Advanced Intrusion Detection System using Deep Learning

Dr. Sathishkumar P, Abhijith S, Abiwaqas Y

Abstract: *Network Intrusion Detection Systems have become more popular as cloud technologies have become more widely adopted. With ever-increasing network traffic, Network Intrusion Detection (NIDS) is a key component of network security, and given the regularity with which new forms of cyber attacks occur, a highly efficient NIDS is necessary. These intrusion detection systems use either a pattern matching system or a machine learning-based anomaly detection system. Pattern matching techniques have a high False Positive Rate, but AI/ML-based systems forecast the potential of an attack by detecting a metric/feature or a link between a set of metrics/features.*

KNN, SVM, and others are the most frequent; however they work on a limited range of characteristics, have relatively poor accuracy, and have higher False Positive Rates. This research developed a deep learning model that combines the strengths of a Convolutional Neural Network and a Bi-directional LSTM to learn spatial and temporal data characteristics. The model in this research is trained and tested using the publicly available dataset NSL-KDD. The proposed model has a high rate of detection and a low incidence of false positives. Many cutting-edge Network Intrusion Detection solutions that leverage Machine Learning/Deep Learning models outperform the suggested model.

Keywords: Network Intrusion, Cyber Attacks, Unsupervised Learning Techniques, Classification, Decision Trees, Deep Learning, Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, Long Short-Term Memory (LSTM), Bi-Directional LSTM (Bi-LSTM), Ensemble Models.

1. INTRODUCTION:

Since the beginning of widespread Internet use, the frequency of intrusion events has increased tremendously due to the disruptive proliferation of cloud technology. Since a single data centre managed by a technology firm like Microsoft, Amazon, Google, or others hosts a multitude of on-demand servers, platforms, and other services in various range of small, medium, and large businesses, the cost of cyber security and firewalls has increased, integrating a wide range of preventative measures and incident handling mechanisms to protect data and prevent service interruptions. Eavesdropping, network infections, probing assaults, and other intrusions are examples of these incursions.

One of the strategies used in Network Intrusion Detection Systems is Prediction Models based on network time series data. Owing to many data points that vary over time due to unpredictable oscillations, the majority of time-series data exhibits non-linear features. NIDS has also made use of empirical Machine Learning techniques such as k-Nearest Neighbours, Support Vector Machine, l Bayes, and others. These statistical approaches do not take into account data relationships and depend primarily on feature extraction or feature selection, making them unsuitable for real-time use and resulting in even lower Detection Rates.

1.1 AIM AND OBJECTIVE:

Deep learning approaches such as Convolution Neural Networks (CNN), Recurrent Neural Networks, and others are now widely used. Due to their high False Positive Rate, these models are currently being researched for practical use. In this research, the suggested model will be evaluated using two datasets: NSL-KDD and UNSW-NB15. NSL-KDD is an advanced version of the KDD99 dataset, which was first released in 1999. In 2015, the University of New South Wales produced the UNSW-NB15 dataset, which highlighted the shortcomings of the KDD98 and KDD99 data sets, particularly the fact that they do not capture recent low footprint assaults.

This research presents a hierarchical model that combines layers of 1DCNN and Bi-LSTM to function on both of these highly unusual datasets. The CNN layers (basically a sub-category of RNN) are used to learn the spatial/high level characteristics of a dataset, while the Bi-LSTM layers (basically a sub-category of RNN) are used to learn the long time-range temporal characteristics of the data and combine them to anticipate assaults. The predictions are made for Binary Classification, which involves forecasting whether or not an attack will occur, as well as the assault's particular category. In NSL-KDD, the analysis has been done on five classes for multi-category attack prediction: normal, denial of service (DoS), probe (probing attacks), R2L (Root to Local Attacks), and U2R. (User to Root Attack).In UNSW-NB15, ten classes were utilised to forecast multicategory attacks: normal, DoS, Exploits, Generic, Reconnaissance, Worms, Shellcode, Analysis, Backdoor, and Fuzzers

2. INTRUSION DETECTION SYSTEM:

We propose a model in this research that combines a 1-Dimensional Convolutional Neural Network (1-D CNN) with numerous layers of Bi-directional LSTM (Bi-LSTM) . The layers of the suggested model's neural network, as well as the datasets and data pre – processing techniques utilised on these datasets, will be explained in this section.

