IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A DEEP LEARNING MODEL TO CLASSIFY ATTACKS IN A NETWORK

[1]Manas Dwivedi, [2]Dr. Urjita Thakar, [3]Dr. Vandan Tewari

[1]Research Scholar, [2]Professor, [3]Associate Professor

[1]Computer Engineering,

[1]Shri Govindram Seksaria Institute of Technology and Science, Indore, India

Abstract: As network services become more widely utilized, security becomes one of the network's primary and most pressing challenges. Several computers connecting to the network serve critical roles in business and other applications that deliver network services. Consequently, we must look for the most effective measures to protect the system, even though several studies employ deep learning (DL)-based intrusion detection algorithms to identify infiltration. Changes in network traffic may result in reduced accuracy of deep learning-based models regarding network assaults. Deep learning has a plethora of strategies for network attacks. The primary goal of this work is to compare and assess the performance of deep learning algorithms on the 1-D data set. The suggested application uses the gated recurrent units(GRU)1-Dimensional Convolutional Neural Network(1-DCNN) hybrid model to identify network attack rates. The preprocessed dataset is trained and tested with the models to provide notable results that increase prediction accuracy. The experiment was conducted using the KDD99 dataset. The deep learning-based hybrid model could reach an accuracy of about 97 percent. On this standard dataset, the GRU-1DCNN model now outperforms the previous model.

Index Terms **- Network Traffic, Attacks, Deep Learning, GRU, 1DCNN, KDD99 dataset.**

I. INTRODUCTION

Network attacks are becoming more common and diversified as internet technology has progressed [1]. Cyber-attacks may lead to data breaches, resulting in significant financial losses. Hackers have cost Facebook millions of dollars in data breaches [2][3]. An attack may be an effort to breach a system's security measures, get unauthorized access to sensitive data, or even destroy the system. More and more network attack activities have been considered a severe danger to network security, particularly wireless communication system security, owing to the openness of wireless channels, which have been expanding alongside these technologies. In the current era of machine learning, users must have secure wireless communication solutions to safeguard their networks, computers, and data against malicious intrusion in order to avoid data loss. Flooding, distributed denial of service, anomalous packet assault, and spoofing are only a few of the many forms of cyber-attacks. Researchers have presented numerous approaches to counter these attack vectors in cyber security. One of the best options is attack detection, which provides a comprehensive and adaptable security system to watch, stop, and even counteract assaults. In particular, attack detection would gather data by keeping tabs on the network, the system's condition, the behavior of system users, and the way they make use of the system, allowing it to identify malicious or unauthorized activity [4] automatically. This alone is insufficient to identify advanced persistent threats or unknown assaults. Using intelligent procedures to recognize and categorize the many different kinds of unknown assaults is necessary. The purpose of this study is to categorize assaults such as DoS (Denial of Service), Probe, R2L (Remote to Local), and U2R (User to Root), all of which create intrusions in networks. A benchmark dataset known as NSL-KDD (Network Security Laboratory-Knowledge Discovery as well as Data) is then used to determine and analyze an incident's root cause. Machine learning and deep learning are used to classify the attack in NSL-KDD dataset[5].

As cyber attacks become more complex and frequent, machine learning (ML) approaches are becoming popular in their detection. Subject matter experts are often required to identify relevant characteristics from available data to identify harmful network traffic when using machine learning in a network setting. Therefore, pre-processing (of feature engineering) is necessary to collect the data required for the deep learning model based on features retrieved from network traffic. As stated before [4] the representation of data packets may have an effect on data-loading (DL) models. There could be a problem with the pre-processing method or with data loss that occurs at that stage. Misleading feature representations can also lead to false positives, where malicious samples are incorrectly labeled as benign [7].

Researchers [8][9] also found that detection systems based on carefully selected and designed characteristics may be fooled in constrained contexts like network communications as well as hostile AI systems. Most of the time, the outcomes are the features that are included in DL models. In contrast, DL models rely not on the raw data but on the precision of a different suite of tools [10, 11].

