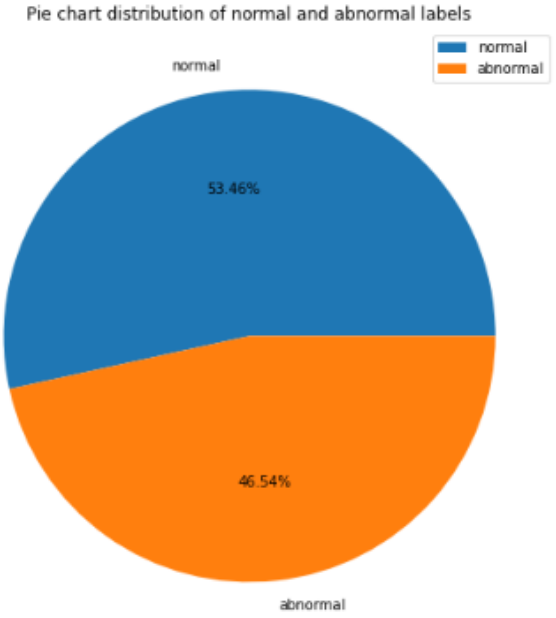


Fig. 1. Pie chart for normal and abnormal labels

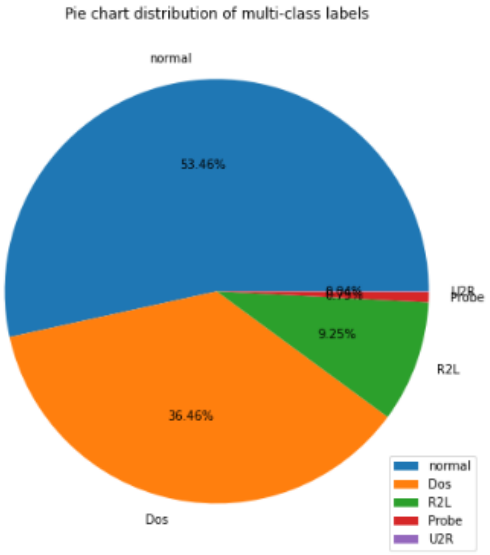


Fig. 2. Pie chart distribution of multi-class labels

D. Training and Testing

The given dataset was trained using 5 different models spread across different deep learning and machine learning models for a comparative analysis to build a robust model for network intrusion detection.

In our research, we used the NSL-KDD dataset. To circumvent the limitations of the KDD Cup dataset, the NSL-KDD dataset was proposed.

The LSVM machine learning model enables us to use a linear support vector machine to classify data. This has been used for binary classification. This approach is very well suited to huge datasets with a large number of predictor fields. We created a rudimentary model using the node's default settings, and it had a 96.55 percent accuracy. The same model was implemented for multi-class classification and achieved an accuracy of 95.39%.[Fig 3]

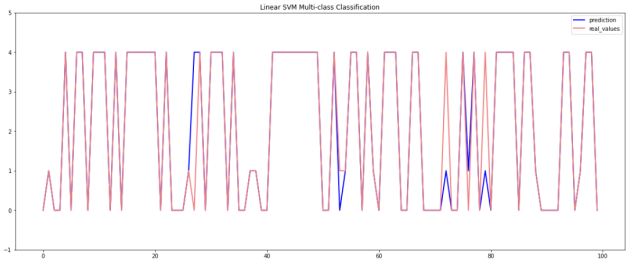


Fig. 3. Graph of real values and predicted values

The next model we used was the multilayer perceptron model which is a supplement of feed forward neural networks. The one-hot encoded dataset was used to train the model. The data is divided into two categories: training and testing of which 75% is used for training and 25% for testing. This dataset is split using the train_test_split() function. The input layer and the first layer consists of 50 neurons where relu was used as the activation function. The output layer or the final layers consists of only 1 neuron where sigmoid activation function is used. The model is implemented for both binary and multiclass classification and achieved accuracies of 97.76% and 96.98% respectively. [Fig 4]

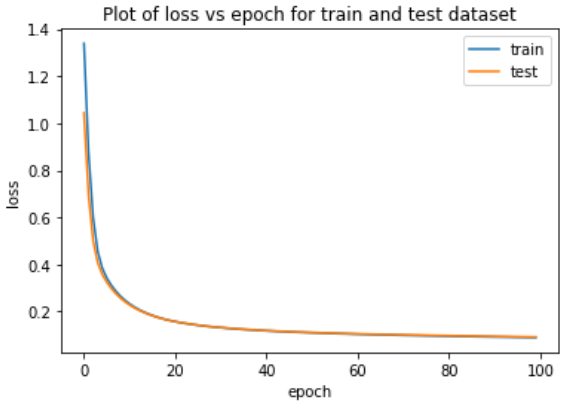


Fig. 4. Plot of loss vs epoch for MLP model

The LSTM model was used for only binary classification and got an accuracy of 97.71%. [6]The LSTM is a sequential network with an advanced RNN that allows information to persist. The LSTM architecture works similar to the RNN architecture. The training and testing data is split into 75% and 25%. The input layer consists of 50 neurons where the dimension is 93.A single neuron serves as the output layer, with sigmoid as the activation function. Binary cross entropy is the loss function employed here. The optimizer is adam. The model was run for 100 epochs with a batch size of 1000. [Fig 5]

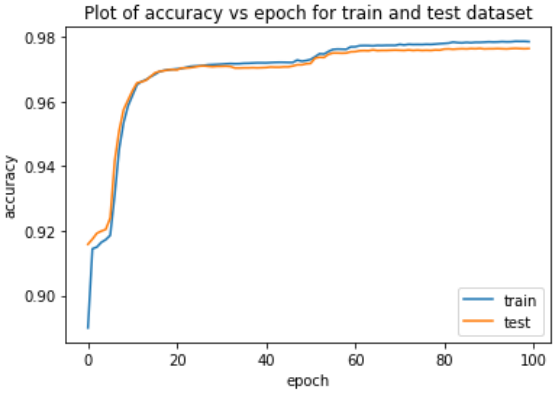


Fig. 5. Plot of accuracy vs epoch for LSTM model

The next deep learning model we implemented was the Auto-Encoder model. We used the auto-encoder model because it proves to be one of the best algorithms to detect anomalies in an unbalanced dataset. Autoencoders are trained to minimize reconstruction error. We performed both binary classification and multi-classification on our dataset.

