Automatic Network Intrusion Detection System Using Machine learning and Deep learning

Preprint · February 2024

DOI: 10.36227/techrxiv.170792293.35058961/v1

7 authors, including:

Mohammed Mynuddin
North Carolina Agricultural and Technical State University
47 PUBLICATIONS **262** CITATIONS
SEE PROFILE

Sultan Uddin Khan
North Carolina Agricultural and Technical State University
26 PUBLICATIONS **30** CITATIONS
SEE PROFILE

Md Jahidul Islam
Tuskegee University
8 PUBLICATIONS **29** CITATIONS
SEE PROFILE

Mohammad Iqbal Hossain
North Carolina Agricultural and Technical State University
12 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Automatic Network Intrusion Detection System Using Machine learning and Deep learning

Mohammed Mynuddin [1], Sultan Uddin Khan [1], Zayed Uddin Chowdhury [1], Foredul Islam[1], Md Jahidul Islam [1], Mohammad Iqbal Hossain [1], and Dewan Mohammed Abdul Ahad[1]

[1]Affiliation not available

February 14, 2024

Automatic Network Intrusion Detection System Using Machine learning and Deep learning

1st Mohammed Mynuddin
Dept. of Electrical Engineering
North Carolina A & T State University
mmynuddin@aggies.ncat.edu

2nd Sultan Uddin Khan
Dept. of Electrical Engineering
North Carolina A & T State University
skhan5@aggies.ncat.edu

3th Zayed Uddin Chowdhury
Data Scientist
Data Bid Machine Inc.
zayed.upal@gmail.com

4th Foredul Islam
Associate Power Systems Engineer-II
Open Systems International Inc.
Foredul.Islam@aspentech.com

5th Md Jahidul Islam
Dept. of Electrical Engineering
Tuskegee University
jahidul.islambd2@gmail.com

6th Mohammad Iqbal Hossain
Dept. of Electrical Engineering
North Carolina A & T State University
mohammad.iqbl.hossain@gmail.com

7th Dewan Mohammed Abdul Ahad
Dept. of Electrical and Computer Engineering
University of North Carolina at Charlotte
dahad@uncc.edu

***Abstract*—In recent years, the popularity of network intrusion detection systems (NIDS) has surged, driven by the widespread adoption of cloud technologies. Given the escalating network traffic and the continuous evolution of cyber threats, the need for a highly efficient NIDS has become paramount for ensuring robust network security. Typically, intrusion detection systems utilize either a pattern-matching system or leverage machine learning for anomaly detection. While pattern-matching approaches tend to suffer from a high false positive rate (FPR), machine learning-based systems, such as SVM and KNN, predict potential attacks by recognizing distinct features. However, these models often operate on a limited set of features, resulting in lower accuracy and higher FPR. In our research, we introduced a deep learning model that harnesses the strengths of a Convolutional Neural Network (CNN) combined with a Bi-directional LSTM (Bi-LSTM) to learn spatial and temporal data features. The model, evaluated using the NSL-KDD dataset, exhibited a high detection rate with a minimal false positive rate. To enhance accuracy, K-fold cross-validation was employed in training the model. This paper showcases the effectiveness of the CNN with Bi-LSTM algorithm in achieving superior performance across metrics like accuracy, F1-score, precision, and recall. The binary classification model trained on the NSL-KDD dataset demonstrates outstanding performance, achieving a high accuracy of 99.5% after 10-fold cross-validation, with an average accuracy of 99.3%. The model exhibits remarkable detection rates (0.994) and a low false positive rate (0.13). In the multiclass setting, the model maintains exceptional precision (99.25%), reaching a peak accuracy of 99.59% for k-value=10. Notably, the Detection Rate for k-value=10 is 99.43%, and the mean False Positive Rate is calculated as 0.214925.**

***Index Terms*—Network Intrusion detection, Cyber-attack, Data preprocessing, K-cross validation, Classification, Bi-LSTM**

I. Introduction

The whole world become a global online market due to the increase in the speed of the internet. People, governments, and businesses today rely on the Internet for their daily activities, operations, and personnel matters. Email is used to exchange documents, wireless handsets are used for phone calls, images are posted to Facebook, Instagram, and Snapchat, and LinkedIn and Twitter are used for social networking. For instance, a lot of companies now sell their products online. Some people might believe that these internet venues are safe. Another important example is people cannot go outside because of the shutdown during COVID-19, they work and attend online meetings from home. Despite the many advantages of using the internet, users are no longer safe from cyberattacks. Today's systems and operations are more effective because to the integration of information technology, but this has also raised the risk of cyberattacks, which is endangering the economic stability of many industrialized nations. In [1], the authors examine the impact of cyber-attack on the economy. According to studies on how cyberattacks affect stock prices, identified target companies suffer losses in the days after an attack that range from 1% to 5%. For the typical New York Stock Exchange corporation, price reductions of this scale result in shareholder losses of $50 million to $200 million. Gandhi et al. [2], analysis the cyberattack on multiple sectors namely social, political, economic, and cultural in the different countries between 1995–2009. A cyberattack is an attack in which the attacker tries to enter an IT system without authorization in order to steal, disrupt, demand money, or carry out other nefarious deeds. Computer systems are targeted in

cyberattacks in an effort to be destroyed, rendered inoperable, disrupted, or taken over. Data stored on the systems is also blocked, modified, or stolen [3]. Cybercrime is increasing substantially year after year as attackers grow more efficient and smarter [4].

