Enhancing Network Security Through Machine Learning-Based Intrusion Detection

Mohan Babu Kunchala
A20524765
mkunchala@hawk.iit.edu

Sanjitha Reddy Pathuri
A20524383
spathuri2@hawk.iit.edu

Sirisha Gandham
A20544789
sgandham1@hawk.iit.edu

Intermediate Project Report

1. Project Introduction

Concerns about safeguarding personal and professional data have escalated significantly in the current era of digitization. Network Intrusion Detection has become a pivotal cybersecurity defense, aiming to identify and prevent vulnerabilities from being exploited by malicious actors within the networks. However, Traditional rule based approaches struggle to keep up with increasingly sophisticated and unique attacks, as attackers continue to improve their methods to evade detection.

Machine Learning has emerged as a vital technology in Network Intrusion Detection to address this problem. Various Machine Learning techniques can be extremely helpful in identifying possible threats by analyzing vast network traffic datasets and can recognise patterns in network traffic. The objective of this project is to use the UNSW-NB15 dataset to create an efficient Machine Learning based method for network intrusion detection. The goal is to precisely categorize the dataset's normal and attack traffic into distinct attack groups.

By investigating the possibilities of neural networks (NNs), like MLP classifiers, Convolutional Neural Networks (CNNs), and Recurrent Convolutional Neural Networks (RCNNs), for network intrusion detection, the proposed method aims to expand on previous research. It will also explore the benefits of feature selection techniques in enhancing classifier performance. Furthermore, the project will also evaluate how well NNs and conventional classifiers perform in comparison.

The initial stage of this project involves data preprocessing, including the Extraction and Cleaning of Data. Before the UNSW-NB15 dataset is fed into machine learning models, it must be cleaned up of missing values and categorical variables. Exploratory Data Analysis (EDA) is also crucial to understand the feature distribution, their correlations, and identifying outliers in the data.

Feature selection aims to identify the most relevant features that enhance classifier accuracy. The project will employ a Mutual Information-based score to identify the top k features. Different Neural Network types will be selected and their effectiveness will be assessed against existing established classifiers using a range of metrics, including F1-Score, AUC-ROC, Accuracy, Precision, and Recall.

In summary, the main goal of this project is to experiment with cutting-edge techniques for Network Intrusion Detection in order to improve the accuracy of the classifier. The ultimate aim is to develop a robust Network Intrusion Detection system capable of identifying even the most innovative and complex attacks, which could bring significant benefits to organizations around the globe.

2. Problem Description

One of the major concerns for organizations globally is cybersecurity. Network Intrusion Detection plays a critical role in cybersecurity, identifying and mitigating attempts by intruders to exploit network vulnerabilities. Since the attackers are often changing their tactics to bypass the detection systems, it is difficult to identify these attacks.

Because they are unable to recognise new and sophisticated attacks, traditional rule based methods are ineffective for detecting network intrusions. Machine learning techniques can learn from massive datasets and recognise patterns in network traffic, they can be extremely helpful in identifying these attacks.

Using the UNSW-NB15 dataset, the proposed project seeks to provide an efficient machine learning based method for network intrusion detection. The objective is to accurately classify the dataset's normal and attack traffic into distinct attack types. The dataset contains both normal and attack traffic, and the aim is to accurately classify the traffic into different attack categories.

3. Description of the Dataset used in the Project

The project utilizes the UNSW-NB15 dataset, available on Kaggle. This network traffic dataset contains labeled traffic data generated by a network simulator and contains both legitimate and illicit traffic. It includes a total of 49 features, encompassing nominal, integer, float, timestamp, and binary data types.

While the original dataset has 49 features, our primary datasets used for training and testing only include 45 features. The excluded features are the first four, representing source and destination IP addresses and ports. The remaining features include protocol type, connection duration, number of bytes transmitted/received, and more.

The UNSW-NB15 dataset is split into 70-30 training and testing sets. The training set contains 175,341 records, and the testing set has 82,332 records. These sets are provided in separate CSV files alongside the dataset.

The dataset contains a binary variable indicating whether the connection was classified as a normal transaction or an attack (0 for normal, 1 for attack) and a categorical variable “attack_cat” indicating the type of attack (if any) associated with each connection. This variable has nine categories such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

The UNSW-NB15 dataset is a valuable resource for network intrusion detection researchers due to its real world nature and diverse range of network traffic. It serves as an excellent benchmark dataset for evaluating and comparing the performance of various machine learning algorithms and intrusion detection systems.

4. Work Completed So Far

The UNSW-NB15 dataset was divided into training and testing sets for analysis which were provided as separate CSV files with the dataset. We have partitioned the code for the Training dataset into one notebook and the Testing dataset into another notebook, so we use each dataset in its own notebook. Milestones achieved thus far include:

1. **Dataset Preparation:** Extraction, Cleaning, and Preprocessing of data in both training and testing datasets. The datasets contained missing values and categorical variables that required addressing before feeding the data into ML models. We addressed this by dropping unwanted columns, using LabelEncoder() to encode categorical variables, and using StandardScaler() to standardize numerical variables. This process was implemented in both the training and testing dataset notebooks.

