DS 600 Capstone

Joyail Tokas and Jennifer Valencia

Topic:AI Powered Phishing Attack Detection

**Problem Statement:**

Phishing is a type of anomaly that consists of malicious attacks intended to trick people into divulging sensitive information (Justindhas et. al., 2023). They are frequently formatted as URLs that mimic legitimate websites. Failure to identify these attacks can result in significant repercussions, such as identity theft, malware distribution, financial threats, and security breaches.

**Business Needs**

Research shows evidence of an increase in phishing attacks in recent years (Alkhalil et. al., 2021). With the rise of phishing attacks, it is essential for businesses to be equipped with robust tools needed to recognize these threats efficiently. Phishing can have detrimental effects on both individual and organizational scales. Unfortunately, these deceptive threats are not easily detected by the human eye, especially as they become more sophisticated. As a result, machine learning must be employed to identify these anomalies quickly and accurately and mitigate potential harm. K-Means Clustering, DBSCAN, Support Vector Machine, and LDA are machine learning approaches that can be used for phishing detection.

**Objective:**

This project will incorporate machine learning algorithms to identify potential phishing attempts through deceptive URLs. Some approaches to be explored include K-Means Clustering, DBSCAN, Support Vector Machine, and LDA. The models will be trained and tested using the PhiUSIIL Phishing URL Dataset from UC Irvine Machine Learning Repository, which comprises 134,850 legitimate and 100,945 phishing URLs. The performance of each of the models will be evaluated to determine which one achieves the highest accuracy when distinguishing between legitimate and illegitimate URLs.

**Methodology:**

1. **Exploratory Data Analysis (EDA)**

The PhiUSIIL Phishing URL Dataset from UC Irvine Machine Learning Repository was imported for analysis. A pandas DataFrame was explored using key methods for in-depth analysis, including data.describe(), data.info(), and data.shape(). The dataset includes 235,795 rows and 56 columns and does not include any null values.

Various visualizations were generated to understand the distribution of the data. Some features such as URL length were plotted, having an average of about 34.57 characters.

The distribution of the dataset was plotted using a bar graph and pie chart. The feature *Label* identifies whether or not a URL is phishing (0) or legitimate (1). The dataset comprises 134,850 legitimate and 100,945 phishing URLs

To understand the relationship between the variables, a correlation matrix was plotted. A box plot of the DomainLength feature was plotted, and the mean was found to be 21.47. A pairplot of the features *Label, DomainLength,* and *URL* length was also plotted.

1. **Feature Analysis**

The dataset was pre-processed. Non-numeric columns were dropped from the DataFrame, null values were accounted for, and categorical columns were encoded using LabelEncoder().

As mentioned previously, the dataset includes 56 features to describe the data. Feature importance was calculated using RandomForestClassifier() and was plotted. Pre-processing revealed that there are 25 meaningful features in the dataset that correlate with *Label*.

A new DataFrame that only included the top 25 features was created to be used for model training. A new correlation matrix was created.

1. **Dataset Enrichment for Domain Name Scoring System**

To enrich the dataset with a Domain Name Scoring System, a combination of third-party libraries and APIs was used to provide security scores for domains. One popular library is pyhunter which interacts with the Hunter.io API for domain security information. Additionally, whois provides domain registration information, and urlscan.io can provide detailed information about the safety of URLs.

The required libraries were installed and API keys for third party services were set up. A function was created to take a domain, query the APIs, and return a safety score based on the responses. After loading the initial dataset, the domain column was extracted to compute the domain score. The DataFrame will be updated to display the domain scores for each URL.

1. **Machine-Learning Model**
   1. K-Means Clustering

The necessary libraries from *sklearn* were imported. K-Means Clustering is an unsupervised machine learning algorithm. Therefore, the *label* feature was dropped from the DataFrame. The data was scaled using *StandardScaler()*, and then split into training and testing dataset. The number of clusters was determined to be 2 since the classification is a binary problem, distinguishing between phishing or legitimate. The DataFrame contains 25 meaningful features for the predictive model. To reduce dimensionality, principal component analysis (PCA) was employed. The silhouette score, which measures the performance of the clustering model, was found to be 0.31.

* 1. Support Vector Machine (SVM)

The necessary libraries were imported from *sklearn* to create the SVM model. The X and y variables were defined, with the former being the DataFrame and the latter being the *label* column. The data was split into training and testing datasets, and scaled using *StandardScaler()*. After training, the performance of the model was evaluated using the testing data. The accuracy was found to be 99.98%.

* 1. DBSCAN

For the DBSCAN clustering, the elbow method was used to determine the optimal value for eps. The k-nearest neighbors algorithm was applied, and the k-distance graph was plotted to identify the elbow point. Based on this analysis, appropriate values for eps and min\_samples were set. DBSCAN was then applied to the scaled training data, and the resulting clusters were visualized using Principal Component Analysis (PCA). The clustering performance was evaluated using the silhouette score, which yielded a value of -0.18, indicating a reasonable level of cluster separation.

* 1. LDA

In the LDA phase, non-numeric columns were filtered out to ensure compatibility. The dataset was split into training and testing sets containing only numeric columns. The LDA model was trained on the training data and evaluated on the test data. The evaluation metrics included accuracy, confusion matrix, and a detailed classification report, with the LDA model achieving an accuracy score of 99.85% on the test set.