2.1 **MODEL ARCHITECTURE**:

The suggested model, as shown in the flow chart, consists of a 1-D CNN layer, numerous layers of Bi-LSTM, and Reshape and Batch Normalization layers in between. The goal is to take use of the 1-D CNN layer's parameter sharing, spatial layout, and local perception features. Parameter Sharing provides for a smaller number of parameters and free variables, resulting in faster extracting features and less processing time.Spatial arrangement allows for the organisation of significant paradigm features in a two – dimensional array, allowing for better detection of feature association.

Finally, domestic perception allows for a smaller set of parameters, resulting in a significant reduction in training time. As a result, 1-D CNN enables rapid spatial learning for supplied time-series data. Following the 1-D CNN layer is a Max Pooling layer, which allows for sample-based partitioning of properties in order to identify significant features, resulting in shortened training time and overfitting mitigation. Following Max Pooling, the Batch Normalization layer allows for parameter normalisation between intermediate layers, resulting in faster training durations. After batch normalisation layers, reshape layers modify the results of the preceding layer for the next pair of Bi-LSTM layers.

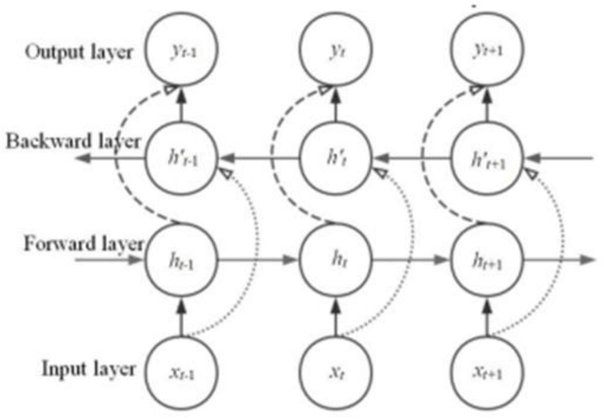


Figure 1: Structure of Bi-LSTM

The convolution layers of Bi-LSTM layers will be used to train from forward as well as backwards time series data, with the hidden layers using two units with the identical outputs and inputs. One system operates with forward time series, while the other works with backward time series. This so-called arrangement is supposed to send future trends to the layers, allowing for faster training time and improved feature engineering, leading to higher accuracy for long-term time-series data.

The model's two Bi-LSTM layers are designed in such a way that the kernel size doubles with each iteration. According to the model's flow chart, the very first Bi-LSTM layer has 64 units, while the following and final Bi-LSTM layer has 128 units. This decision was made to imitate the usage of coarse grain to fine grained learning in order to better comprehend the connection between long-range time-dependent features learned by the initial 1-D CNN Layer, which allows for improved extracting features and quicker training periods.

There is a Max Pooling layer between each Bi-LSTM layer to exclude the weakest relevant attributes and a Batch Normalization layer between each intermediary to normalise the output data of the preceding intermediary layer to improve performance and reduce training durations.

The Densely Integrated layer is next, followed by a Dropout Layer. Despite the fact that the model employs Max Pooling in between each layer, the Dropout Layer is used to account for Over Fitting. The reason for this is that combining CNN and RNN results in a larger likelihood of overfitting and poor performance on the testing set. The model is tested using k-fold cross validation to ensure that this is not the case