2. **Exploratory Data Analysis (EDA):** EDA is essential to understand the data's distribution, feature correlations, and identify outliers. Visualizations like histograms, heatmaps, and box plots were generated using EDA in both the training and testing dataset notebooks. These visualizations provided valuable insights into the data and its distribution across various factors.

3. **Feature Selection:** Feature selection aims to identify the most relevant features that can improve classifier performance. We employed Mutual Information Score to select the top k features(k=10) using SelectKBest. This approach helps eliminate noise and redundancy in the data, ultimately enhancing the efficiency of our ML models. Feature selection was implemented in both training and testing dataset notebooks.

4. **Model Selection, Training and Evaluation (Traditional classifiers):** We have selected and trained traditional classifiers/supervised learning models like Logistic Regression, Decision Tree, Random Forest, SVM, and XGBoost. The performance of the traditional classifiers was evaluated on both training and testing data using metrics like Accuracy, Precision, Recall, F1-Score, Precision-Recall Curves, ROC Curves, and AUC Score Plots. Classification reports and confusion matrices were generated for each supervised model to identify misclassified samples and analyze the results.

5. **Model Selection, Training and Evaluation (Using neural networks):** Following feature selection, we will move on to selecting and training supervised learning models using neural networks (both the training and testing dataset). As planned, different Neural Network types like CNNs, RCNNs, and NNs (MLP Classifier) will be chosen and compared with the already evaluated traditional classifiers. The performance of the supervised learning models using neural networks will be evaluated using the same metrics employed for supervised models: Accuracy, Precision, Recall, F1-Score, Precision-Recall Curves, ROC Curves, and AUC Score Plots. Classification reports and confusion matrices will also be generated for each model to analyze misclassified samples and results. This evaluation will be conducted in both training and testing dataset notebooks.

