Report

Faiq Aslam [*] Nazim Sulman [†]

April 30, 2023

Abstract

The rising number of cyber assaults has resulted in the requirement for efficient measures for cyber security. One of these measures is the creation of machine learning models that can predict and categorize cyber assaults. In this project, we examine a dataset that encompasses 23 various categories of cyber assaults and strive to produce a classification model that can categorize these assaults into 5 categories. The dataset is preprocessed to handle missing values, outliers, and feature scaling. We determine the most relevant features for classification using correlation analysis. Three classification algorithms, namely, Decision Tree, K-Nearest Neighbors, and Artificial Neural Networks are employed to create the classification models. The efficiency of each model is assessed using metrics such as accuracy, precision, recall, and F1-score. We have got 0.96 accuracy using Decision Tree, and K- Nearest Neighbors.

1 Introduction

The severity of cyber attacks on network traffic cannot be underestimated as they pose a substantial danger to businesses and organizations. Cybercriminals can take advantage of weaknesses in network traffic to gain illegal entry to confidential data, disrupt business activities, and pilfer valuable intellectual property.

Several types of cyber attacks can target network traffic some of these attack that is common are Distributed Denial of Service (DDoS) attacks, Man-in-the-Middle (MitM) attacks, Spoofing attacks, and Phishing attacks. To protect against these types of cyber attacks, businesses and organizations need to implement strong security measures, such as firewalls, intrusion detection and prevention systems, and security monitoring tools.

One of these measures is the creation of machine learning models that can predict and categorize cyber assaults. In this project, we analyze a dataset that includes 23 different categories of cyber attacks and use three different algorithms - Decision Tree, K-Nearest Neighbors, and Artificial Neural Networks (ANN) - to create a classification model that can detect these attacks. By using these algorithms, we aim to improve our ability to identify and prevent such attacks, ultimately enhancing the security of the network and safeguarding sensitive information.

2 Data Preprocessing

Preprocessing is an essential step in the data analysis process. It involves cleaning and preparing the data for analysis. In this project, the preprocessing step is divided into two sections.

2.1 Extraction

The first step in preprocessing is extracting the relevant data from the dataset. This involves identifying the columns or features that will be used for analysis. Some of these columns are identified as relevant for analysis. 1) Duration 2) src byte 3) dst byte

[*] 19P-0082
[†] 20P-00639

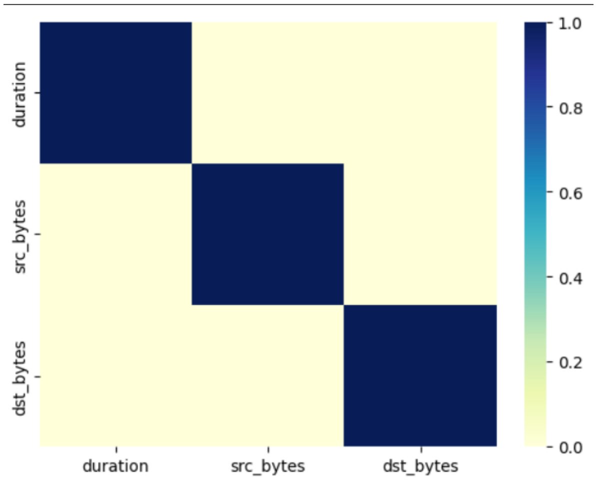


Figure 1: Relevent Features

2.2 Cleaning

Once the relevant data has been extracted, the next step is to clean the data. This involves handling missing values, outliers, and feature scaling.

2.2.1 Missing Value

Missing values are handled with two methods. 1) For Categorical: if dataset contains categorical column then the missing value is replaced by the most frequent value in that column. 2) For Continuous variable: Missing values are replaced by the mean of that column.

2.2.2 Outliers

Outliers are the data point that differs with other data points. Outliers can be caused by various reasons, such as data entry errors, measurement errors, or simply by chance. Outliers were identified using box plots and are removed from the dataset.

