International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I6P114

Volume 72 Issue 6, 136-145, June 2024

Original Article

Multilayer Perceptron and Auto-Encoder Based Intrusion Detection System

G. Sahitya[1], C. Kaushik[2], K. Karthik[3], V. Anemesh Chandra[4], C. Janaki Ram[5]

[1,2,3,4,5]*Department of ECE, VNRVJIET, Telangana, India.*

[3]*Corresponding Author : karthikkolli17@gmail.com*

Received: 22 March 2024 Revised: 16 May 2024 Accepted: 08 June 2024 Published: 29 June 2024

Abstract *- In recent times, with the rapid growth of the internet, every day a lot of data is being generated. Along with this growth, there are advancements in cybersecurity attacks and the technologies through which security attacks are taking place; as a result, there is an increase in security and privacy concerns for users. An Intrusion Detection System can be developed to address this issue. The Intrusion Detection System can be made by evaluating several advanced computational deep learning and machine learning models for intrusion detection using datasets containing features extracted from network traffic; in this paper, using Deep Learning (DL) Techniques such as Multi-layer Perceptron (MLP) and Auto encoders (AE). These classifiers are being trained and evaluated on the dataset, and their performance metrics, including accuracy and classification reports, are being computed by using only the features which are necessary and useful. The Intrusion Detection Model, through these classifiers, improves the accuracy of intrusion detection.*

Keywords *- Auto-encoders (AE), Cybersecurity, Deep Learning Techniques, Multi-layer Perceptron (MLP), Network Intrusion Detection System (NIDS).*

1. Introduction

In the contemporary digital era, the exponential increase in data has led to a corresponding rise in cyber threats. These threats pose significant risks to the integrity, confidentiality, security, and availability of data and network systems. Network Intrusion Detection Systems (NIDS) have emerged as a critical line of defence, monitoring network traffic for suspicious or anomalous activities.

NIDS can be broadly classified into two categories: signature-based and anomaly-based. Signature-based NIDS scrutinize network traffic against a repository of recognized attack patterns or signatures, triggering an alert upon detection of correspondence. However, this approach is limited by its dependence on known attack patterns, rendering it ineffective against novel threats. On the other hand, anomaly-based NIDS learn the patterns in network traffic and detect any deviation from the norm as a potential intrusion, offering a promising solution to the limitations of signature-based systems.

Despite the potential of anomaly-based NIDS, their effectiveness is often hampered by the complexity and diversity of network traffic patterns. This is where the application of Deep Learning (DL) techniques, an extension of Machine Learning that uses multi-layered artificial neural networks, comes into play. Recent advancements in DL techniques have paved the way for their application in network intrusion detection, aiming to enhance performance.

In this context, the work introduces a novel approach to anomaly-based NIDS using a combination of Multilayer Perceptron and Auto-encoder-based models. This approach aims to classify network behaviour as ‘Normal’ or ‘Anomaly’, with anomalies further categorized into four distinct types: Denial of Service (DoS), Probe, Root to Local (R2L), and User to Root (U2R). While existing research has explored the use of DL techniques in network intrusion detection, the application of a combined Multilayer Perceptron and Auto-encoder model represents a novel contribution to the field. This work aims to bridge the gap in the literature by providing a comprehensive evaluation of this approach, comparing its performance with existing methods, and exploring its potential for enhancing the effectiveness of anomaly-based NIDS.

2. Related Work

The document delves into the use of deep learning, particularly Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), for network intrusion detection to tackle the escalating intricacy of cyber-security threats. It offers an exhaustive review of employing deep neural networks in intrusion detection, assessing their effectiveness through trials with the NSL-KDD dataset. Findings reveal the dominance of CNN over RNN variants, attaining an accuracy exceeding 97%. Despite encouraging outcomes, issues such as extended training durations due to resource limitations are acknowledged. Future studies aim to

enhance the performance of DL models, considering computational needs and hyperparameters. In essence, the document emphasizes the potential of DL in enhancing NIDS while also underlining the necessity for further exploration to surmount implementation obstacles and boost efficiency. [1]

The document outlines a study on intrusion detection employing DL, specifically focusing on recurrent neural networks (RNN-IDS). It addresses the challenge of accurately identifying various network attacks by proposing an RNN-based classifier and assessing its performance in binary and multiclass classification scenarios. Through experimentation and analysis of factors like neuron count and learning rates, the document demonstrates the RNN-IDS model's superiority over traditional ML methods, showcasing its potential to enhance intrusion detection accuracy. Emphasizing the escalating severity of security threats and the limitations of conventional methodologies, the study underscores the relevance of DL, particularly RNNs, in effectively addressing the challenges of intrusion detection. Detailed descriptions of the dataset, preprocessing steps, methodology, and evaluation metrics are provided, alongside discussions on the implications of the research findings and future research directions, indicating the promise of the RNN-IDS model in advancing intrusion detection technology. [2]

The paper introduces an innovative Intrusion Detection System (IDS) for wireless networks that merges Convolutional Neural Network (CNN) classification with a newly proposed feature selection method called Conditional Random Field and Linear Correlation Coefficient-Based Feature Selection (CRF-LCFS). The goal of this system is to enhance the security of data communication by accurately identifying intruders. The CRF-LCFS algorithm, using the KDD'99 Cup dataset, selects the most suitable attributes and groups features based on correlation coefficient values using Euclidean distance measurement and CRF. The system's effectiveness is demonstrated by experimental results, achieving a remarkable detection accuracy of 98.8% when combined with CNN, surpassing other IDSs in detecting various types of attacks. The system stands out for its high detection accuracy and low false alarm rates, indicating its strong potential for improving network security. In summary, the combination of CRF-LCFS and CNN offers a sophisticated strategy for detecting intruders in wireless networks, with promising enhancements in detection accuracy and false alarm rates. Future research will focus on refining the algorithm and exploring its applicability in different network environments. [3]

