NETWORK INTRUSION DETECTION SYSTEM

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

ANJITH S M

ICT ACADEMY OF KERALA
THIRUVANANTHAPURAM, KERALA, INDIA
OCT 2024

LIST OF FIGURES

LIST OF ABBREVIATIONS

NIDS	:	Network Intrusion Detection System
DoS	:	Denial of Service
ML	:	Machine Learning
IoT	:	Internet of Things

CONTENTS

ABSTRACT

The Network Intrusion Detection System (NIDS) plays a vital role in the realm of cybersecurity by providing continuous monitoring of network traffic to identify suspicious and potentially harmful activities. In this project, we developed a machine learning-based NIDS that utilizes an available dataset to classify network traffic as either normal or indicative of an attack. By analysing various features—such as protocol type, traffic flow, and packet details—the system aims to detect a range of attack types, including denial-of-service (DoS) attacks, probing activities, exploits, and more.

Our NIDS automates the detection process by training a machine learning model to recognize patterns in network data. We evaluated several classification algorithms, including Random Forest, Logistic Regression, and Neural Networks, to determine the most accurate model for deployment. Once trained, this model can analyse incoming traffic in real-time, providing immediate alerts and insights into any ongoing security threats.

To improve usability, we integrated a web-based platform that serves as a real-time monitoring and detection interface. This user-friendly website allows network administrators to visualize live traffic data, access reports on detected attacks, and upload new network logs for analysis. Featuring interactive visualizations, alert notifications, and comprehensive logs for in-depth traffic analysis, the platform enhances the overall monitoring experience. Additionally, users have the option to retrain the model with new data directly through the website, ensuring that the intrusion detection system stays current and effective against evolving threats.

This project not only demonstrates the practical application of machine learning in the field of cybersecurity but also emphasizes the significance of real-time monitoring and user accessibility through the developed web interface.

1. PROBLEM DEFINITION

1.1 OVERVIEW

In today's interconnected world, the importance of network security cannot be overstated. As organizations increasingly rely on digital systems to store and manage sensitive information, they face a growing array of cyber threats. Network Intrusion Detection Systems (NIDS) are critical in this landscape, providing the ability to monitor network traffic continuously and identify suspicious activities that may indicate security breaches.

To address these challenges, this project explores the integration of machine learning (ML) into NIDS These systems utilize various detection techniques, including signature-based detection, which relies on predefined attack patterns, and anomaly-based detection, which establishes a baseline of normal behavior to identify deviations. By training a model on diverse network traffic data, the system can classify traffic as either normal or attack-related in real-time, enhancing the overall security posture. This project not only highlights the potential of ML in improving intrusion detection but also emphasizes the importance of a user-friendly web interface for effective real-time monitoring and management of network security.

1.2 PROBLEM STATEMENT

Despite the advancements in cybersecurity measures, many organizations continue to suffer from security breaches due to the limitations of traditional Network Intrusion Detection Systems (NIDS). These systems often rely on outdated signature-based methods that struggle to keep pace with the rapidly evolving threat landscape. This project was initiated to address the challenges posed by increasing network traffic volume and the sophistication of modern attacks, which frequently evade conventional detection techniques. The aim is to develop a more effective machine learning-based NIDS that can accurately classify network traffic, thereby enhancing the ability to identify and respond to security threats in real time.

2. INTRODUCTION

2.1 INTRODUCTION

The increasing frequency and sophistication of cyber threats underscore the critical importance of robust network security measures. In this context, Network Intrusion Detection Systems (NIDS) serve as essential tools for monitoring network traffic and identifying potentially malicious activities. This report focuses on the development of a machine learning-based NIDS that utilizes advanced algorithms to classify network traffic effectively. By analysing various features of network packets, the system aims to differentiate between normal traffic and various types of attacks, including denial-of-service (DoS) and probing attempts.

