School of Engineering

Cyber Security Attacks Prediction

Project Report

Performed by:

Yani Lala

Saibou Keita

Ashish Singh

Aicha El Gueddari

Ravichandan Kodijuttu

Durga Bhavani Kowrada

Chapter 1: Exploratory Data Analysis (EDA) and Preprocessing

1.1 Data Insights and Recommendations for Preprocessing:

After performing the initial Exploratory Data Analysis (EDA), we've noticed that that the dataset includes so many missing values notably Malware Indicators, Alerts/Warnings, Proxy Information and Firewall Logs fields. This helped us take some measurements regarding the dataset:

1.1.1 Unnecessary fields removal:

- Payload Data: This column consists of meaningless Latin phrases that do not provide any value for predictive modeling.

- User Information: The field has too many unique and specific entries, which makes it overly detailed and likely not useful for the model.

1.1.2 Sensitive fields:

- Severity Level & Anomaly Scores: These fields might contain incorrect calculations that could introduce noise, affecting the model's performance. These should be managed carefully or excluded if needed.

- Action Taken: This field contains information about actions taken after an event, which can lead to overfitting by making the model overly reliant on post-event outcomes instead of predicting future attacks.

1.1.3 Required modifications:

- Alerts/Warnings, Firewall Logs, IDS/IPS Alerts, Malware Indicators: We made the hypothesis that the empty values in these fields can be interpreted as "no detection/alert" instead of missing data. So, we decided to keep them at this stage.

- Source IP Address, Destination IP Address, Timestamp: These fields require further decomposition and analysis, including categorizing IP addresses by class and assessing whether they are public or private. Additionally, timestamp analysis should focus on identifying time patterns and ranges.
- Proxy Information: We also made the hypothesis that missing values can be replaced with "no proxy" to reflect cases where no proxy was used.

Chapter 2: Statistical Analysis

The aim of statistical analysis is to study behaviors in our dataset that are not perceptible with simple observations. It enables us to test hypotheses, identify relationships and trends, detect outliers, assess data quality and support model development. The choice of the appropriate test depends on the type of data. That's why, in this step, we separate our dataset into several subsets according to column type (Boolean, categorical, numerical).

2.1 Boolean features distribution:

This part is about plotting the Boolean values of the dataset to analyze the distribution of the data. The plots are very important because they give us information on our data and its distribution.

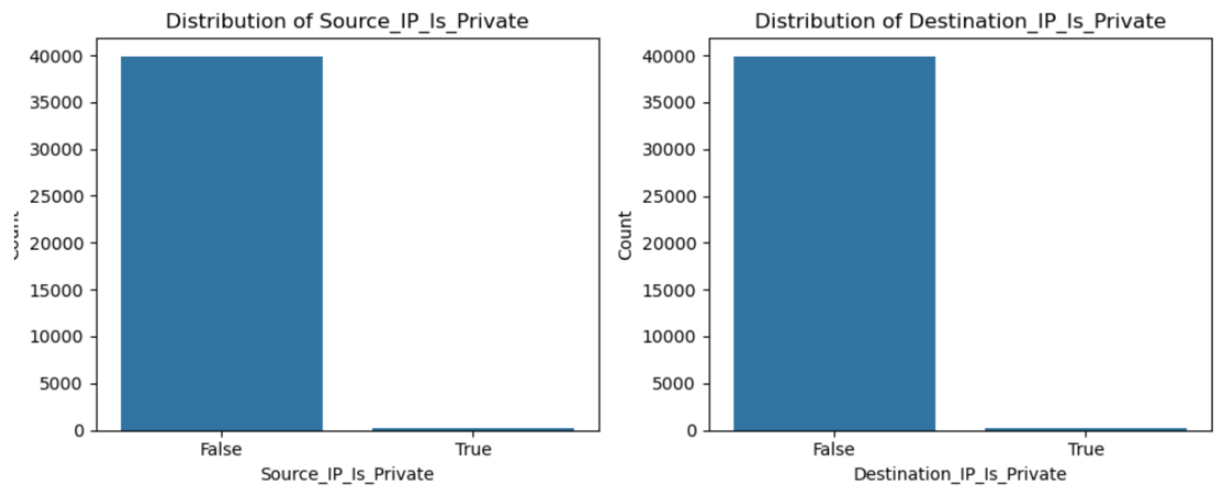


Figure 1: The distribution of private IP addresses

2.1.1 Categorical features distribution

We followed the same approach as in the previous section to gain insights from the distribution of categorical features across the dataset.

2.1.2 Numerical features distribution

This step is performed to check whether our numerical features follow a normal distribution which we later found out that they don't. It provides valuable information for statistical analysis.

