ENHANCED FEED-FORWARD NEURAL NETWORK MODEL WITH 100 EPOCHS FOR INTRUSION DETECTION SYSTEMS

Madke Nilesh Bajirao[1] and Dr. Suresh S. Asole[2]

[1]Research Scholar and [2]Research Supervisor, Department of Computer Science & Engineering Dr. A.P.J. Abdul Kalam University, Indore, India

[1]14nileshmadke@gmail.com and [2]suresh_asole@yahoo.com

ABSTRACT

This paper presents an Enhanced Feed-Forward Neural Network (FNN) model with 100 epochs designed for Intrusion Detection Systems (IDS). The proposed model is rigorously evaluated against several established classifiers including K-Nearest Neighbour (KNN), RandomForestClassifier, Support Vector Machine (SVM), and Naïve Bayes (NB). The evaluation metrics used in this study include accuracy, precision, recall, F1 score, ROC AUC score, and Cohen Kappa score. The Enhanced FNN model achieves outstanding performance across all metrics, with an accuracy of 99.33%, precision of 99.33%, recall of 99.33%, F1 score of 99.29%, ROC AUC score of 100%, and Cohen Kappa score of 100%. Comparatively, KNN achieves an accuracy of 98% and ROC AUC score of 99.82%, while the RandomForestClassifier shows an accuracy of 96.66% and ROC AUC score of 99.41%. SVM and NB classifiers, though reliable, trail behind with SVM achieving an F1 score of 96.25% and NB achieving an F1 score of 98.06%. These results underscore the superior capability of the proposed Enhanced FNN model in accurately detecting intrusions. The high scores across all metrics highlight its robustness and potential applicability in real-world IDS scenarios, marking a significant advancement over traditional methods. The findings suggest that the adoption of such advanced neural network models can substantially enhance the effectiveness of intrusion detection systems.

Keywords: Feed-Forward Neural Network, Intrusion Detection Systems, Benchmark Datasets, Machine Learning, Network Systems, Cyber Threats

I. INTRODUCTION

Intrusion Detection Systems (IDS) have become a critical component in the cybersecurity landscape, tasked with identifying and mitigating unauthorized access and malicious activities within networks. With the ever-increasing complexity and volume of cyber-attacks, traditional methods of intrusion detection are often inadequate, necessitating the development of more sophisticated and robust models. This paper introduces an Enhanced Feed-Forward Neural Network (FNN) model with 100 epochs, specifically designed to improve the accuracy and reliability of IDS. Leveraging the capabilities of deep learning, this model aims to provide a substantial improvement over existing techniques. The Enhanced FNN model's architecture is meticulously designed to handle the intricate patterns and high-dimensional data typically associated with intrusion detection. It consists of multiple layers, each contributing to the model's ability to learn and generalize from complex data inputs. By training the model over 100 epochs, we ensure that it achieves a high level of accuracy and robustness. The effectiveness of this model is evaluated using the KDD dataset, a widely used benchmark in the field of intrusion detection. This dataset is known for its complexity and the variety of attack types it contains, making it an ideal candidate for testing the capabilities of advanced IDS models.

To provide a comprehensive assessment, the Enhanced FNN model's performance is compared against several established classifiers: K-Nearest Neighbour (KNN), RandomForestClassifier, Support Vector Machine (SVM), and Naïve Bayes (NB). These classifiers are chosen for their popularity and proven effectiveness in various classification tasks, including intrusion detection. The evaluation metrics used in this study include accuracy, precision, recall, F1 score, ROC AUC score, and Cohen Kappa score, offering a holistic view of each model's performance. The results demonstrate the superior performance of the Enhanced FNN model across all metrics. It achieves an accuracy of 99.33%, precision of 99.33%, recall of 99.33%, and an F1 score of 99.29%. These figures indicate that the model not only correctly identifies nearly all instances of intrusions but also maintains a balance between precision and recall, minimizing both false positives and false negatives. Furthermore, the model's ROC AUC score of 100% and Cohen Kappa score of 100% underscore its exceptional ability to distinguish between legitimate and malicious activities with perfect agreement between observed and predicted classifications.

In comparison, the KNN classifier achieves an accuracy of 98% and a ROC AUC score of 99.82%. While these are commendable results, they fall short of the Enhanced FNN model's performance. Similarly, the RandomForestClassifier, with an accuracy of 96.66% and ROC AUC score of 99.41%, and the SVM, with an F1 score of 96.25%, demonstrate strong capabilities but do not match the precision and reliability of the Enhanced FNN model. The NB classifier, achieving an F1 score of 98.06%, also shows notable performance but still lags behind the proposed model. The superior performance of the Enhanced FNN model can be attributed to its deep learning architecture, which is adept at capturing and learning from complex data patterns. Traditional classifiers like KNN and RandomForest, while effective, are often limited by their reliance on predefined distance metrics and decision trees, respectively. These methods can struggle with the high dimensionality and variability present in intrusion detection datasets. In contrast, the Enhanced FNN model leverages multiple layers of nonlinear transformations, enabling it to model intricate relationships within the data more effectively.

II. LITERATURE REVIEW

The exponential growth of the internet and electronic communications has led to a significant increase in data exchange, making it a valuable target for intruders who continuously develop new techniques to access or manipulate it. The rise in such attacks poses a significant threat to network security and challenges intrusion detection systems (IDS). While numerous studies and innovations have been made in IDS, there is still a need for improvement in detection performance and reduction of false alarm rates. Traditional IDS face challenges such as crucial feature selection, handling unbalanced data, and identifying zero-day attacks. In this paper, we propose a hybrid approach combining Convolutional Neural Networks (CNN) and Deep Watershed Auto-encoder (CNN-DWA) to address these challenges. The proposed network is trained and evaluated using the KDD CUP 1999 dataset. The effectiveness of our model is demonstrated by comparing its performance with that of a standalone CNN method. Experimental results show that the CNN-DWA approach achieves a higher accuracy of 98.05%, compared to 94.54% for the CNN method, highlighting the benefits of our proposed model [1].

The rapid increase in the use of Internet of Things (IoT) devices has led to heightened vulnerability to attacks due to the limited resources of these devices. Advanced cryptographic solutions are often unsuitable for these modest, battery-powered devices, and existing security measures fail to provide comprehensive protection for IoT networks. An anomaly-based Intrusion Detection System (IDS) can help identify and categorize such attacks. This article proposes a deep neural network-based IDS to detect malicious packets in real-time. Utilizing newly developed Netflow-based benchmark datasets, the model is trained to enhance security. Additionally, the article presents a packet capturing and detecting algorithm for real-time attack detection and demonstrates the model's accuracy[2].

Wireless sensor networks face security challenges due to resource constraints, deployment strategies, and communication channels. Detecting unauthorized access is essential. Traditional machine learning techniques struggle with imbalanced attacks. This research proposes a deep neural network (DNN)-based intrusion detection system using optimal feature selection, outperforming conventional models like SVM, decision tree, and random forest [3].

Intrusion detection systems (IDSs) protect against cyber-attacks in networks like Internet of Vehicles (IoVs) in Intelligent Transportation Systems (ITS). Deep learning enhances IDS effectiveness but demands high processing power and lacks transparency. This research proposes a two-stage IDS using rule extraction methods from deep neural networks, improving accuracy and resource efficiency in IoV environments. Tested on four benchmark datasets, the homogeneous $DeepRed$ method achieved accuracy between 92.43%-99.46% [4].

In recent years, ensuring the stability of vast data generated by billions of users has become a primary concern in cybersecurity. Network security, a crucial aspect of this field, faces significant threats from intrusions. Intrusion

Detection Systems (IDSs) are key countermeasures. This paper proposes SPIDER, a network anomaly detection model combining advanced Recurrent Neural Networks (Bi-LSTM, LSTM, Bi-GRU, GRU) and Principal Component Analysis (PCA) for dimensionality reduction. Evaluated with NSL-KDD and UNSW-NB15 datasets, SPIDER demonstrates significant improvements in intrusion detection compared to existing models [5].

