SE495 Project: Monitoring and Detecting Cyber Attacks Based on Traffic Data and DNS Information

Course: SE495

Section: 1311

Instructor: Dr. AlMuthanna Alageel

Done by:

RAMY GAMAL ABDALLA IBRAHIM (ID:221110521)

1. Introduction

1.1-Project Overview

Cybersecurity threats are evolving at a very fast pace, becoming more complex and frequent. Threats such as botnets, DDoS attacks, and malicious traffic are becoming more sophisticated, thus posing greater risks to organizations and becoming more challenging for them to protect their systems. That’s why it's very important for organizations to be able to accurately detect and classify various types of cyberattacks in real-time, to ensure network security is maintained. The project aims to leverage machine learning models to monitor and detect cyberattacks using network traffic data and DNS information. By analyzing features like traffic duration, packet size, and timestamps, the models can classify traffic as normal or malicious and identify specific attack types.

1.2-Problem Definition

Traditional rule-based threat detection systems are struggling to keep up with the evolving cyber threats. They are slowly failing to adapt to the increasingly sophisticated and more complex attack patterns. These systems are generally trained on predefined patterns. That’s why they are unable to deal with zero-day attacks. Additionally, some of these systems are trained on datasets that contain much more normal traffic than malicious traffic, since it is less frequent than normal. The imbalance in classes is causing bias in these systems.

1.3-Impact

This project is very beneficial to cybersecurity professionals and organizations that seek to improve their network & system security. Let’s take a look at some of the impacts:

1) **Automating threat detection:** Less reliance on manual monitoring.

2) **Improving accuracy:** Machine learning models outperform traditional signature-based methods.

3) **Reduced Response Time:** Using automation threat detection, the time taken to detect the attack once the attack is initiated will be reduced, minimizing potential damage.

4) **Improved Resource Allocation:** Cyber attack defense teams can prioritize their responses based on the type and severity of detected threats. In other words, the teams can focus more on the high-risk or most damaging attacks first.

1.4-How Will Security Teams & Organizations Benefit Exactly?

1.4.1-Actionable Insights

By building various machine learning models, we can gain insights about the most critical features that help decide whether this network traffic is legitimate or malicious. A good example would be traffic spikes at odd hours

1.4.2-Scalability

In today's digital world, networks generate massive amounts of traffic data in real time. In order for cybersecurity systems to process and analyze these huge amounts of traffic data without delays, they would require high-level scalability. Through leveraging machine learning models, thousands of network packets can be analyzed per second, while ensuring that the performance stays consistent.

1.4.3-Adaptability

Because cyber threats are constantly evolving, it's critical for cybersecurity systems to adapt and improve their detection capabilities in real time. The machine learning models built in this project are designed to adapt to constantly changing threats. By continuously training them on new threat data, these models can learn the latest attack patterns, making them effective in detecting zero-day attacks.

2. Objectives

The main objective of this project is to build, train, and evaluate various machine learning models that are capable of accurately detecting and classifying cyberattacks based on network traffic data. Let’s explore some of the sub-objectives of this project.

2.1-Train Binary and Multi-Class Classifiers

Firstly, machine learning models will be trained for binary classification tasks, in order to differentiate between malicious or normal network traffic. Secondly, the same machine learning models will be then trained for multi-class classification tasks to classify the network traffic as botnet, DDoS, or DGA. This will help identify specific attack types. Finally, a deep learning model will be trained for multiclass classification tasks in order to capture complex non-linear relationships.

2.2-Compare Model Performance

Specific classification metrics will be used (mentioned in the methodology section) to evaluate all the models for both binary and multiclass classification. Once every model is tested for evaluation, they will be compared against each other to find the best performing model.

2.3-Identify Most Important Features

Feature importance was calculated using two main model-based methods. Firstly, for tree-based models like Random Forest and Gradient Boosting, feature importance is calculated using the built-in feature_importances attribute. The goal in tree-based models is to decrease impurity, a measure of uncertainty, as much as possible. Features that reduce impurity more are considered more important. Secondly, for linear models like Logistic Regression, feature importance is calculated by computing the absolute value of the model coefficients. The bigger the absolute coefficient value is, the more important a feature is.

2.4-Dashboard Visualization

Create a dashboard using html to demonstrate the important visualizations (like Confusion Matrix and Feature Importance) as well as visually demonstrate a Model Performance Metrics table to show the performance of different models. This dashboard makes it easier to draw insights and conclusions.

3. Methodology

3.1-Data Collection

3.1.1-Dataset Acquisition

Network traffic data was collected from the Stratosphere IPS Project. 3 datasets were collected total from the Stratosphere repository. The first dataset was Botnet Data, which contains traffic labeled as botnet. It's found in the CTU-13 Dataset. The second dataset was DDoS Data, which contains traffic labeled as botnet as well, since DDoS is a special type of botnet. The third dataset was also Botnet, which contains traffic labeled as both botnet and normal. It’s found in the Zeus (Botnet-25-4) in the Stratosphere repository. After collecting these 3 datasets, they were combined into 1 dataset in CSV format. These 3 datasets were binetflow files. It's important to ensure that all datasets have the same structure before merging them into 1 dataset, through concatenating them into a single data frame.