The report outlines the objectives, methodologies, and findings of the project, emphasizing the effectiveness of machine learning techniques in enhancing intrusion detection capabilities. It discusses the limitations of traditional signature-based detection systems and highlights how machine learning models can adapt to evolving threats.

Furthermore, the report details the implementation of a user-friendly web interface that enables real-time monitoring and reporting, empowering network administrators to respond promptly to identified threats. By combining advanced detection algorithms with a practical monitoring solution, this project aims to contribute significantly to improving organizational cybersecurity frameworks, providing a more dynamic and effective approach to safeguarding network integrity against emerging threats.

2.2 OBJECTIVES

i. To develop a machine learning-based Network Intrusion Detection System (NIDS) capable of accurately classifying network traffic as either normal or malicious.
ii. To create an interactive platform that helps to classify the network as Normal or Malicious based on the data provided.

2.3 ORGANISATION OF THE REPORT

The report is divided into six chapters. The first chapter deals with the problem statement, an overview of the project and the problem that was identified which led to the project. The second chapter is an introduction to the project, the objective of the project. The third chapter is the literature survey, it shows the recent developments and the different

technologies that are available, and the technologies that can be used in the project. The fourth chapter is the methodology, i.e., it defines the flow of the project, the different processes that have been done during the project and the methods that have been followed for the success if the project. The fifth chapter is the result which explains about the result of the project and the sixth chapter is the conclusion chapter that explains about the relevant areas where the projects is a great fit and the most useful.

3. LITERATURE SURVEY

Recent studies emphasize the growing role of machine learning (ML) in Network Intrusion Detection Systems (NIDS) due to its ability to adapt to complex and evolving cyber threats. Traditional NIDS often rely on signature-based methods, which, while effective for known threats, struggle against novel attack patterns. As a result, researchers have explored various ML algorithms to enhance intrusion detection, including Decision Trees, Support Vector Machines (SVM), and Random Forests, all of which offer robust classification capabilities.

A prominent approach is the use of ensemble methods, particularly Random Forest, which combines multiple decision trees to improve prediction accuracy and is particularly effective on imbalanced datasets—a common issue in network traffic data. Neural networks and deep learning models have also shown promise in detecting complex, multi-stage attacks, though they often require significant computational resources and large datasets.

The importance of real-time detection has led to studies on ML-based NIDS that can process and classify traffic efficiently. Researchers have noted that web-based monitoring interfaces can enhance user interaction and usability, allowing administrators to respond faster to threats. This project builds on these findings by developing an ML-based NIDS with a practical, real-time monitoring solution, addressing both the need for adaptive detection and accessible user interfaces in cybersecurity.

4. METHODOLOGY

4.1 INTRODUCTION

This chapter outlines the step-by-step methodology used to develop the Network Intrusion Detection System (NIDS), covering data preprocessing, feature selection, model training, and deployment. Each step has been carefully designed to ensure the accuracy and effectiveness of the machine learning model, along with a user-friendly deployment interface for real-time monitoring.

4.2 DATA COLLECTION AND DATASET DESCRIPTION

The dataset forms the foundation for training the NIDS. This project utilized a well-established intrusion detection dataset that contains various features, such as protocol type, source and destination, source and destination packet details, labels, etc., indicating normal or attack-related traffic.

4.3 DATA PREPROCESSING

To ensure that the dataset is optimized for machine learning, preprocessing steps were conducted:

- **Data Visualization Using Histogram Plot**

To comprehend the dataset's structure, histogram plots were employed to illustrate the distribution of various features, including protocol type, source bytes, and packet length. These plots enabled a detailed view of data spread and frequency, revealing patterns and helping to spot outliers or skewed distributions in the dataset. Additionally, histograms were instrumental in highlighting class imbalance by visualizing the relative frequencies of normal versus attack-related instances. This visualization facilitated adjustments during preprocessing, ensuring a balanced dataset that promotes accurate and fair model learning across both categories.

