SE495 Milestone Project

Topic: Monitoring and detecting cyberattacks based on traffic data.

Student: Ahmed Yasser Ibrahim

ID: 222110758

Introduction

Network traffic is regularly targeted by various malware threats. In this project, we develop and evaluate machine learning models to detect malicious flows and classify attack families using labeled datasets sourced from the Stratosphere research repository. We address two core tasks: binary classification (malicious vs. legitimate) and multiclass classification (identifying Botnet v1, Zeus, or DDoS families).

Data

As mentioned before, our data is obtained from Stratosphere website. To collect this data, we browsed through many datasets of malwares available on Stratosphere. One challenge we faced was finding suitable data. The reason for this is that the Stratosphere website mostly has unlabelled data. Many of the datasets available had a labels column, but it was empty. After a lengthy search process, we obtained three datasets that are labelled:

1- Botnet v1

2- Zeus Botnet

3- DDoS Malware

Those three datasets include malicious and legitimate packets. Botnet is a type of malware that creates bots that can browse and use websites and services on the internet. Zeus, on the other hand, is a common type of malware that targets banking credentials. It also creates bots to automate financial fraud. Finally, DDoS is a type of malware that floods servers with a huge number of requests from different parts of the globe. This overwhelms the server and may breakdown websites, leading to some vulnerabilities. All three of our malware types may be categorized under Botnet malware. Unfortunately, due to

unavailability of other labelled datasets, we are unable to obtain datasets regarding DGA or other types of malware.

Preprocessing

During data preparation, we encountered several challenges that required careful handling before model training:

- **Misaligned headers in Botnet-v1:** The original CSV contained commas within the ‘label’ field, which caused the parser to split that column in two and misalign all subsequent headers. We resolved this by applying a fixed-split parser to realign the columns, then concatenated the cleaned Botnet-v1 data with the Zeus and DDoS datasets.

- **Data volume and missing values:** With over eight million records—and more than three million missing dTos entries—our Colab environment frequently crashed. We migrated development to a local VS Code setup, freed up RAM by stopping some processes, and dropped all rows containing critical nulls to streamline the dataset and stabilize processing.

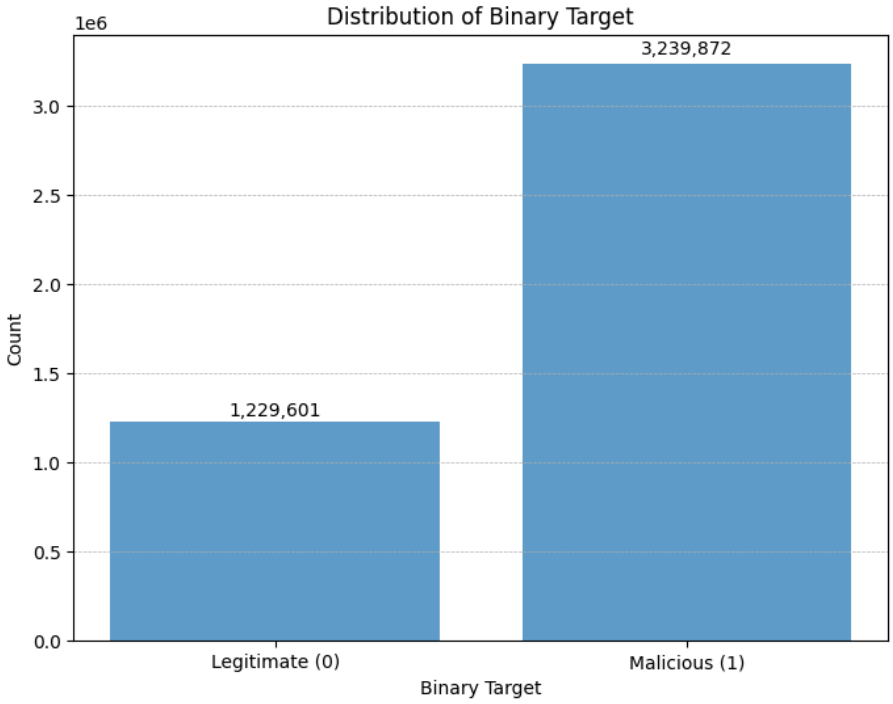
- **Categorical encoding:** We one-hot encoded most protocol and state features, grouping rare state values under a single “other” category to control dimensionality. IP address fields were transformed into numeric embeddings using a specialized encoding library.