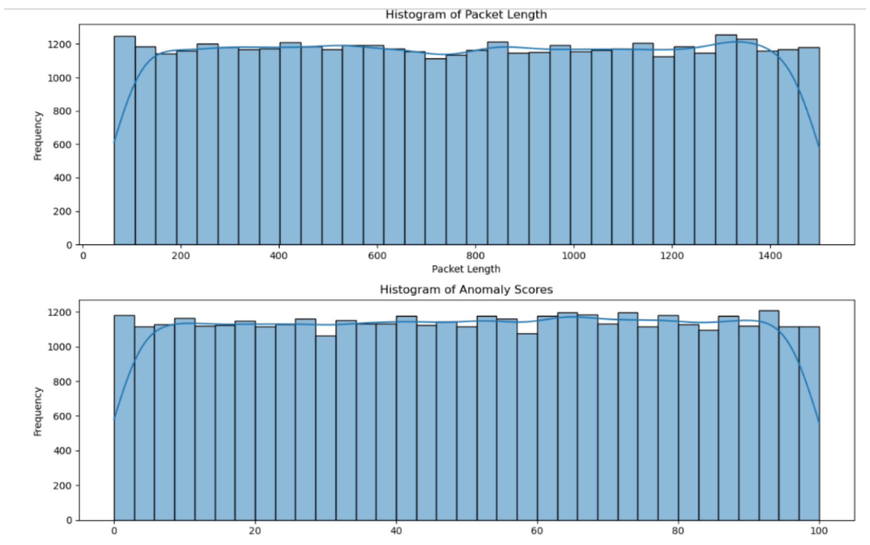


Figure 2: The distribution of numerical features

2.1.3 Correlations and statistical analysis

The purpose of computing and visualizing the correlations between boolean features is to identify relationships in order to improve the model efficiency and make relevant decisions on feature engineering.

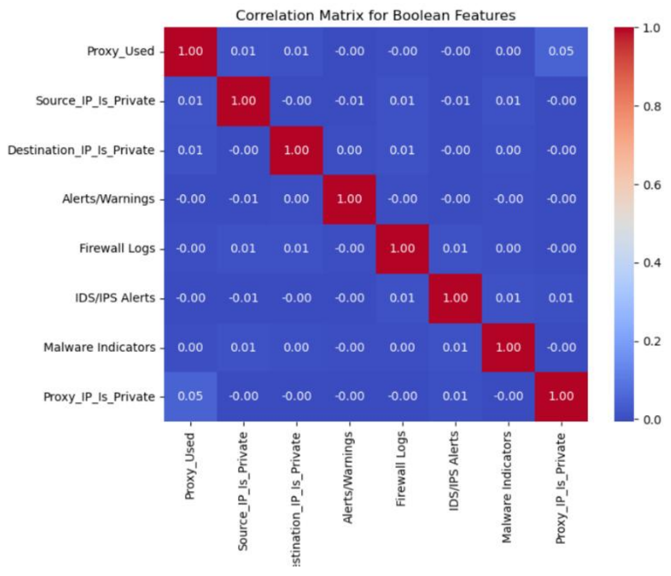
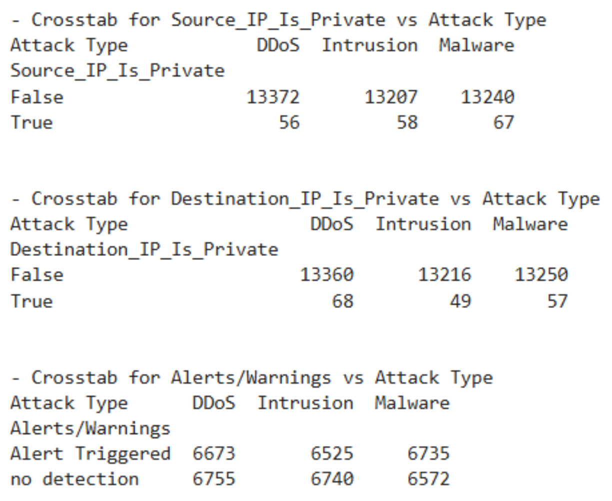


Figure 3: Correlation matrix for Boolean Features

Nevertheless, the Boolean columns didn't show any significant correlations, making it difficult to merge them and extract meaningful insights.