Fig 4.1: Histplot of the dataset

- **Handling Missing Values:** Missing data in network logs can hinder model performance. We used methods like mean imputation or, where necessary, removed rows with excessive missing data to maintain dataset integrity.
- **Encoding Categorical Variables:** Since network traffic includes both categorical (e.g., protocol type) and numerical data, encoding categorical variables was necessary. We applied one-hot encoding to convert these categorical features into numerical format, enabling compatibility with machine learning models.
- **Feature Scaling:** Differences in feature scales can cause biased learning. To address this, we used the Min-Max scaler, standardizing features to a consistent range, thus ensuring efficient and unbiased training across different algorithms.
- **Outlier Detection:** Outliers in the dataset can lead to skewed results. Boxplots were used to analyse and remove outliers, ensuring that the model's performance was not negatively impacted by extreme values.

4.4 FEATURE SELECTION AND CORRELATION ANALYSIS

- **Correlation Matrix Analysis Using Heatmap Plot**

To assess feature relationships, a correlation matrix visualized through a heatmap was utilized, showcasing the degree of association between different features. This heatmap revealed high correlations between specific features and the target variable, indicating features that hold significant predictive value for accurate classification. Additionally, the heatmap highlighted inter-feature correlations, identifying redundant features that could be removed to simplify model complexity while preserving performance. This step was essential in refining the feature set, ensuring the model focuses on attributes with the greatest impact on accurate classification.