A variety of datasets can be detected more accurately using existing network IDS work. However, there aren't any publicly available intrusion detection labeled datasets because of concerns about confidentiality and security. Because of this issue, DARPA's KDDCup [12]project was the first to gather IDS datasets. KDDCup was designed by Lincoln Lab using tcpdump to replicate traffic in US Air Force facilities using traffic generated by a closed network with human injection assaults. As the previous version of NSL-KDD had no way to ensure that the simulated dataset faithfully reproduced real network traffic [13], and as there were so many duplicate records, especially for attacks, NSL-KDD was developed to address these issues. Despite the huge increase in redundancy, the new dataset is nevertheless challenged for its depiction of real traffic. It is frequently cited as the most widely used benchmark dataset for NIDS studies. In addition to NSL-KDD [14] and Kyoto, other network IDS datasets exist, such as WSN-DS and CICIDS[15]. Researchers from the University of New South Wales in Australia generated the UNSW-NB15 dataset, which is the most current to be made publicly available [16][17] [18]. Over 2.5 million records have been included in the UNSW-NB15 dataset, which includes more than 40 attributes that mimic real network traffic, including nine contemporary attack methods. The architects offer condensed versions of the testing and training datasets typically used in academic studies. Consequently, some sources have inverted the partitioning of test and training datasets, using the smaller dataset for training, resulting in worse classification accuracy because of limited instances for specific attack types. As a result of the data cleansing, editing, reduction, and wrangling procedures, only limited deep learning pre-processing is required for the datasets.

The following is a breakdown of the remaining sections: Section 2 presents related work on intrusion detection and class imbalance research leveraging deep learning. Methods, as well as a conceptual framework, are presented in Section 3 of the final system. Section 4 elaborates on the research's results and conclusions. The KDD99 benchmark public datasets associated with evaluation measures are detailed in Section 4. Section 5 concludes this study report with a look at what's to come in the future.

II. RELATED WORK

Since the birth of the internet, many studies have been conducted to better understand how to keep networks safe. Infectious disease syndrome is a topic with a lot of published research. ' An intrusion detection system (IDS) is a tool that can identify potential threats and block them before they even happen. There was less processing work for the classifier to do. For NIDS, L. Chen et al. [19] introduce a brand-new system that uses convolutional neural networks. They build detection algorithms based on deep learning using the extracted features and actual network traffic. In-depth studies can be conducted because scientists use widely-used benchmark datasets. The results indicate that our approach is effective, and that the model trained with raw traffic is more accurate than the model trained with extracted features.

Potential network threats are exposed in a novel framework presented by A. Dawoud et al. [20] that uses unsupervised deep learning techniques to detect anomalies. Our research explores the feasibility of applying deep learning to the problem of anomaly detection by using RBM as generative energy-based models as well as Autoencoders as non-probabilistic techniques. This article presents the results of an extensive study into unsupervised deep learning techniques. The simulation tests show a detection performance of at least 99 percent, a significant improvement over the related work.

Using a slightly skewed version of the same dataset as CIC-IDS2017 and CSE-CIC-IDS2018, D. Raju et al. [21] evaluated the performance of three distinct deep learning models. Standard metrics were used to assess the model's accuracy; these metrics were used for the FCL, the Seq2Seq LSTM, and the Autoencoder. To evaluate the efficacy of the interventions, researchers introduce new evaluation measures (the Modified Rankin Coefficient and Cohen's Kappa Coefficient). Because the well-known earlier did not produce any results that could be considered conclusive, this is the case. The experimental results confirmed that the proposed evaluation measures can generate more precise assessments of the DL models' stability.

T-IDS was developed by M. S. Koli and M. K. Chavan [22] using a randomised data partitioned learning model (RDPLM) that is sensitive to the features set as well as the feature selection method, a simplified sub-spacing, and a number of randomised meta-learning strategies. Our model achieves a 99.984% accuracy rate on the popular botnet dataset in just 21.38 seconds of training time. Several distinct machine learning methods have been found to be useful, including deep neural networks, features that reduce errors while performing the tree detection job's sequential minimum optimization tasks, and models that create random Trees.

With their [23] proposal, H. Dhillon and A. Haque show how to build an IDS for a network using a new method that uses multiple deep learning approaches (NIDS). To further improve our proposed scheme and make it much more successful in real-world scenarios, they may use deep transfer learning to import the model's acquired knowledge from one domain to another; this, however, calls for voluminous amounts of data and computational power. We improved classification times while achieving a 98.30% source domain classification performance and a 98.43% target domain classification accuracy on the UNSW -15 dataset.

