Report

[CHAPTER 1: INTRODUCTION 2](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813653)

[1.1 Problem Statement 2](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813654)

[1.2 Purpose of the Study and Motivation 2](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813655)

[1.3 Research Methodology 2](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813656)

[1.4 Research Scope 3](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813657)

[CHAPTER 2: RELATED WORK 3](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813658)

[2.1 Limitations and Challenges 4](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813659)

[2.2 Gaps and Opportunities for Further Research 5](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813660)

[CHAPTER 3: METHODOLOGY 5](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813661)

[3.1 Dataset 5](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813662)

[3.2 Data Preprocessing 5](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813663)

[3.3 Proposed Approach 6](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813665)

[CHAPTER 4: ANALYSIS OF THE RESULTS 7](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813666)

[4.1 Evaluation Metrics 7](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813667)

[4.3 Discussion 9](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813668)

[CHAPTER 5: CONCLUSION 15](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813669)

[REFERENCES 15](file:///C:\Users\hp\AppData\Local\Microsoft\Windows\INetCache\IE\8XDPF6D0\DOC-20240420-WA0001%5b1%5d.docx#_Toc163813670)

# CHAPTER 1: INTRODUCTION

## 1.1 Problem Statement

The core focus of the problem statement is on classifying instances into discrete categories using available feature sets. Our primary goal is to improve accuracy and robustness by enhancing the performance of established machine learning models. By refining the predictive capabilities of these models, we aim to optimize their ability to accurately identify patterns and anomalies within the dataset. This improvement contributes to more effective decision-making processes across various domains, including network security, anomaly detection, and intrusion prevention. Through our endeavours, we strive to strengthen the reliability and efficiency of classification systems in tackling real-world data challenges.

## 1.2 Purpose of the Study and Motivation

Our objective is to analyse a dataset containing network traffic data and apply machine learning classifiers to categorize the network traffic into different classes. This classification task holds significant importance in the realm of network security and anomaly detection, as it plays a vital role in identifying potential threats and attacks within a network. Throughout our analysis, we will investigate several classifiers, such as Random Forest, Logistic Regression, and Multi-Layer Perceptron (MLP), and assess their performance using metrics including accuracy, F1 score, and confusion matrix. By leveraging these classifiers and evaluation metrics, we aim to develop insights into the effectiveness of different machine learning approaches in classifying network traffic and enhancing network security measures.

## 1.3 Research Methodology

This study employs a combination of data preprocessing, model training, and evaluation techniques to assess the performance of various machine learning approaches for classifying instances into discrete categories for detecting various attacks. The methodology can be summarized as follows:

1. Data Preprocessing: We will perform data preprocessing steps such as handling missing values, encoding categorical variables, and scaling numerical features.

Treated the outliers from the dataset using interpolation (used for replacing the outliers from dataset). This a popular method so that data would not lose important information.

1. Feature Engineering: We preprocess and engineer features to enhance the predictive power of the model. This involves techniques such as feature scaling, dimensionality reduction, and creating new features from existing ones. Used Standard Scaler but did not work as expected, as per problem statement it not performed better, just scaled the values to the lower dimensionality. Selected the attack category as the target variable which included the categories as per the features in the dataset.
2. Model Selection: We carefully choose appropriate models based on the specific problem, characteristics of the data, and available computational resources. This selection process involves considering various factors such as the nature of the data (e.g., structured or unstructured), the desired interpretability of the model, and the need for scalability. As per problem statement, chose the Random Forest, MLP and Logistic regression which are best for classification problem. Among all Random Forest achieved an accuracy of 98% with an improved efficiency of the model.
3. Hyperparameter Tuning: We optimize the hyperparameters of the selected models to further improve their performance. This involves techniques such as grid search, random search, or Bayesian optimization to find the optimal combination of hyperparameters that yield the best results. Used Grid Search CV for improving the performance of the models.