This led us to shift our approach and examine the distributions of the Boolean columns in relation to the attack type.



```
- Crosstab for Source_IP_Is_Private vs Attack Type
Attack Type            DDoS  Intrusion  Malware
Source_IP_Is_Private
False                 13372      13207    13240
True                     56         58       67


- Crosstab for Destination_IP_Is_Private vs Attack Type
Attack Type                 DDoS  Intrusion  Malware
Destination_IP_Is_Private
False                      13360      13216    13250
True                          68         49       57


- Crosstab for Alerts/Warnings vs Attack Type
Attack Type      DDoS  Intrusion  Malware
Alerts/Warnings
Alert Triggered  6673       6525     6735
no detection     6755       6740     6572
```

Figure 4: The distribution of Boolean features based on Attack Types

Variations between different attack types are too small to determine whether it’s a fundamental feature or not. A statistical test is a conceivable solution in this case.

2.1.3.1 P Value:

The **p-value** is a statistical measure that helps determine the significance of results in hypothesis testing.

P_value < 0.05 => There is a significant difference in the true/false proportion with respect to the attack types

P_value > 0.05 => the result is not statistically significant

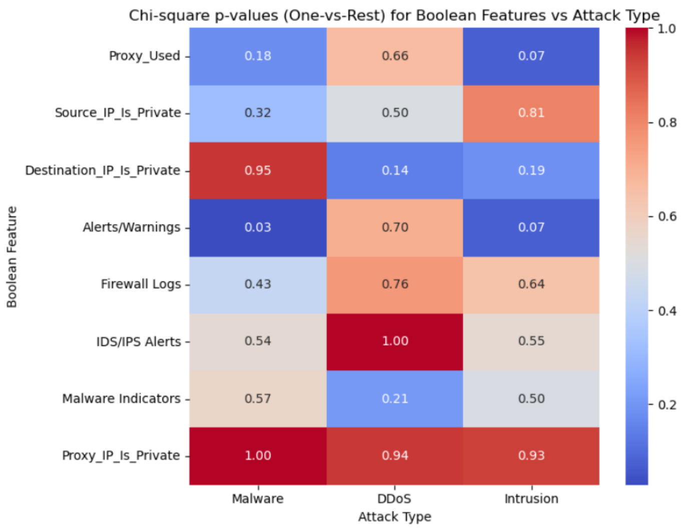


Figure 5: P Values of Boolean features vs Attack Type distribution

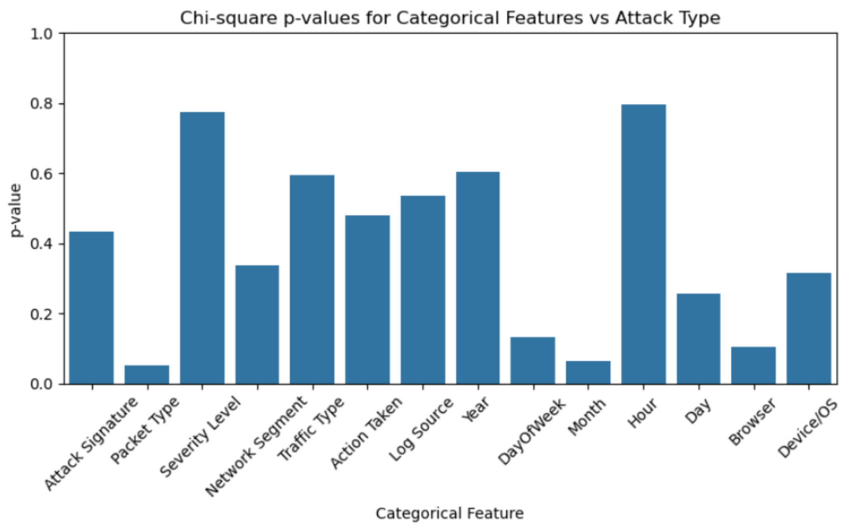


Figure 6: P Values of categorical features vs Attack Type distribution

- The first test shows no statistically significant difference but suggests that the boolean distribution varies more based on the attack type for the fields: Alerts/Warnings, Proxy_Used, and Destination_IP_Is_Private, compared to the other fields. On the other hand, the fields Firewall Logs and IDS/IP Alerts almost entirely reject this hypothesis.
- In the second test, we observe distribution differences that are influenced by the attack type, helping us identify which attacks may be linked to these distribution differences.
- Specifically, these tests indicate that three fields could be valuable for our model. We decide to retain Alerts/Warnings, Proxy_Used, Destination_IP_Is_Private, and Source_IP_Is_Private (which may be related to other variables).
- Some other P Value tests will be available in the Jupyter Notebook for numerical values.

2.2 Feature engineering:

These are the key modifications made to the dataset during the feature engineering process. These decisions were based on the results of the dataset exploration phase, the statistical analysis, and a reflection on the coherence of these observations and results.