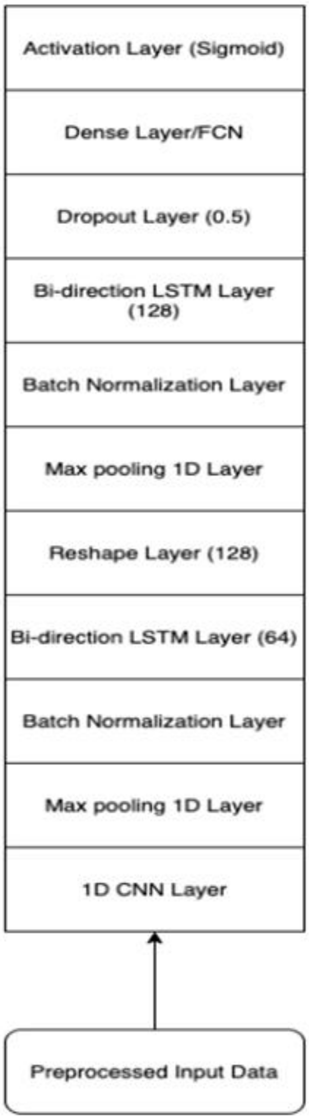

Activation Layer (Sigmoid)
Dense Layer/FCN
Dropout Layer (0.5)
Bi-direction LSTM Layer (128)
Batch Normalization Layer
Max pooling 1D Layer
Reshape Layer (128)
Bi-direction LSTM Layer (64)
Batch Normalization Layer
Max pooling 1D Layer
1D CNN Layer
Preprocessed Input Data

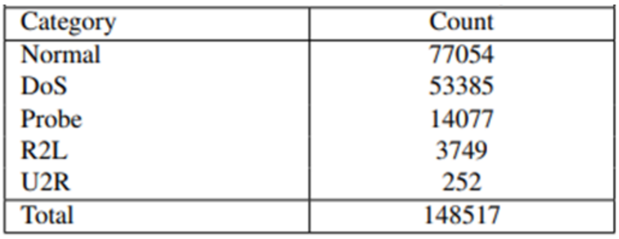
Figure 2: Block Diagram of Model

2.2 DATASET:

2.2.1 NSL-KDD Dataset:

The University of New Brunswick made the NSL-KDD dataset public. The NSL-KDD Database is an upgrade to the KDDCup'99 Dataset, which has intrinsic flaws that have been discovered by numerous analyses.

NSL-KDD is among the most popular datasets for Network Intrusion Detection Systems analysis since it comprises all of the relevant information from the whole KDD Dataset. NSL-KDD differs from its predecessor in several ways, including: The proportion of statistics in the core KDD Dataset is inversely based on number of records picked from each difficulty category.There are four types of attacks in the data set: denial of service (DoS), probe, user to root (U2R), and remote to local (R2L). The feature types in this data set can be broken down into 4 types: 4 Categorical, 6 Binary, 23 Discrete, 10 Continuous.The attributes of a traffic record are classified into four types and reveal data about the contact with the traffic input by the IDS, they are Intrinsic, Content, Host-based, and Time-based.



Category	Count
Normal	77054
DoS	53385
Probe	14077
R2L	3749
U2R	252
Total	148517

Figure 3: Table of NSL-KDD Dataset Attack Categories

2.3 PREPROCESSING:

Data augmentation is commonly performed by Normalization of numerical features and One Hot Encoding of continuous values. However, as previously mentioned, the NSL-KDD Dataset includes a refined number of entries for each attack category.

2.3.1 One Hot Encoding:

The NSL-KDD dataset include categorical characteristics that must be translated to numerical values for our deep learning model to produce accurate predictions. As a result, in the pre-processing section, these columns were transformed to numerical values using the pandas python library's get-dummies function. One-hot encoding was preferred over label encoding because label encoding may yield numerous numbers in the same column, and the model may misinterpret these values as being in a certain order, affecting categorization.

2.3.2 Normalization:

Normalization is the process of downscaling data into a certain range in order to eliminate duplication and speed up model training. The study employs Min-Max Normalization, which rescales the data range to [0,1], the formula for the Min-Max normalization is presented at Eq.1

To address the diverse data scale ranges with the least misinterpretation errors, a Min-Max normalisation strategy was suggested. Because anomaly detection applications have no predefined distribution to follow, Min-Max scaling can deal with non-Gaussian feature distributions, unlike the signature-based technique in NIDS, which is extremely suited in our NSL-KDD dataset.

The Min-Max normalisation strategy is presented to avoid gradients while optimising the loss function from the un-smoothing route to the global minimum.

$$X[i] = \frac{X[i] - X_{min}}{X_{max} - X_{min}}$$

Equation 1: Min-Max Normalization

2.3.3 Stratified K-Fold Cross Validation:

Stratification is the act of organizing data so that each fold accurately represents the whole dataset. The stratified K-cross fold validating approach divides the dataset into K sets, with the training phase on K-1 folds and validation on the Kth fold.