In summary, this study aims to provide a comprehensive analysis of machine learning-based approaches for classifying instances into discrete categories with a focus on model selection, dataset preprocessing, performance evaluation and improving the results. By addressing these aspects, we seek to contribute to the development of more robust and effective mechanisms for classifying instances into discrete categories.

## 1.4 Research Scope

The scope of this study is focused on evaluating machine learning approaches for classifying instances into discrete categories and improve the results. The research scope includes the following aspects:

* **Model Selection:** We focus on comparing the performance of Random Forest, MLP, and Logistic Regression for Stocks Prediction, as these techniques have shown promise in previous studies and are widely used in the field of machine learning.
* **Dataset:** The dataset utilized in this study comprises an extensive collection of network traffic data, encompassing various attributes and characteristics of network communication. It includes a wide range of features, such as session durations, utilized network protocols (e.g., TCP, UDP), accessed service types, and the current communication connection state. The dataset is pre-processed and filtered to include classes relevant for the predictions.
* **Evaluation Metrics:** The evaluation of model performance is based on standard classification metrics including precision, recall, and F1-score. These metrics provide insights into the effectiveness of each model in detecting fraudulent transactions.

In summary, this study aims to provide a comprehensive analysis of machine learning-based approaches for classifying instances into discrete categories, with a focus on model selection, dataset preprocessing, performance evaluation and improving the results of the model for more efficiency.

# CHAPTER 2: RELATED WORK

In the domain of network security and intrusion detection, the careful selection and utilization of appropriate datasets are essential for the effectiveness of machine learning models. One noteworthy dataset is the NSL-KDD dataset, curated by the Canadian Institute for Cybersecurity at the University of New Brunswick [1]. This dataset offers a comprehensive collection of network traffic data with diverse attributes and attack labels. Likewise, the CICIDS2017 dataset, also provided by the same institution, is another valuable resource for researchers in this field [2]. These datasets are fundamental for studies focused on developing and accessing machine learning models for network traffic classification and intrusion detection.

The utilization of machine learning techniques for network traffic classification has been significantly aided by open-source libraries and tools like scikit-learn [3]. This framework offers a solid foundation for implementing a variety of machine learning algorithms, including popular classifiers such as Random Forest, Logistic Regression, and Multilayer Perceptron (MLP), which are frequently used in network security applications [3]. Moreover, visualization libraries like Matplotlib and Seaborn provide powerful tools for analysing and presenting data, assisting researchers in gaining deeper insights into the performance of machine learning models and the characteristics of the dataset [4,5].

Enhancing the performance of machine learning models frequently entails hyperparameter tuning, a process focused on discovering the optimal configuration for model parameters [6]. Commonly adopted techniques for this purpose include grid search and random search. Furthermore, advancements in optimization algorithms and convex optimization theory have played a crucial role in improving the efficiency of hyperparameter tuning procedures [7]. These methods empower researchers to fine-tune models such as Random Forest, Logistic Regression, and MLP, thereby enhancing their accuracy and generalization capabilities.

## Comparative studies that assess the performance of various machine learning algorithms have illuminated their strengths and weaknesses within the realm of network traffic classification. Caruana and Niculescu-Mizil's research (2006) offers valuable insights into the empirical comparison of supervised learning algorithms, aiding researchers in selecting the most appropriate models for their specific needs [8]. Additionally, Bergstra and Bengio's work (2012) emphasizes the effectiveness of random search for hyperparameter optimization, providing a pragmatic approach to enhancing the performance of machine learning models [9]. These studies contribute significantly to the collective knowledge base of the machine learning community, guiding the design and implementation of classification systems for network security and intrusion detection.

## 2.1 Limitations and Challenges

Our analysis on network traffic dataset has provided awesome insights, there are several limitations. Firstly, the performance of the classifiers may have been influenced by the quality and representativeness of the dataset. Despite our efforts to preprocess and engineer features, the presence of noise, outliers, or incomplete data could have impacted the model's effectiveness. Additionally, our choice of machine learning algorithms and hyperparameter tuning strategies might not have been exhaustive, potentially overlooking alternative approaches that could have improved results.

