Description of the Cryptojacking Project

Cryptojacking is a cyber-attack utilizing malicious scripts similar to those from large cryptocurrency houses to illegally mine data without users being aware. These attacks are stealthy and difficult to detect or analyze, often leading to decreased computing speeds for users as well as crashes due to straining of computational resources.

My goal for this project is to classify network activity from various websites as either cryptojacking or not based on features related to both network-based and host-based data.

Here, I will take you through the process of building the model from the data loading stage up to the deployment stage. I built this model using the Jupyter Notebook and I deployed it using Streamlit. The data was gotten from the \_\_\_\_\_\_ source

Contents

1. Loading Data

2. Statistical Summaries

3. Missing Values and Duplicates

4. Target Distribution and Visualization

5. Scaling and Modeling

6. Classification Report and Feature Importance

7. Model Prediction and Accuracy

8. Model Deployment

1. LOADING DATA

Okay, to begin, I import the Numpy and Pandas libraries to load and analyze the data. Then, I load the csv files for the train set and test set using the pd.read\_csv function. And I preview the first 5 rows of the datasets. I checked for the shape and the information about each column to gain an overview of the data. I also located my label. My label has

2. Statistical Summaries

After loading the data, I checked for the statistical summaries of the data using the describe function. The output table below shows the count of values, the mean value, the maximum value and other important features.

3. Missing Values and Duplicates

I also checked for duplicates and missing values in the data. Luckily for me, there were no duplicates but there were NaN values (NaN means Not a Number). I solved the NaN values by

4. Target Distribution and Visualization

Then, I plot a countplot to show the rate of distribution of my targets in the label. And I discovered there is a class imbalance in the data. Then, I did a correlation heatmap of the data and I discovered that the features are not so strongly correlated with the target.

5. Scaling and Modeling

Then, I scaled the numerical features in the data using the Standard Scaler and then, I split the data using the train\_test\_split module from model selection from Sci-kit learn. I split the dataset to a 70% train set and 30% test set and a random state of

And then, I trained the model using the

6. Classification Report and Feature Importance

After the modeling process, I created a Report to check for the accuracy of my model