This process is repeated until all of the folds have been utilized to validate the model. Stratification signifies that each fold accurately represents the whole dataset, allowing for parameter fine tuning and improved attack classification by the model. The K-cross fold method is preferred over other validation methods because it works better and takes less computing resource.

3. RESULTS AND DISCUSSION:

3.1 EVALUATION METRICS:

Accuracy (ACC), Detection Rate (DR), False Positive Rate (FPR), F1-Score, and ROC-AUC curve are some of the measures used to assess the efficiency of the suggested model.

The model's ability to forecast all classes and attacks is measured by accuracy and DR. FPR, along with DR and ACC, is a critical indicator that measures the fraction of normal records categorized as attack. The model may not be useful if the FPR is high, even if the DR and ACC are strong. Because accuracy and recall alone may not provide a comprehensive set of results, the F1-score provides a more precise measure of performance. Equations (2), (3),(4) and (5) provide definitions for the metrics described above .

Where TP (True Positive) is the number of attacks correctly classified, TN (True Negative) is the number of normal traffic correctly classified, FN (False Negative) is the number of attacks misclassified as normal traffic and FP (False Positive) is the number of normal traffic misclassified as attack.

Finally, the ROC-AUC curve assesses the model's capacity to differentiate between different dataset categories when the criterion is changed. The AUC (Area Under Curve) is a number that ranges from 0 to 1 and represents the full area beneath the ROC curve. The higher the AUC, the better the model is in properly categorizing distinct kinds.

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN}$$

Equation 2: Accuracy

$$DetectionRate = \frac{TP}{TP + FN}$$

Equation 3: Detection Rate

$$FalsePositiveRate = \frac{FP}{FP + TN}$$

Equation 4: False Positive Rate

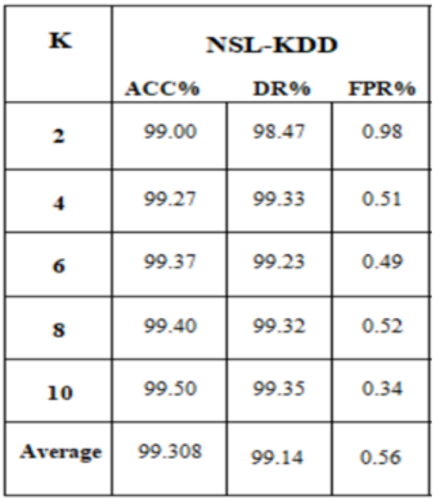
$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Equation 5: F1 Score

3.2 MODEL RESULTS:

3.2.1 Binary Classification:

The model implies whether the data is an invasion or falls to the usual class in Binary Classification. The mean detection rate (DR percent) for NSL-KDD dataset is 99.14 percent, accuracy (ACC) is 99.308 percent, and false positive rate (FPR percent) is 0.56 percent, according to the results of binary classification by the suggested model under distinct Stratified K-Fold Cross Validation for k=2 to 10.



K	NSL-KDD		
	ACC%	DR%	FPR%
2	99.00	98.47	0.98
4	99.27	99.33	0.51
6	99.37	99.23	0.49
8	99.40	99.32	0.52
10	99.50	99.35	0.34
Average	99.308	99.14	0.56

Figure 4: Result of Binary Classification

The model has a high detection rate (DR percent) and a low false positive rate (FPR percent), as shown in Fig.4 and Fig.5 for various k-values. F1-Scores for k ranging from 2 to 10, with 0.9548 being the best F1-Score of binary classification for k=10.

When k is 10, the highest accuracy is 94.21 percent, and the detection rate is 95.92 percent. This is right because as the size of folds increases, the model will have more samples of every attack/normal class to train, and thus will be able to categorize them better.