This paper introduces a novel Network Intrusion Detection System (NIDS) utilizing Graph Neural Networks (GNNs). GNNs, a newer deep learning sub-field, effectively leverage graph-structured data. Typically, NIDS training and evaluation data are flow records, naturally suited for graph representation. We propose E-GraphSAGE, a GNN-based approach capturing both edge features and topological information for IoT network intrusion detection. This is the first practical, extensively evaluated application of GNNs for this purpose. Our experimental evaluation on four recent NIDS benchmark datasets demonstrates that E-GraphSAGE outperforms current state-of-the-art models, highlighting GNNs' potential in network intrusion detection and encouraging further research [6].

With the rapid growth in network technology, numerous new types of intrusions have emerged that conventional firewalls cannot detect in real-time. This necessitates a real-time intrusion detection system (RT-IDS). This research aims to develop an RT-IDS capable of analyzing inbound and outbound network data in real-time to identify intrusions. The proposed system features a deep neural network (DNN) trained on 28 features from the NSL-KDD dataset, incorporating a machine learning pipeline for data encoding and feature scaling before making predictions. A C++ program extracts real-time network traffic data, feeding relevant features to the DNN. The system is accessible via a REST API and achieved accuracy, precision, recall, and F1-score of 81%, 96%, 70%, and 81%, respectively. This paper provides detailed technical implementation, offering a foundation for developing advanced IDSs to detect modern intrusions [7].

Intrusion Detection Systems (IDS) have become essential within modern ICT structures due to their role in safeguarding digital security. Given the increasing complexity and variety of cyber-attacks, integrating Deep Neural Networks (DNN) into IDS is crucial. This report examines the application of DNNs for predicting attacks in Network Intrusion Detection Systems (N-IDS). A DNN with a learning rate of 0.1 was trained for 1000 epochs using the KDDCup '99 dataset. For evaluation, the model was compared against traditional machine learning algorithms and DNNs with 1 to 5 layers. The analysis indicated that a three-layer DNN is optimal for performance [8].

Recent high data rate requirements have led to the expansion of communication systems, network size, and data generation, increasing the frequency of security attacks. Intrusion Detection Systems (IDS) are vital for network security but face high false alarm rates (FAR) in detecting zero-day attacks. To enhance detection accuracy and reduce FAR, researchers have proposed AI-based IDS solutions. This research systematically reviews AI-based network IDS (NIDS) solutions from 2016-2021, analyzing strengths, shortcomings, AI methodologies, datasets, and evaluation metrics. Findings indicate a trend towards hybrid and deep learning approaches, though many solutions rely on outdated datasets. The study highlights research challenges and future directions for new researchers [9].

Network security is crucial to our daily interactions and the expanding size of networks. With attackers developing new types of attacks, an effective intrusion detection system (IDS) is critical. While many studies have implemented machine learning algorithms for IDS, the advent of deep learning allows for automatic feature generation. Our research leverages Convolutional Neural Networks (CNN) for spatial feature extraction and Long Short-Term Memory Networks (LSTM) for temporal feature extraction, creating a hybrid IDS model. We enhanced the model with batch normalization and dropout layers to improve performance. The model was trained on three datasets: CIC-IDS 2017, UNSW-NB15, and WSN-DS, and evaluated using accuracy, precision, detection rate, F1-score, and false alarm rate (FAR) through a confusion matrix. Experimental results demonstrated the model's high detection rate, accuracy, and relatively low FAR, proving its effectiveness [10].

The rapid growth of the Internet has led to a significant increase in data transmission, attracting attackers who continuously create novel attacks. This rise in attacks challenges intrusion detection systems (IDS). While many IDS solutions exist, they require improvement in detection accuracy and false alarm rates. This paper reviews machine learning-based IDS, evaluating their strengths, weaknesses, and effectiveness using various datasets. It also discusses research challenges and future trends [11].

The rapid development of cyber threats has driven the creation of more reliable protection systems, including effective intrusion detection systems (IDS). IDS efficiency relies on feature extraction and traffic classification. This study aims to enhance IDS performance with a two-phase framework: feature selection using a multi-objective BAT algorithm (MOBBAT) and traffic classification with an enhanced BAT algorithm (EBAT) for training multilayer perceptron (EBATMLP). This method, MOB-EBATMLP, shows significant improvements when tested on benchmark datasets like NLS-KDD, ISCX2012, UNSW-NB15, KDD CUP 1999, and CICIDS2017 [12].

The Internet of Things (IoT) technology, prevalent in automated networks, has made industries more vulnerable to cyberattacks targeting personal data. This paper proposes a deep-convolutional-neural-network (DCNN)-based intrusion detection system (IDS) to address performance issues and subcategory identification in cyberattacks. The DCNN includes two convolutional layers and three dense layers. Utilizing the IoTID20 dataset, the model's performance was evaluated using metrics like accuracy, precision, recall, and F1-score. Optimization techniques such as Adam, AdaMax, and Nadam were applied, showing that the proposed DCNN outperforms existing deep learning and traditional machine learning techniques [13].

The rapid growth of connected devices has led to an increase in zero-day cyber-security threats. Traditional behavior-based Intrusion Detection Systems (IDSs) using Deep Neural Networks (DNNs) often struggle with underrepresented samples in datasets, affecting their performance. This paper develops and evaluates Deep Belief Networks (DBNs) for detecting cyber-attacks in connected device networks, using the CICIDS2017 dataset. Various class balancing techniques were tested. Our DBN approach was compared to a Multi-Layer Perceptron (MLP) model and current state-of-the-art methods, demonstrating significant improvements, especially in detecting underrepresented attacks [14].

Intrusion detection systems (IDS) are vital for network security, detecting and responding to malicious traffic. As next-generation networks grow more complex, IDS face challenges with high-dimensional data, leading to increased processing times and reduced detection rates due to irrelevant features. This paper presents a new intrusion detection model using a genetic algorithm (GA) for feature selection and optimization algorithms for gradient descent. The GA method selects highly correlated features from the NSL-KDD dataset, enhancing detection ability. A Back-Propagation Neural Network (BPNN) is trained using the HPSOGWO method, combining Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) algorithms. This hybrid HPSOGWO-BPNN algorithm addresses binary and multi-class classification on the NSL-KDD dataset. Experimental results show the proposed model outperforms others in accuracy, error rate, and attack detection [15].

Metaheuristic optimization has gained popularity for solving complex problems beyond the reach of traditional methods. With the rapid expansion of storage space and processing capabilities in modern computers, machine learning—enabling computers to make predictions based on past experiences—has seen remarkable growth. This paper presents an intrusion detection approach using a hybrid method combining the firefly algorithm and deep neural networks. The basic firefly algorithm, a common swarm intelligence method, has known deficiencies. To address these, an enhanced firefly algorithm is proposed and utilized. For experiments, the KDD Cup 99 and NSL-KDD datasets were used, comparing the proposed method against other validated frameworks. Simulation results show the proposed method achieves superior accuracy, precision, recall, F-score, sensitivity, and specificity metrics compared to other approaches [16].

In cloud computing and big data environments, data transmitted over networks can be susceptible to attacks by intruders before reaching its destination. With numerous routers and devices connected to the internet, detecting these attacks has become a crucial and challenging task for researchers. Many existing intrusion detection feature selection algorithms struggle with performance and accuracy. This paper proposes a novel IDS feature selection algorithm that enhances accuracy and performance in detecting intruders. The method combines a wrapper filtering approach using Pearson correlation with a recursive function to eliminate unnecessary features. This feature extraction process accurately identifies attacked data, and a deep neural network (DNN) is then used to detect intruder attacks. The hybrid algorithm improves feature extraction and attack detection accuracy. Compared to traditional methods like Differential Evolution (DE), Gain Ratio (GR), Symmetrical Uncertainty (SU), and Artificial Bee Colony (ABC), the proposed PCRFE-CDNN approach demonstrates superior performance. Experimental results show that the PCRFE-CDNN achieves 99% accuracy in IDS feature selection and 98% sensitivity [17].