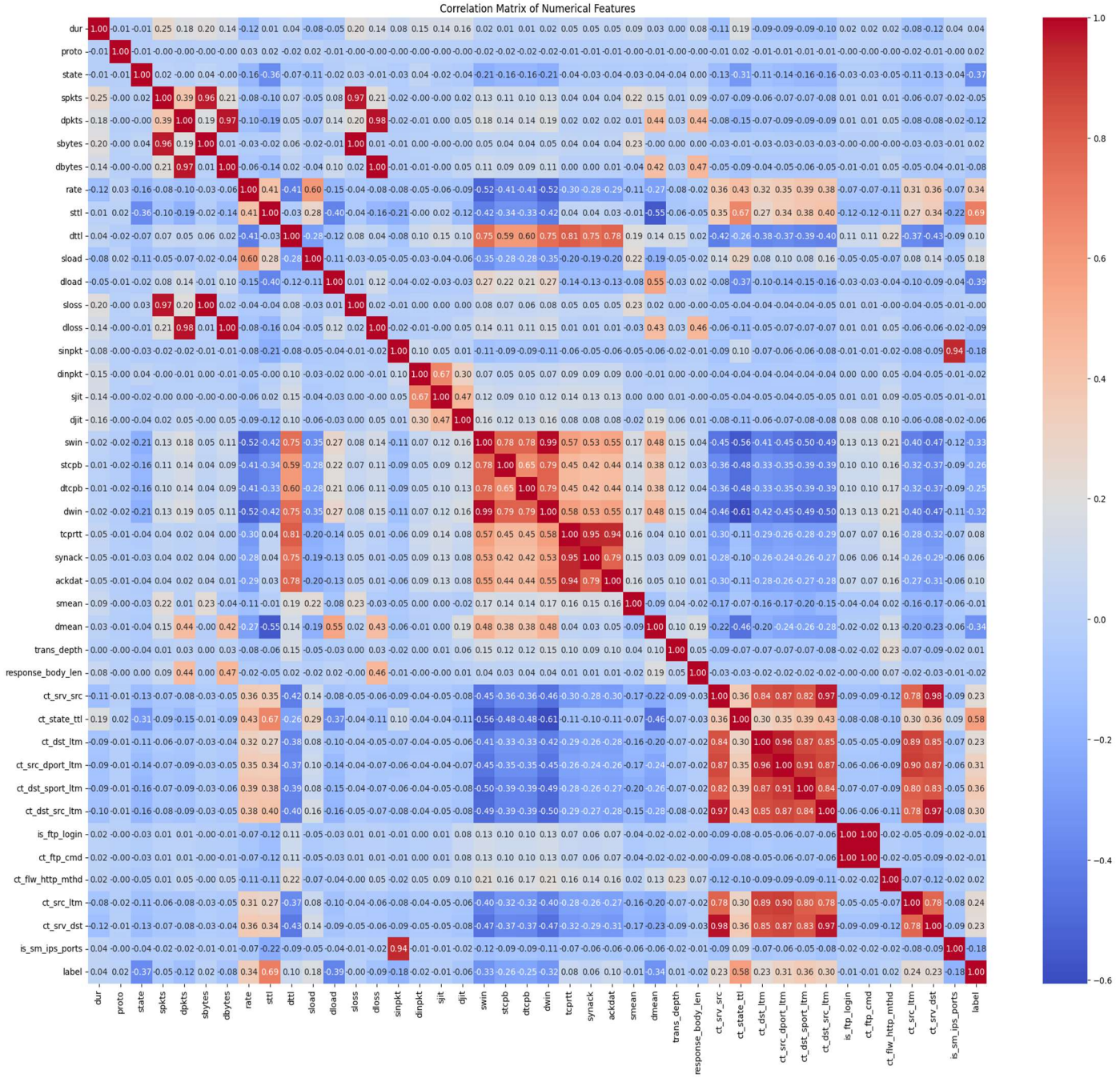

Fig 4.2: Heat map of the correlation matrix

4.5 MODEL SELECTION AND TRAINING

Multiple machine learning algorithms were evaluated to determine the most effective model for classifying network traffic. The models assessed included:

- **Random Forest Classifier:** Selected for its robustness and accuracy, Random Forest is an ensemble learning method that combines multiple decision trees to provide reliable results, even with imbalanced data. This model also provided insights into feature importance, which aided in the refinement of the feature selection process. The model's performance, evaluated using accuracy, precision, recall, and F1-score, is summarized below:
    - **Class 0 (normal traffic):**
        - Precision: 0.93
        - Recall: 0.89
        - F1-score: 0.91
        - Support: 11,169
    - **Class 1 (attack-related traffic):**
        - Precision: 0.95
        - Recall: 0.97
        - F1-score: 0.96
        - Support: 23,900
    - **Overall Accuracy:** 0.94 (across 35,069 instances)
    - **Macro Average:**
        - Precision: 0.94
        - Recall: 0.93
        - F1-score: 0.94
    - **Weighted Average:**
        - Precision: 0.94
        - Recall: 0.94
        - F1-score: 0.94

- **Logistic Regression:** Logistic Regression was chosen due to its simplicity and efficiency, serving as a baseline model for comparison. Although less complex than ensemble methods, it demonstrated reasonable performance, particularly in handling the binary classification of network traffic. The results are as follows:
    - **Class 0 (normal traffic):**
        - Precision: 0.87
        - Recall: 0.78
        - F1-score: 0.82
        - Support: 11,169
    - **Class 1 (attack-related traffic):**
        - Precision: 0.90
        - Recall: 0.95
        - F1-score: 0.92
        - Support: 23,900
    - **Overall Accuracy:** 0.89 (across 35,069 instances)
    - **Macro Average:**
        - Precision: 0.89
        - Recall: 0.86
        - F1-score: 0.87
    - **Weighted Average:**
        - Precision: 0.89
        - Recall: 0.89
        - F1-score: 0.89
- **Decision Tree Classifier:** This model was tested due to its interpretability and ease of visualization, which allowed for straightforward understanding of decision rules.

Although not as robust as Random Forest, the Decision Tree provided competitive performance. The evaluation metrics for the Decision Tree Classifier are as follows:

- **Mean Squared Error (MSE):** 0.0770
- **R-squared (R^2):** 0.6452
- **Mean Absolute Error (MAE):** 0.0770

Each model was assessed based on its performance metrics, and the Random Forest Classifier was chosen as the final model for deployment due to its superior accuracy and balanced performance across both normal and attack traffic classes.