Recently, machine learning (ML) applications have witnessed unprecedented growth across various fields [5]–[7], revolutionizing the way tasks are performed and problems are addressed [8]–[10]. One notable domain where machine learning, particularly Deep Learning (DL), has made substantial contributions is in the realm of intrusion detection systems (IDS) [11]. ML techniques, including DL, offer advanced capabilities in recognizing patterns, anomalies, and trends within vast datasets, enabling the development of more robust and adaptive intrusion detection mechanisms. By leveraging the power of neural networks, these systems can autonomously learn and adapt to evolving cyber threats, providing a dynamic defense against sophisticated attacks. The application of ML and DL in intrusion detection systems enhances the accuracy and efficiency of threat detection, aiding cybersecurity professionals in safeguarding critical information systems [12]. This integration not only improves the speed of threat identification but also allows for proactive responses, thereby fortifying the security posture of various industries in the face of an ever-evolving threat landscape.

In this paper, we propose a network intrusion detection method based on machine learning and deep learning to identify malicious connections. This is achieved by leveraging the strengths of a Convolutional Neural Network (CNN) combined with a Bi-directional Long Short-Term Memory (Bi-LSTM) network [13]. The summary of contributions is outlined below:

- **Network Intrusion Detector:** This paper introduces a network intrusion detection system, a predictive model designed to identify malicious connections within a network.
- **Deep Learning Model for Spatial and Temporal Features:** In this research, we develop a deep learning model that leverages the combined strengths of a Convolutional Neural Network (CNN) and a Bi-directional Long Short-Term Memory (Bi-LSTM) network. The model is trained and evaluated using the NSL-KDD dataset, demonstrating high detection rates and low False Positive Rates (FPR). K-fold cross-validation is employed to enhance model accuracy.
- **CNN with Bi-LSTM for Improved Effectiveness:** This paper employs the CNN with Bi-LSTM algorithm to enhance the effectiveness of the intrusion detection model. The approach leads to improved accuracy, F1-score, precision, and recall, making it a robust solution for network security.

The subsequent sections of this paper follow a structured path: In Section II, we delve into an in-depth discussion of existing research challenges. Moving on to Section III, we describe the related work. Section IV provides insights into our methodology, shedding light on its system design and components. Continuing in Section V, we present the compelling results obtained from our simulations. The future direction of our research work is addressed in Section VI. Finally, we draw conclusions in Section VII.

II. GAP CHALLENGES

A recent study shows that most research is conducted on NSL_KDD and KDDCup'99 datasets, limiting the performance in real-time environments. These datasets are outdated for modern network attacks. To achieve improved performance in terms of intrusion detection accuracy, we should focus on updated datasets. Intrusion detection systems face challenges such as a high false alarm rate, low detection rate, unbalanced datasets, and slow response times.

III. RELATED WORKS

A defense system called an intrusion detection system (IDS) is used to find intruders who are attempting to access a network without authorization. A hardware or software IDS that scans network inbound and outbound traffic for suspicious activity using intrusion signatures may be used. There are three different types of IDS, including network-based, host-based, and hybrid-based detection systems. HIDS's primary function will often rely on the fact that cybercriminals or other attackers who wish to take over the machines they have compromised are real. The software was being installed by hackers to claim "control" of the machine [14]. To keep track of the traffic, NIDS is positioned strategically inside the network. It aids in the detection of network-related security concerns [15]. In [16] the authors show a comparison between NIDS and HIDS using deep learning algorithm. Kurundkar et al. [17] used Snort to identify the network intrusion. In [18] this research, the authors tackle the escalating global cybersecurity concern, focusing on critical sectors. They propose a research framework to identify and categorize various cyber threats, while also presenting a fuzzy logic-based intrusion detection system for issuing warnings to administrators. This empowers them to proactively respond and safeguard interconnected and distributed information systems. In [19] the authors developed a Fuzzy logic-based intrusion detection system, but this system does not have the guarantee to detect abnormalities online. To effectively identify different kinds of malicious activity, authors describe a Genetic Algorithm technique throughout this research with an enhanced initial population and selection operator [20]. NIDS employs statistical or pattern-based algorithms to find the good things happening in the network. Amanoul, Sandy Victor, et al. [21] used machine learning-based algorithms such as Random Forest, Bayes Net, and Neural Network for detecting the network intrusions. Deep learning algorithms have been built in NIDS prediction models in this study to automatically identify abnormalities and dangers [22].

IV. METHODOLOGY

In this project, we will develop a network intrusion detection system by using Machine learning (ML) and Deep learning (DL) techniques. The overall process is divided into three

steps as shown in Fig. 1, that are 1) Data preprocessing 2) Training, and 3) Testing [12]. The raw data contains lots of duplicate entries and null data, so we need to preprocess the data before feeding to the model. Then, we will divide the whole dataset into groups. Nearly 80% of the total dataset size is typically made up of the training dataset, with the remaining 20% representing the testing dataset. The model will predict either benign or attack after being well-trained on the training dataset. We will verify the performance of the proposed model in terms of F1 score, precision, and accuracy. We will use Python programming language for implementing this project. We will use different tools and libraries in Python such as Keras, numpy, sklearn, matplotlib, pandas, scipy, and so on.

A. Algorithm

The algorithm I outlines a structured approach for detecting network intrusions using machine learning techniques. It begins with initialization, where necessary Python libraries for data processing, machine learning, and visualization are loaded. The data preparation phase involves loading the NSL-KDD dataset, which is a widely used data set for network intrusion detection, and then preprocessing it by removing an unnecessary column, normalizing, and encoding categorical features.