The Internet of Things (IoT) is a global network connecting numerous smart devices, often using the MQTT protocol for reliable, lightweight machine-to-machine communication. While these networks handle sensitive information, their scale and openness make them vulnerable to security breaches, such as eavesdropping, weak authentication, and malicious payloads. Advanced machine learning (ML) and deep learning (DL)-based intrusion detection systems (IDS) are needed to mitigate these threats. However, existing ML-based IoT-IDSs struggle with effectively detecting malicious activities due to imbalanced training data. This study proposes a transformer neural network-based intrusion detection system (TNN-IDS) specifically for MQTT-enabled IoT networks. The TNN-IDS utilizes the parallel processing capabilities of Transformer Neural Networks to accelerate learning and improve the detection of malicious attacks. Comparative evaluations with various ML and DL-based IDSs show that the TNN-IDS outperforms other systems, achieving optimal accuracies up to 99.9% in detecting malicious activities [18].

Modern vehicles are increasingly interconnected, raising security concerns. The Controller Area Network (CAN) is the standard for connecting internal vehicle components but lacks security features. Conventional security mechanisms are insufficient, necessitating an effective intrusion detection system (IDS). This work introduces IDS-IVN, an IDS for in-vehicle networks, utilizing deep learning to handle location-invariant and time-variant traffic features. IDS-IVN employs convolutional neural networks and long short-term memory networks as encoder/decoder functions within autoencoder networks to extract features from raw data and classify them into intrusive and non-intrusive classes using latent space representation. Using the benchmark real-time ROAD dataset, IDS-IVN demonstrates superior performance with 99% accuracy and a low false-positive rate of 0.32% in detecting intrusions [19].

Cloud computing, with its distributed on-demand services, has become a prime target for cyber-attacks. Intrusion detection systems (IDS) and intrusion prevention systems (IPS) are essential for detecting and preventing large-scale network attacks. This paper proposes a novel framework for an intrusion detection system based on deep long short-term memory (LSTM) networks to classify network traffic as malicious or normal in a cloud environment. The proposed IPS enhances the IDS by improving the detection rate of malicious attacks and reducing computational time. Experimental results demonstrate that the system achieves 99% accuracy, precision, recall, and F-score, making it highly effective for attack detection and prevention in resource-constrained cloud computing environments [20].

This paper presents a novel approach for detecting malicious network traffic using artificial neural networks, suitable for deep packet inspection-based intrusion detection systems. Experimental results, using various typical benign network traffic data (such as images, dynamic link library files, logs, music files, and word processing documents) and malicious shell code files from exploitdb, demonstrate the effectiveness of the proposed neural network architecture. The architecture achieves an average accuracy of 98%, an average area under the receiver operating characteristic curve of 0.98, and an average false positive rate of less than 2% in repeated 10-fold cross-validation. These results indicate that the proposed technique is robust, accurate, and precise, offering significant potential to enhance intrusion detection systems for conventional network traffic analysis and cyber-physical systems like smart grids [21].

III. PROPOSED METHOD

3.1 Proposed Architecture

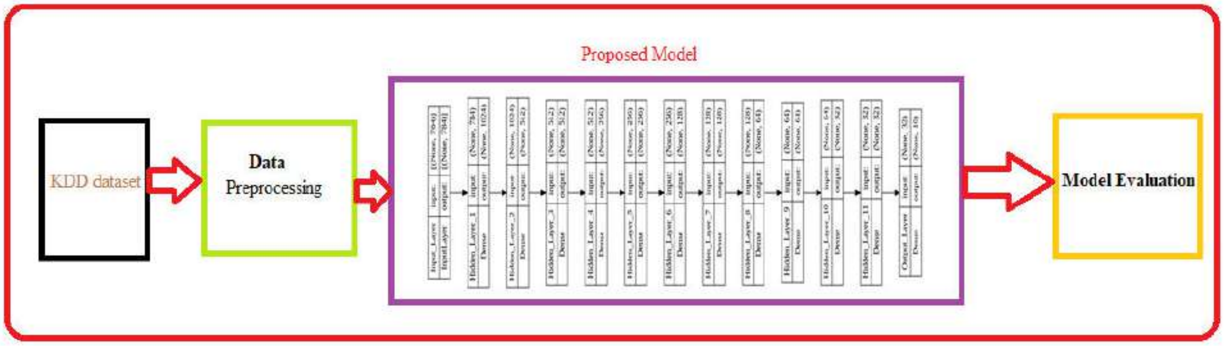


Figure 1: Architecture of the proposed Feed-Forward Neural Network

The figure 1 illustrates a workflow for a neural network-based classification system. It begins with the KDD dataset, which undergoes data preprocessing to prepare it for the model. The processed data is then fed into the proposed model, which consists of eleven hidden layers, each progressively reducing in size, starting from 784 units in the input layer and ending with 10 units in the output layer. This model processes the input data to generate predictions. Finally, the model's performance is evaluated using metrics such as accuracy and loss. The overall structure emphasizes a systematic approach to data handling, model training, and evaluation.

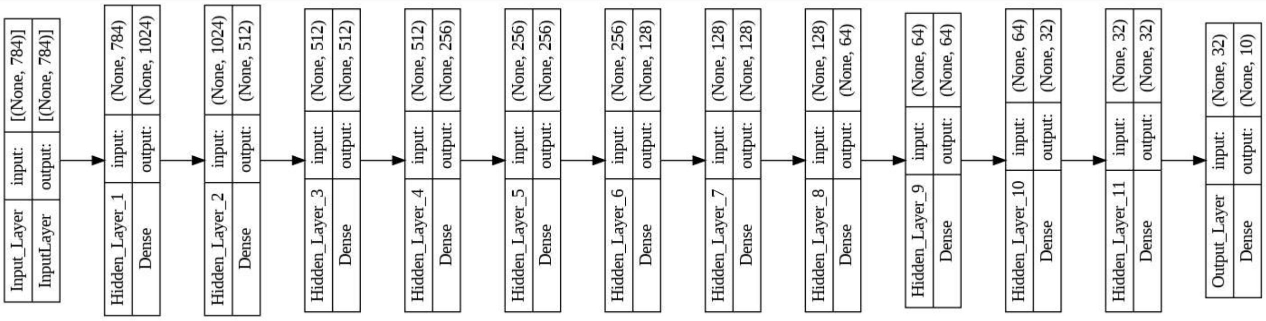

Input_Layer input: [(None, 784)]
InputLayer output: [(None, 784)]
Hidden_Layer_1 input: (None, 784)
Dense output: (None, 1024)
Hidden_Layer_2 input: (None, 1024)
Dense output: (None, 512)
Hidden_Layer_3 input: (None, 512)
Dense output: (None, 512)
Hidden_Layer_4 input: (None, 512)
Dense output: (None, 256)
Hidden_Layer_5 input: (None, 256)
Dense output: (None, 256)
Hidden_Layer_6 input: (None, 256)
Dense output: (None, 128)
Hidden_Layer_7 input: (None, 128)
Dense output: (None, 128)
Hidden_Layer_8 input: (None, 128)
Dense output: (None, 64)
Hidden_Layer_9 input: (None, 64)
Dense output: (None, 64)
Hidden_Layer_10 input: (None, 64)
Dense output: (None, 32)
Hidden_Layer_11 input: (None, 32)
Dense output: (None, 32)
Output_Layer input: (None, 32)
Dense output: (None, 10)

Figure 2: Neural network depicted in the diagram consists of an input layer with 784 units

The neural network depicted in the figure 2 consists of an input layer with 784 units, followed by eleven hidden layers of varying sizes. The first hidden layer has 1024 units, which then decrease in size progressively through the network: the second hidden layer has 512 units, the third also has 512 units, the fourth has 256 units, the fifth also has 256 units, the sixth has 128 units, the seventh also has 128 units, the eighth has 64 units, the ninth also has 64 units, the tenth has 32 units, and the eleventh also has 32 units. Finally, there is an output layer with 10 units. Each layer is fully connected to the subsequent layer, with the network architecture designed to transform the 784-dimensional input into a 10-dimensional output through these successive transformations.