2.2.3 Feature Scaling

Feature scaling is an important step in preprocessing as it ensures that the features are on the same scale. We have used StandardScaler, MinMaxScaler - sklearn libraries for feature scaling.

After completing the cleaning process on the raw dataset, a preprocessed dataset was generated. This preprocessed dataset was then saved as a separate file, which can be used for further analysis and classification tasks. The purpose of saving the preprocessed dataset separately is to avoid repeating the preprocessing steps every time the dataset is used for a new task. This also helps to maintain consistency in the data processing pipeline and facilitates reproducibility of the results.

2.3 Algorithms

Three classification algorithms, namely, Decision Tree, K-Nearest Neighbors, and Artificial Neural Networks are employed to create the classification models.

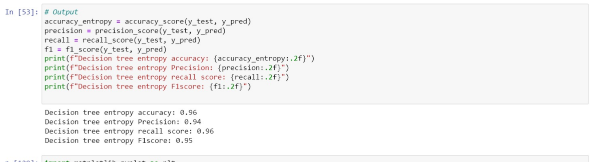
2.3.1 Decision Tree

In this project, we first checked the preprocessed dataset for any missing values and ensured that it was clean before proceeding. The dataset was then split into a training set and a test set, which were used for model training and evaluation respectively.

To develop the decision tree model, we used entropy as a criterion for determining the best splitting points. Entropy measures the randomness or uncertainty in the dataset, and the goal is to choose the split that maximizes the information gain and reduces the most uncertainty in the data.

After building the decision tree model, we evaluated its performance using various metrics such as accuracy, F1 score, and recall. The model achieved an accuracy of 0.96, precision of 0.94, recall score of 0.96, and F1 score of 0.95, indicating that it was able to accurately classify instances of cyber attacks.

Overall, our approach using the decision tree algorithm was effective in developing a model for classifying cyber attacks with high accuracy and precision. Further optimization and refinement of the model could lead to even better performance and improved cyber security.



```
In [53]: # Output
         accuracy_entropy = accuracy_score(y_test, y_pred)
         precision = precision_score(y_test, y_pred)
         recall = recall_score(y_test, y_pred)
         f1 = f1_score(y_test, y_pred)
         print(f"Decision tree entropy accuracy: {accuracy_entropy:.2f}")
         print(f"Decision tree entropy Precision: {precision:.2f}")
         print(f"Decision tree entropy recall score: {recall:.2f}")
         print(f"Decision tree entropy F1score: {f1:.2f}")

         Decision tree entropy accuracy: 0.96
         Decision tree entropy Precision: 0.94
         Decision tree entropy recall score: 0.96
         Decision tree entropy F1score: 0.95
```

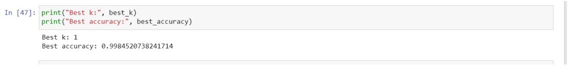
Figure 2: Decision Tree

2.3.2 K-Nearest Neighbor

In our project, we applied the K-nearest neighbors (KNN) algorithm on the preprocessed dataset, using a randomly selected value of k=3. This algorithm classifies new instances by finding the k closest instances in the training data and assigning the class label based on the majority class of those neighbors.

To optimize the KNN model, we evaluated its performance for different values of k and selected the value that resulted in the highest accuracy. We then trained the KNN model on the preprocessed dataset using this optimal value of k and evaluated its accuracy.

Our KNN model achieved a high accuracy rate of 0.99, which was determined to be the best among all the tested values of k. and best value of k is found to be 1.