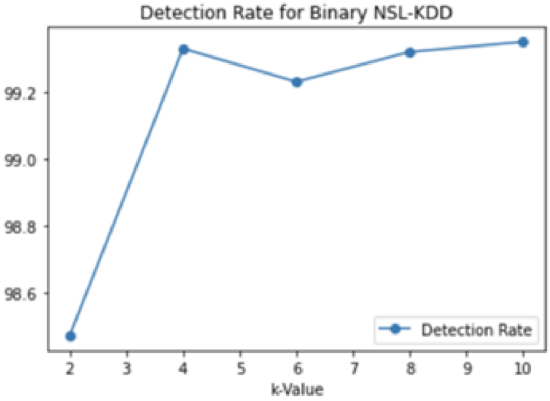


Figure 5: Detection Rate for Binary Classification

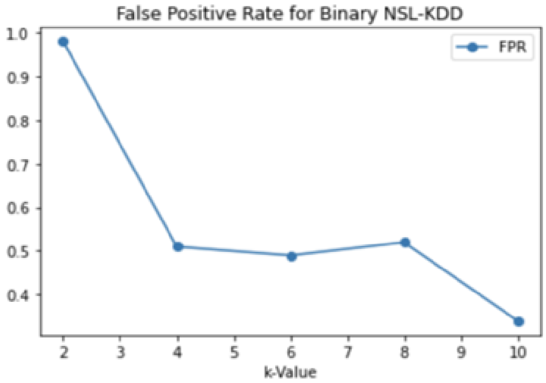
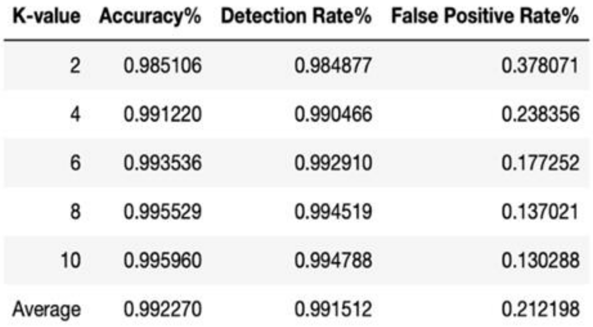


Figure 6: FPR for Binary Classification

3.2.2 Multi-Class Classification:

The model yields an overall precision of 99.22 percent for the NSL-KDD dataset in multiclass, with the greatest accuracy of 99.4 percent for k-value=10. By k-value = 10, the mean Detection Rate is 98.882 percent, with the best performance of 99.13 percent. By k-value=10, the mean FPR percent is 0.0043, with the best rate of 0.0033.

The chart of individual class DR in Fig.11 demonstrates that for k-values larger than 4, the ratio of DR for the U2R category drops, which is owing to the fact that as the k-value grows, the proportion of train samples reduces. Normal, DoS, and Probe are three classes with a significant DR. Figures 7, 8, 9 show the unique values for Accuracy, DR, and FPR, as well as plots.



K-value	Accuracy%	Detection Rate%	False Positive Rate%
2	0.985106	0.984877	0.378071
4	0.991220	0.990466	0.238356
6	0.993536	0.992910	0.177252
8	0.995529	0.994519	0.137021
10	0.995960	0.994788	0.130288
Average	0.992270	0.991512	0.212198