The research introduces a supervised learning model that enhances network intrusion detection classifiers using Generative Adversarial Networks (GANs). These GANs address the limitations of traditional machine learning techniques in handling complex intrusion detection tasks. The proposed framework improves classifier generalization and effectiveness by continuously generating labeled samples for adversarial training. Experimental results demonstrate that the improved classifier, trained with the ID-GAN framework, achieves better accuracy, precision, recall, and F1 score. Also, the study investigates the impact of training parameters, such as prior training and noise distribution selection, and discusses data augmentation effects on training time. Overall, the ID-GAN framework significantly enhances classifier performance and generalization. It also provides insightful information about how GANs might be used to improve multi-class classification in intrusion detection and suggests promising directions for further research in the field. [4]

In their research titled "Performance evaluation of DL techniques for DoS attacks detection in wireless sensor network," Salmi and Ough Dir assess the efficiency of DL methods in identifying DoS attacks in Wireless Sensor Networks (WSNs). They develop and implement several DL-based IDSs using the WSN-DS dataset, which encompasses four distinct types of DoS attacks: Blackhole, Grayhole, Scheduling, and Flooding attacks. The models are evaluated based on standard comparison metrics such as accuracy, precision, recall, and F1-score and utilize a range of DL algorithms, including Dense Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and hybrid CNN and RNN architectures. The paper highlights the increasing demand for WSNs across various industries for monitoring physical and environmental conditions but also underscores their susceptibility to security threats, particularly denial-of-service attacks, due to their resource constraints. The project's objective is to experiment with DL-based algorithms to devise a lightweight, accurate, and efficient technique for detecting DoS attacks in WSNs. As per the methodology, which involves data pre-processing, normalization, transformation, and splitting, the CNN model yields the highest accuracy and F1 score. [5]

The document presents the application of a Big Data-based Deep Learning System (BDHDLS) to intrusion detection, aiming to address the challenges posed by complex network attacks. BDHDLS is proposed as a solution to enhance the performance of ML-based IDS. The system utilizes hierarchical DL to learn distinctive data distribution patterns for specific attack families, employing big data techniques for feature selection and clustering, as well as parallel strategies to reduce model construction time. Incorporating both behavioral and content features, BDHDLS improves the robustness of learning algorithms, as demonstrated through experimental evaluations using datasets like ISCX2012, CICIDS2017, and DARPA1998. Results show BDHDLS outperforming traditional models in terms of detection rate and accuracy, with statistically significant performance gains. The document concludes by discussing potential future research directions, such as advanced decision fusion algorithms and resource optimization, emphasizing BDHDLS's potential to enhance intrusion detection through DL and big data methodologies. [6]

The document explores the development of a deep neural network-based multi-class classification IDS, focusing on network security and cyber threat detection. Proposing a DL model with multiple stacked fully connected layers, the study implements a flow-based anomaly detection IDS for multi-class classification, utilizing the known dataset, i.e. CICIDS2017, for training and evaluation. The IDS landscape, including signature-based, anomaly-based, and hybrid systems, is examined, emphasizing flow feature-based classification for network traffic analysis. Discussions on ML and DL techniques in cybersecurity underscore the importance of comprehensive datasets for experimentation and model evaluation. Detailed analysis of the CICIDS2017 dataset, data cleansing, transformation, and feature reduction techniques, along with deep neural network architecture and model evaluation based on 10-fold cross-validation, are provided. The document concludes with future research directions, suggesting further feature reduction, dataset extension, and exploration of different model architectures for improved performance. Overall, the study offers valuable insights into IDS development, emphasizing the significance of network security, data quality, and model efficacy in cyber threat detection. [7]

The paper, authored by Zeeshan Ahmad, Johari Abdullah, and others, explores challenges in NIDS amidst increasing network size and data. It stresses the significance of IDS in network security and aims to clarify IDS concepts and provide an organization based on ML and DL techniques. Methodologies for selecting recent articles on ML- and DL-based NIDS are discussed, followed by detailed classifications of IDS and methodologies employed for NIDS, focusing on ML and DL algorithms. Overall, the paper offers a comprehensive overview of recent trends in designing efficient network-based IDS. [8]

The document explores utilizing DL architectures for adaptive NIDS to address evolving security threats, highlighting the importance of network security and proposing Deep Neural Networks (DNNs) for attack detection and classification. It showcases the effectiveness of the proposed model using the UNSW-NB15 dataset, discussing IDS, detection techniques, and challenges. By employing Convolutional Neural Networks (CNN) and regularized multi-layer perceptrons, it achieves significant performance improvements, concluding with avenues for future research in feature reduction, transfer learning, bootstrapping techniques, and DL anomaly detection models to enhance cybersecurity architectures. [9] [15]

The article from Yarmouk University proposes an ML-based model for intrusion detection, addressing network security challenges. Using KNIME and the CICIDS2017 dataset, it evaluates SVM, RProp, and decision tree classifiers, achieving accuracy rates from 90.59% to 98.6%. By emphasizing the significance of IDS in modern networks and detailing the research methodology, including data preprocessing and feature selection, the study offers a practical approach to enhancing intrusion detection. The results demonstrate promising accuracy metrics, suggesting potential benefits for network security enhancement through ML. [10]