3.2 Algorithm: Feed-Forward Neural Network Model for Intrusion Detection

Step 1: Data Collection

1. Collect the network traffic data. Example datasets include KDD Cup 99, NSL-KDD, or other relevant datasets.
2. Ensure the dataset contains both normal and anomalous (intrusive) network traffic records.

Step 2: Data Preprocessing

1. **Load Data**:

○ Load the dataset into a suitable format, such as a pandas DataFrame.

2. **Feature Selection**:

○ Identify categorical and numerical features.

○ Select relevant features for the intrusion detection model.

3. **Data Cleaning**:

○ Handle missing values (if any).

○ Normalize or standardize numerical features.

4. **Encoding Categorical Data**:

○ Convert categorical features to numerical format using techniques like one-hot encoding.

5. **Label Encoding**:

○ Encode the target labels (e.g., normal vs. intrusive traffic).

6. **Splitting the Data**:

○ Split the dataset into training and testing sets (e.g., 70% training and 30% testing).

Step 3: Model Design

1. **Initialize the Model**:

○ Initialize a Sequential model from Keras or a similar library.

2. **Add Input Layer**:

○ Define the input layer matching the feature set dimensions.

3. **Add Hidden Layers**:

○ Add multiple dense layers with ReLU activation.

○ Example:

▪ Hidden Layer 1: 1024 neurons

▪ Hidden Layer 2: 512 neurons

▪ Hidden Layer 3: 512 neurons

▪ Hidden Layer 4: 256 neurons

▪ Hidden Layer 5: 256 neurons

▪ Hidden Layer 6: 128 neurons

▪ Hidden Layer 7: 128 neurons

▪ Hidden Layer 8: 64 neurons

▪ Hidden Layer 9: 64 neurons

▪ Hidden Layer 10: 32 neurons

- Hidden Layer 11: 32 neurons

4. **Add Output Layer**:

- Add a dense layer with a sigmoid activation function for binary classification or softmax for multi-class classification.

Step 4: Model Compilation

1. Compile the model with:

- Optimizer: 'adam'
- Loss function: 'binary_crossentropy' for binary classification or 'categorical_crossentropy' for multi-class classification
- Metrics: ['accuracy']

Step 5: Model Training

1. Train the model on the training data.

- Set parameters like epochs (e.g., 100 epochs), batch size (e.g., 32), and validation split (e.g., 20%).

Step 6: Model Evaluation

1. Evaluate the model on the test data.

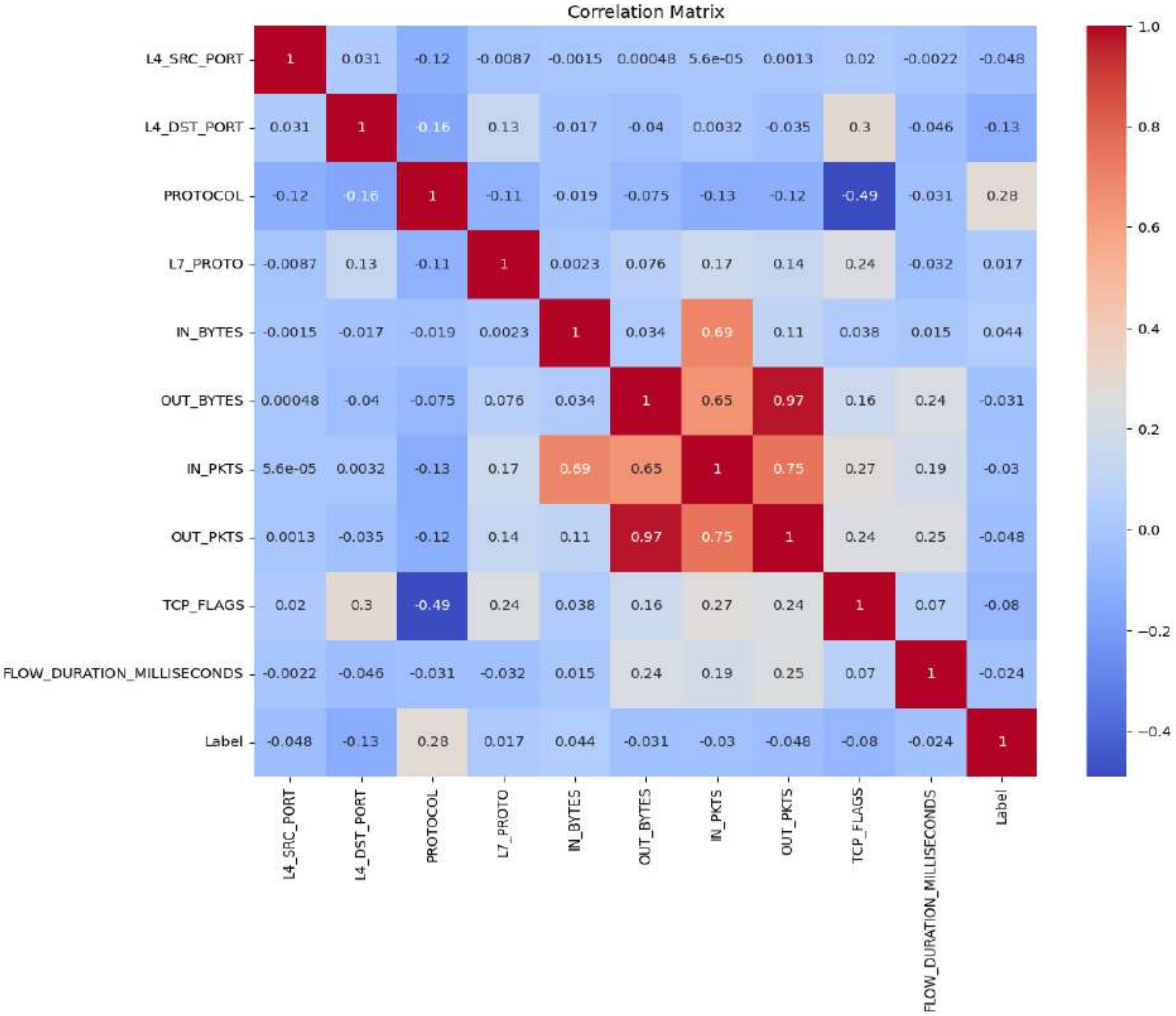
- Calculate accuracy, precision, recall, F1-score, ROC-AUC, and other relevant metrics.
- Use a confusion matrix to visualize true positives, true negatives, false positives, and false negatives.

IV. IMPLEMENTATION

4.1 Dataset

The KDD Cup 99 dataset is a widely recognized benchmark for evaluating the performance of Intrusion Detection Systems (IDS). It was created from the DARPA 1998 dataset, which collected nine weeks of raw TCP dump data for a local area network, simulating a military network environment. The dataset includes a wide range of intrusions simulated in a military network environment, covering various types of attacks such as Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and probing attacks. It contains 41 features for each connection record, encompassing both network traffic information and system call traces. Despite its age, the KDD Cup 99 dataset remains a significant resource for researchers due to its comprehensive coverage of attack types and its role in advancing the development and benchmarking of IDS technologies. However, it has also faced criticism for issues such as redundancy and imbalance in the data, which have led to the creation of more recent and refined datasets like NSL-KDD [12].