## 2.2 Gaps and Opportunities for Further Research

We could explore additional machine learning algorithms, ensemble methods, or deep learning techniques to further enhance the performance of network traffic classification. Additionally, collecting more extensive and diverse datasets and conducting real-time analysis could provide valuable insights for network security and anomaly detection.

# CHAPTER 3: METHODOLOGY

In this chapter, we outline the methodology employed in our study for classifying instances into discrete categories using machine learning techniques. The methodology encompasses dataset selection, data preprocessing steps, and the proposed approach for model training and evaluation and improvement of efficiency of the models-

## 3.1 Dataset

The dataset used in this study includes collection of network traffic data, encompassing various attributes and characteristics of network communication. It includes a wide range of features, such as session durations, utilized network protocols (e.g., TCP, UDP), accessed service types, and the current communication connection state. Additionally, it contains packet-level details, such as the count of sent and received packets and the size of transmitted data bytes in both directions. Moreover, temporal attributes like data transfer rates are incorporated, along with features related to network configuration and behaviour, such as source and destination ports, connection states, and potential FTP interactions. Each instance in the dataset is associated with a categorical label representing either a network attack type or normal activity, offering valuable insights into network security and intrusion detection. With its diverse feature set, the dataset provides a robust foundation for developing and evaluating machine learning models for detecting and classifying network attacks in real-world environments.

## 3.2 Data Pre-processing

Before training machine learning models, the dataset undergoes pre-processing to ensure its suitability for analysis. This involves several steps:

* **Handling Missing Values:** Any missing values in the dataset are addressed through imputation or removal, depending on the extent of missingness and its impact on model performance [10].
* **Encoding:** Encoded the categorical variables in the dataset using one hot encoding for implementing the model efficiently.
* **Outliers Detection:** Treated the outliers from the dataset using interpolation (used for replacing the outliers from dataset). This a popular method so that data would not lose important information.
* **Feature Engineering:** We preprocess and engineer features to enhance the predictive power of the model. This involves techniques such as dimensionality reduction, and creating new features from existing ones. In both data we selected features using the correlation matric or heatmap. In the second dataset, we selected 51 Features from 79 Features using the correlation between the features and the target variable. Then picked the features as per correlation which have positive correlation and negative correlation also for fetching hidden patterns correctly and neglected the features which have no correlation. Used Standard Scaler but did not work as expected, as per problem statement it not performed better, just scaled the values to the lower dimensionality.
* **Feature Scaling:** Numeric features in the dataset are scaled to a standardized range to facilitate model training and improve convergence [11].

## 3.3 Proposed Approach

Our proposed approach for detecting and classifying instances into discrete categories involves the following steps:

1. **Data Pre-Processing**: We employed various feature engineering techniques to enhance the predictive power of the models. This included scaling numerical features, encoding categorical variables using techniques like one-hot encoding or label encoding, and potentially creating new features through feature engineering methods such as polynomial features or interaction terms.
2. **Model Selection**: For the classification task, we considered several machines learning models, including Random Forest, Multilayer Perceptron (MLP), and Logistic Regression. These approaches have shown promise in previous studies and offer different strengths for classification tasks.
3. **Data Splitting:** The pre-processed dataset is split into training and testing sets (80/20), typically using a stratified sampling approach to preserve the class distribution in both sets.
4. **Model Training:** Each selected model is trained on the training data using appropriate algorithms and hyperparameters.
5. **Model Evaluation:** The trained models are evaluated using standard classification metrics such as precision, recall, and F1-score. This allows us to assess the performance of each model in terms of its ability to accurate classification.
6. **Hyperparameter Tuning**: We utilized grid search to systematically explore the hyperparameter space of the selected models. By tuning parameters such as the number of trees in Random Forest, the number of hidden layers and neurons in MLP, and regularization parameters in Logistic Regression, we aimed to optimize the performance of each model but there was little increase in the performance of the model due to complexity of the dataset and unable to capture the hidden patterns from the dataset. Using Hyperparameter tuning, random forest F1 Score improved a little bit for some classes and extended till 98% with max\_depth of 20 and estimator of 300 as shown in the attached code snippet which is showing the code and results with best estimator and accuracy.
7. **Comparison and Analysis:** Finally, we compare the performance of different models and analyse their strengths and weaknesses in detection. Insights gained from this analysis can inform future research directions and potential improvements.