Automatically detecting and classifying encrypted communications that used a lightweight deep learning framework is described by Y. Zeng, H. Gu et al. [24]. (DFR). As a result of using deep learning, DFR can pick up new tricks simply by observing traffic patterns, without any intervention from humans or collection of private information. Consequently, the methods are put to the test on two open- source datasets, where they are compared to state-of-the-art alternatives. The approach outperforms the sophisticated algorithms while requiring significantly less storage space, with an average of 13.49 percent for encrypted traffic classification as well as 12.15 percent for the IDS F1 score.

Across the board, the detection performance that these methods provided improved after adopting any of the ones we suggested. Most of them were tested on publicly available datasets, such as KDD99 and NSL-KDD; however, results may change

when applied to different settings. Based on our review of the relevant literature, we concluded that a mixture of more than two deep learning techniques may improve the accuracy with which anomalies in network traffic can be detected.

III. RESEARCH METHODOLOGY

In today's more complex and severe computer network security threats, researchers are using various machine learning techniques to defend the data and reputation of their clients. Deep learning is known as one of the fascinating methods lately seen widespread use in network attacks to enhance the performance of these systems in terms of safeguarding computer networks, including hosts. In this study, we used the KDD99 dataset, which is accessible to the general public online. Following this, we removed null and missing values from the dataset using a standard scale tool. We also performed an outlier check and extracted features from the data. The dataset was then separated into a training component and a testing half, with an 80:20 ratio reflecting this division. Last but not least, we bring in the deep learning-based GRU and 1DCNN models, which improve the overall performance of the IDS system in terms of accuracy, precision, recall f1-score, and ROC curve.

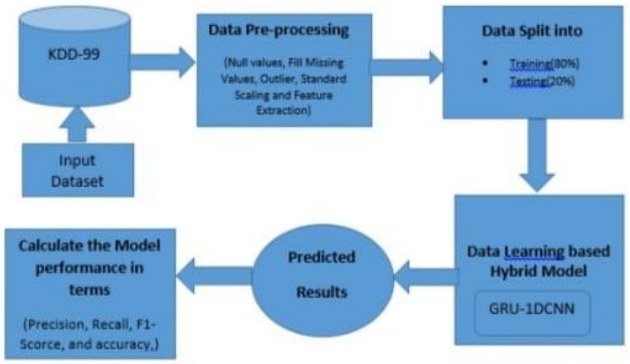

Figure 1: Block Diagram o proposed methodology

The above figure 1 shows the flow of the proposed methodology. The figure shows that the process starts with the data collection, preprocessing, data splitting and proposed hybrid model. All these steps are described below very straightforward manner.

A. Data collection (KDD99)

We utilized the KDD99 dataset in this study. This dataset provides information on intrusion websites. The KDD99 is a tool that many researchers engaged in IDS research utilize. Without the labels, the dataset has 41 characteristics that fit into one of five categories: normal, denial-of-service (DoS), probe, remote-to-local (R2L), or user-to-root (U2R). KDD99 (ten percent variant) contains 494,021 and 311,029 records in the train and test set, correspondingly. Both KDD99 training and test sets include classes that are not balanced. DoS class has the most records, with the Normal class coming second. Furthermore, the number of R2L records in the testing set is much higher. It was revealed that this collection of records had a high number of duplicated records.