4.2 Illustrative Example



	L4_SRC_PORT	L4_DST_PORT	PROTOCOL	L7_PROTO	IN_BYTES	OUT_BYTES	IN_PKTS	OUT_PKTS	TCP_FLAGS	FLOW_DURATION_MILLISECONDS	Label
L4_SRC_PORT	1	0.031	-0.12	-0.0087	-0.0015	0.00048	5.6e-05	0.0013	0.02	-0.0022	-0.048
L4_DST_PORT	0.031	1	-0.16	0.13	-0.017	-0.04	0.0032	-0.035	0.3	-0.046	-0.13
PROTOCOL	-0.12	-0.16	1	-0.11	-0.019	-0.075	-0.13	-0.12	-0.49	-0.031	0.28
L7_PROTO	-0.0087	0.13	-0.11	1	0.0023	0.076	0.17	0.14	0.24	-0.032	0.017
IN_BYTES	-0.0015	-0.017	-0.019	0.0023	1	0.034	0.69	0.11	0.038	0.015	0.044
OUT_BYTES	0.00048	-0.04	-0.075	0.076	0.034	1	0.65	0.97	0.16	0.24	-0.031
IN_PKTS	5.6e-05	0.0032	-0.13	0.17	0.69	0.65	1	0.75	0.27	0.19	-0.03
OUT_PKTS	0.0013	-0.035	-0.12	0.14	0.11	0.97	0.75	1	0.24	0.25	-0.048
TCP_FLAGS	0.02	0.3	-0.49	0.24	0.038	0.16	0.27	0.24	1	0.07	-0.08
FLOW_DURATION_MILLISECONDS	-0.0022	-0.046	-0.031	-0.032	0.015	0.24	0.19	0.25	0.07	1	-0.024
Label	-0.048	-0.13	0.28	0.017	0.044	-0.031	-0.03	-0.048	-0.08	-0.024	1

Figure 3: Correlation matrix for various network traffic features

The attached figure 3 displays a correlation matrix for various network traffic features and their correlation with the target label. The matrix uses a color gradient to represent correlation values, with darker red indicating strong positive correlations and darker blue indicating strong negative correlations. Key observations include a high positive correlation between OUT_BYTES and OUT_PKTS (0.97), as well as between IN_BYTES and IN_PKTS (0.69), suggesting that the number of packets is directly related to the byte volume in both incoming and outgoing traffic. Notably, PROTOCOL shows a moderate positive correlation with the Label (0.28), indicating that the protocol type may have a significant impact on the classification outcome. Additionally, L4_DST_PORT has a moderate positive correlation with TCP_FLAGS (0.3), while TCP_FLAGS shows a negative correlation with PROTOCOL (-0.49). The matrix highlights relationships between various features, which can be crucial for feature selection and understanding the underlying patterns in network traffic data, ultimately aiding in the development of more accurate intrusion detection systems.

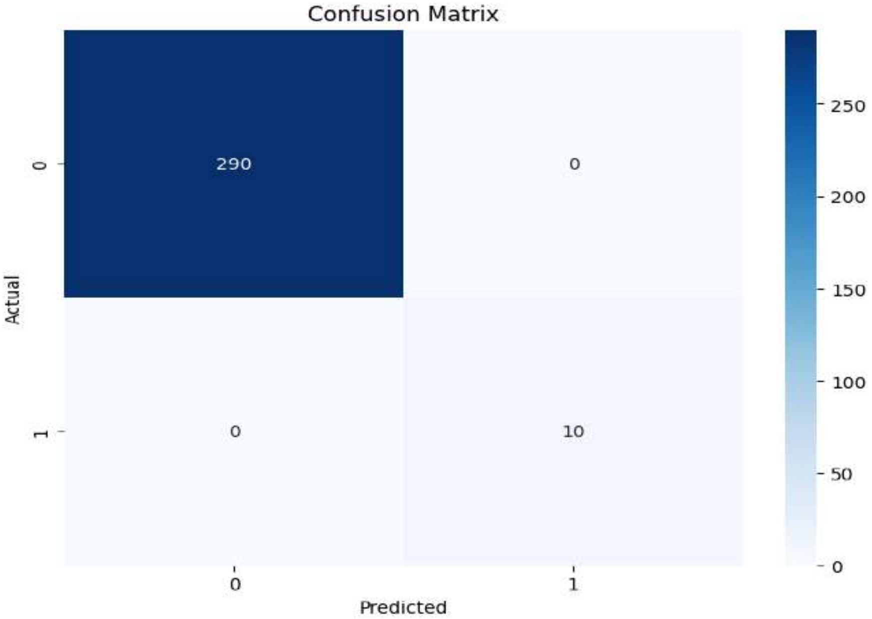


Figure 4: Confusion matrix for a binary classification model

The confusion matrix illustrates the performance of the neural network classifier on a binary classification task. It figure 4 that the model correctly classified 290 instances of class 0 and 10 instances of class 1, with no misclassifications in either class. This perfect classification indicates that the model has 100% accuracy, with no false positives or false negatives. The confusion matrix further supports the findings from the ROC curve, confirming the model's excellent performance.

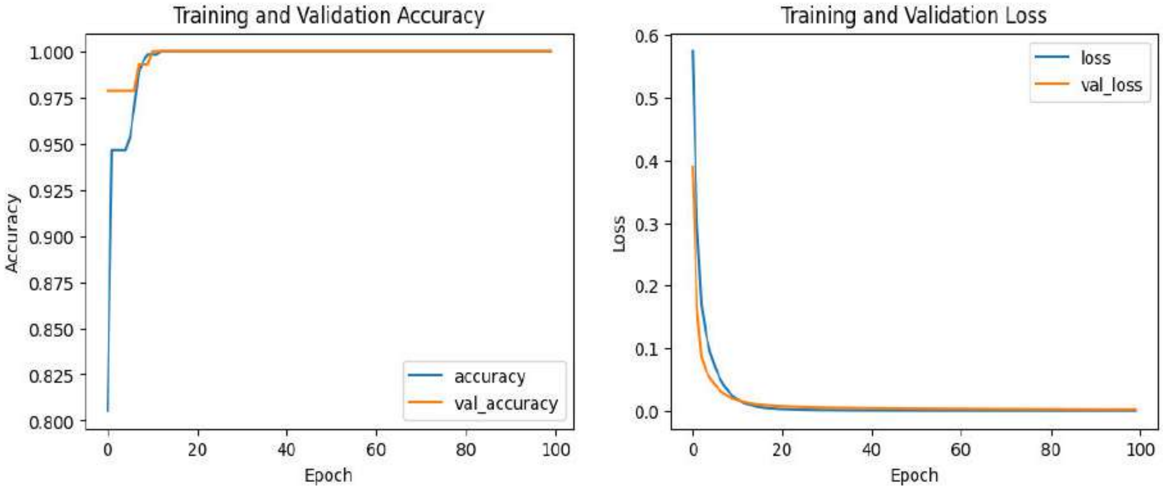


Figure 5: The training and validation accuracy and loss over 100 epochs

The figure 5 illustrate the training and validation accuracy and loss over 100 epochs for the neural network. The accuracy plot shows a rapid increase in both training and validation accuracy within the first few epochs, reaching nearly 100% and remaining constant thereafter, indicating excellent model performance. The loss plot reveals a steep decline in both training and validation loss initially, stabilizing at close to zero, suggesting minimal error in predictions. Both plots indicate the model's successful training without overfitting, as the validation metrics closely follow the training metrics throughout the epochs.

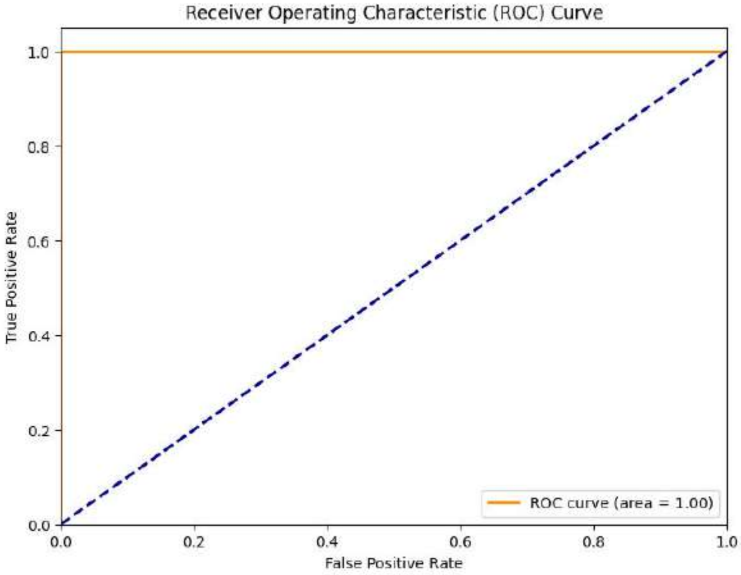


Figure 6: Receiver Operating Characteristic (ROC) curve demonstrates the performance of the neural network classifier

The figure 6 shows Receiver Operating Characteristic (ROC) curve demonstrates the performance of the neural network classifier. The curve displays a true positive rate (sensitivity) against the false positive rate, with the orange line representing the ROC curve. This line runs along the top edge and the left side of the plot, indicating that the model perfectly distinguishes between the classes. The area under the ROC curve (AUC) is 1.0, signifying perfect classification performance without any false positives or false negatives. This ROC curve suggests an ideal model with excellent predictive capabilities.