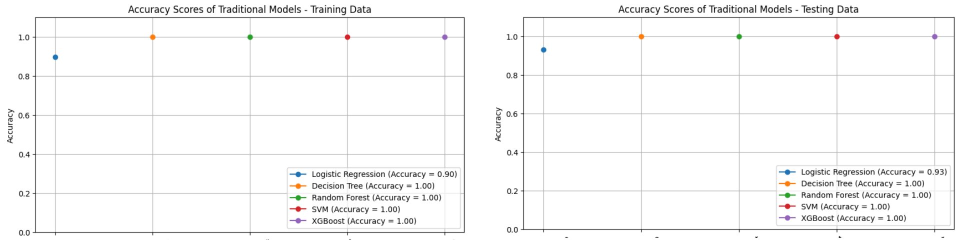

Figure 1: Accuracy of Traditional Models - Training V/S Testing Dataset

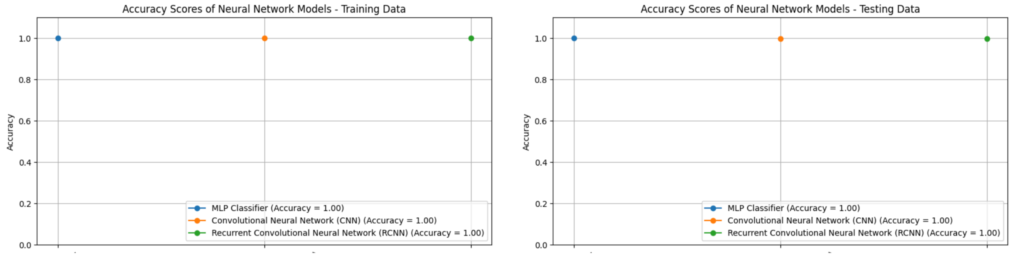

Figure 2: Accuracy of Neural Network Models - Training V/S Testing Dataset

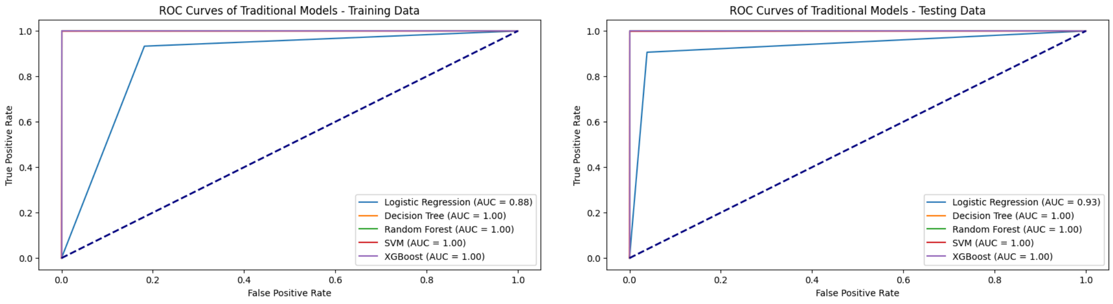

Figure 3: ROC Curves of Traditional Models - Training V/S Testing Dataset

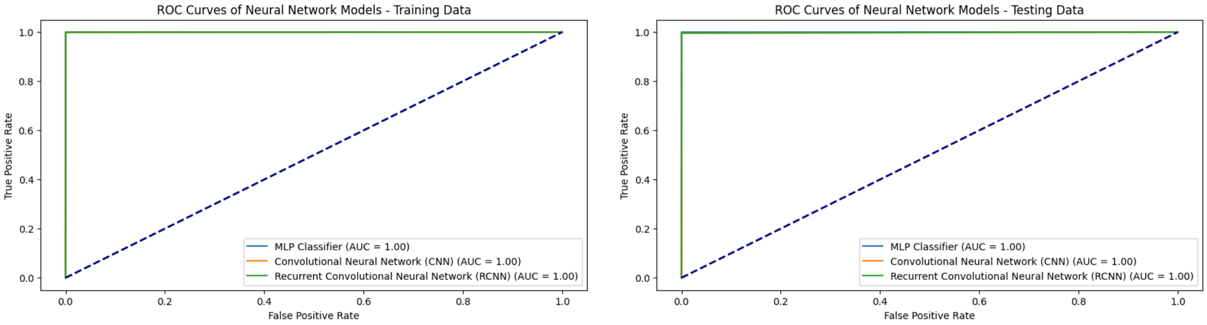

Figure 4: ROC Curves of Neural Network Models - Training V/S Testing Dataset

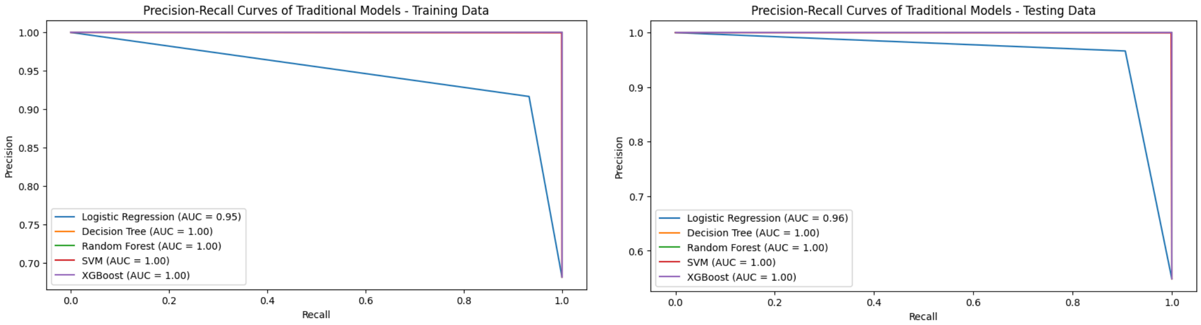

Figure 5: Precision-Recall Curves of Traditional Models - Training V/S Testing Dataset

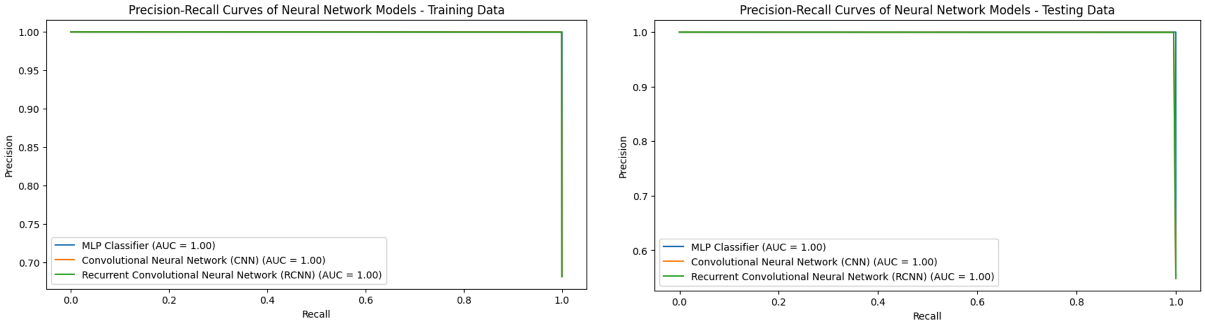

Figure 6: Precision-Recall Curves of Neural Network Models - Training V/S Testing Dataset

5. What Remains To Be Done

The next steps involve working with supervised learning models using neural networks:

Comparison and Conclusion: We will compare the performance of Neural network models with the traditional models. This comprehensive analysis will form the basis of our project conclusions. By comparing the effectiveness of both learning approaches for Network Intrusion Detection, we can draw valuable insights and determine the optimal approach for this specific task.

By following these steps, we aim to develop a robust Network Intrusion Detection system leveraging Machine Learning's capabilities. This system should be effective in identifying even the most sophisticated attacks, ultimately enhancing network security for organizations.