Table 1. Features of NSL-KDD dataset

SI No.	Attribute	Type	SI No.	Attribute	Format
1	duration	Continuous	22	is_guest_login	Symbolic
2	protocol_type	Symbolic	23	count	Continuous
3	service	Symbolic	24	srv_count	Continuous
4	flag	Symbolic	25	serror_rate	Continuous
5	src_bytes	Continuous	26	srv_serror_rate	Continuous
6	dst_bytes	Continuous	27	rerror_rate	Continuous
7	land	Symbolic	28	srv_rerror_rate	Continuous
8	wrong_fragment	Continuous	29	same_srv_rate	Continuous
9	urgent	Continuous	30	diff_srv_rate	Continuous
10	hot	Continuous	31	srv_diff_host_rate	Continuous
11	num_failed_logins	Continuous	32	dst_host_count	Continuous
12	logged_in	Symbolic	33	dst_host_same_srv_count	Continuous
13	num_compromised	Continuous	34	dst_host_same_srv_rate	Continuous
14	root_shell	Continuous	35	dst_host_diff_srv_rate	Continuous
15	su_attempted	Continuous	36	dst_host_same_src_port_rate	Continuous
16	num_root	Continuous	37	dst_host_srv_diff_host_rate	Continuous
17	num_file_creations	Continuous	38	dst_host_serror_rate	Continuous
18	num_shells	Continuous	39	dst_host_srv_serror_rate	Continuous
19	num_access_files	Continuous	40	dst_host_rerror_rate	Continuous
20	num_outbound_cmds	Continuous	41	dst_host_srv_rerror_rate	Continuous
21	is_host_login	Symbolic			

3. Dataset

3.1. Features and Types of Attacks

The NSL-KDD dataset is a dataset which also contains the records from the KDD dataset [11], [13]. The dataset is divided into a training dataset and a testing dataset consisting of 125,973 and 22,544 samples, respectively.

Table 1 shows all the information about the samples. It tells about the attribute and the type. Each sample has 41 features. The test dataset consists of the classified attacks:

- DoS attack: Attacks involve sending multiple packet requests to a network, which overloads the network and makes the network unresponsive, denying services to users.
- Probe attack: Attacks involve scanning a machine or network to obtain information about its vulnerabilities to exploit them and attack the network.
- U2R attack: Attacks occur when a non-privileged user gains access to a user on a specific computer or system.
- R2L attack: Attackers send packets to an unauthorised machine over a network, targeting to exploit vulnerabilities and gain unauthorised access.

Table 2. Types of attacks

Attack	Attack Type
DoS Attack	apache2, back, land, neptune, mailbomb, pod, processtable, smurf, teardrop, udpstrom, worm
Probe Attack	ipsweep, mscan, nmap, portsweep, saint, satan
U2R Attack	buffer_overflow, loadmodule, perl, ps, rootkit, sqlattack, xterm
R2L Attack	ftp_write, guess_passwd, httptunnel, imap, multihop, named, phf, sendmail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, xlock, xsnoop

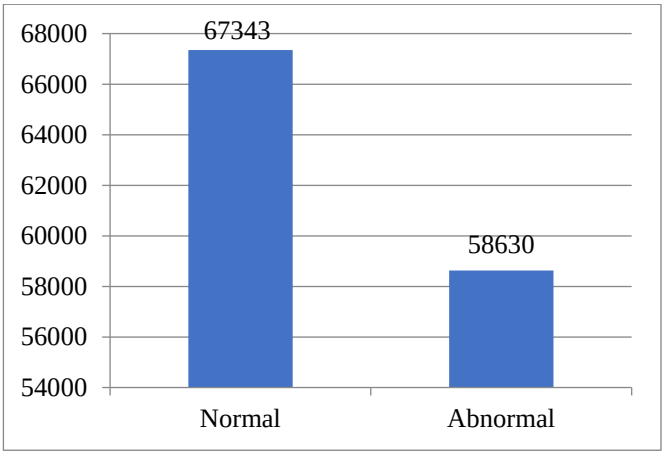

Fig. 1 Binary classification of attacks

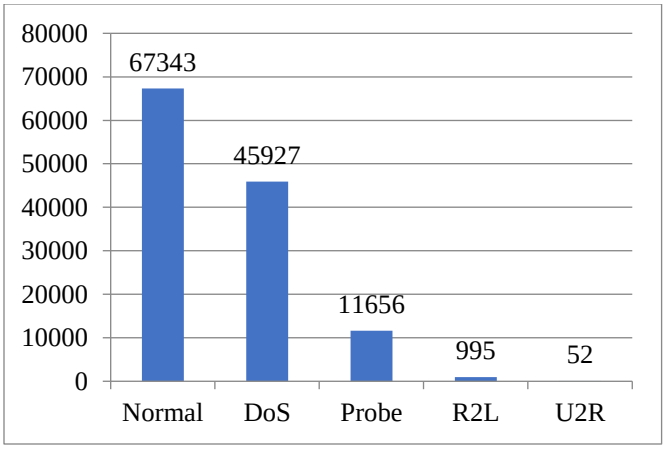

Fig. 2 Multi-classification of attacks

Table 2 describes the 39 types of attacks, which are categorised under 4 attacks, namely “DoS Attack”, “Probe Attack”, “U2R attack”, and “R2L attack” [12].

3.2. Data Pre-processing

Data pre-processing is one of the main steps under data mining. Data pre-processing involves transforming, cleaning, and making it ready for the process of analysis. The importance of data pre-processing is to provide quality data for the analysis. Data cleaning includes fixing inconsistencies, correcting errors, and filling missing values. Data Transformation is transforming data into a format which is best for the analysis. Data Reduction is another step under data pre-processing, which involves reducing the data size to preserve quality data. Another step under data pre-processing is the Data Integration process, combining data from various sources to from a dataset for analysis and prediction.

3.3. Data Normalization

Data Normalisation is the process of transforming features in a way that they are in the same scale, most common scales are 0 and 1 or 1 and -1. Through the normalisation of data, the performance and stability of the model increases. There are various methods for Data normalization.