3.1.2-Dataset Details

1) **Dataset Name:** combined_dataset (CSV format)
2) **Size:** 130,502 samples & 17 features
3) **Target Variable:** attack_type
4) **Some Features Considered**:
    - **StartTime:** Time the network connection started.
    - **Dur:** Duration of the connection.
    - **Proto:** Protocol used for the connection (e.g., TCP, UDP).
    - **SrcAddr:** Source IP address.
    - **Sport:** Source port number.
    - **DstAddr:** Destination IP address.
    - **Dport:** Destination port number.
    - **TotPkts:** Total packets in the connection.
    - **TotBytes:** Total bytes transferred.

3.2-Data Preprocessing

1) **Handling Missing Values in Numeric Columns:** Median imputation
2) **Handling Missing Values in Non-Numeric Columns:** Mode imputation
3) **One-Hot Encoding:** Label encoding was used for any categorical data
4) **Dropping Irrelevant Columns:** The ‘Label’ column was dropped for redundancy.
5) **Feature Scaling:** Numeric features were standardized using StandardScaler, to ensure that all features contribute equally to the analysis.

3.3-Models Trained

1) **Logistic Regression**
2) **Random Forest**
3) **Gradient Boosting**
4) **SVM**
5) **Neural Network (Multi-layer Perceptron classifier)**
6) **Neural Network (Deep Learning)**

3.4-Train-Test Split

- **70% training & 30% testing**

3.5-Evaluation Metrics

1) **Primary: F1-score (weighted and macro)**
2) **Secondary: Precision, recall, confusion matrices.**

4. Results

4.1-Models Performance Comparison (Multiclass)

Model	F1 (Weighted)	F1 (Macro)	Precision	Recall
Logistic Regression	1	0.9937	1	1
Random Forest	1	1	1	1
Gradient Boosting	1	1	1	1
SVM Performance	0.9999	0.9737	0.9999	0.9999
Neural Network	1	0.9937	1	1