4.6 DEPLOYMENT AND REAL-TIME MONITORING PLATFORM

The final model was integrated into a real-time monitoring platform developed with Streamlit, a user-friendly web application framework that enables quick deployment of machine learning models. The deployment process involved several steps:

- **Model Serialization:** The trained model and scaler were saved as .pkl files, allowing for efficient loading and deployment within the application without re-training.
- **User Interface Design:** A clean, intuitive interface was developed to display real-time data, model predictions, and alert notifications. Key components include interactive visualizations, logs of detected anomalies, and options for administrators to upload new logs or retrain the model as needed.
- **Real-Time Data Ingestion:** The platform allows for continuous data input, simulating real-time network monitoring where administrators can monitor live traffic and receive immediate alerts on potential intrusions.
- **Periodic Model Evaluation:** Regular evaluation of model performance is recommended to maintain accuracy and reliability over time. This involves periodically checking metrics and, if necessary, conducting additional tuning.

5. RESULT

The performance of the machine learning models developed for network traffic classification was evaluated using key metrics such as accuracy, precision, recall, and F1-score. Among the models tested, the Random Forest Classifier emerged as the most effective, achieving an overall accuracy of 94%. This high accuracy demonstrates its ability to reliably distinguish between normal and attack-related traffic, even when dealing with imbalanced datasets. The precision for detecting attacks was 95%, and recall reached 97%, indicating the model's strength in identifying a significant portion of network attacks while maintaining a low false positive rate.

Logistic Regression, while computationally efficient, performed moderately with an overall accuracy of 89%. It provided a baseline for comparison but was less effective in handling complex patterns in the data, especially when differentiating between normal traffic and various types of attacks.

The Decision Tree Classifier also performed reasonably well, though its accuracy and error metrics were slightly lower than those of the Random Forest. Its Mean Squared Error was 0.077, and the R-squared value was 0.645, indicating room for improvement in capturing complex decision boundaries.

In summary, the Random Forest Classifier was selected for deployment due to its superior balance of precision, recall, and overall accuracy, making it the most reliable option for real-time network intrusion detection.

6. CONCLUSION

The Network Intrusion Detection System (NIDS) developed in this project effectively demonstrates the power of machine learning in cybersecurity. By leveraging a well-structured dataset and applying multiple algorithms, the project identified the Random Forest Classifier as the most suitable model for real-time intrusion detection. With its high accuracy of 94% and robust performance in identifying network attacks, the system proves reliable in distinguishing between normal and malicious traffic. The Decision Tree and Logistic Regression models provided valuable insights but were ultimately less effective than the Random Forest approach.

From a broader perspective, the project showcases the importance of automating the process of detecting network anomalies and attacks, which is crucial in today's evolving cybersecurity landscape. The successful integration of data visualization techniques, feature selection, and model evaluation into a streamlined machine learning pipeline highlights how machine learning can enhance network security protocols.

This NIDS has significant potential for integration into various network platforms, such as cloud environments, enterprise networks, and IoT systems. In cloud computing, the system can be adapted to monitor virtualized network infrastructure, detecting threats across distributed resources. Similarly, in large-scale enterprise networks, the NIDS can serve as a crucial defence mechanism, continuously monitoring internal traffic for advanced persistent threats (APTs). The adaptability of the system also makes it ideal for securing IoT networks, where numerous devices operate with minimal security, increasing vulnerability to attacks.

In conclusion, this project offers a scalable and adaptable solution to network security challenges across multiple platforms, making it a valuable tool for enhancing cybersecurity defences.

REFERENCES

i. https://www.linkedin.com/learning-login/share?account=78898396&forceAccount=false&redirect=https%3A%2F%2Fwww.linkedin.com%2Flearning%2Fartificial-intelligence-foundations-machine-learning-22345868%3Ftrk%3Dshare_ent_url%26shareId%3D3THzjCBcS1iv4zjbfKyktA%253D%253D