There are various methods like min-max normalization, z-score normalization, and decimal scaling, which are used regularly. In this project, implemented normalization was implemented using the Standard Scaler. The Standard Scaler normalizes features by subtracting the mean and then scaling to unit variance.

$$z = (x - u)/s$$

Where “z” is standardised value, “x” is featuring value, “u” is mean value of the feature values, and “s” is the standard deviation of the feature values.

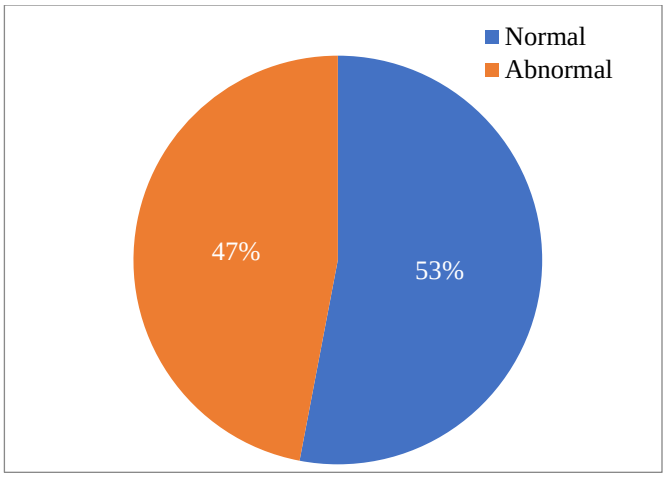

Fig. 3 Pie chart distribution of attacks under the binary-class classification

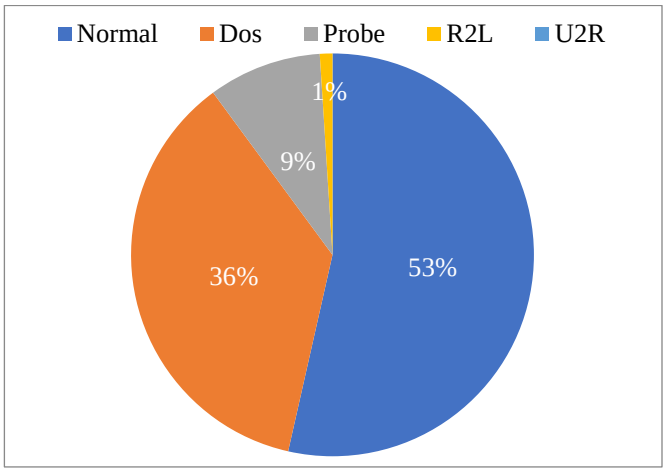

Fig. 4 Pie chart distribution of types of attacks under multi-class classification

3.4. Data Encoding

The process of transforming data from one form to a necessary format is called data encoding. One Hot Encoding is converting categorical data variables. Each categorical value is changed to a new categorical value and allocated a binary value of 1 and 0.

3.5. Binary Classification

Binary classification is a category of ML task where the data is classified into two distinct groups, such as spam or not spam, positive or negative, etc. [14]. It is a simple decision between two outcomes. Binary classification is used in many algorithms such as regression, support vector machine, etc. In this experiment, there are the classes "normal" and "abnormal". Figure 1 can observe the number of attack counts with the respective label, and Figure 3 shows the distribution of the attack labels.

3.6. Multi-Class Classification

Multi-class classification refers to an ML process where data is categorized into multiple classes beyond just two [14]. It is a more complex decision that involves assigning data to one of the multiple possible classes. Multi-class classification can be performed using various strategies, such as one-vs-rest and one-vs-one, which involves splitting the multi-class dataset into multiple binary datasets and training a binary classifier on each. For the experiment, the chosen ones are "normal", "DoS", "probe", "U2R", and "R2L" as the classes to be classified. Figure 2 shows the number of attacks and the labels that correspond to them. The distribution of assault labels is shown in Figure 4.

In feature engineering, the features that are chosen are through the Pearson coefficient.

Pearson Coefficient - The coefficient can only be between -1 to 1, which helps in the Detection for the project which is used. The features that are selected for the project are chosen by a factor of equal or greater than 0.8.

4. Methodology

4.1. Workflow

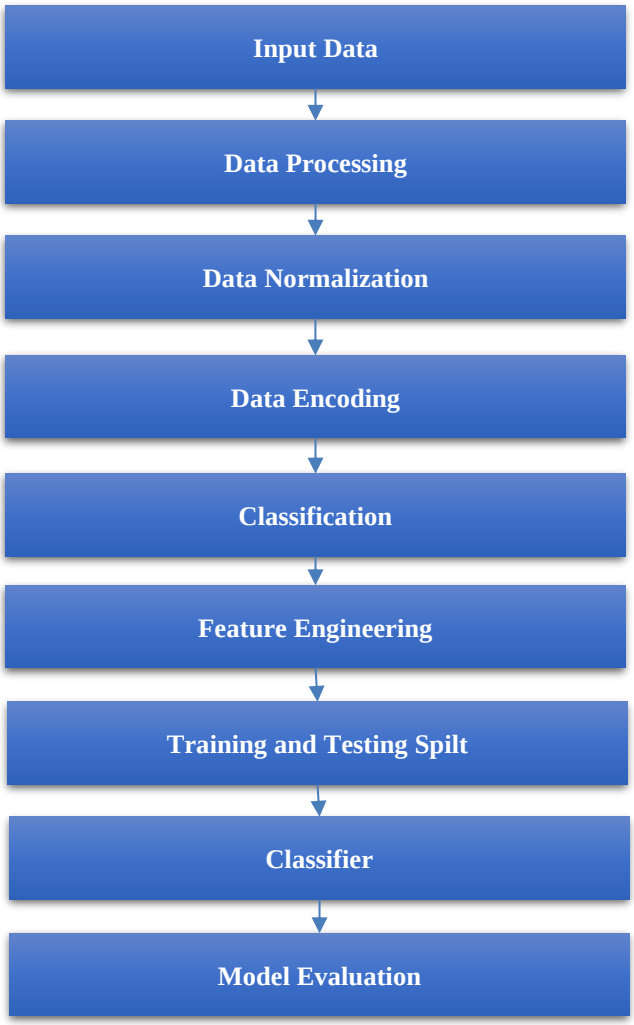

Fig. 5 Workflow diagram of proposed IDS

Figure 5 shows the workflow that is exercised while doing this experiment.