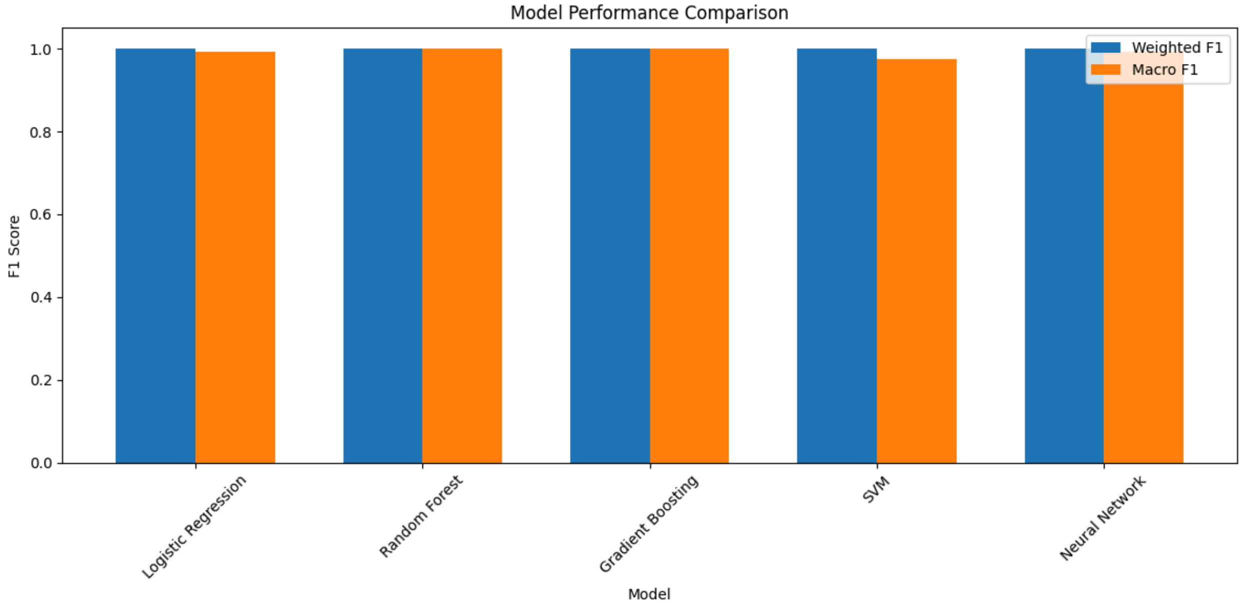

Figure 1

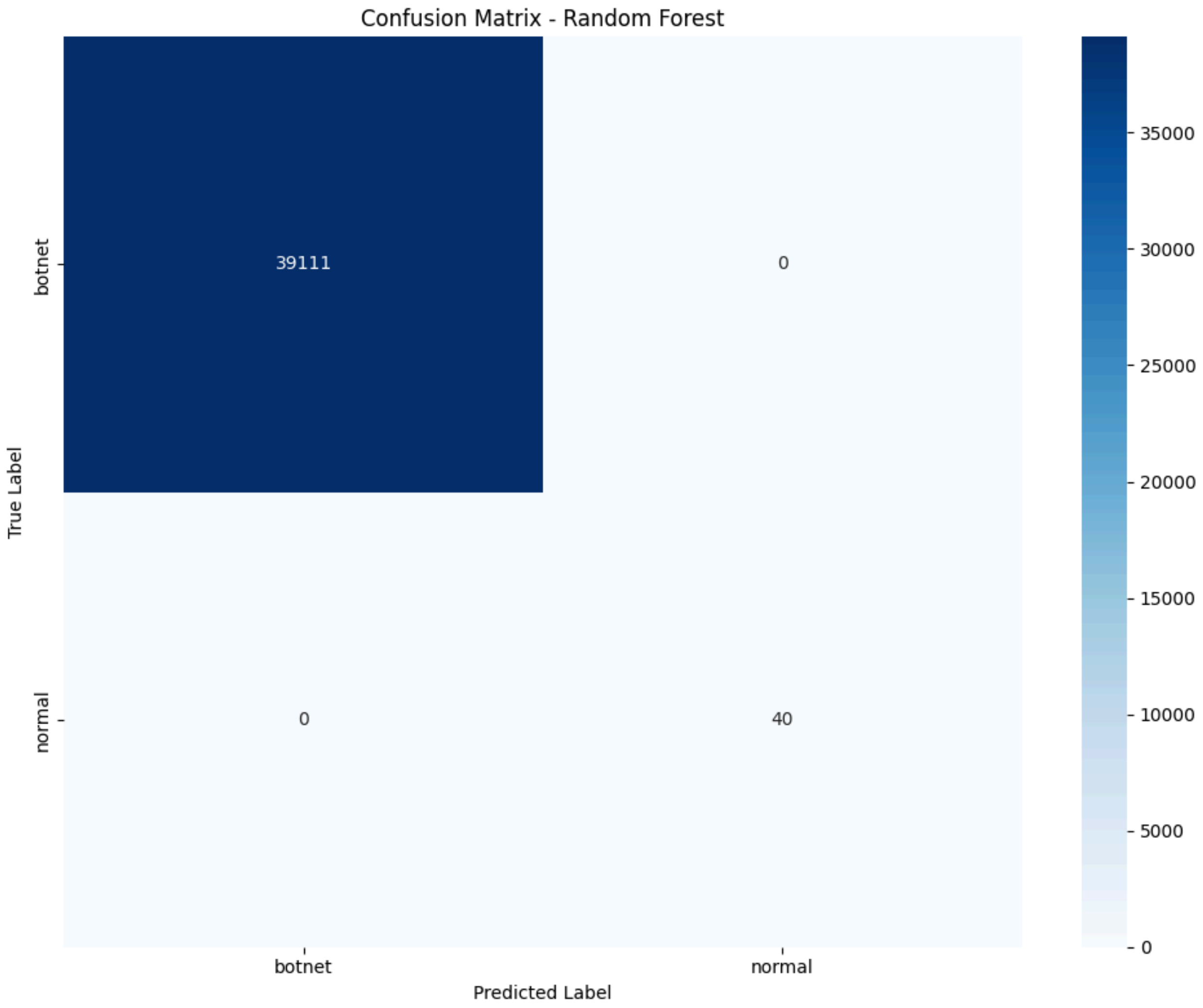

Figure 2: The Random Forest model demonstrated near-perfect classification performance, correctly identifying almost all 'botnet' and 'normal' traffic instances with minimal misclassifications.

4.2-Key Findings

1) **Best Model:** Random Forest performed the best in both tasks (Perfect 1s)
2) **Top 5 Most Important Features: 1-TotBytes 2-SrcBytes 3-TotPkts 4-Dport 5-hour**
3) **Data imbalance: Oversampling improved minority class detection**

4) **Deep Learning:** The deep learning model achieved comparable results to traditional machine learning approaches, with only slightly lower performance. This suggests that additional complexity may not be necessary for this particular dataset.

5) **Class Imbalance:** The extreme imbalance in the dataset (99.9% botnet vs. 0.1% normal) is significant which risks overfitting.

5. Conclusion

In this project, I was able to develop and evaluate multiple machine learning and deep learning models for cyber attack detection based on network traffic data. All models performed exceptionally well which demonstrates the effectiveness of machine learning approaches for cybersecurity applications. The Random Forest classifier emerged as the optimal model, achieving perfect scores across all evaluation metrics for both binary and multi-class classification tasks. Key features like total bytes, source bytes, and Total packets were critical. The dashboard enables real-time monitoring, aiding security teams in threat response.