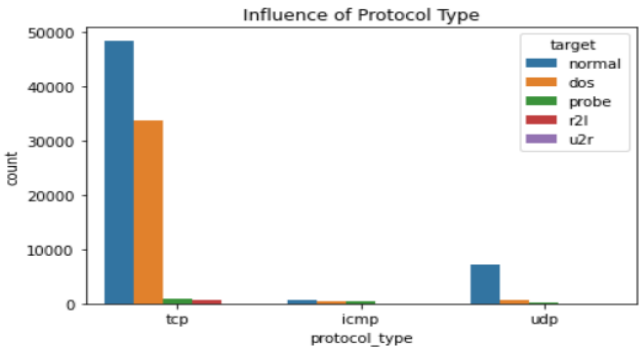

Figure 2: Influence of protocol type

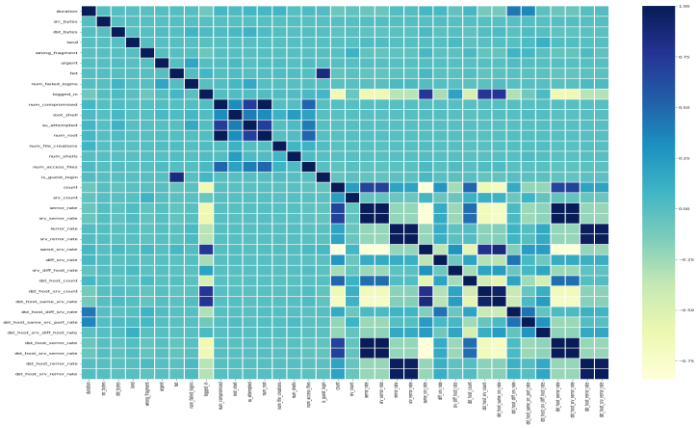

Figure 3: Correlation between whole data

Figure 3 shows the Correlation between data. Correlation is the statistical term for the connection between two variables. A correlation can be positive or negative, indicating that as one variable's value rises, the values of the other variables also drop.

B. Data Preprocessing

After collecting the dataset, Data preprocessing techniques were applied. The data that were obtained are then preprocessed using a few different methods. Cleansing the data entails developing analytical and visual methods that apply to several domains. The data collected from the actual world need to be preprocessed to eliminate noise, fill in gaps with new values, and transform the information into the right format. Some preprocessing techniques are as follows:

StanderScaler: StandardScaler first eliminates the mean before scaling each feature or variable according to its unit variance. This procedure is executed in a decentralized manner based on the feature. Because StandardScaler calculates each feature's empirical means and standard deviations, it is susceptible to being influenced by outliers (provided that they are present in the dataset).

Feature Extraction: The process of converting raw data into numerical features that may be processed while still maintaining the integrity of the data contained in the original data set is referred to as feature extraction. Compared to applying machine learning immediately to the raw data produces superior outcomes.

A. Dataset Splitting

There is a distinction made between the training data and the testing data. Eighty percent of the features are incorporated into the machine learning model throughout the training process. The machine learning model is tested with the remaining twenty percent of the information.

B. Deep Learning Models

Deep learning [25] is one of the most recent developments in the field of ML. In the context of artificial neural networks, it is a branch of machine learning. We can handle a huge number of objects to be taught if we use the deep learning method in the application field. The procedure is carried out on millions of data points. Learning characteristics from the data is what deep learning entails. The system's performance may suffer if a huge volume of data is accessible. Deep learning is a learning process ideally suited for gaining more precision in performance. We accomplished this by using a hybrid deep learning model consisting of the GRU and 1DCNN models. Both models will be discussed briefly below.

A. Gated Recurrent Units(GRU)

A recurrent neural network (RNN) is a type of neural network that takes as its input sequence data, processes that data in a recursive fashion along the evolutionary direction of the sequence, and links together its nodes in a chain. The LSTM network is a more advanced variant of the RNN network that can deal with the gradient disappearing and the gradient explosion problems. The GRU network is an improved version of the LSTM that requires less training time. Gated recurrent units (GRUs) were first introduced by [27], and they are another [26] popular type of recurrent network. Control and management of information flow between NN cells is performed by GRUs using gating strategies. Comparable to a long short-term memory unit (LSTM), the GRU lacks an output gate and thus has fewer parameters. Alternately, it is equipped with a reset gate and an update gate, as shown in Figure 4. As a result, the primary distinction between a GRU and an LSTM is that the former only has two gates (the reset and update gates) and the latter has three (namely input, output, and forget gates). Because of how it's built, the GRU can efficiently and flexibly extract dependencies from massive data sequences. Unlike some other methods, this one doesn't require discarding data collected from earlier in the process. As a result, GRU is a slightly more efficient variant that, in many cases, achieves the same or better performance as its alternatives while also being significantly faster to calculate. While it is true that GRUs perform better on some smaller and less frequent datasets, both variants of RNN have shown to be effective when it comes to delivering the result.