4.2. Multi-Layer Perceptron

A multi-layer perceptron is a kind of Feed-Forward Neural Network (FFNN) that consists of multiple layers. The layers include an input layer, hidden layers which can be one or multiple, and an output layer. Every layer is connected fully to the other layers, i.e. every neutron in the layer is connected to every neutron in the next layer.

The feed-forward neural network only sends the information in one direction. Information is passed from one layer to the other through the interconnected neurons.

In this execution of a Model on a Neural Network applied to the NSL-KDD dataset, the Keras library is utilised. The input features consist of 93 attributes, excluding the target attribute, which is binary, indicating whether an intrusion or an attack has occurred or not.

The dataset is separated into a training and a testing dataset in 75% - 25% split. The model is majorly trained on the available data and further tested on the remaining data.

Figure 6 represents a Multi-layer perceptron, demonstrating the input layer, hidden layer, and output layer.

The MLP model is a kind of Feed forward Artificial Neural Network (FF-ANN) implemented using the Sequential model from the Keras library. The multi-layer perceptron model which is implemented consists of one input layer followed by two hidden layers and one output layer.

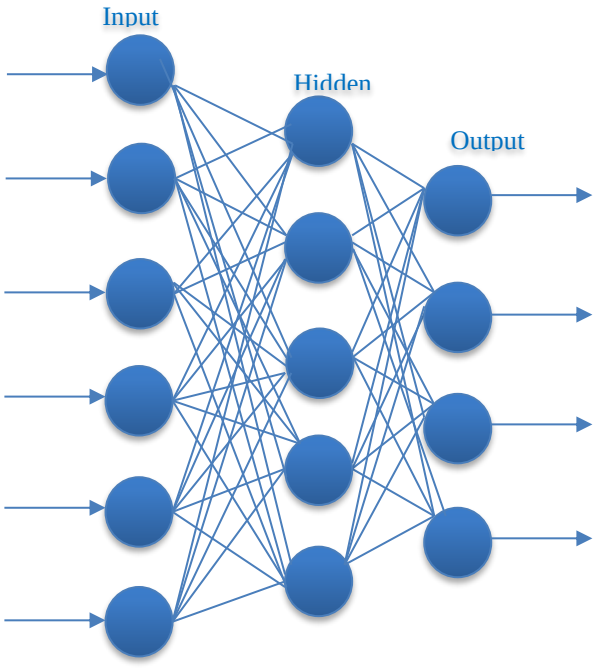

Fig. 6 Multi-layer perceptron

The input layer in the model has been implemented using 48 neurons, which is also equal to the number of features in the training data.

The activation function used at the input layer is the "hyperbolic tangent function (tanh)," it outputs values between -1 and 1 and is particularly useful for bringing the output of the neurons into a normalised range.

The second layer of the MLP is another hidden layer with 30 neurons, also implementing the tanh activation function. There is only one single neuron at the output layer of the MLP for the binary classification (normal or abnormal). The "sigmoid function" is the activation function utilized at the output layer. It compresses the output values to a range between 0 and 1, offering a probabilistic understanding of tasks involving binary classification.

Initially, the model is configured with the Stochastic Gradient Descent (SGD) optimizer, which has a learning rate of 0.01 and a momentum of 0.8. The model uses binary cross-entropy as the loss function, which is appropriate for binary classification tasks. The model's performance is assessed based on its effectiveness. After the first round of training, the model is recompiled with the 'Adam optimizer' due to its ability to adapt the learning rate.

The model has been trained over 120 epochs, each epoch using 4500 samples from the training set. Additionally, 20% of the training data was kept for a validation set. After training, the model's performance is evaluated on the testing set. This implementation demonstrates the application of a neural network model for intrusion detection using the NSL-KDD dataset. The choice of activation functions, optimisers, and the architecture of the MLP are crucial aspects of the model's performance. The use of the tanh activation function helps normalise the output of the neurons, while the binary cross-entropy loss function, SGD and the Adam optimisers ensure the model learns to classify intrusions effectively. The model's performance is assessed based on its accuracy on the test dataset, which provides a quantitative measure of the effectiveness of intrusion detection. The graphical representation of the model's layers provides a visual understanding of the model's architecture. This implementation serves as a comprehensive guide for applying neural networks to intrusion detection tasks.

4.3. Auto-Encoders

Autoencoders are a type of Artificial Neural Network (ANN) which are used for learning efficient coding of input data. They are unsupervised learning models that use backpropagation to generate a target output that matches the input. The central idea is to learn an encoding for a set of data, characteristically for the goal of having to do dimensionality reduction.

The construction of an autoencoder consists mainly of two components: an encoder part and a decoder part. The encoder part compresses the input data and produces a lower-dimensional code, while the decoder part reconstructs the original data from this code. The objective of an autoencoder is to minimize the differences between the original input and the reconstructed output, often referred to as reconstruction error.

