Phishing URL Detection Using Machine Learning

Course Project Report

Project Overview

For this project, I worked with the PhiUSIIL Phishing URL Dataset, which contains a large collection of URLs along with different features that describe each URL. The dataset labels each entry as either a phishing URL or a legitimate URL. I picked this dataset because phishing attacks are still one of the most common problems in cybersecurity. Many people fall for fake websites, and being able to automatically detect suspicious URLs can help reduce these attacks.

The main goal of this project was to build several supervised machine learning models that can predict whether a given URL is phishing or not. The dataset includes things like URL length, domain length, the number of special characters, and other numeric indicators that can help a classifier make a decision. Since these features are already extracted, the project focuses on comparing different models and seeing which one performs best.

Methodology

I trained four different supervised learning models and then used an ensemble model to compare performance. The models I used were Logistic Regression, Random Forest, Support Vector Machine with an RBF kernel, and a small neural network using the MLPClassifier. I chose Logistic Regression as a baseline because it is simple and easy to understand. Random Forest was included because it usually handles non linear relationships well and works nicely on tabular data. SVM was chosen because it is good at creating strong boundaries between classes. The neural network was added as a more modern approach that can sometimes pick up patterns that simpler models miss.

Before training the models, I removed several text based columns from the dataset because they cannot be used directly as numeric features. After that, I split the dataset into training and testing sets using an 80 to 20 split. I also scaled the features for the models that needed it, like Logistic Regression, SVM, and the neural network. The ensemble model used a soft voting system that combined the predictions from Logistic Regression, Random Forest, and SVM.

Results and Analysis

Each model produced slightly different results. In general, the models all performed fairly well, but some stood out more than others. Logistic Regression did a good job for a baseline model and showed that the dataset is learnable with simple methods. Random Forest performed even

better because it can capture more complex patterns. The SVM model also had strong accuracy and F1 score after scaling the features. The MLP neural network did reasonably well, but it did not outperform the Random Forest or SVM models.

The ensemble model gave one of the best overall performances. Since it combines the strengths of multiple models, it was able to improve the predictions compared to any single model alone. Looking at the metrics like accuracy, precision, recall, and F1 score helped highlight the differences. In particular, the F1 score was useful because it balances both precision and recall. The confusion matrices also made it clear that the ensemble classifier was better at reducing misclassified phishing URLs, which is very important in a security context.

Conclusion

In this project, I was able to compare several machine learning models on the phishing URL dataset. Random Forest and SVM performed the best among the single models, and the soft voting ensemble gave the strongest overall results. This shows that combining different models can help improve prediction accuracy, especially for binary classification problems like phishing detection.

If I had more time, I would focus on tuning the hyperparameters for each model to try to push the accuracy even higher. I would also explore feature engineering or try more advanced ensemble methods. Overall, this project helped me understand how machine learning can be applied in cybersecurity and how different models can perform differently on the same dataset.