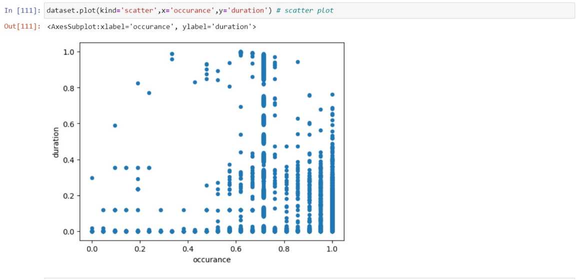


```
In [47]: print("Best k:", best_k)
         print("Best accuracy:", best_accuracy)

         Best k: 1
         Best accuracy: 0.9984520738241714
```

Figure 3: KNN

2.3.3 ANN

We used a Multilayer Perceptron (MLP) Artificial Neural Network (ANN) model for our preprocessed labeled data. However, we dropped the labeled column and utilized K-means clustering to label the dataset instead. This clustering method is unsupervised, and it identifies patterns and structures in the data. First, we selected relevant features from the dataset and then randomly chose four clusters. After applying the K-means clustering algorithm, we obtained labels, which we then used to update the dataset with the clustered label information. After getting labels we apply MLP on the dataset



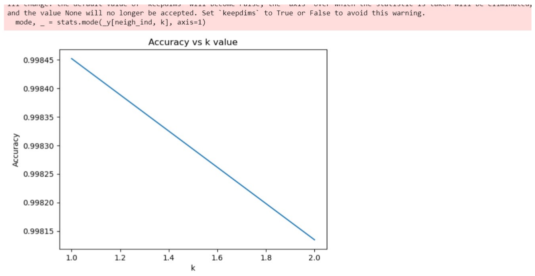
```
In [111]: dataset.plot(kind='scatter',x='occurance',y='duration') # scatter plot

Out[111]: <AxesSubplot:xlabel='occurance', ylabel='duration'>
```

Figure 4: K means Clustering

To change the spell check language, simply open the Overleaf menu at the top left of the editor window, scroll down to the spell check setting, and adjust accordingly.

2.4 Comparison

We have applied three classification model on the dataset and best accuracy so far is of K Nearest Neighbor(KNN) it shows 0.99 accuracy. Decision tree gives accuracy of 0.96 while MLP gives not such results.



```
ill change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated,
and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.
  mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

Figure 5: KNN

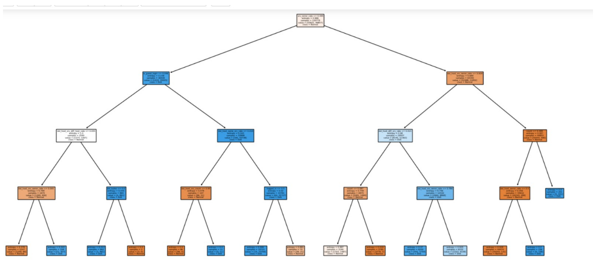


Figure 6: DT

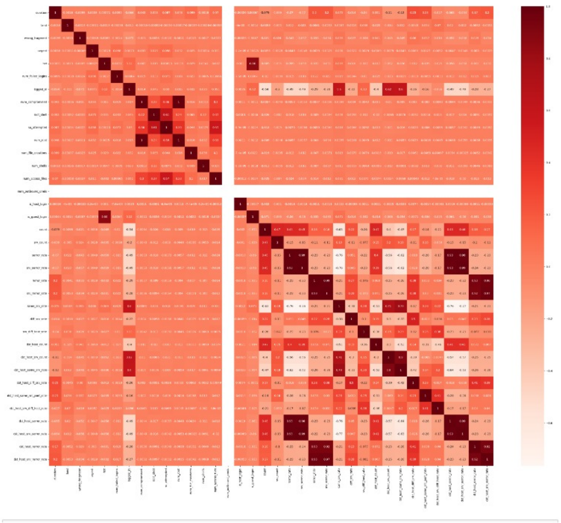


Figure 7: K means Clustering

2.5 Conclusion

The project successfully developed a classification model for cyber attacks using decision tree, K-nearest neighbors, and artificial neural networks. The model's performance was evaluated using appropriate metrics, and its hyperparameters were tuned to optimize its performance. Clustering was also performed on the dataset to visualize the results. Overall, the project demonstrated the importance of data preprocessing, feature selection, and algorithm selection for developing an accurate and effective classification model.

References