One of the key applications of autoencoders is in the topic of anomaly detection. By training an autoencoder on normal data, it can learn to reconstruct it accurately. However, when presented with anomalous data, the autoencoder will likely produce a high reconstruction error, indicating an anomaly. Other applications include noise reduction, image denoising, and feature extraction.

The dataset is first divided into testing and training datasets in a 25% to 75% ratio. This ensures that the model is trained on the majority of the available data while also providing a distinct subset of data to assess the model's performance. To keep the auto-encoder focused on learning representations of the input data without being influenced by these specific labels, the remaining attributes—"intrusion," "abnormal," "normal," and "label"—are excluded from both the training and testing datasets. The target attribute, "intrusion," is removed from testing.

Figure 7 is an auto-encoder, and it is visible that the left side is the encoder part, followed by a code part and to the right end is the decoder part.

The auto-encoder's architecture is designed with an input layer that aligns with the dimensionality of the input data. This is followed by an encoding layer composed of 50 neurons, which condenses the input into a more compact representation. The Rectified Linear Unit (ReLu) activation function is employed at this encoding layer, introducing non-linearity to the model and enabling it to learn complex patterns within the data.

Following the encoding layer is the decoding layer, which aims to reconstruct the original input data from its encoded form. The output layer utilizes the SoftMax activation function, which is commonly used for classification tasks. For binary data, the Sigmoid activation function is typically used, while the SoftMax activation function is fitting for multi-class problems.

Using the Adam optimiser, the model is then compiled due to its adaptive learning rate properties. It employs Mean Squared Error (MSE) as its loss function, which is typically used for tasks involving reconstruction. The training procedure aims to adjust the classifier to the training data in such a way that the reconstruction error is minimized.

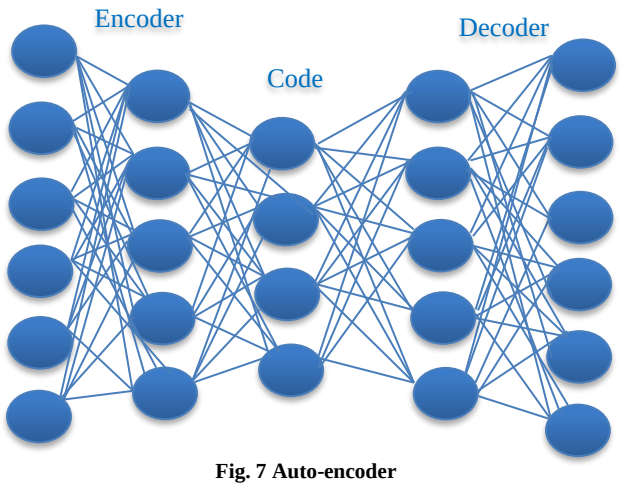

Fig. 7 Auto-encoder

5. Results

5.1. Multi-layer Perceptron Results

In the experiment, after implementing the multi-layer perceptron technique, the accuracy is around 96.8% for binary classification and around 97.1% for multi-class classification.

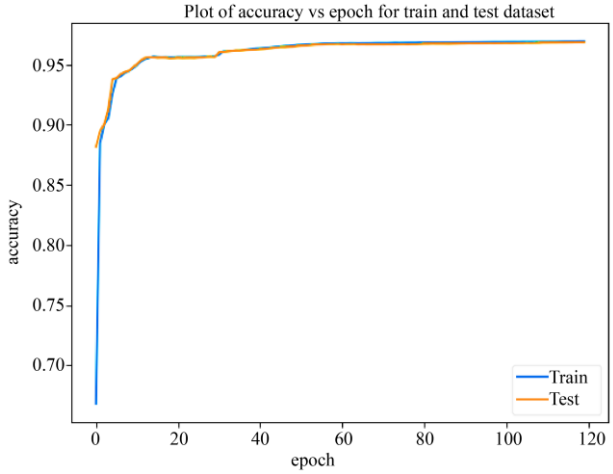

Fig. 8 Accuracy Vs Epoch for Multi-layer perceptron binary classification

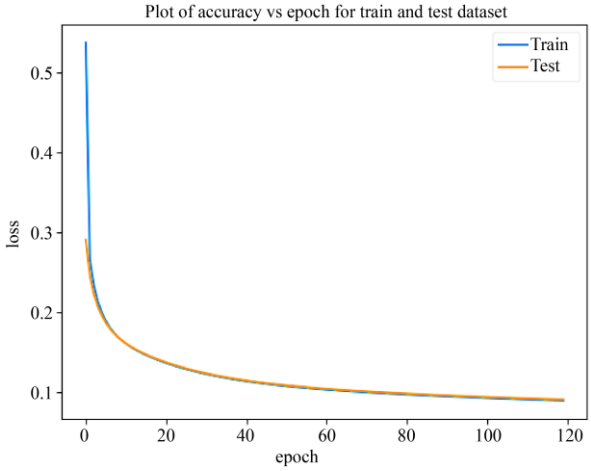

Fig. 9 Loss Vs Epoch for Multi-layer perceptron binary classification

In Figure 8, a comparative analysis of accuracy in accordance with the epochs for training and testing datasets for the binary classification is observed. In Figure 9, a comparative analysis of loss in accordance with the epochs for training and testing datasets for the binary classification is observed.

In Figure 10, a comparative analysis of accuracy in accordance with the epochs for training and testing datasets for the multi-class classification is observed. In Figure 11, a comparative analysis of loss in accordance with the epochs for training and testing datasets for the multi-class classification is observed.

As can be seen in Figures 8 and 10, the testing accuracy of the model in binary and multi-class classifications is in sync with the training accuracy of the dataset. This shows that the model is stable, and no sign of overfitting can be observed. Overfitting is one of the major issues that is caused during the process of implementing the DL models, and the hyperparameters have adjusted it along with choosing the hidden layers in such a way that overfitting does not occur.