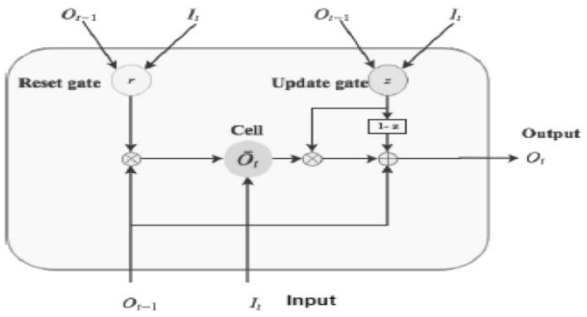

Figure 4: The reset and update gates make up the basic structure of a gated recurrent unit (GRU) cell.

B. 1D-CNN Model Architecture

Due to its one-dimensional nature, CNN is ideally suited for processing signals of this type. Because the 1D-CNN can only take in one dimension of data, its convolution kernel is also flat. Each CL and PL produces a feature vector of a single dimension as its output. In Figure 5[28], we can see the structure of the refined 1D-CNN network. Optimised 1D-CNNs consist of an input layer, a hidden layer (including a CL and a PL), and a fully connected layer (FCL). As part of the input layer, raw Traffic data are segmented into their corresponding time steps. The feature extraction layer consists of the CL and the PL. Features are

extracted from the Traffic data using the input layer's Traffic data and the convolution layer's multiple convolution kernels. Multiple feature vectors are generated as a consequence. Using the pooling operator can reduce the dimensionality of the eigenvectors, which in turn improves the robustness of the nonlinear features. Nonlinear features of input signals can be extracted in a hierarchical fashion using alternating convolution pooling layers. These capabilities have wide-ranging potential uses. There are multiple interconnected layers that make up the categorization layer; the first of these is what 'flattens' the characteristics. It means that all of the feature vectors are joined together, tails to heads, to produce a single-dimensional vector. In the fully connected layer, the total number of neurons is the same as in the other networks[29]. CNN's output is made up of both the full connection output and the final PL output. The next step involves classifying the faults using a Soft-max classifier.

The primary objective of this study was to develop a hybrid neural network using a combination of GRU and LSTM. Current neural networks, such as Deep ANN based on TensorFlow, can train and detect the classes, but they lose the previously trained information when creating continuous incoming data, rendering them unable to recognize just previous attacks. To remedy this, a GRU and LSTM hybrid model is recommended.

Figure 5: 1D-CNN Architecture

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 93, 32)	128
max_pooling1d (MaxPooling1D)	(None, 23, 32)	0
gru (GRU)	(None, 32)	6336
layer_normalization (LayerNormalization)	(None, 32)	64
flatten (Flatten)	(None, 32)	0
dense (Dense)	(None, 16)	528
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 5)	85

Total params: 7,141
Trainable params: 7,141
Non-trainable params: 0

Figure 6: Hybrid model summary

The above figure 6 shows the model summary of the proposed model. The summary shows several layers, output shape and parameters. The figure total and trainable parameters are the same, which is 7,141, and non-triable parameters are 0.

IV. RESULTS AND DISCUSSION

The results of these experiments of suggested deep learning models were presented in this part. According to the simulation findings, the presented is the most effective solution. Both models have good accuracies, according to the findings of the evaluation. This section illustrates the research results using the Jupiter notebook and the python programming language. In this section, we'll look through the findings in further detail.

There are several labels in the dataset used in this experiment. When seeking to classify an attack, accuracy is a useful statistic to use. A model's prediction accuracy is measured using Precision, Recall, and F1-Score as indicated in Equations 1 to 4.

A confusion matrix shows how well an algorithm performs visually. When a classifier has been trained, it is tested on real data. Using a confusion matrix, it is feasible to visualize the predicted probability. It is possible to measure accuracy, precision, and recall by applying the formulae in the following paragraphs. And each of the many sorts of assaults is represented in this way.

- **True Positive:** The instances that have a favorable outcome are reliably predicted to have a positive outcome by the classifiers.
- **False Positive:** The instances considered are negative. However, the classifier declares them positive by errors.
- **True Negative:** The instances being looked at are negative and belong to the negative group.
- **False Negative:** The instances that were being appeared had good outcomes, but the classifier put them in the wrong category.