- **Temporal feature engineering:** We removed the raw Start Time column and extracted the year into a new feature, which later proved valuable for capturing temporal patterns.

- **Normalization and splitting:** Finally, we standardized all continuous variables with Standard Scaler and performed an 80/20 stratified train–test split. With these steps complete, the data was fully prepared for downstream machine-learning workflows.

Methodology (Binary Classification Task)

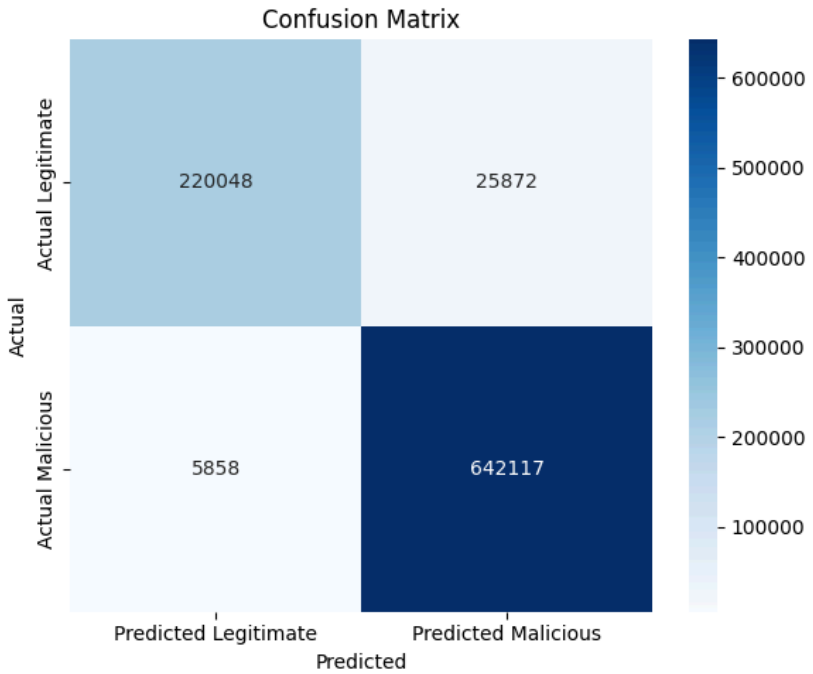
After our preprocessing, our data distribution was as follows.

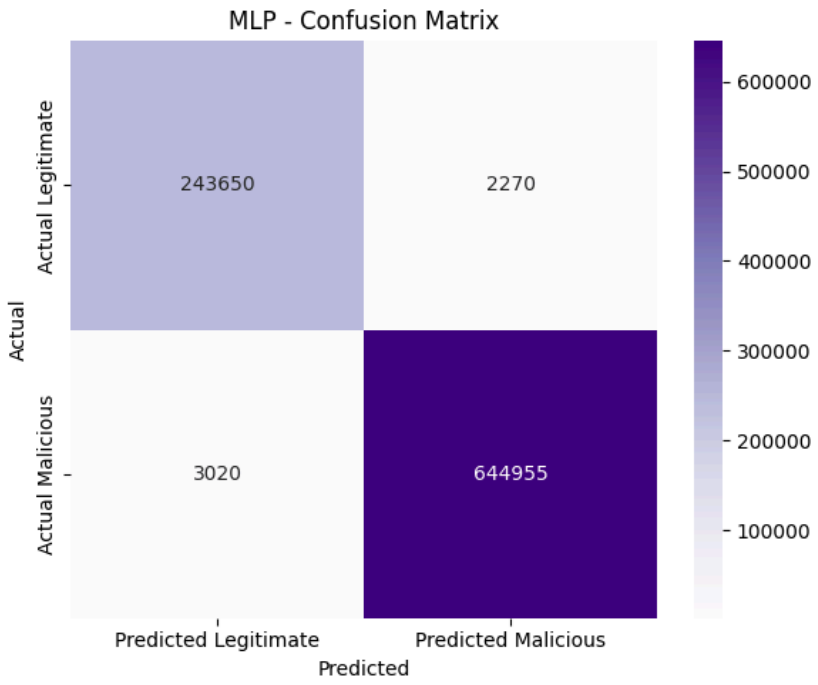


In this task, we created a column ‘label’ that has values of either 0 or 1. Then we trained two different models: Logistic Regression and Neural Networks. The performance of our models and our interpretations are shown below.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9645	0.9613	0.9910	0.9759
Neural Network (MLP)	0.9900	0.9900	0.9900	0.9900