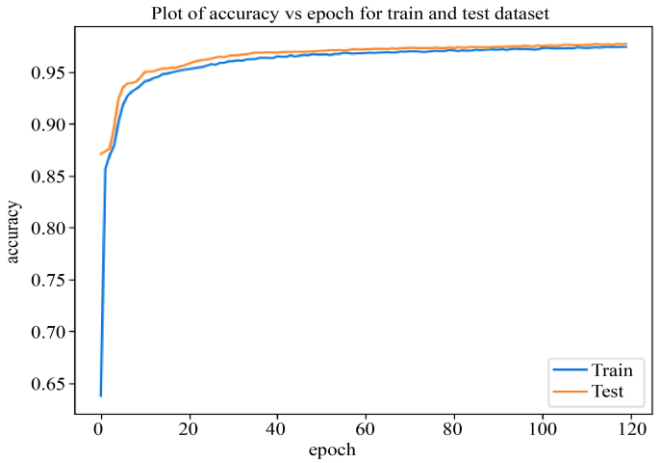

Fig. 10 Accuracy Vs Epoch for Multi-layer perceptron multi-class classification

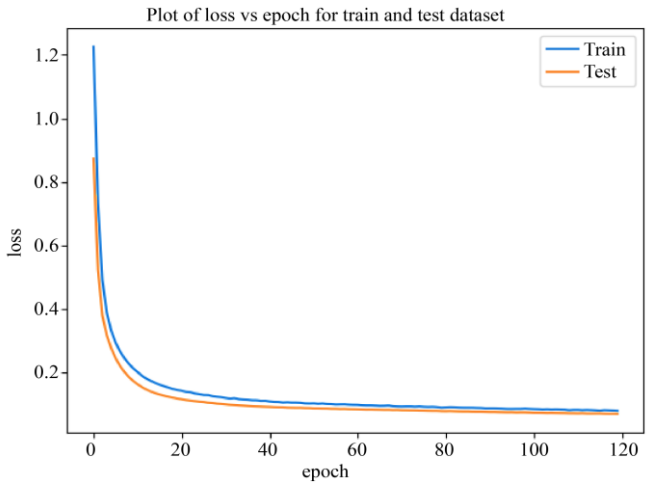

Fig. 11 Loss Vs Epoch for Multi-layer perceptron multi-class classification

5.2. Auto-Encoder Results

In the experiment, after implementing the multi-layer perceptron technique, the accuracy is observed to be around 95.6% for the binary classification and around 91.22% for the multi-class classification. The autoencoder is implemented with minimal overfitting.

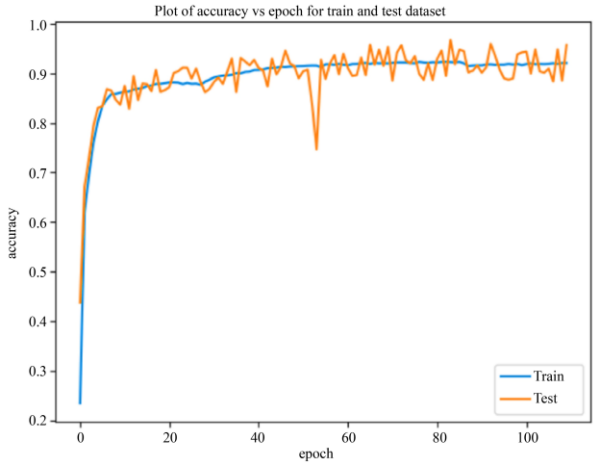

Fig. 12 Accuracy Vs Epoch for auto-encoders binary classification

In Figure 12, a comparative analysis of accuracy in accordance with the epochs for training and testing datasets for the binary classification can be observed.

In Figure 13, a comparative analysis of accuracy in accordance with the epochs for training and testing datasets for the multi-class classification can be observed.

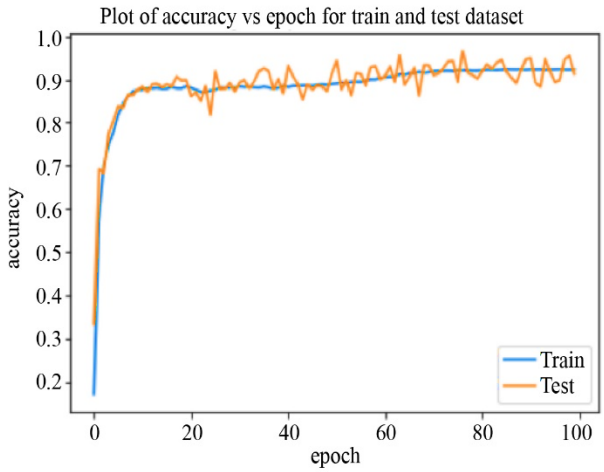

Fig. 13 Accuracy Vs Epoch for auto-encoders multi-class classification

6. Conclusion and Future Scope

Through this experiment, implemented DL techniques, specifically Multi-layer perceptron and Autoencoders for Network intrusion detection. Figure 14 demonstrates the comparative analysis between both of them under the binary classification, which classifies the attack as either normal or abnormal. This approach leverages the strengths of both

Multilayer Perceptron (MLP) and Auto-encoder models, which has led to improved performance in network intrusion detection.