Data exploration is conducted by analyzing the distribution and occurrence of different attack types. Feature engineering and label engineering follow, where features are one-hot encoded, normalized, and attack labels are consolidated into generalized classes. The core of the algorithm is in building and training a deep learning model with convolutional and LSTM layers, followed by compiling the model with performance metrics.

The model is rigorously tested using stratified k-fold cross-validation and evaluated using various metrics such as F1-score, detection rates, and false positive rates. The algorithm also includes a comparison of the constructed classifier with various machine-learning classifiers. It concludes with a final visualization and evaluation phase where confusion matrices are created and the performance of different classifiers is compared before finishing the algorithm with all necessary evaluations and visualizations completed.

Algorithm 1 Network Traffic Classification for Intrusion Detection

Initialize:
Import necessary Python libraries for data processing, machine learning, and visualization.
Load and Prepare Data:
Load the NSL-KDD dataset from CSV files for both training (KDDTrain+.txt) and testing (KDDTest + .txt).
Assign column names to the datasets.
Remove the 'difficulty_level' column from both datasets.
Normalize and encode categorical data.
Data Exploration:
Count the occurrences of different attack types.
Visualize the distribution of attack classes.
Analyze the protocol type against the subclass of attacks.
Feature Engineering:
One-hot encode categorical features like 'protocol_type', 'service', and 'flag'.
Normalize the numerical features to scale the data.
Label Engineering:
Consolidate various attack labels into generalized attack class names ('Dos', 'R2L', 'Probe', 'U2R', and 'Normal').
Model Building and Compilation:
Construct a Sequential deep learning model with convolutional and LSTM layers.
Compile the model with metrics to evaluate performance.
Model Training:
Perform stratified k-fold cross-validation.
Train the model using the training set and validate using the test set.
Record the model's performance metrics.
Model Evaluation:
Calculate and visualize F1-scores, detection rates, and false positive rates.
Compare the performance across different k-values.
Classifier Comparison:
Train and evaluate various machine learning classifiers.
Compare classifier performance based on accuracy, precision, recall, and F1-score.
Visualization and Final Evaluation:
Visualize confusion matrices for each classifier.
Tabulate the performance of different classifiers for a comprehensive comparison.
Finish Algorithm:

B. Bi-LSTM Structure

The proposed model consists of a 1-D CNN layer, several layers of BiLSTM, reshape, Dropout layer, Activation layer, Batch Normalization layers, dense layer, and Max pooling layers (See Fig. 2). The purpose is to exploit the parameter sharing, spatial architecture, and local perception aspects of the 1-D CNN layer. Parameter Sharing reduces the number of parameters and free variables, resulting in faster feature extraction and reduced processing time. The spatial organization of critical paradigm characteristics in a two-dimensional array allows for enhanced detection of feature connection.

C. Dataset

We use NSL-KDD Dataset for our designed model. The NSL-KDD dataset was made accessible by the University of New Brunswick [23]. The KDDCup'99 Dataset [24], which has inherent faults that have been found by various analysis, has been replaced by the NSL-KDD Database. Due to the fact that it contains all of the pertinent data from the entire KDD Dataset, NSL-KDD is one of the most widely used datasets for Network Intrusion Detection Systems analysis. Several things set NSL-KDD apart from its predecessor, such as: Depending on how many records are selected from each difficulty group, the percentage of statistics in the core KDD Dataset is in-

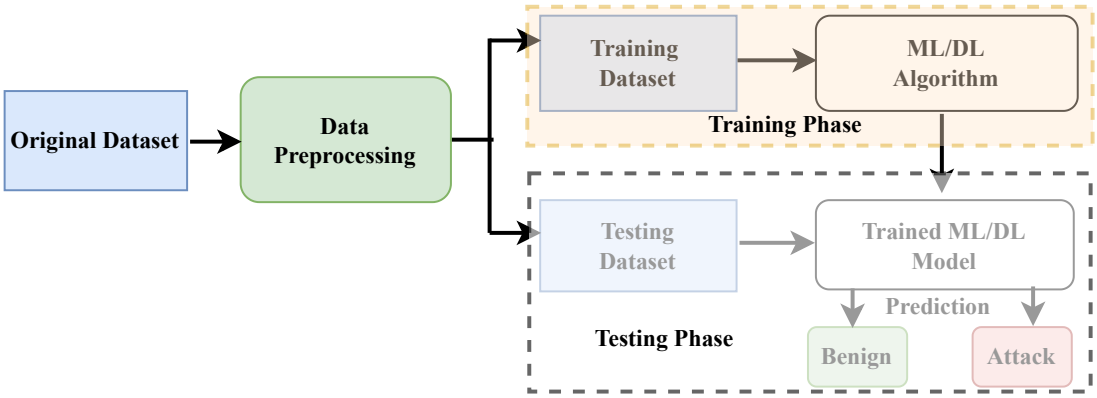

Figure 1: Block diagram of the system

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 122, 64)           7872

 max_pooling1d (MaxPooling1D  (None, 24, 64)           0
 )

 batch_normalization (BatchN  (None, 24, 64)           256
 ormalization)

 bidirectional (Bidirectiona  (None, 128)              66048
 l)

 reshape (Reshape)           (None, 128, 1)            0

 max_pooling1d_1 (MaxPooling  (None, 25, 1)            0
 1D)