A. Simulation Results

In this part, we evaluated the experimental criteria for putting the IDS model into action and applying the deep learning approach to detecting network attacks. In this study, we implemented a hybrid GRU-1DCNN model, and the findings are shown and discussed more below.

After applying preprocessing tasks such as label encoding, we obtained three additional columns based on the binarization of bales. As a result, we did not lose any information of this kind, despite the fact that there were no null or nan values displayed in the data. This was the case even though the preprocessing tasks yielded 125973 rows as well as 42 columns with respect to the data.

Table I: Parameter Setting

Model	Sequential
Hybrid Model	GRU-1DCNN
Pooling	Max pool
Activation	Relu
	Softmax
Optimizer	Adam
Loss	Categorical cross entropy
Epochs	30
Batch size	512
Train data	94479, 93, 1
Test data	31494, 93, 1
Train, test ratio	80:20
Metrics	Accuracy

Table II: Comparison between the base and proposed model using performance parameters

Model	Existing Model	Proposed Hybrid Model GRU-1DCNN
Accuracy	0.82	0.9705
Precision	0.83	0.9711
Recall	0.82	0.9701
F1-Score	0.81	0.9706

The performance of the existing model is compared to the performance of the new model that is being proposed in table II which can be found above. The metrics that are compared include accuracy, precision, recall, as well as f1-score. The accuracy of the minimum viable model is rated at 82%, its precision is rated at 83%, its recall rating is rated at 82%, and its f1 score is 81%. On the other hand, the proposed hybrid model was able to achieve an accuracy of 97.05%, precision of 98.11%, f1-score of 97.01%, and recall of 97.06%. As a consequence of this, it is abundantly clear that the GRU-1DCNN hybrid model that I presented performs significantly better than the initial GRU model.

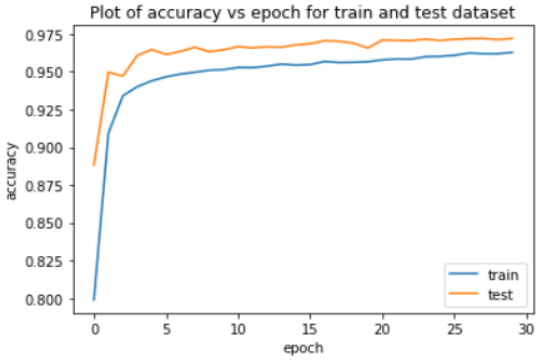

Figure 7: Plot of GRU-1DCNN for train and test dataset Accuracy

The GRU-1DCNN model's suggested testing as well as training accuracy is shown in Figure 7. The accuracy value is represented by the Y-axis, while the X-axis represents the number of epochs (30). The model's training accuracy shows how well it can distinguish between the two inputs during training just on training sample.

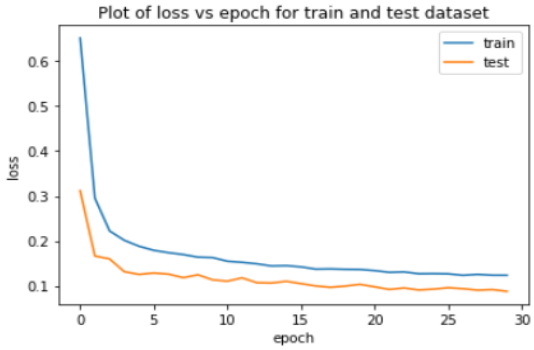

Figure 8: Plot of GRU-1DCNN for train and test dataset loss

Figure 8 depicts the proposed module's GRU-1DCNN training and testing data losses. The model was trained across ten epochs. The train loss value indicates how well or badly a model performed after every optimization iteration. A test loss measurement is often used to assess the algorithm's performance readably.