Figure 7: Result of Multiclass Classification

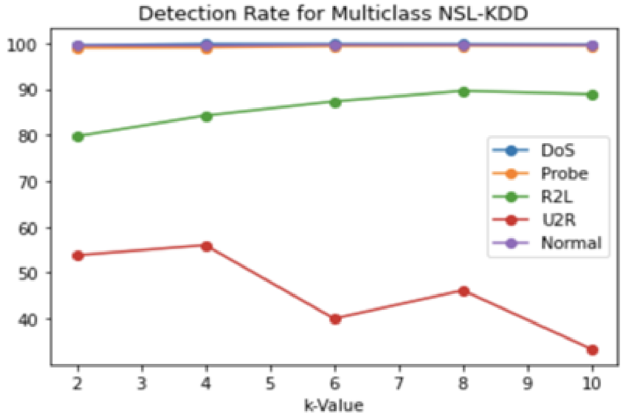


Figure 8: Detection Rate for Multiclass Classification

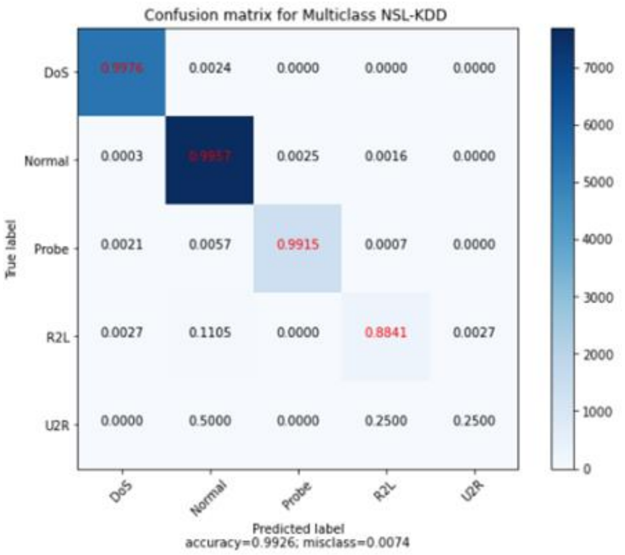


Figure 9: Confusion Matrix for Multiclass classification

The F1-score of multiclass NSL-KDD analysis indicates a gradual increase in values from k-value 2 to 8, with a tiny decrease at k-value=10.As previously stated, the F1-Score, which is 0.9929 when k-value = 8, is a fairly accurate statistic for evaluating a model. Fig.10 depicts the plot.

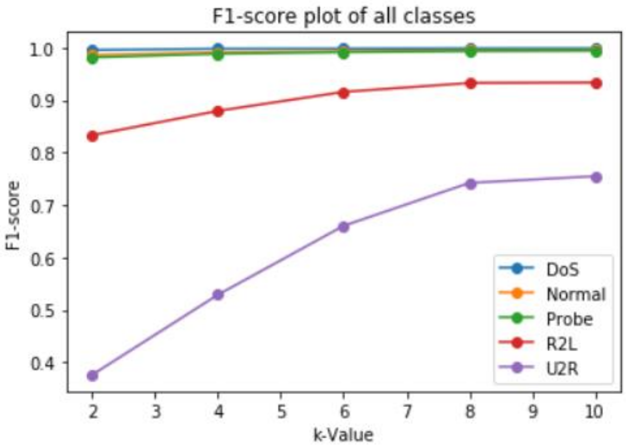


Figure 10: F1 Score for Multiclass Classification

The Area Under Curve (AUC) of all classes ranges 0.94-1.0, and the mean AUC is 0.971, as shown in Fig.11. This indicates that the model is extremely efficient and reliable in discriminating across different dataset classifications.

The AUC for multiclass NSL-KDD is 1.00 across all classes, as shown in Fig.17. As previously stated, AUC is a metric of a model's capacity to discriminate between various classes within a dataset, and it accomplishes well on the NSL-KDD dataset.