 batch_normalization_1 (Batc  (None, 25, 1)            4
 hNormalization)

 bidirectional_1 (Bidirectio  (None, 256)              133120
 nal)

 dropout (Dropout)           (None, 256)               0

 dense (Dense)               (None, 5)                 1285

 activation (Activation)     (None, 5)                 0

=================================================================
Total params: 208,585
Trainable params: 208,455
Non-trainable params: 130
```

Figure 2: Model summary of Bi-LSTM

versely correlated. The data collection contains four different types of attacks: user to root (U2R), remote to local (R2L), and denial of service (DoS) (R2L)(See Table I) . This data set's feature types can be divided into 4 categories: 4 categorical, 6 binary, 23 discrete, and 10 continuous. The characteristics of a traffic record are divided into four categories and provide information about the interactions with the traffic entered by the IDS.

Table I: NSL-KDD dataset

Category	Count
Normal	77054
DOS	53385
Probe	14077
R2L	3749
U2L	252
Total	14517

D. Data Visualization

In this section, we visualize the data. In Fig. 3. represents the correlation matrix of all attributes after one hot encoding. All categorical variables are converted into binary features using One-Hot-Encoding. For Binary Classification, a duplicate of DataFrame is produced. The attack label is divided into two categories: normal and abnormal. LabelEncoder () is used to encode 'label,' and encoded labels are saved in 'intrusion.' 'label' is encoded just once. The multiclass classification is visualized in Fig.4. Binary class classification is shown in Fig. 5.

E. Preprocessing data

Data augmentation is frequently carried out via One Hot Encoding of continuous data and Normalization of numerical features. The NSL-KDD Dataset, however, has a more precise number of entries for each attack category, as was already indicated. Fig. 6 shows some samples of cleaning data after cleaning the raw data.

F. One Hot Encoding

For our deep learning model to generate precise predictions, the NSL-KDD dataset contains categorical properties that need to be converted to numerical values. Therefore, in the pre-processing stage, these variables were converted to numerical values using the get-dummies function of the Pandas python package. Label encoding was favored over one-hot encoding since the latter could produce multiple integers in the same column, which could lead the model to fail.

G. Normaization

In order to reduce duplicate data and expedite model training, normalization involves downscaling data into a specific

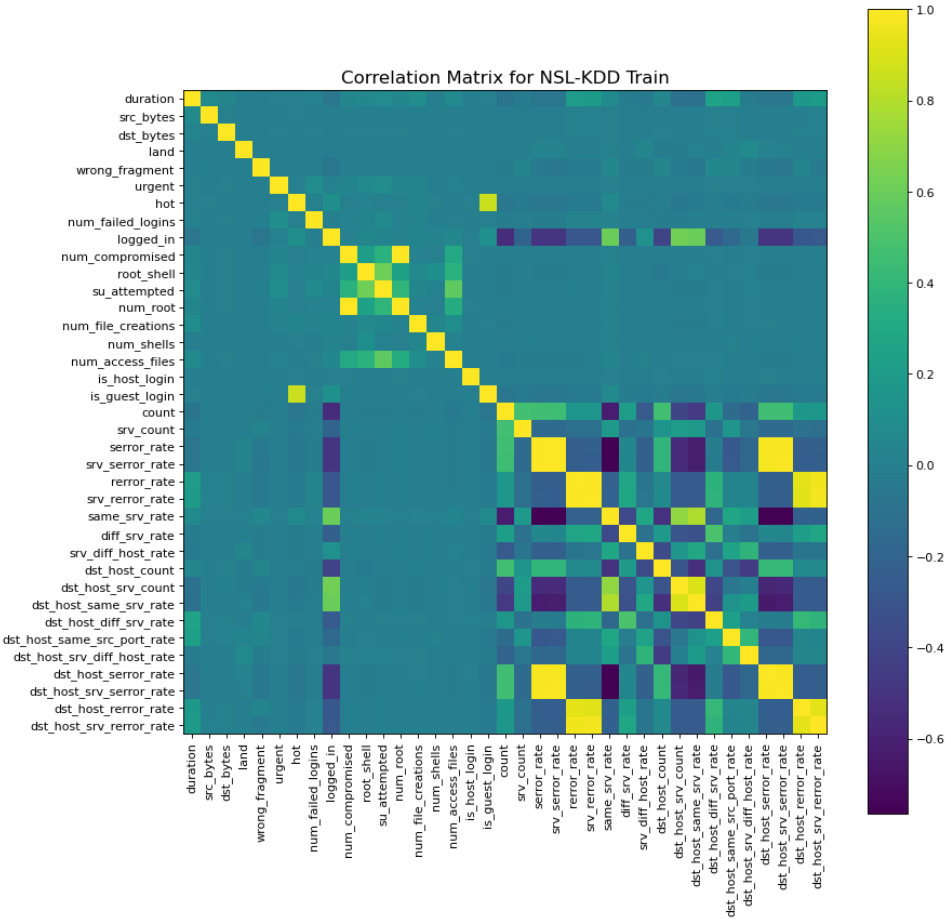

Figure 3: Correlation matrix

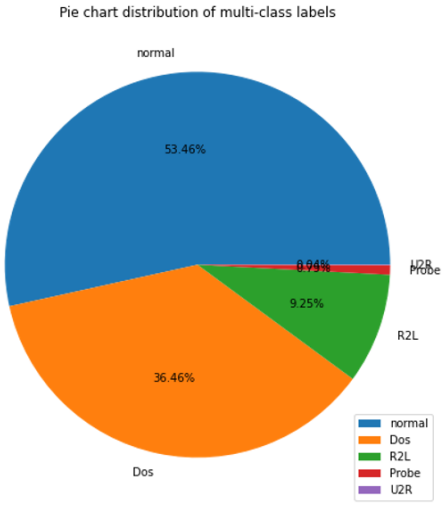

Figure 4: Multiclass classification

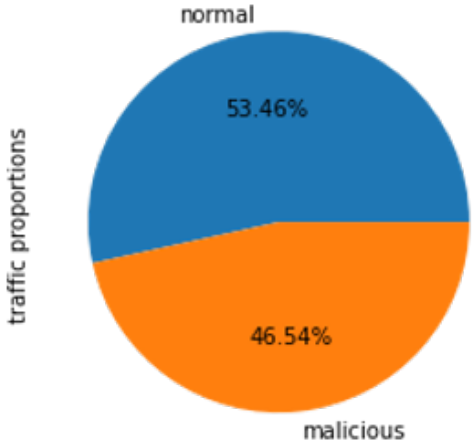

Figure 5: Binary Classification

range. The Min-Max normalization method used in the study rescales the data range to [0,1]. Eq. (i) contains the formula for Min-Max normalization.

$$Min-Max\_Normalization = \frac{(X[i]-X\_min)}{(X\_max-X\_min)} \quad (1)$$

A Min-Max normalization approach was recommended to