On the other hand, the Auto-encoder is an unsupervised artificial neural network that learns how to compress and encode data efficiently and then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. This model is used to detect any deviation from the norm as a potential intrusion. The combination of these two models allowed us to leverage the strengths of both supervised learning (through the MLP) and unsupervised learning (through the Auto-encoder). This hybrid model was able to learn complex patterns in the network traffic and detect anomalies more effectively. In comparison to state-of-the-art techniques, this approach offers several advantages. Firstly, it can detect both known and unknown attacks, overcoming the limitation of signature-based NIDS. Secondly, it can learn complex and non-linear patterns in network traffic, which is often not possible with traditional machine learning techniques. Finally, this approach is more adaptable to evolving threats, as it can learn from new data and update its model accordingly. Through rigorous testing and validation, it was found that this chosen approach outperformed existing methods in terms of accuracy, precision, recall, and F1 score. This suggests that the combination of MLP and Auto-encoder models holds significant promise for enhancing the effectiveness of anomaly-based NIDS.

In this experiment, the model's accuracy has been increased. The multi-layer perceptron's accuracy is higher than the Autoencoder's. A comparative study of the two models under the multi-class categorization is similarly shown in Figure 15. Attacks are divided into five categories: "normal," "DoS attack," "Probe," "U2R," and "R2L." It has been noted that both models, each with unique qualities, are useful for network intrusion detection.

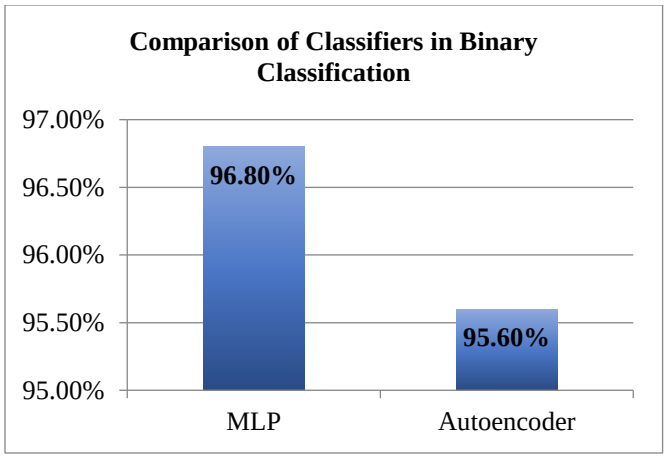

Fig. 14 Accuracy comparison of classifiers in binary classification

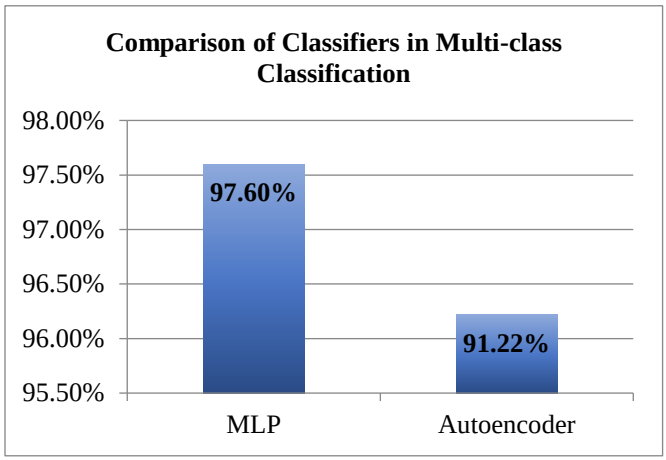

Fig. 15 Accuracy comparison of classifiers in Multi-class classification

For future work, the plan is to work and implement other DL techniques along with trying to combine various techniques with having different hyperparameters, multiple layers and other factors that would help in improvising the process of network intrusion detection.

References

[1] Sara Al-Emadi, Aisha Al-Mohannadi, and Felwa Al-Senaid, "Using Deep Learning Techniques for Network Intrusion Detection," *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies*, Doha, Qatar, pp. 171-176, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Chuanlong Yin et al., "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] B. Riyaz, and Sannasi Ganapathy, "A Deep Learning Approach for Effective Intrusion Detection in Wireless Networks Using CNN," *Soft Computing*, vol. 24, no. 22, pp. 17265-17278, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Chuanlong Yin et al., "Enhancing Network Intrusion Detection Classifiers Using Supervised Adversarial Training," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 6690-6719, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Salim Salmi, and Lahcen Oughdir, "Performance Evaluation of Deep Learning Techniques for DoS Attacks Detection in Wireless Sensor Network," *Journal of Big Data*, vol. 10, no. 1, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Wei Zhong, Ning Yu, and Chunyu Ai, "Applying Big Data Based Deep Learning System to Intrusion Detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181-195, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Petros Toupas et al., "An Intrusion Detection System for Multi-class Classification Based on Deep Neural Networks," *2019 18th IEEE International Conference on Machine Learning and Applications*, Boca Raton, FL, USA, pp. 1253-1258, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zeeshan Ahmad et al., "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 1, pp. 1-29, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Lirim Ashiku, and Cihan Dagli, "Network Intrusion Detection System using Deep Learning," *Procedia Computer Science*, vol. 185, pp. 239-247, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Panda Mrutyunjaya et al., "Network Intrusion Detection System: A Machine Learning Approach," *Intelligent Decision Technologies*, vol. 5, no. 4, pp. 347-356, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[11] ISCX NSL-KDD Dataset, UNB, 2009. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[12] Mohammed Maithem, and Ghadaa A. Al-Sultany, "Network Intrusion Detection System Using Deep Neural Networks," *Journal of Physics: Conference Series*, vol. 1804, no. 1, pp. 1-11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sydney Mambwe Kasongo, "A Deep Learning Technique for Intrusion Detection System Using a Recurrent Neural Networks Based Framework," *Computer Communications*, vol. 199, pp. 113-125, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ankit Thakkar, and Ritika Lohiya, "A Review of the Advancement in Intrusion Detection Datasets," *Procedia Computer Science*, vol. 167, pp. 636-645, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] V.K. Navya et al., "Intrusion Detection System Using Deep Neural Networks (DNN)," *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation*, Coimbatore, India, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]