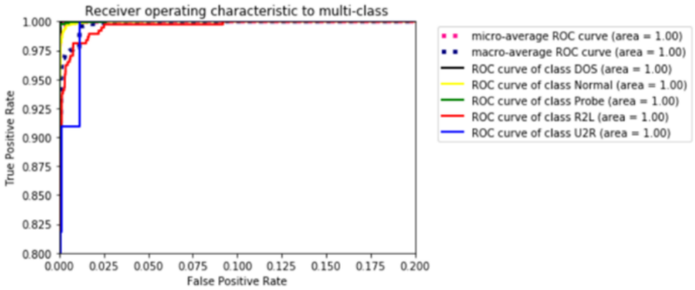


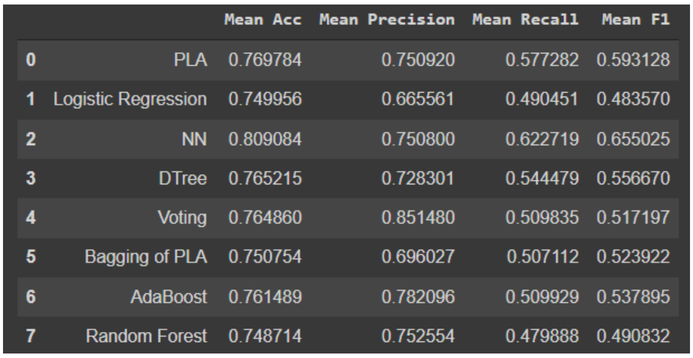
Figure 11: ROC Curve

3.3 COMPARISON:

The suggested model is compared to existing models such as Logistic Regression, Perceptron Learning Algorithm (PLA), Bagging of PLA, Decision Trees, Random Forest, AdaBoost, Multi-Layer Perceptron Neural Networks. In contrast, the suggested model clearly outperforms the competition on all parameters, particularly Detection Rate.

In relation, the Logistic Regression is less, but when other criteria are considered, including DR and precision, the suggested model emerges as the best option.

The Voting Classifier model pops out to be highest performing classifiers among other comparative models with Mean Precision of 85.14 in overall performance, although it produces somewhat poorer outcomes.



		Mean Acc	Mean Precision	Mean Recall	Mean F1
0	PLA	0.769784	0.750920	0.577282	0.593128
1	Logistic Regression	0.749956	0.665561	0.490451	0.483570
2	NN	0.809084	0.750800	0.622719	0.655025
3	DTree	0.765215	0.728301	0.544479	0.556670
4	Voting	0.764860	0.851480	0.509835	0.517197
5	Bagging of PLA	0.750754	0.696027	0.507112	0.523922
6	AdaBoost	0.761489	0.782096	0.509929	0.537895
7	Random Forest	0.748714	0.752554	0.479888	0.490832

Figure 12: Comparison of Existing Models

3.4 FUTURE SCOPE:

The expanding IoT must be factored into intrusion detection system, algorithms, and data analysis. Attackers can get access to organizations using webcams, automobiles, and smart device. Qualitative methodologies from all IoT devices in the company must be abstracted into a single location in order to infer the intrusion trail.

As monitoring and promptly neutralizing breaches becomes more arduous and expensive, solutions which do not rely on detecting an assault to restrict harm to a corporation will be included to the security stack. One way is to decrease or obscure the attack interface itself, making it impossible to find target weaknesses.

Hacker-style deception methods will be increasingly used by cyber security firms. Moving Target Defense is a broad category that encompasses such preventative techniques (MTD).

4. CONCLUSION:

This study provides a model for analyzing network traffic that takes into account a variety of characteristics such as protocol type, service type, and so on. In order to adjust for unbalanced datasets, oversampling was used. Learning of temporal and spatial information was made possible by combining CNN and Bi-directional LSTM layers. After training and validation on the NSL-KDD dataset, the proposed model produces impressive tangible results for Intrusion Detection systems. As a result of the analysis, the need to improve the model for U2R and Worms attacks will be examined in the future to enable for experimentation in a honeypot system.

As far as the results are concerned, the Accuracy Rate, Detection Rate, False Positive Rate for both binary and Multiclass classification are exceptional as evident from the comparison with existing models. However, the data insufficiency is minor setback, because the data points of U2R are insignificant relative to other attack flags. Hence the DR and F1 score were started off with diminishing points comparatively to other classes

5. REFERENCES:

1. Chao Liang, Bharanidharan Shanmugam, Sami Azam, MirjamJonkman, Friso De Boer, GanthanNarayansamy, “Intrusion Detection System for Internet of Things based on a Machine Learning approach”, International Conference on Vision Towards Emerging Trends in Communication and Networking, IEEE, 2019.

2. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J.C. “An intrusion detection and prevention system in cloud computing: a systematic review”. J. Netw. Comput. Appl. Vol. 36, pp. 25–41, 2018.

3. Y. Farhaoui and A. Asimi, “Performance assessment of tools of the intrusion detection/prevention systems,” International Journal of Computer Science and Information Security, vol. 10, issue 1, pp. 1-7, 2020

4. Ma T et al, “A hybrid methodologies for intrusion detection based deep neural network with support vector machine and clustering technique”, International conference on frontier computing. Springer, 2019.

5. Manjula C. Belavagi and BalachandraMuniyal, Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection, Procedia Computer Science, Elsevier, 2017