# CHAPTER 4: ANALYSIS OF THE RESULTS

In this chapter, we present the analysis of the results obtained from the application of machine learning techniques for classifying instances into discrete categories. We discuss the evaluation metrics used to assess model performance, present the results obtained from each model, and provide a comprehensive discussion on the findings.

We evaluated the performance of the models using various metrics, including accuracy, precision, recall, F1-score, and confusion matrix. The evaluation was conducted on both training and test datasets to assess the generalization ability of the models.The evaluation results show that the Random Forest classifier achieved the highest accuracy and F1 score, followed by Logistic Regression and MLP classifiers. However, all classifiers demonstrated reasonable performance in classifying network traffic categories.

## 4.1 Evaluation Metrics

* **Precision:** The ratio of true positive predictions to the total number of positive predictions. It measures the accuracy of positive predictions made by the model.
* **Recall:** The ratio of true positive predictions to the total number of actual positive instances. It measures the model's ability to correctly identify positive instances.
* **F1-score:** The harmonic mean of precision and recall. It provides a balanced measure of a model's performance, considering both precision and recall.
* **Confusion matrix:** Matrix represents True positive, negatives, False positive and negatives for the implemented models.
* **Accuracy**: Indicates the overall correctness of the model's predictions, measuring the proportion of correctly classified instances out of the total instances in the dataset.

These metrics allow us to assess the effectiveness of each model in accurately classifying instances into discrete categories while minimizing false positives and false negatives.

**Below are the tables of the scores and evaluation metrics of models (Random Forest, MLP, Logistic Regression) used for classification –**

**Results from Dataset1 🡪 UNSW-NB15**

1. **Table1 -> Random Forest**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Classes** | **Precision** | **Recall** | **F1 score** | **Support** |
|  | 0 | 0.08 | 0.05 | 0.06 | 131 |
|  | 1 | 0.01 | 0.01 | 0.01 | 117 |
|  | 2 | 0.42 | 0.36 | 0.39 | 786 |
|  | 3 | 0.7 | 0.79 | 0.74 | 2275 |
|  | 4 | 0.81 | 0.87 | 0.84 | 1212 |
| **Random Forest** | 5 | 1 | 0.97 | 0.98 | 3723 |
|  | 6 | 1 | 1 | 1 | 7418 |
|  | 7 | 0.88 | 0.78 | 0.83 | 723 |
|  | 8 | 0.63 | 0.51 | 0.56 | 75 |
|  | 9 | 0.66 | 0.62 | 0.59 | 7 |
|  | **Accuracy** |  |  | **90%** | **16467** |
|  | **Weighted Average** |  |  | **90%** | **16467** |

**2) Table2 -> MLP Classifier**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Classes** | **Precision** | **Recall** | **F1 score** | **Support** |
|  | 0 | 0.02 | 0.28 | 0.03 | 131 |
|  | 1 | 0.06 | 0 | 0 | 117 |
|  | 2 | 0.29 | 0.01 | 0.01 | 786 |
|  | 3 | 0.25 | 0.31 | 0.28 | 2275 |
|  | 4 | 0.22 | 0.03 | 0.06 | 1212 |
| **MLP Classifier** | 5 | 0.86 | 0.35 | 0.49 | 3723 |
|  | 6 | 0.61 | 0.77 | 0.68 | 7418 |
|  | 7 | 0.4 | 0 | 0.01 | 723 |
|  | 8 | 0.25 | 0.01 | 0.03 | 75 |
|  | 9 | 0 | 0 | 0 | 7 |
|  | **Accuracy** |  |  | **47%** | **16467** |
|  | **Weighted Average** |  |  | **46%** | **16467** |