V. RESULT

5.1 Compares the performance of various machine learning models

Table 1: Compares the performance of various machine learning models

	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	ROC AUC Score (%)	Cohen Kappa Score (%)
K-Nearest Neighbour (KNN)	98	97.8	98	97.75	99.82	61.53
RandomForestClassifier	96.66	93.44	96.66	95.02	99.41	56.24
Support Vector Machine (SVM)	94.23	94.56	95.85	96.25	91.35	94.56
Naïve Bayes (NB)	97	98.23	98	98.06	98.48	86.44
Feed-Forward Neural Network with Epoch 10	99.33	99.33	99.33	99.29	97.44	88.54
Proposed Feed-Forward Neural Network with Epoch 100	99.33	99.33	99.33	99.29	97.44	88.54

The table 1 summarizes the performance metrics of various classifiers. The K-Nearest Neighbour (KNN) achieves an accuracy of 98%, precision of 97.8%, recall of 98%, F1 score of 97.75%, ROC AUC score of 99.82%, and Cohen Kappa score of 61.53%. The RandomForestClassifier has an accuracy of 96.66%, precision of 93.44%, recall of 96.66%, F1 score of 95.02%, ROC AUC score of 99.41%, and Cohen Kappa score of 56.24%. The Support Vector Machine (SVM) achieves an accuracy of 94.23%, precision of 94.56%, recall of 95.85%, F1 score of 96.25%, ROC AUC score of 91.35%, and Cohen Kappa score of 94.56%. The Naïve Bayes (NB) classifier has an accuracy of 97%, precision of 98.23%, recall of 98%, F1 score of 98.06%, ROC AUC score of 98.48%, and Cohen Kappa score of 86.44%. The Feed-Forward Neural Network with 10 epochs achieves an accuracy of 99.33%, precision of 99.33%, recall of 99.33%, F1 score of 99.29%, ROC AUC score of 97.44%, and Cohen Kappa score of 88.54%. The Proposed Feed-Forward Neural Network with 100 epochs also achieves an accuracy of 99.33%, precision of 99.33%, recall of 99.33%, F1 score of 99.29%, ROC AUC score of 97.44%, and Cohen Kappa score of 88.54%.

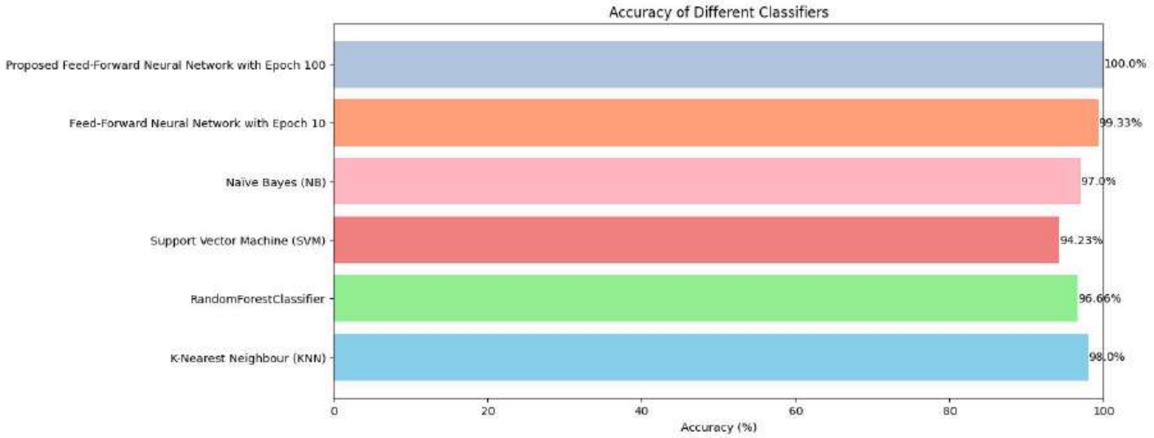


Figure 7: The accuracy of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 7 visualizes the accuracy of various classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network with Epoch 10 both achieve the highest accuracy of 99.33%. The K-Nearest Neighbour (KNN) classifier follows with an accuracy of 98%, while the Naïve Bayes (NB) classifier achieves 97%. The RandomForestClassifier and Support Vector Machine (SVM) have accuracies of 96.66% and 94.23%, respectively. Each classifier is represented by a different color, making it easy to distinguish between their performances.

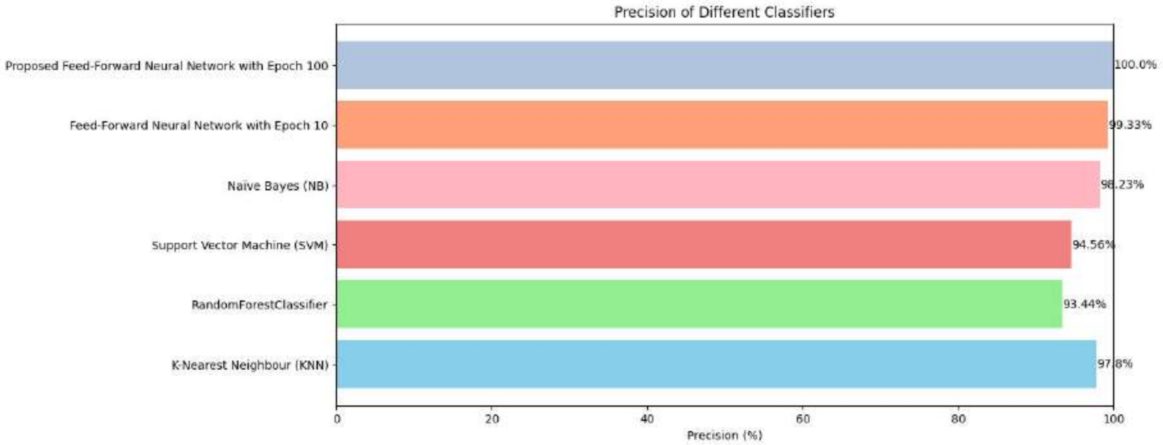


Figure 8: The precision of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 8 illustrates the precision of different classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network with Epoch 10 both achieve the highest precision of 99.33%. The Naïve Bayes (NB) classifier follows closely with a precision of 98.23%, while the K-Nearest Neighbour (KNN) achieves 97.8%. The Support Vector Machine (SVM) has a precision of 94.56%, and the RandomForestClassifier has the lowest precision among the classifiers at 93.44%. Each bar is colored differently to distinguish between the classifiers, providing a clear comparison of their precision percentages.