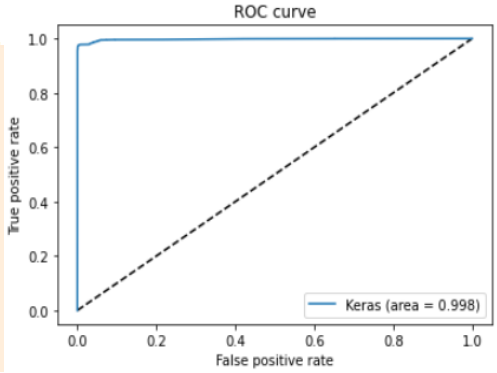

Figure 9: ROC curve of GRU-1DCNN hybrid model

The ROC curve for the suggested hybrid mode can be seen above in figure number 9. ROC is a graph that displays the accuracy of the model's overall analysis method. It is also often referred to as the ROC curve. ROC is 99% for the hybrid model GRU-1DCNN

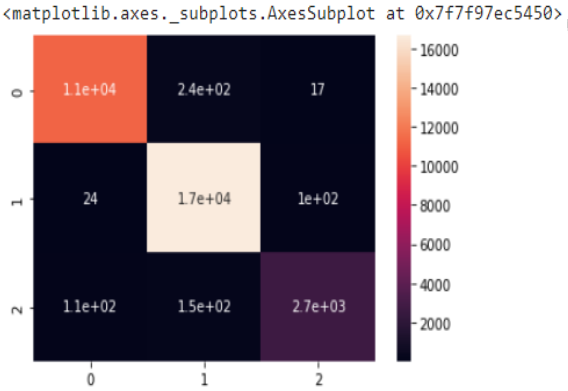

Figure 10: Confusion matrix of hybrid model

The confusion matrix for the hybrid DL model can be seen shown in Figure 10. The genuine positive values are shown by the diagonal value of the confusion matrix, which is 11e+04 for the 0 labels, 17e+04 for the one label, then 2.7e+03 for the two labels, correspondingly

V. CONCLUSION AND FUTURE WORK

Global network traffic is increasing at an alarming rate. The contemporary architecture renders traditional network intrusion detection algorithms ineffective for classifying a massive network data stream. This research aims to offer a contemporary process for developing a classification of network attacks utilizing different DL approaches. This paper classifies the network attacks by adding important elements and prevalent cyber security vulnerabilities and exposures into the most recent modeled network traffic dataset. Compared to the corresponding DL-based network attacks, the suggested deep learning classification framework combined with the hyper parameter tuning technique exhibited considerable improvements to the hybrid model.

According to the models, our suggested technique for network traffic detection utilizing network attacks achieved an overall accuracy of 97.05 percent. While comparison with some other hybrid models that achieved results nearby, our results like LSTM-RNN[30] model get 96.51% accuracy and Long Short Term Memory (LSTM) [31] model yielded a 79.2% accuracy, and CNN-LSTM [32] obtained 94% accuracy. But over method shows higher performance than baseline methods and other research work.

Although the suggested models produced impressive outcomes, we agree that there is potential for enhancement, primarily via feature reduction strategies. Future work asks for transfer learning using relevant current datasets to serve as a baseline for model classification improvements with the UNSW-NB15 dataset & to increase our models' capacity to withstand zero-day assaults. In addition to transfer learning, bootstrapping approaches will be examined to provide a well-balanced dataset for training a multiclass classification model. Adaptive and resilient NIDS that can identify common vulnerabilities, exposures, and zero-day network behavioral aspects will be developed using deep learning anomaly detection methods. Minimizing the possibility of a conflict, In the future, we will also use some other hybrid deep learning models like CNN-RNN, DNN-RNN, DNN-LSTM and ANN etc.

REFERENCES

[1] K. N. Mukherjee, B.; Heberlein, L.T.; Levitt, "Network intrusion detection," *IEEE*, no. 8, pp. 26–41, 1994.

[2] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, and J. Xu, "Malicious network traffic detection based on deep neural networks and association analysis," *Sensors (Switzerland)*, 2020, doi: 10.3390/s20051452.

[3] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2020.3048198.

[4] Y. Wu, D. Wei, and J. Feng, "Network attacks detection methods based on deep learning techniques: A survey," *Security and Communication Networks*. 2020. doi: 10.1155/2020/8872923.

[5] P. R. Mol, "Classification of Network Intrusion Attacks Using Machine Learning and Deep Learning ^ 1," vol. 25, no. 2, pp. 1927–1943, 2021.

[6] P. Maxwell, E. Alhajjar, and N. D. Bastian, "Intelligent Feature Engineering for Cybersecurity," 2019. doi: 10.1109/BigData47090.2019.9006122.

[7] L. Tong, B. Li, N. Zhang, C. Hajaj, C. Xiao, and Y. Vorobeychik, "Improving the robustness of ML classifiers against realizable evasion attacks using conserved features," 2019.