```
df.isnull().values.any()

False
```

```
df.isnull().sum()

duration                0
protocol_type           0
service                 0
flag                    0
src_bytes               0
dst_bytes               0
land                    0
wrong_fragment          0
urgent                  0
hot                     0
num_failed_logins       0
logged_in               0
num_compromised         0
same_srv_rate           0
diff_srv_rate           0
srv_diff_host_rate      0
dst_host_count          0
dst_host_srv_count      0
dst_host_same_srv_rate  0
dst_host_diff_srv_rate  0
dst_host_same_src_port_rate  0
dst_host_srv_diff_host_rate  0
dst_host_serror_rate    0
dst_host_srv_serror_rate  0
dst_host_rerror_rate    0
dst_host_srv_rerror_rate  0
subclass                0
difficulty_level        0
dtype: int64
```

Figure 6: Few samples of cleaning data

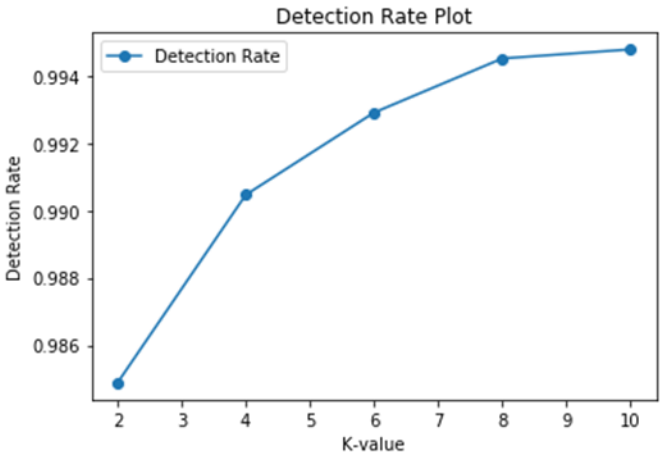

Figure 7: Detection Rate (DR)

handle the varied data scale ranges with the fewest misunderstanding errors. Min-Max scaling can deal with non-Gaussian feature distributions, unlike the signature-based technique in NIDS, which is ideally suited in our NSL-KDD dataset, because anomaly detection applications have no specified distribution to follow. In order to prevent gradients and optimize the loss function from the un-smoothing path to the global minimum, the Min-Max normalization technique is provided.

H. K-Fold Cross Validation

Data must be arranged through stratification so that each fold appropriately represents the entire dataset. The dataset is split into K sets using the stratified K-cross fold validating approach, with the training phase using K-1 folds and the validation using the K^th fold. Until all of the folds have been used to validate the model, this process is repeated. Stratification means that each fold correctly represents the entire dataset, enabling the model to better classify attacks and fine-tune its parameters. Because it performs better and uses fewer computational resources, the K-cross fold approach is favored over alternative validation techniques.

V. RESULTS ANALYSIS

A. Evaluation metrics

Some of the metrics used to evaluate the effectiveness of the proposed model include Accuracy, Detection Rate, False Positive Rate, F1-Score, and ROC-AUC curve. Accuracy and detection rate are the metrics used to assess the model's capacity to predict all classes and attacks. False Positive Rate is a crucial metric that gauges the percentage of typical records classified as attacks, along with detection rate and accuracy. Even if the detection rate and accuracy are strong, the model could not be effective if the False Positive Rate is large. The F1-score offers a more accurate way to assess performance because accuracy and recall by themselves might not yield a complete set of findings. The metrics mentioned above have definitions in Eq. (ii), (iii), (iv), (v) and (vi).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$$\text{Detection rate or Recall} = \frac{TP}{TP+FN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \tag{5}$$

$$F1-\text{ score} = \frac{2*\text{ Precision } * \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{6}$$

where, TP is true positive, TN is true negative, FN is false negative, and FP is false negative. TP indicates the number of attacks correctly classified, and False Positive (FP) indicates the number of normal traffic misclassified as attack. When the criterion is altered, the model's ability to distinguish between various dataset categories is evaluated using the ROC-AUC (Area Under Curve) curve. The whole area under the ROC curve is represented by the AUC, which is a number between 0 and 1. The model performs better at correctly identifying various kinds of data the higher the AUC.