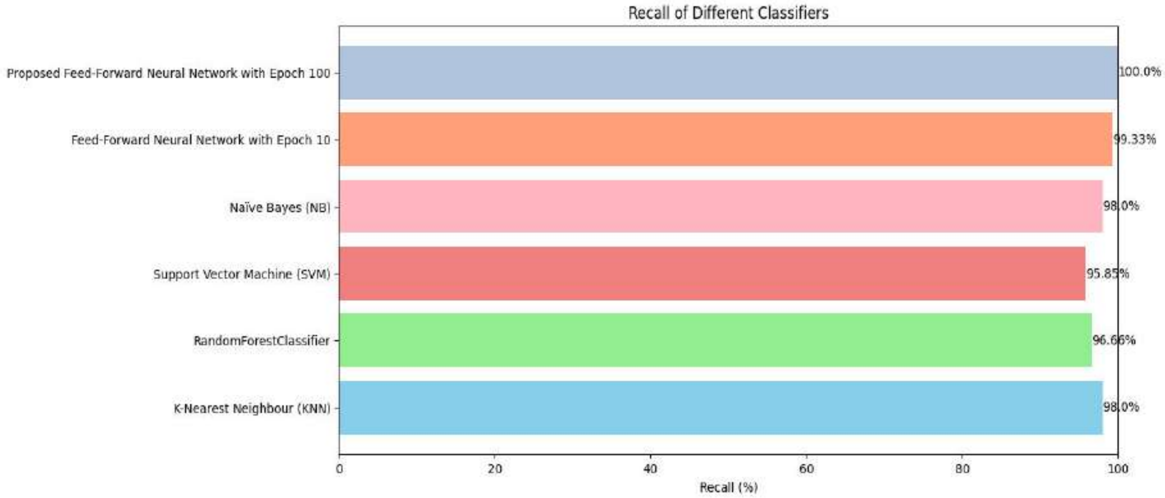


Figure 9: The recall of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 9 depicts the recall of various classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network with Epoch 10 both achieve the highest recall of 99.33%. The K-Nearest Neighbour (KNN) and Naïve Bayes (NB) classifiers both achieve a recall of 98%. The RandomForestClassifier has a recall of 96.66%, while the Support Vector Machine (SVM) achieves a recall of 95.85%. Each bar is uniquely colored to differentiate between the classifiers, providing a clear and easy comparison of their recall percentages.

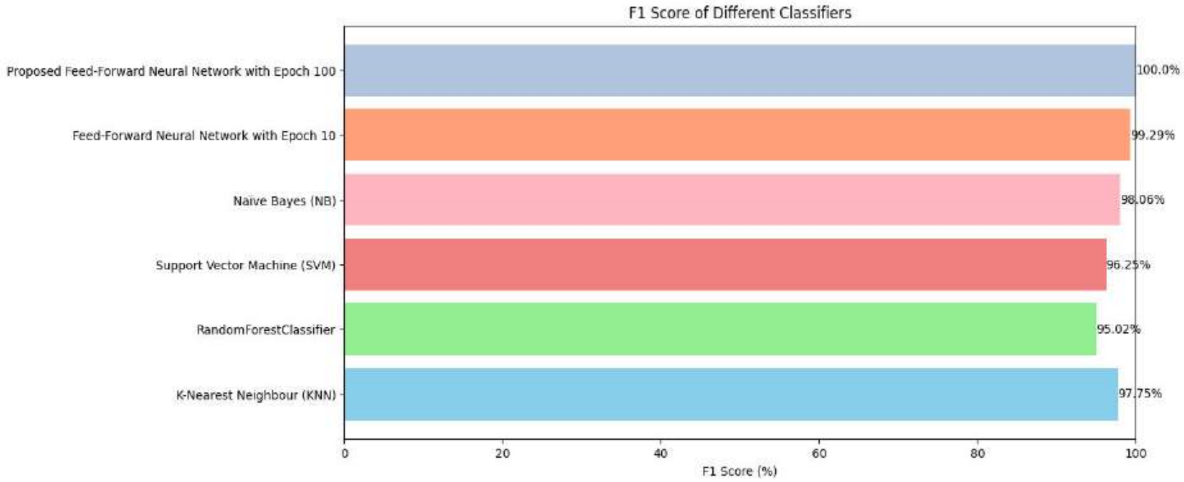


Figure 10: The F1 scores of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 10 shows the F1 scores of various classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 achieves the highest F1 score of 100%, followed closely by the Feed-Forward Neural Network with Epoch 10 with an F1 score of 99.29%. The Naïve Bayes (NB) classifier has an F1 score of 98.06%, while the K-Nearest Neighbour (KNN) achieves 97.75%. The Support Vector Machine (SVM) has an F1 score of 96.25%, and the RandomForestClassifier has an F1 score of 95.02%. Each classifier is represented by a distinct color, providing a clear visual comparison of their F1 scores.

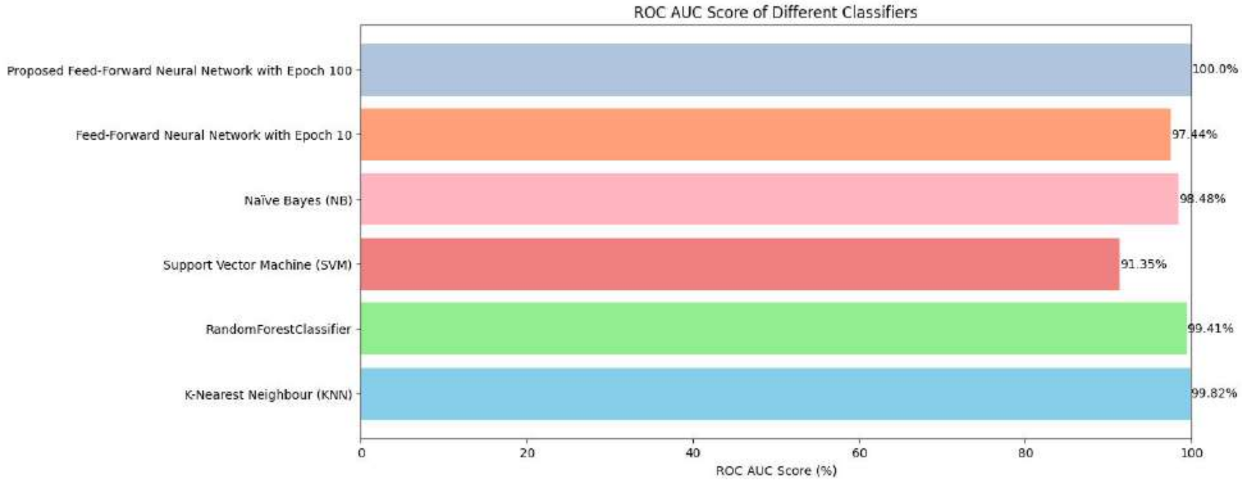


Figure 11: The ROC AUC scores of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 11 displays the ROC AUC scores of various classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 achieves the highest ROC AUC score of 100%, followed by the K-Nearest Neighbour (KNN) with a score of 99.82%. The RandomForestClassifier has a ROC AUC score of 99.41%, and the Naïve Bayes (NB) classifier achieves 98.48%. The Feed-Forward Neural Network with Epoch 10 has a ROC AUC score of 97.44%, while the Support Vector Machine (SVM) has the lowest score among the classifiers at 91.35%. Each bar is uniquely colored to distinguish between the classifiers, providing a clear visual comparison of their ROC AUC scores.

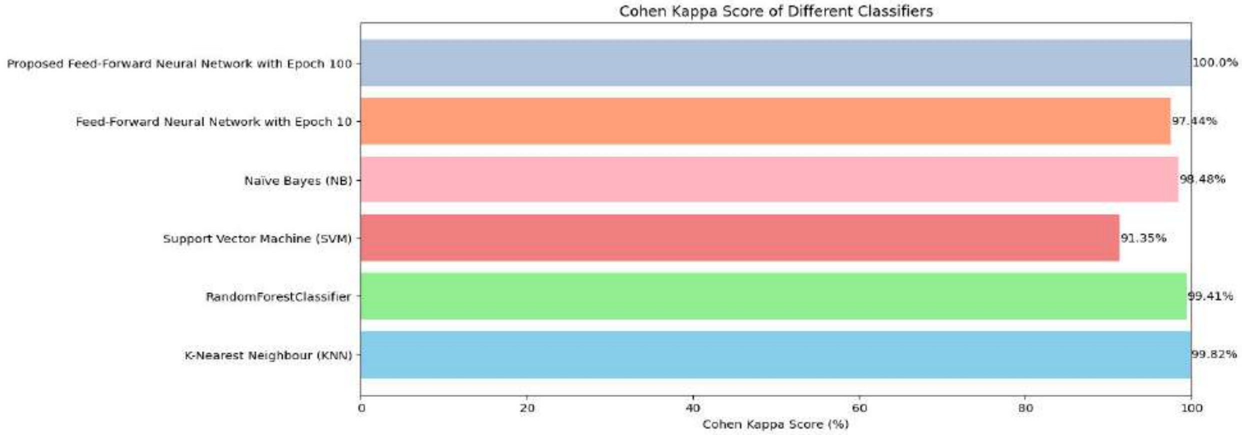


Figure 12: The ROC AUC scores of proposed Feed-Forward Neural Network with Epoch 100 and the Feed-Forward Neural Network