- Convert Hours into Four Time-of-Day Categories: Categorize hours into different parts of the day.
- Create Binary Feature for Business vs Non-Business Hours: Define business hours as 9 AM to 5 PM and create a binary feature.
- Test "One vs Rest" Method: Apply the "one vs rest" method for certain features.
- Cyclical Encoding for Time Features: Use cyclical encoding for time-based variables (e.g., hours of the day).
- Transformation of Numerical Variable (Packet Length): Apply a logarithmic transformation (log(1+x)) to handle outliers in packet length.
- Convert Alerts/Warnings: Map "Alert Triggered" to 1 and "No Detection" to 0.
- Convert Action Taken: Map "Blocked" and "Logged" to 1 (response), "Ignored" to 0 (no response).

- Create New "Alert_Response" Column: Set to 1 if either "Alert Triggered" or "Action Taken" indicates an alert.

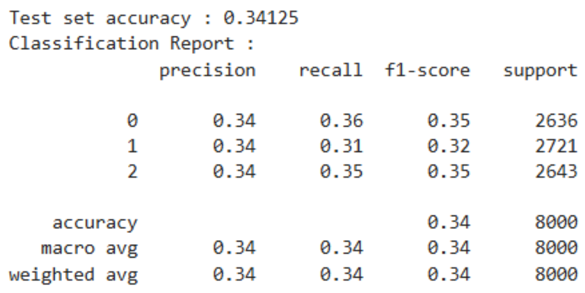
Chapter3: Supervised learning and model selection for classification

For this part of the project, two different models will be trained and evaluated for the dataset explained before: **Random Forest** and **XGBoost**.

Before any training, a preprocessing for categorical and boolean columns by applying **one-hot encoding** to categorical columns and **mapping** or converting boolean columns to numeric form is necessary. It ensures that all columns are correctly formatted for modelling.

3.1 Random Forest:

After splitting data into training and testing data, we performed hyperparameter tuning for a Random Forest model using RandomizedSearchCV. It randomly selects parameter combinations from a predefined grid and evaluates them through cross-validation to identify the best-performing set of parameters. Then we evaluated the model.



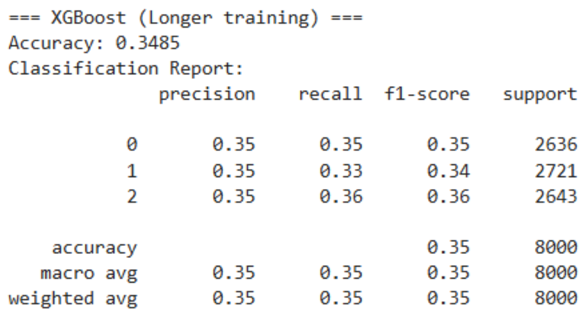
```
Test set accuracy : 0.34125
Classification Report :
              precision    recall  f1-score   support

           0       0.34      0.36      0.35      2636
           1       0.34      0.31      0.32      2721
           2       0.34      0.35      0.35      2643

    accuracy                           0.34      8000
   macro avg       0.34      0.34      0.34      8000
weighted avg       0.34      0.34      0.34      8000
```

Figure 7: Random Forest classification report

As shown in the previous figure, the model's performance was suboptimal across all attack types, achieving an accuracy of only 34%, which is considered quite low for our use case. There are 3 Attack Types, so if we pick one randomly, we have ~33% chances, which is almost as good as what our model provides.

3.2 XGBoost:

We followed the same approach as previously described, this time using XGBoost, considering this model supposed to be more performant.



```
=== XGBoost (Longer training) ===
Accuracy: 0.3485
Classification Report:
              precision    recall  f1-score   support

           0       0.35      0.35      0.35      2636
           1       0.35      0.33      0.34      2721
           2       0.35      0.36      0.36      2643

    accuracy                           0.35      8000
   macro avg       0.35      0.35      0.35      8000
weighted avg       0.35      0.35      0.35      8000
```

Figure 8: XGBoost classification report

As shown in the figure, it is a little bit better but still not really but still only slightly better than a random choice.

3.3 Observations:

We could have trained our model for longer with more sophisticated machine learning algorithms, but the time we tried, we came across a case of overtraining. The amount of data with only 40,000 rows limits us in this respect. But more than that, the quality of the data itself seems to leave something to be desired. Indeed, several tests enabled us to detect relationships between the type of attack and the fields we selected. If our model is so imprecise, it surely means that there are inconsistencies in the relationships present and that we have measured and indicated a potential data quality problem.

GITHUB LINK TO THE PROJECT:

https://github.com/ashish18oct/Cyber-Security-Attack-Prediction