**Work Summary on Using Machine Learning Detect Malicious cybersecurity threats**

**Overview**

In this work I analyze whether ads URL is prone to Phishing (malicious) or not. Some ads URL could contain a malicious link that can trick any receipt and lead to a malware installation, freezing the system as part of ransomware attack or revealing sensitive information.

**Research Objectives**

The objective of this research is to determine the most efficient methodology in detection fraud. The chosen model will be used in generating the most important features that influences fraud.

**Dataset Details:**

The input dataset contains an 11k sample corresponding to the 11k URL. Each sample contains 32 features that give a different and unique description of URL ranging from -1,0,1. The target binary variable which is 1 for phishing (Malicious) and -1 for non-phishing(non-malicious). The non-malicious activities 6,157 outnumber the malicious activities 4,898.

**Exploratory Data Analysis.**

For the exploratory data analysis, I examine the missing rate of each feature in the data and all the features had zero missing rate. To ensure accurate modeling, one of any two features with a correlation coefficient of 0.99 is excluded from the model training process. Also, to address the class imbalance associated with the target variable (results), I used the synthetic minority oversampling techniques (SMOTE). This techniques tackles class imbalance in machine learning by generating synthetic sample for the minority class. The final target class contains 6,157 for both malicious and non-malicious after applying the SMOTE technique.

**Model Building**

The primary research methodology involves comparing various machine learning approaches and selecting the best model based on evaluation metrics. To achieve this goal, an automatic algorithm with 10-fold cross-validation is implemented in the building process. This process comprises the following models:

**Logistic Regression** (LR) Is a linear classification model that plays a crucial role in binary classification tasks, where it predicts the relationship between a dependent variable and one or more independent variables. This model leverages the logistic function, also known as the sigmoid function, to transform a linear combination of independent variables into a probability score.

**Linear Discriminant Analysis** (LDA): This model predicts the class of the dependent variable by utilizing the linear combination of the independent variables.

**K-nearest Neighbors** (KNN): KNN uses proximity to make classifications or predictions about the grouping of an individual data point. It can be used for either classification or regression problems.

**Classification and Regression Tree** (CART): Is a variation of the decision tree algorithm. It can handle both [classification and regression](https://www.geeksforgeeks.org/ml-classification-vs-regression/) tasks. It explains how an outcome variable’s values can be predicted based on other values.

**Naive Bayes** (NB): Naive Bayes uses Bayes' theorem to assign a probability to every possible value in the target class, and the resulting distribution is then condensed into a single prediction.

**Support Vector Machine** (SVM): SVM finds a hyperplane that best fits the data points in a continuous space, instead of fitting a line to the data points. It can be used in both regression and classification tasks, but it generally works best for classification problems.

**Random Forest** (RF): Random Forest involves the creation of multiple decision trees, each constructed using distinct random subsets of both the data and its features. Each decision tree functions as an individual 'expert,' offering its own perspective on the classification of the data. To make predictions, the algorithm computes predictions from each decision tree and ultimately selects the most frequently occurring outcome among these individual results.

**XGBoost** (XGB): XGBoost is particularly popular in various data science and machine learning competitions on platforms like Kaggle due to its high predictive accuracy and versatility. It is designed for both classification and regression tasks and is known for its efficiency, scalability, and ability to handle complex structured data.

**Model Selection Metrics**

Given the class imbalance, accuracy is not the most appropriate metric to evaluate performance. Therefore, other metrics such as AUC, F1 score, Precision, and Recall were also assessed. Thus, Accuracy is not the sole focus when selecting the best-performing model. Also, to ensure an effective evaluation of the model's performance in addressing the class imbalance, PRUAC is also included. Lastly, the KS that measures the maximum separation between the cumulative distribution of the fraud and cumulative distribution of nonfraud was also included. The following provides descriptions of the performance metrics.

**Precision**= : Precision calculates the ratio of correctly classified fraud transactions to all transactions classified as fraud. The range ∈ [0,1].

**Recall**=: This measures the ratio of correctly classified fraud transactions to all actual fraudulent transactions. The range ∈ [0,1].

: This combines the precision and recall using the harmonic mean. It provides a balanced measure of a model's performance. The range ∈ [0,1].

Where:

**TP** is the number of transactions correctly classified as fraud.

**TN** is the number of transactions correctly classified as non-fraud.

**FN** is the number of fraud transaction wrongly classified as nonfraud.

**FP** is the number of nonfraud transaction wrongly classified as fraud

**AUC**: This metric summarizes the trade-off between the true positive rate and the false positive rate for a classifier. It quantifies a classifier’s ability to distinguish between positive and negative classes. The range ∈ [0,1].

**PRAUC**: PRAUC is a metric that summarizes the precision-recall trade-off across different classification thresholds. It calculates the area under the precision-recall curve, which plots precision against recall. A high PRAUC indicates a model that maintains high precision while achieving high recall. This metric is often used for tasks like fraud detection, anomaly detection, and imbalanced classification problems. The range ∈ [0,1].

**Results and Discussion**

**Model Performance in Train Data**

The Random Forest model emerged as the best-performing model, achieving the highest AUC of 99.03%, demonstrating its robust ability to effectively distinguish between malicious and non-malicious activities. Additionally, the Random Forest model yielded the highest accuracy rate of 97.03%, along with a precision and recall rates of 99.00% and 99.31%, respectively. This high F1 score of 99.15% signifies a well-balanced trade-off between precise positive predictions (precision) and the comprehensive capture of positive instances (recall) by the RF model. Furthermore, the PRAUC value of 99.35% obtained from the Random Forest model underscores its superior capability to differentiate between positive and negative classes compared to all other models.