1. **Model Evaluation**
2. KMeans Clustering Evaluation

The KMeans clustering model was employed to identify distinct clusters within our dataset, aimed at differentiating between phishing and legitimate URLs. The **Silhouette Score** was used to evaluate how well the data points were clustered, with a higher score indicating better cluster separation. Our results showed a 0.31 score, suggesting that the clusters were well-separated, which is promising for understanding the underlying structure of the data.

1. SVM Classification Evaluation

The Support Vector Machine (SVM) classifier was utilized to classify URLs as phishing or legitimate, focusing on improving the model’s prediction accuracy. The **Accuracy** metric was a key performance indicator, showing an impressive 99.98%, which signifies that the SVM model correctly classified a high percentage of URLs. This result highlights the model's effectiveness in distinguishing between the two classes and its potential suitability for practical deployment.

We also examined **Precision**, **F1 Score**, **Mean Squared Error (MSE)**, and **Root Squared Error (RSE)** to evaluate the classifier's performance comprehensively. The Precision and F1 Score were 99.67% and 99.98%, reflecting the model’s strong performance in minimizing false positives and balancing recall with precision. The MSE and RSE values were 0.00019 and 0.014, indicating that the model's prediction errors were relatively low. The **ROC AUC** and **USRC** metrics further supported the model’s robustness, demonstrating its ability to effectively identify phishing URLs while maintaining a good balance between detecting true positives and avoiding false positives.

1. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) Evaluation

The DBSCAN clustering model was employed to identify distinct clusters within our dataset, aimed at differentiating between phishing and legitimate URLs. The **Silhouette Score** was used to evaluate the model. Our results showed a -0.18 score, which suggests there are potential areas of improvement for the model.

1. LDA (Latent Dirichlet Allocation) Evaluation

For the Linear Discriminant Analysis (LDA) model, the evaluation metrics reveal impressive performance. The AUC-ROC score, which is not explicitly mentioned but can be derived from the ROC curve plot, indicates how well the model distinguishes between phishing and non-phishing URLs, with a higher AUC value reflecting better performance. The model achieved an accuracy of 0.9989, precision of 0.9982, and F1 Score of 0.9991, demonstrating exceptional performance in correctly classifying phishing URLs. The Mean Squared Error (MSE) of 0.0011 and Residual Sum of Squares (RSE) of 50.0000 suggest minimal prediction errors, though the Mean Absolute Percentage Error (MAP) of 4.77e+12 indicates potential issues with scaling or data representation. Overall, the LDA model exhibits outstanding performance in classification, though attention may be needed to address the MAP anomaly.

**VI. Streamlit Application**

Streamlit applications were created based on the KMeans, LDA, and SVM models. The models were saved as pickle files. The code loads the model from the pickle library, accepts a URL input, transforms it using the vectorizer, and predicts whether or not a URL is phishing.

**Conclusion and Future Recommendations**

Based on the various metrics, SVM, DBSCAN, K-Means Clustering, and LDA are useful models for determining whether or not a URL is phishing.

Due to the cost of Hunter API (~$500/month), the scoring mechanism was not pursued. However, the code remains in this project for potential future updates of this project.

**References**

Justindhas, Y., Raghul, V., Pramadeish S., Prakash, S.. (2024). A Comprehensive review on An Ensemble-Based Machine Learning Approach for Phishing Website Detection. In 2024 2nd International Conference on Computer, Communication and Control (IC4), 1-6, <https://ieeexplore.ieee.org/document/10486561>

Prasad,Arvind and Chandra,Shalini. (2024). PhiUSIIL Phishing URL (Website). UCI Machine Learning Repository. https://doi.org/10.1016/j.cose.2023.103545.

Althobaiti, K., Wolters, M.K., Nawal, A., Vaniea, K. (2023, August 31). Using Clustering Algorithms to Automatically Identify Phishing Campaigns. IEEE Access, 11, 96502-96513. doi: 10.1109/ACCESS.2023.3310810

Alkhalil, Z., Hawage, C., Nawaf, L., Khan, I. (2021, March 9). Phishing Attacks: A Recent Comprehensive Study and a New Autonomy. Frontiers in Computer Science 3:563060. doi: 10.3389/fcomp.2021.563060

Catal, C., Giray, G., Tekinerdogan, B., Kumar, S., Shukla, S. (2022, May 23). Applications of Deep Learning for Phishing Detection: A Systematic Literature Review. Knowledge and Information Systems, 64, 1457-1500. https://doi.org/10.1007/s10115-022-01672-x

Sahu, K. & Shrivastava, S.K. (2015, February). Kernel K-Means Clustering for Phishing Website and Malware Categorization. International Journal of Computer Applications 111(9).

Sindhu, S., Patil, S.P., Sreevalsan, A., & Rahman, F. (2020). Phishing Detection Using Random Forest, SVM, and Neural Network with Backpropagation. 2020 International Conference on Smart Technologies in Computing, Electrical, and Electronics (ICSTCEE), 391-394. doi: 10.1109/ICSTCEE49637.2020.9277256