B. Binary Classification

In Binary Classification, the model suggests whether the data is an invasion or belongs into the normal class. We use K-cross validation to improve the accuracy of the binary class classification. The value of K is selected from 2 to 10 by increasing 2 steps. Finally, we take the average for all the k values. The mean detection rate is calculated for NSL-KDD dataset. The accuracy of the binary classification model is 99.5% after 10-flod cross validation. The average accuracy of the model is 99.3%. It is clear from Fig.8 and Fig. 9, the model has a high detection rate and the low false positive rate which are 0.994 and 0.13 respectively.

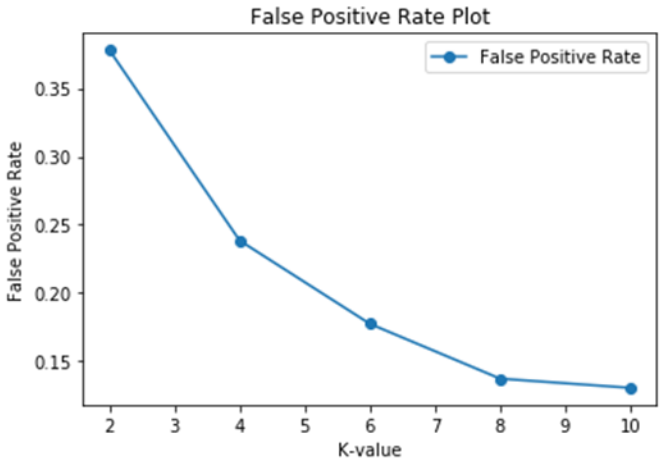

Figure 8: False Positive Rate (FPR)

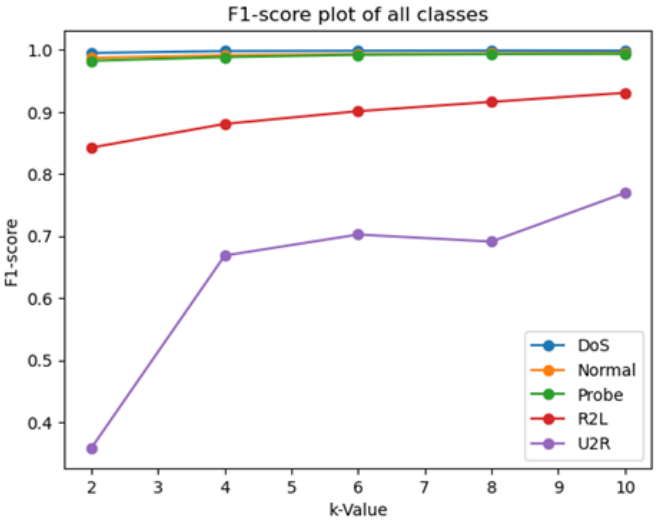

Figure 9: F1-score of multiclass classification

C. Multi-Class Classification

For the NSL-KDD dataset in multiclass, the model achieves an overall precision of 99.25%, with the maximum accuracy of 99.59% for k-value=10. The Detection Rate for k-value = 10 is 99.43%, with the greatest accuracy being 99.59%. The mean FPR is 0.214925. The Table II shows the individuals accuracy, detection rate, and FPR for different K-values. Fig. 10. describes the F1-score for DoS, Normal, Probe, R2L, and U2R. F1-score is higher for DoS, Normal, and Probe. The U2R has lower F1-score compared to others. Confusion matrix for multiclass classification is presented in Fig.11.

D. Comparison of different Machine Learning Algorithms

We develop a network intrusion detection system by using Machine learning (ML). K-Nearest Neighbor (KNN), Decision Tree (DTree), Support Vector Machine (SVM), Artificial Neural Network (ANN), Fast Learning Network, K-Mean Clustering, and Ensemble Methods are the machine learning

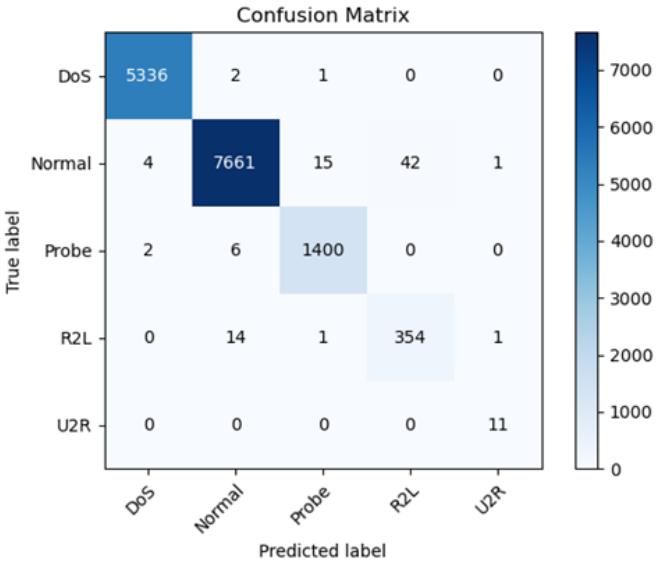

Figure 10: Confusion Matrix for Multiclass classification

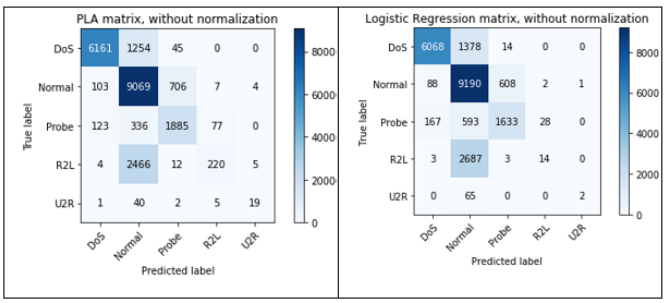

Figure 11: Confusion Matrix for different algorithms

methods most frequently employed for IDS. For the Machine learning part, we select several algorithms. We have done the preprocessing the original NSL-KDD dataset. In order to reduce duplicate data and expedite model training, normalization involves downscaling data into a certain range. Min-max normalization has been used for downscaling the data. Then, we have compared the results of different ML models such as Perceptron Learning Algorithm (PLA), Logistic Regression, Decision Trees, Bagging of PLA, Random Forest, Multi-Layer Perceptron Neural Networks and AdaBoost. We have used NSL-KDD-Dataset. By addressing a number of the KDD Cup'99 dataset's core problems, NSL-KDD has been updated and improved. The performance of different algorithms is shown in Table III. Fig. 12, 13, 14 and 1 represents the confusion matrices for PLA, Logistic Regression, NN, DTree, Voting, Baggong of PLA, AdaBoost, and Random Forest algorithm.

VI. FUTURE WORK

The escalating presence of the Internet of Things (IoT) necessitates the incorporation of IoT devices into Intrusion Detection System (IDS) algorithms and data analysis. Attack-

Table II: Multiclass Classification Results

K-value	Accuracy	Detection Rate	False Positive Rate	Mean F1 Score