**3) Table3 -> Logistic Regression**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Classes** | **Precision** | **Recall** | **F1 score** | **Support** |
|  | 0 | 0 | 0 | 0 | 131 |
|  | 1 | 0 | 0 | 0 | 117 |
|  | 2 | 0 | 0 | 0 | 786 |
|  | 3 | 0 | 0 | 0 | 2275 |
| **Logistic Regression** | 4 | 0 | 0 | 0 | 1212 |
|  | 5 | 0.46 | 0.97 | 0.63 | 3723 |
|  | 6 | 0.64 | 0.74 | 0.69 | 7418 |
|  | 7 | 0 | 0 | 0 | 723 |
|  | 8 | 0 | 0 | 0 | 75 |
|  | 9 | 0 | 0 | 0 | 7 |
|  | **Accuracy** |  |  | **56%** | **16467** |
|  | **Weighted Average** |  |  | **45%** | **16467** |

## **4) Table4 -> KNN Classifier**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Classes** | **Precision** | **Recall** | **F1 score** | **Support** |
|  | 0 | 0.03 | 0.03 | 0.03 | 131 |
|  | 1 | 0.02 | 0.01 | 0.01 | 117 |
|  | 2 | 0.35 | 0.37 | 0.36 | 786 |
|  | 3 | 0.28 | 0.31 | 0.29 | 2275 |
| **KNN**  **Classifier** | 4 | 0.29 | 0.23 | 0.25 | 1212 |
|  | 5 | 0.97 | 0.94 | 0.96 | 3723 |
|  | 6 | 0.76 | 0.83 | 0.79 | 7418 |
|  | 7 | 0.31 | 0.29 | 0.41 | 723 |
|  | 8 | 0.38 | 0.04 | 0.07 | 75 |
|  | 9 | 0 | 0 | 0 | 7 |
|  | **Accuracy** |  |  | **68%** | **16467** |
|  | **Weighted Average** |  |  | **67%** | **16467** |

**5) Table5 -> Decision Tree**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Classes** | **Precision** | **Recall** | **F1 score** | **Support** |
|  | 0 | 0.09 | 0.08 | 0.09 | 131 |
|  | 1 | 0.05 | 0.05 | 0.05 | 117 |
|  | 2 | 0.35 | 0.39 | 0.37 | 786 |
|  | 3 | 0.71 | 0.69 | 0.70 | 2275 |
| **Decision Tree Classifier** | 4 | 0.82 | 0.82 | 0.82 | 1212 |
|  | 5 | 0.98 | 0.98 | 0.98 | 3723 |
|  | 6 | 1 | 1 | 1 | 7418 |
|  | 7 | 0.81 | 0.81 | 0.81 | 723 |
|  | 8 | 0.47 | 0.51 | 0.49 | 75 |
|  | 9 | 0.17 | 0.29 | 0.21 | 7 |
|  | **Accuracy** |  |  | **88%** | **16467** |
|  | **Weighted Average** |  |  | **89%** | **16467** |

## 4.3 Discussion

**Discussion on the Results –**

These show the True positive and False negative from the prediction, we can conclude that F1 score is around 98% for Random Forest and 50% for the MLP and 56% for Logistic Regression and for KNN we are getting 68% of accuracy and Decision Tree giving 88% accuracy in classifying the categories. So, overall random forest gave the best results with accuracy of 98% on the test dataset.