[8] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in Network Intrusion Detection Systems," *Expert Syst. Appl.*, 2021, doi: 10.1016/j.eswa.2021.115782.

[9] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-Bayes classifier," 2009. doi: 10.1145/1655008.1655013.

[10] Q. Ma, C. Sun, B. Cui, and X. Jin, "A novel model for anomaly detection in network traffic based on kernel support vector machine," *Comput. Secure.*, 2021, doi: 10.1016/j.cose.2021.102215.

[11] L. M. Ibrahim, D. B. Taha, and M. S. Mahmod, "A comparison study for intrusion database (KDD99, NSL-KDD) based on self-organization map (SOM) artificial neural network," *J. Eng. Sci. Technol.*, 2013.

[12] G. Creech, "Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks," 2014.

[13] J. Mchugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory," *ACM Trans. Inf. Syst. Secure.*, 2000, doi: 10.1145/382912.382923.

[14] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secure.*, 2012, doi: 10.1016/j.cose.2011.12.012.

[15] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," 2018. doi: 10.5220/0006639801080116.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015. doi: 10.1109/MilCIS.2015.7348942.

[17] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for Network Intrusion Detection Systems," 2017. doi: 10.1109/BADGERS.2015.14.

[18] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," 2017. doi: 10.1109/ISIE.2017.8001537.

[19] L. Chen, X. Kuang, A. Xu, S. Suo, and Y. Yang, "A Novel Network Intrusion Detection System Based on CNN," 2020. doi: 10.1109/CBD51900.2020.00051.

[20] A. Dawoud, O. A. Sianaki, S. Shahristani, and C. Raun, "Internet of Things Intrusion Detection: A Deep Learning Approach," 2020. doi: 10.1109/SSCI47803.2020.9308293.

[21] D. Raju, S. Sawai, S. Gavel, and A. S. Raghuvanshi, "DEVELOPMENT OF ANOMALY-BASED INTRUSION DETECTION SCHEME USING DEEP LEARNING IN DATA NETWORK," 2021. doi: 10.1109/ICCCNT51525.2021.9579510.

[22] M. S. Koli and M. K. Chavan, "An advanced method for detection of botnet traffic using intrusion detection system," 2017. doi: 10.1109/ICICCT.2017.7975246.

[23] H. Dhillon and A. Haque, "Towards network traffic monitoring using deep transfer learning," 2020. doi: 10.1109/TrustCom50675.2020.00144.

[24] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-Full-Range: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2908225.

[25] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*. 2021. doi: 10.1109/ACCESS.2021.3097247.

[26] C. Xu, J. Shen, X. Du, and F. Zhang, "An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2867564.

[27] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014. doi: 10.3115/v1/d14-1179.

[28] M. De Lucia, P. E. Maxwell, N. D. Bastian, A. Swami, B. Jalaian, and N. Leslie, "Machine learning raw network traffic detection," no. April, p. 24, 2021, doi: 10.1117/12.2586114.

[29] Y. Li, L. Zou, L. Jiang, and X. Zhou, "Fault Diagnosis of Rotating Machinery Based on Combination of Deep Belief Network and One-dimensional Convolutional Neural Network," *IEEE Access*, 2019, doi: 10.1109/ACCESS.2019.2953490.

[30] S. B. Naik and R. Sasikala, "Machine Learning Algorithms to Classify Network Attacks," no. 6, pp. 38–44, 2022, doi: 10.46647/ijetms.2022.v06i01.006.

[31] B. Lee, S. Amaresh, C. Green, D. Engels, and D. W. Engels, "Comparative Study of Deep Learning Models for Network Intrusion Detection," *Other Comput. Eng. Commons, Other Comput. Sci. SMU Data Sci. Rev.*, vol. 1, no. 1, 2018, [Online]. Available: https://scholar.smu.edu/datasciencereviewAvailableat:https://scholar.smu.edu/datasciencereview/vol1/iss1/8

[32] M. Y. Alzahrani and A. M. Bamhdi, "Hybrid deep-learning model to detect botnet attacks over internet of things environments," *Soft Comput.*, 2022, doi: 10.1007/s00500-022-06750-4.