**Table 1.0 Model Performance Metrics on Train Data Sets**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Metric** | **AUC** | **F1-Score** | **Recall** | **Precision** | **PRAUC** | **Accuracy** |
| **LR** | 92.73% | 93.70% | 94.68% | 92.75% | 95.19% | 92.85% |
| **LDA** | 91.87% | 93.00% | 94.29% | 91.74% | 94.60% | 92.05% |
| **KNN** | 96.46% | 96.90% | 97.33% | 96.48% | 97.65% | 94.17% |
| **CART** | 99.04% | 99.15% | 99.20% | 99.10% | 99.37% | 96.08% |
| **NB** | 64.49% | 45.08% | 29.13% | 99.58% | 84.00% | 60.61% |
| **SVM** | 95.22% | 95.90% | 96.96% | 94.87% | 96.76% | 94.84% |
| **RF** | 99.03% | 99.15% | 99.31% | 99.00% | 99.35% | 97.03% |
| **XGB** | 97.55% | 97.87% | 98.27% | 97.47% | 98.35% | 96.66% |

**Detection of Overfitting (Performance on test set)**

Assessing the performance of the models on the test data is crucial because it provides an unbiased evaluation of how well each model generalizes to unseen data. This evaluation ensures that the models have not merely memorized the training data (a phenomenon known as overfitting) but can accurately predict new, real-world examples. Evaluating performance on a separate, unseen dataset helps determine the model's reliability and its potential to perform well in practical applications, aiding in the selection of the best model.

The fitted models are employed to predict the test dataset, and the performance results from the test dataset are compared to those obtained in the training data to detect the possibility of overfitting. Overfitting occurs when a model performs exceptionally well on the training data but fails to generalize effectively to new, unseen data.

Among the models, the Random Forest (RF) model exhibits the lowest reduction in performance metrics when transitioning from the training dataset to the test dataset. A comparison of performance metrics in the training data, as shown in Table 1 above, to those in the test data, as presented in Table 2.0 below, reveals a reduction of less than 3% for all metrics in the test dataset. Consequently, the RF model has been selected as the final model for further in the analysis.

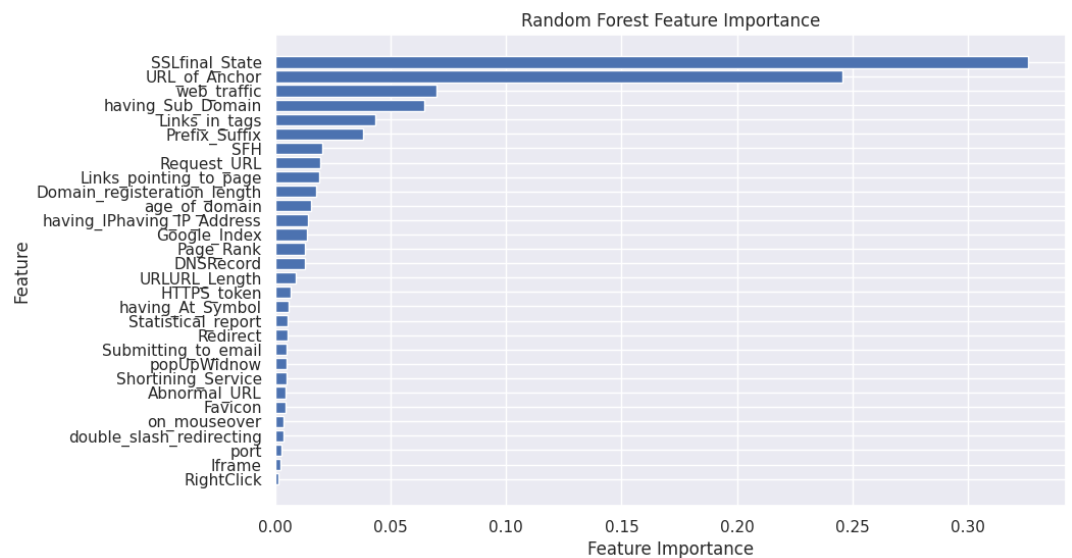
**Table 2.0 Model Performance on the Test Data Set.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **AUC** | **F1-Score** | **Recall** | **Precision** | **PRAUC** |
| **LR** | 92.16% | 93.35% | 93.94% | 92.76% | 95.07% |
| **LDA** | 91.66% | 93.15% | 94.82% | 91.54% | 94.65% |
| **KNN** | 93.88% | 94.81% | 95.30% | 94.32% | 96.14% |
| **CART** | 95.74% | 96.34% | 96.49% | 96.19% | 97.34% |
| **NB** | 63.25% | 42.01% | 26.61% | 99.70% | 83.99% |
| **SVM** | 94.42% | 95.40% | 96.57% | 96.57% | 96.38% |
| **RF** | 96.35% | 96.95% | 97.61% | 96.31% | 97.64% |
| **XGB** | 96.14% | 96.80% | 97.61% | 96.00% | 97.48% |

\*RF declined by 2.69% 2.20% 1.70% 2.70% 1.71%

**Feature importance Score**

Finally, SSL Final State is the most important feature, with a feature importance score of 0.30, followed by the URL Anchor with a score of 0.24 and the least important feature being Right click with feature importance score less than 0.05.

****