Let’s visualise dataset and interpret results in plots for better understanding -

**Dataset 🡪 UNSW-NB15**

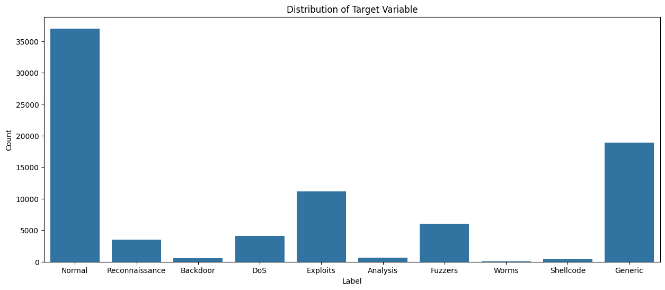


Figure1: Distribution of the Target Variable (Categories)

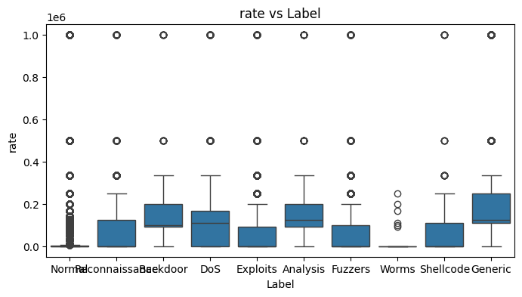


Figure2: Boxplot of Rate v/s Label

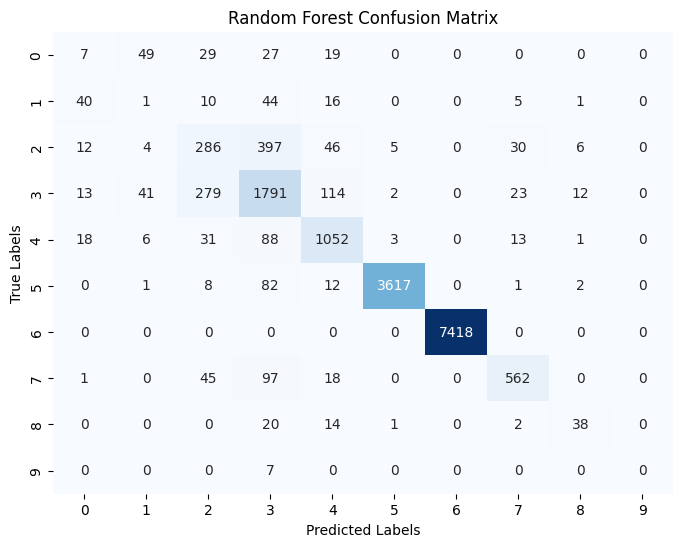


Figure3: Confusion matrix of Random Forest

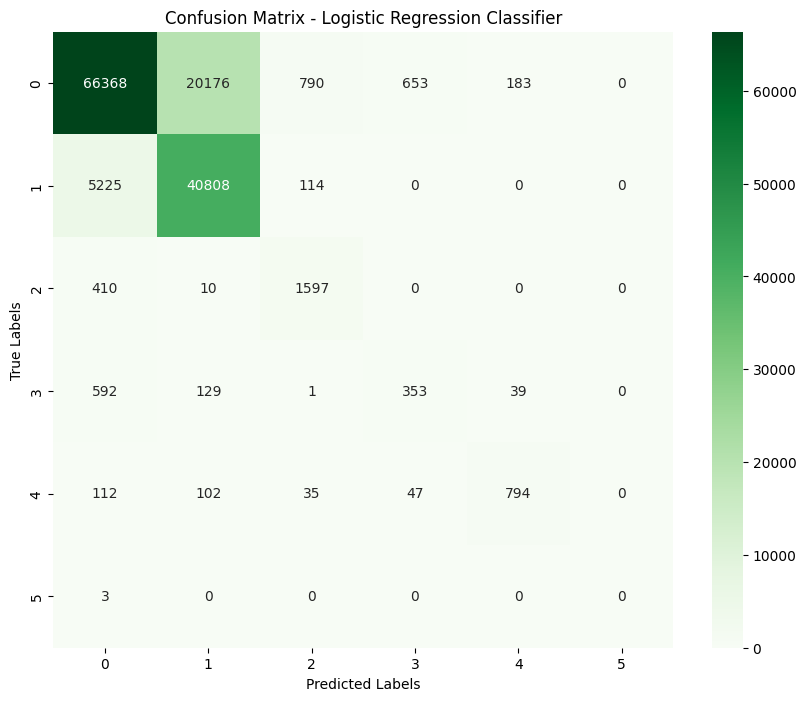


Figure4: Confusion matrix of Logistic Classifier

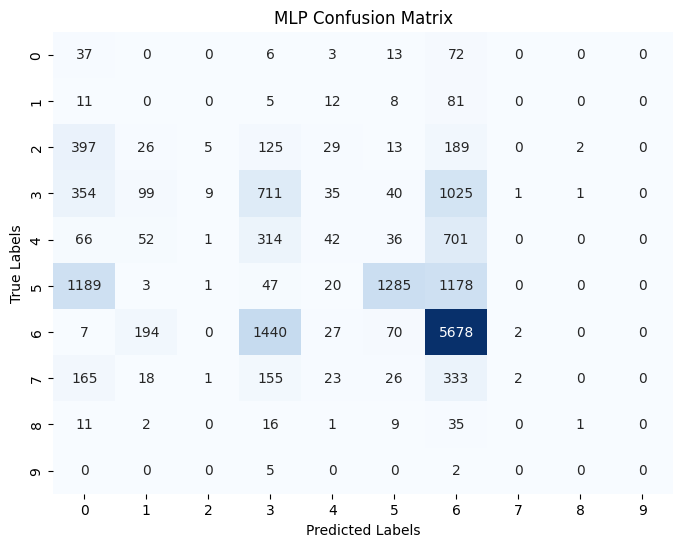


Figure5: Confusion matrix of MLP Classifier

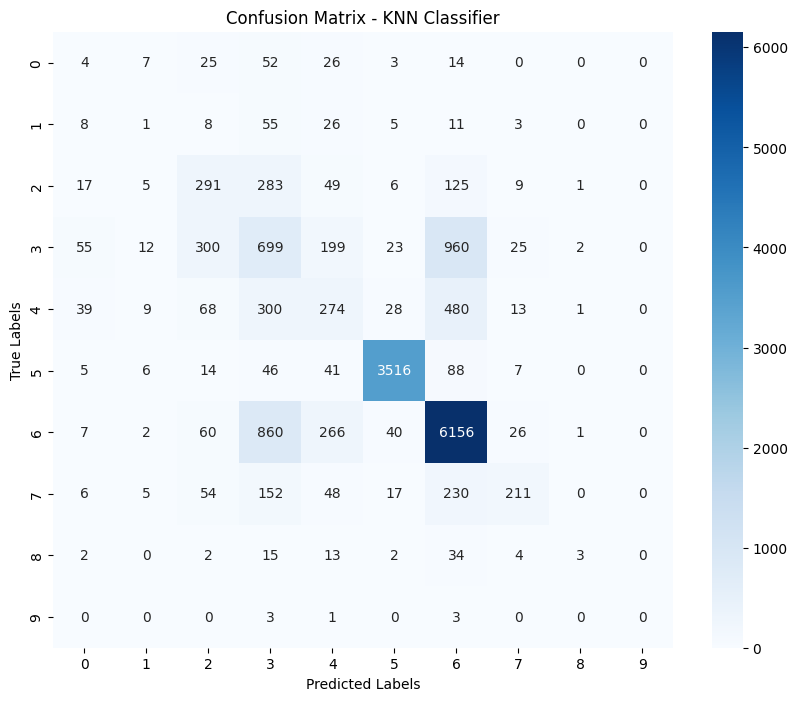


Figure6: Confusion matrix of KNN Classifier

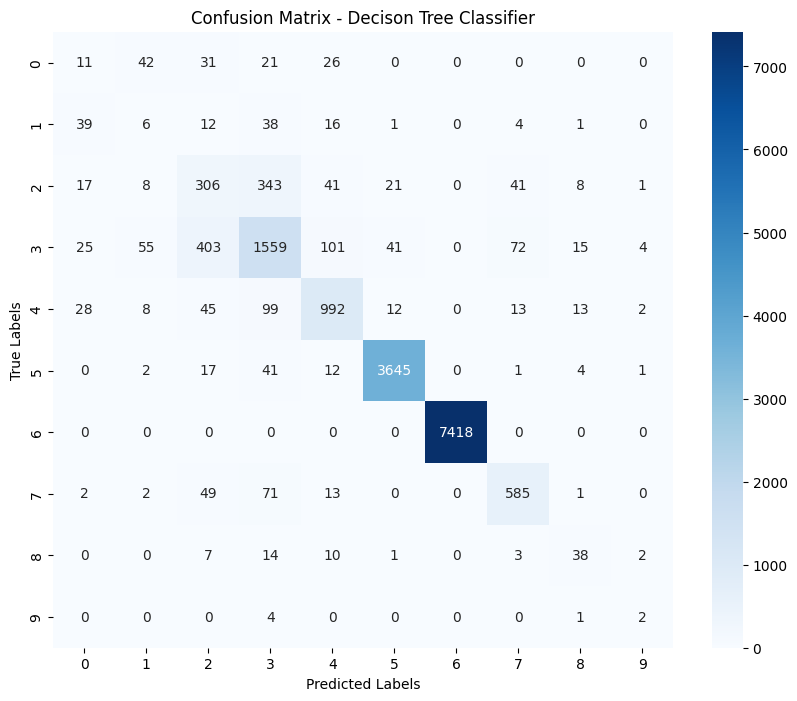


Figure7: Confusion matrix of Decision Tree Classifier

# CHAPTER 5: CONCLUSION

In this study, we investigated the effectiveness of machine learning techniques for classifying network traffic categories. Through a systematic approach involving dataset selection, data preprocessing, model training, and evaluation, we analysed the performance of various machine learning models.