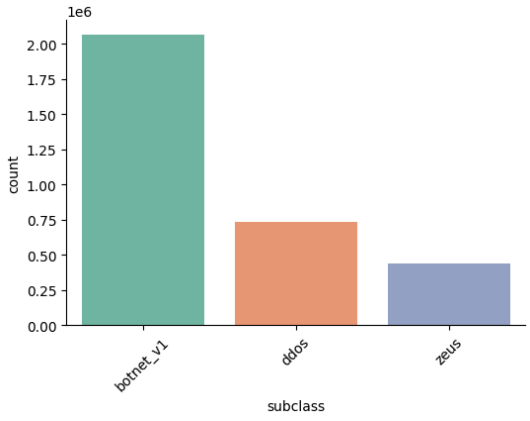
The confusion matrices for Logistic Regression model and Neural Network model and shown below respectively. They help us visualize the predicted and actual classes to which our data belongs.





Methodology (Multiclass Classification Task)

In this task, we created a column ‘subclass ’that has values of either 0-3 representing botnet v1, ddos, zeus, and none respectively. Our data distribution after preprocessing is shown below.



Then we trained two different models: Decision Trees and Random Forests. The performance of our models and our interpretations are shown below.

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.9995	0.9995	0.9995	0.9995
Random Forest	0.9995	0.9995	0.9995	0.9995

Conclusion

Our data-preprocessing pipeline enabled robust model training across both detection tasks. In the binary setting, a simple logistic regression already achieved strong accuracy, while our multilayer perceptron pushed performance nearly to perfection. For multiclass classification, both the decision tree and random forest models produced identical, near-perfect metrics—a result that, frankly, raises some suspicion. One interpretation is that our datasets, drawn from legacy Stratosphere samples, contain "traditional" malware patterns that are exceptionally easy for modern machine-learning algorithms to capture. At the same time, the presence of a few outliers prevents absolute accuracy, suggesting that our feature engineering faithfully encodes the dominant behaviors while edge cases remain. Overall, these results confirm the effectiveness of our approach on historical traffic data, and they underscore the importance of validating against more diverse, contemporary threat samples in future work.

Challenges

Throughout our project, we faced numerous challenges that we were mostly able to overcome. We highlight some of the challenges below:

1. **Misaligned headers in Botnet-v1**: The original CSV had commas in the label column, which split that field into two and misaligned the headers. We resolved this by parsing with a fixed split and realigning to match our original schema.

2. **Memory constraints on our combined dataset**: Loading over eight million records into Colab and later VS Code caused out-of-memory crashes. After freeing RAM and ending other processes, we removed rows with missing values to stabilize processing.

3. **High null rate in dTos**: More than three million entries in the dTos column were nulls, so we dropped any rows with nulls in critical features to reduce our dataset's size and ensure model robustness.

4. **High-cardinality categorical encoding**: One-hot encoding of protocol and state features risked exploding dimensions. We used our numerical columns list to isolate continuous features and grouped rare state values under a single "other" category to control feature-space size.

5. **IP address embedding**: Converting textual IP columns into numeric form required a specialized encoding library and careful insertion back into the DataFrame so that all

features remained aligned.

6. **Temporal feature handling**: We dropped the full start time column and created a new year variable extracted from it—capturing essential temporal patterns without retaining the high-cardinality timestamp.

Future Improvements

To ensure that our models generalize well to unseen data, especially regarding newer internet traffic, we may look for newer datasets and test our models. Moreover, we would like to test our model on predicting different types of internet traffic malwares, such as DGA, if we find the appropriate data for it.