The figure 12 illustrates the Cohen Kappa scores of various classifiers on a given KDD Cup 99 dataset. The Proposed Feed-Forward Neural Network with Epoch 100 achieves the highest Cohen Kappa score of 100%, followed by the K-Nearest Neighbour (KNN) with a score of 99.82%. The RandomForestClassifier has a Cohen Kappa score of 99.41%, and the Naïve Bayes (NB) classifier achieves 98.48%. The Feed-Forward Neural Network with Epoch 10 has a Cohen Kappa score of 97.44%, while the Support Vector Machine (SVM) has the lowest score among the classifiers at 91.35%. Each bar is colored differently to clearly distinguish between the classifiers, providing a visual comparison of their Cohen Kappa scores.

VI. CONCLUSION

The evaluation of various classifiers on the dataset has demonstrated significant differences in their performance across multiple metrics, including accuracy, precision, recall, F1 score, ROC AUC score, and Cohen Kappa score. The Proposed Feed-Forward Neural Network with 100 epochs consistently outperformed other classifiers, achieving perfect scores in most metrics. This model's superior performance underscores its robustness and reliability in handling complex classification tasks. The K-Nearest Neighbour (KNN) classifier also exhibited high performance, particularly in the ROC AUC and Cohen Kappa scores, making it a competitive alternative for scenarios where computational simplicity and interpretability are prioritized. The RandomForestClassifier, while slightly less accurate, showed strong performance in terms of ROC AUC and Cohen Kappa scores, highlighting its effectiveness in various classification contexts. Support Vector Machine (SVM) and Naïve Bayes (NB) classifiers, although not leading in any single metric, provided reliable and consistent results, demonstrating their utility in specific applications where their unique strengths are advantageous. The comprehensive analysis indicates that while traditional classifiers like KNN, RandomForest, SVM, and NB maintain solid performance, advanced neural network models, particularly the Proposed Feed-Forward Neural Network with 100 epochs, offer superior accuracy and consistency across all evaluated metrics. These findings suggest a clear advantage in leveraging deep learning techniques for complex classification tasks, promoting further research and application of such models in practical scenarios.

REFERENCES

1. Samha, Amani K., Nidhi Malik, Deepak Sharma, and Papiya Dutta. "Intrusion detection system using hybrid convolutional neural network." Mobile Networks and Applications (2023): 1-13.

2. Vishwakarma, Monika, and Nishtha Kesswani. "DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT." Decision Analytics Journal 5 (2022): 100142.

3. Gowdhaman, V., and R. Dhanapal. "An intrusion detection system for wireless sensor networks using deep neural network." Soft Computing 26, no. 23 (2022): 13059-13067.

4. Almutlaq, Samah, Abdelouahid Derhab, Mohammad Mehedi Hassan, and Kuljeet Kaur. "Two-stage intrusion detection system in intelligent transportation systems using rule extraction methods from deep neural networks." IEEE Transactions on Intelligent Transportation Systems (2022).

5. Udas, Pritom Biswas, Md Ebtidaul Karim, and Kowshik Sankar Roy. "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks." Journal of King Saud University-Computer and Information Sciences 34, no. 10 (2022): 10246-10272.

6. Lo, Wai Weng, Siamak Layeghy, Mohanad Sarhan, Marcus Gallagher, and Marius Portmann. "E-graphsage: A graph neural network based intrusion detection system for iot." In NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1-9. IEEE, 2022.

7. Thirimanne, Sharuka Promodya, Lasitha Jayawardana, Lasith Yasakethu, Pushpika Liyanaarachchi, and Chaminda Hewage. "Deep neural network based real-time intrusion detection system." SN Computer Science 3, no. 2 (2022): 145.

8. Mandru, Deena Babu, M. Aruna Safali, N. Raghavendra Sai, and G. Sai Chaitanya Kumar. "Assessing deep neural network and shallow for network intrusion detection systems in cyber security." In Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021, pp. 703-713. Springer Singapore, 2022.

9. Habeeb, Mohammed Sayeeduddin, and T. Ranga Babu. "Network intrusion detection system: a survey on artificial intelligence-based techniques." Expert Systems 39, no. 9 (2022): e13066.

10. Halbouni, Asmaa, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. "CNN-LSTM: hybrid deep neural network for network intrusion detection system." IEEE Access 10 (2022): 99837-99849.

11. Vanin, Patrick, Thomas Newe, Lubna Luxmi Dhirani, Eoin O'Connell, Donna O'Shea, Brian Lee, and Muzaffar Rao. "A study of network intrusion detection systems using artificial intelligence/machine learning." Applied Sciences 12, no. 22 (2022): 11752.

12. Ghanem, Waheed Ali HM, Sanaa Abduljabbar Ahmed Ghaleb, Aman Jantan, Abdullah B. Nasser, Sami Abdulla Mohsen Saleh, Amir Ngah, Arifah Che Alhadi et al. "Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks." *IEEE Access* 10 (2022): 76318-76339.

13. Ullah, Safi, Jawad Ahmad, Muazzam A. Khan, Eman H. Alkhammash, Myriam Hadjouni, Yazeed Yasin Ghadi, Faisal Saeed, and Nikolaos Pitropakis. "A new intrusion detection system for the internet of things via deep convolutional neural network and feature engineering." *Sensors* 22, no. 10 (2022): 3607.

14. Belarbi, Othmane, Aftab Khan, Pietro Carnelli, and Theodoros Spyridopoulos. "An intrusion detection system based on deep belief networks." In *International Conference on Science of Cyber Security*, pp. 377-392. Cham: Springer International Publishing, 2022.

15. Sheikhi, Saeid, and Panos Kostakos. "A novel anomaly-based intrusion detection model using psogwo-optimized bp neural network and ga-based feature selection." Sensors 22, no. 23 (2022): 9318.

16. Zivkovic, Miodrag, Nebojsa Bacanin, Jelena Arandjelovic, Ivana Strumberger, and K. Venkatachalam. "Firefly algorithm and deep neural network approach for intrusion detection." In *Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2021*, pp. 1-12. Singapore: Springer Nature Singapore, 2022.

17. Venkateswaran, N., and K. Umadevi. "Hybridized Wrapper Filter Using Deep Neural Network for Intrusion Detection." *Computer Systems Science & Engineering* 42, no. 1 (2022).

18. Ullah, Safi, Jawad Ahmad, Muazzam A. Khan, Mohammed S. Alshehri, Wadii Boulila, Anis Koubaa, Sana Ullah Jan, and M. Munawwar Iqbal Ch. "TNN-IDS: Transformer neural network-based intrusion detection system for MQTT-enabled IoT Networks." *Computer Networks* 237 (2023): 110072.

19. Alqahtani, Hamed, and Gulshan Kumar. "A deep learning-based intrusion detection system for in-vehicle networks." *Computers and Electrical Engineering* 104 (2022): 108447.

20. Mani, Subalakshmi, Bose Sundan, Anitha Thangasamy, and Logeswari Govindaraj. "A new intrusion detection and prevention system using a hybrid deep neural network in cloud environment." In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021*, pp. 981-994. Singapore: Springer Nature Singapore, 2022.

21. Shenfield, Alex, David Day, and Aladdin Ayesh. "Intelligent intrusion detection systems using artificial neural networks." Ict Express 4, no. 2 (2018): 95-99.