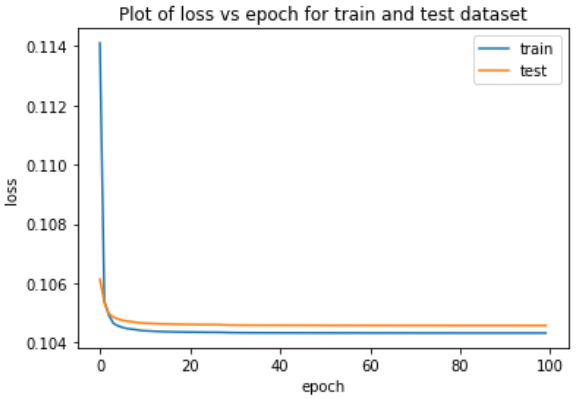


Fig. 6. Plot of loss vs epoch of binary classification of Auto Encoder model

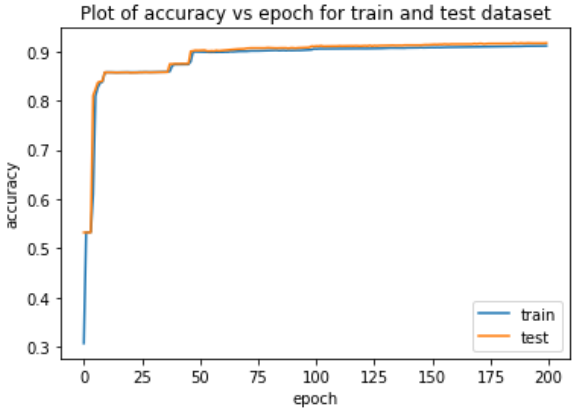
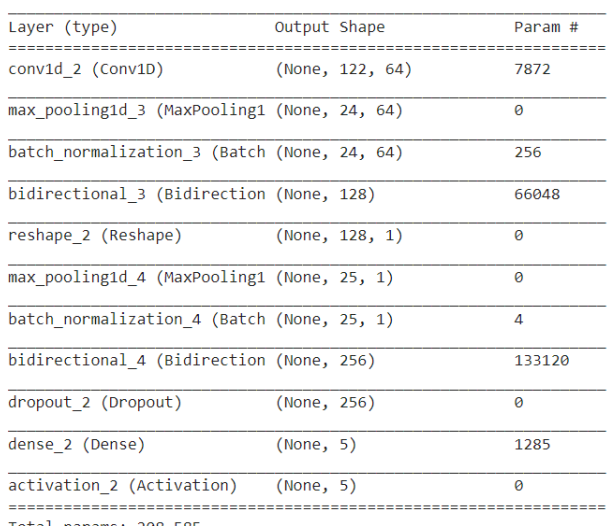


Fig. 7. Plot of accuracy vs epoch of multiclass classification of Auto Encoder model

With the auto-encoder model, we achieved 89.66% for binary classification [Fig 6] and 91.41% accuracy for multi-classification.[Fig 7]

The final model we implemented was the CNN-BiLSTM model. To incorporate learning of spatial

and temporal characteristics of the data, this model combines the strengths of Convolutional Neural Networks and Bi-directional LSTMs. This model is implemented to perform multi-classification on the dataset. [Fig 8]



Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 122, 64)	7872
max_pooling1d_3 (MaxPooling1	(None, 24, 64)	0
batch_normalization_3 (Batch	(None, 24, 64)	256
bidirectional_3 (Bidirection	(None, 128)	66048
reshape_2 (Reshape)	(None, 128, 1)	0
max_pooling1d_4 (MaxPooling1	(None, 25, 1)	0
batch_normalization_4 (Batch	(None, 25, 1)	4
bidirectional_4 (Bidirection	(None, 256)	133120
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 5)	1285
activation_2 (Activation)	(None, 5)	0

Fig. 8. Summary of CNN-BiLSTM model

The model is run twice during the training with epochs=15 and batch size 1000.

IV. Results and Conclusion

In this paper, our proposed solution looked to make a robust predictive model which would serve as the underlying model for network intrusion detection. In the process, we compared 5 models to build the best predictive model.

From the above models we came to a conclusion that CNN-BiLSTM gave very promising results. The detection rate of the attacks using this model is 99.04%. The true positive rate achieved with this model is 98.97%.

V Future Work

As part of future work, we can work on using ensemble learning on the above implemented models and get better results for both binary classification and multi-classification. Ensemble learning has recently been identified as a significant factor in improving the performance of deep learning models. Therefore with the help of this methodology, we can achieve better accuracies for the unbalanced datasets as well.

VI References

[1] https://iopscience.iop.org/article/10.1088/1742-6596/1804/1/012138/meta

[2] https://www.sciencedirect.com/science/article/pii/S1877050921011078

[3] http://eprints.eudl.eu/id/eprint/2057/

[4] https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150

[5] https://ieeexplore.ieee.org/abstract/document/9040718/

[6] https://en.wikipedia.org/wiki/Long_short-term_memory