2	**0.987086**	**0.985833**	**0.354168**	**0.593**
4	**0.991705**	**0.990493**	**0.237683**	**0.4834**
6	**0.993132**	**0.992553**	**0.186174**	**0.656**
8	**0.994667**	**0.993752**	**0.156211**	**0.567**
10	**0.995960**	**0.994384**	**0.140388**	**0.5287**
Average	**0.992510**	**0.991403**	**0.214925**	**0.4964**

Table III: Comparison of different ML models performance on NSL-KDD dataset

Algorithm Name	Mean accuracy	Mean Precision	Mean Recall	Mean F1 Score
PLA	**0.7697**	**0.7509**	**0.577**	**0.593**
Logistic Regression	**0.7498**	**0.665**	**0.4903**	**0.4834**
NN	**0.7951**	**0.7453**	**0.6229**	**0.656**
DTree	**0.7602**	**0.712**	**0.5576**	**0.567**
Voting	**0.7644**	**0.8035**	**0.5202**	**0.5287**
Bagging of PLA	**0.7538**	**0.6154**	**0.4888**	**0.4964**
AdaBoost	**0.7372**	**0.7534**	**0.4889**	**0.5115**
Random Forest	**0.7598**	**0.7397**	**0.4861**	**0.4988**

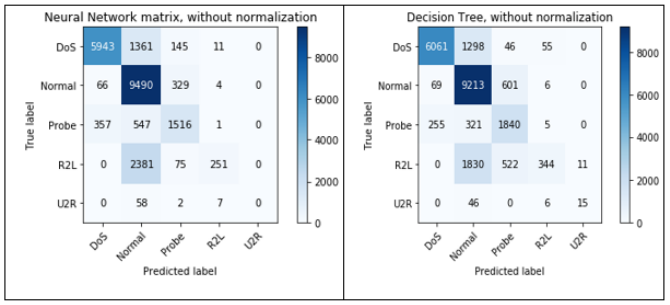

Figure 12: Confusion Matrix for different algorithms

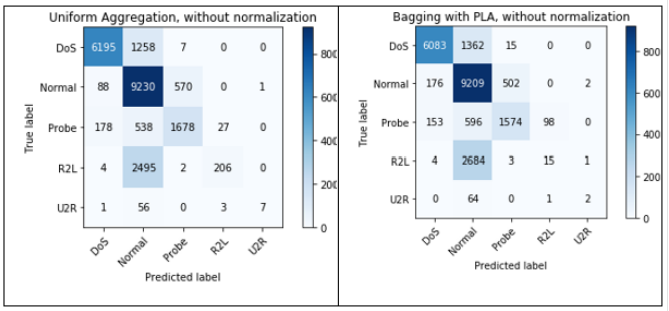

Figure 13: Confusion Matrix for different algorithms

ers exploit smart devices, webcams, and autonomous vehicles to gain unauthorized access to business organizations. To deduce the intrusion path, qualitative information from all IoT devices within the enterprise needs to be abstracted into a

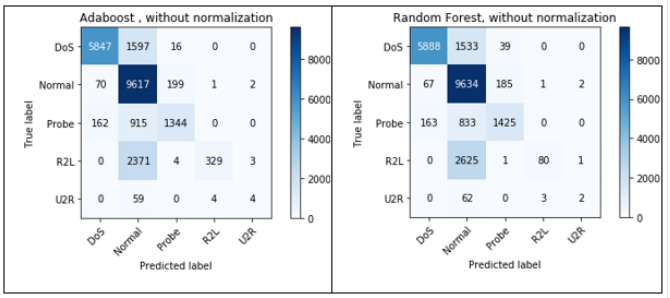

Figure 14: Confusion Matrix for different algorithms

centralized system. As the challenges of intensive care and rapid breach neutralization increase in complexity and cost, security solutions that do not rely solely on attack detection for damage limitation will gain prominence. One approach involves reducing or obfuscating the attack surface, making it challenging to detect target vulnerabilities. Cybersecurity companies are increasingly adopting hacker-style deception strategies. Moving Target Defense encompasses a broad range of protective measures. Future investigations will focus on updating models to address worm attacks and User-to-root (U2R), enabling testing in a honeypot system.

VII. Conclusion

This research introduces a comprehensive model for evaluating network traffic, taking into account various factors such as service type, protocol type, etc. By combining Convolutional Neural Network (CNN) and Bidirectional Long Short-

Term Memory (Bi-LSTM) layers, the model captures spatial information and learns temporal patterns. The proposed model demonstrates exceptional performance for Intrusion Detection Systems (IDS) after training and validation on the NSL-KDD dataset. In terms of results, the accuracy, False Positive Rate (FPR), and Detection Rate (DR) are remarkable for both binary and multiclass classification compared to existing models. However, the U2R class lacks sufficient data compared to other flags, resulting in lower DR and F1 scores compared to the other classes.

REFERENCES

[1] Cashell B, Jackson WD, Jickling M, et al. The economic impact of cyber-attacks. Congressional research service documents, CRS RL32331 (Washington DC). 2004;2.

[2] Gandhi R, Sharma A, Mahoney W, et al. Dimensions of cyber-attacks: Cultural, social, economic, and political. IEEE Technology and Society Magazine. 2011;30(1):28–38.

[3] Agrafiotis I, Nurse JR, Goldsmith M, et al. A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. Journal of Cybersecurity. 2018;4(1):tyy006.

[4] Bendovschi A. Cyber-attacks–trends, patterns and security countermeasures. Procedia Economics and Finance. 2015;28:24–31.