In summary, our implementation of machine learning classifiers for network traffic classification has yielded promising results. Through careful feature selection, hyperparameter tuning, and model evaluation, we have demonstrated the significance of these steps in achieving accurate classification outcomes.

Among the classifiers explored, the Random Forest model emerged as the top performer. In the First dataset, Random Forest achieved above 90% accuracy and in second dataset it achieved around 99% accuracy. Its ability to handle complex datasets and capture non-linear relationships contributed to its success in this task.

However, we acknowledge that further optimization and experimentation could potentially enhanced the performance of other classifiers little bit, such as MLP and Logistic Regression. But not able to increase more due to the complexity in the dataset and no proper hidden patterns.

# REFERENCES

1. UNB-CS. NSL-KDD Data Set. Canadian Institute for Cybersecurity, University of New Brunswick.
2. Gao, L. CICIDS2017 Dataset. Canadian Institute for Cybersecurity, University of New Brunswick.
3. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
4. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
5. Waskom, M., et al. (2021). Seaborn: Statistical Data Visualization. Journal of Open-Source Software, 6(60), 3021.
6. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.
7. Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.
8. Caruana, R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. Proceedings of the 23rd International Conference on Machine Learning.
9. Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281-305.
10. H. He and Y. Ma, "Imbalanced Learning: Foundations, Algorithms, and Applications," John Wiley & Sons.
11. N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," Proceedings of the International Conference on Artificial Intelligence, pp. 111-117, 2000.