[5] Bala D, Hossain MA, Islam MA, et al. Effective recognition system of american sign language alphabets using machine learning classifiers, ann and cnn. In: 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI); IEEE; 2022. p. 1–6.

[6] Khan SU, Mynuddin M, Ahad DMA, et al. A comparative analysis of deep learning models for power quality disturbance classification. In: 2023 IEEE World AI IoT Congress (AIIoT); IEEE; 2023. p. 0317–0323.

[7] Bala D, Mynuddin M, Hossain MI, et al. A robust plant leaf disease recognition system using convolutional neural networks. In: 2022 International Conference on Engineering and Emerging Technologies (ICEET); IEEE; 2022. p. 1–6.

[8] Bala D, Islam MA, Hossain MI, et al. Automated brain tumor classification system using convolutional neural networks from mri images. In: 2022 International Conference on Engineering and Emerging Technologies (ICEET); IEEE; 2022. p. 1–6.

[9] Khan SU, Mynuddin M, Islam MA, et al. Deep learning based power quality disturbance recognition using residual network in smart grid. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME); IEEE; 2023. p. 1–6.

[10] Bala D, Islam MA, Hossain MA, et al. Efficient epileptic seizure recognition system using the multi-model ensemble method from eeg. In: 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI); IEEE; 2022. p. 1–6.

[11] Lansky J, Ali S, Mohammadi M, et al. Deep learning-based intrusion detection systems: a systematic review. IEEE Access. 2021;9:101574–101599.

[12] Ahmad Z, Shahid Khan A, Wai Shiang C, et al. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. 2021;32(1):e4150.

[13] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997;9(8):1735–1780.

[14] Deshpande P, Sharma SC, Peddoju SK, et al. Hids: A host based intrusion detection system for cloud computing environment. International Journal of System Assurance Engineering and Management. 2018; 9:567–576.

[15] Kumar BS, Raju TCSP, Ratnakar M, et al. Intrusion detection system-types and prevention. International Journal of Computer Science and Information Technologies. 2013;4(1):77–82.

[16] Fernández GC, Xu S. A case study on using deep learning for network intrusion detection. In: MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM); IEEE; 2019. p. 1–6.

[17] Kurundkar G, Naik N, Khamitkar S. Network intrusion detection using snort. International Journal of Engineering Research and Applications. 2012;2(2):1288–1296.

[18] Mynuddin M, Hossain MI, Khan SU, et al. Cyber security system using fuzzy logic. In: 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME); IEEE; 2023. p. 1–6.

[19] Shanmugavadivu R, Nagarajan N. Network intrusion detection system using fuzzy logic. Indian Journal of Computer Science and Engineering (IJCSE). 2011;2(1):101–111.

[20] Benaicha SE, Saoudi L, Guermeche SEB, et al. Intrusion detection system using genetic algorithm. In: 2014 Science and Information Conference; IEEE; 2014. p. 564–568.

[21] Amanoul SV, Abdulazeez AM, Zeebare DQ, et al. Intrusion detection systems based on machine learning algorithms. In: 2021 IEEE international conference on automatic control & intelligent systems (I2CACIS); IEEE; 2021. p. 282–287.

[22] Sstla V, Kolli VK, Voggu LK, et al. Predictive model for network intrusion detection system using deep learning. Revue d'Intelligence Artificielle. 2020;34(3).

[23] NSL-KDD. Nsl-kdd dataset. available: http://nslcsunbca/nsl-kdd/. —;.

[24] Cup K. Kdd cup 1999 dataset. Available: http://KddIcsUciEdu/Databases/Kddcup99. 1999;.