MIS 637 B

Stevens Institute of Technology

**Name –**

**Project:** A real-world data science project including problem statement, data, methodology/algorithm, software, execution and analysis, references, documentation, and presentation.

# Problem Statement –

The advent of the digital age has resulted in previously unheard-of levels of connectedness and convenience, but it has also made room for a growing threat like phishing. Phishing is a widespread problem in the field of cybersecurity. It is a fraudulent attempt to get sensitive information, including financial details, usernames, and passwords, by pretending to be a reliable source in electronic communication. Phishing attacks pose a greater threat than in the past as more and more people and organizations transact their business online. Such attacks can have serious repercussions, such as identity theft, money loss, compromised data security, and harm to one's reputation. Phishing attacks persist in their evolution of sophistication, rendering them more challenging to identify and prevent, even with the implementation of security measures and user education campaigns.

Strong and efficient phishing detection systems are crucial in this situation. Conventional techniques for spotting phishing efforts, such manually reviewing dubious emails or webpages, are time-consuming, labor-intensive, and frequently ineffectual against the ever-evolving strategies used by cybercriminals. Moreover, humans cannot keep up with the ever-expanding threat landscape due to the sheer number of internet transactions and communications. Therefore, there is an urgent need for automated systems that can quickly and reliably identify phishing attempts in real time, reducing the danger posed by bad actors and preserving the integrity of online transactions and communications. Therefore, the challenge at hand is to create an intricate phishing detection system that makes use of cutting-edge technological tools and approaches in order to successfully detect and neutralize phishing efforts. To separate authentic emails from fraudulent ones, this system needs to be able to analyze a ton of data, including user interactions, website URLs, and email content. Furthermore, it needs to be resilient and adaptive to the constantly changing strategies and tools that cybercriminals use, always learning from and enhancing its detection abilities over time. The system must also find a balance between efficiency and accuracy, reducing false positives while making sure that legitimate phishing efforts are not disregarded or overlooked.

In response to the pervasive threat of phishing, I initiated a comprehensive initiative with dual objectives. Firstly, the initiative seeks to construct a predictive model capable of identifying potentially fraudulent websites based on various features and characteristics. Going beyond mere identification, the project endeavors to delve deep into the intricate web of behavioral patterns exhibited by likely perpetrators of phishing attacks. By analyzing historical data and key variables, the study aims to uncover subtle indicators of fraudulent activity, providing valuable insights into the modus operandi of malicious actors. Ultimately, this endeavor aims to bolster cybersecurity measures and safeguard individuals and organizations against the pernicious effects of phishing attacks.

# Data –

The dataset we are using offers a wide range of attributes designed to identify possible phishing attempts, which is helpful in detecting and analyzing phishing domains buried within URLs. The dataset covers a wide range of attributes retrieved from URLs, all of which contribute to a more nuanced knowledge of the traits associated with phishing activities, with the goal of improving the ability to detect and neutralize phishing attempts on the web.

**This dataset's salient characteristics are:**   
  
  
**NumDots:** This attribute indicates how many dots are in the URL, which is a measure of its structure and complexity.   
  
**PathLevel:** Provides information about the website's structure and organization by indicating the path's level within the URL hierarchy.   
  
**UrlLength:** Indicates the total length of the URL and provides important details regarding its possible legitimacy or dubious character.   
  
**NumDash:** Indicates the quantity of dashes in the URL, which may be a sign of particular URL patterns that are frequently connected to phishing attempts.   
  
**AtSymbol:** Denotes the existence of the "@" sign in the URL, which may be a sign that phishing attempts are based on emails.

**IpAddress:** Indicates whether an IP address is present in the URL, which may indicate attempts to hide the website's actual destination.   
  
**HttpsInHostname:** Indicates if the hostname has the "https://" prefix, offering information about the website's security protocols.   
  
  
**PathLength:** Indicates the path's length within the URL and provides information about the hierarchy and intricacy of the website's directory.

**NumNumericChars**: The number of numeric characters (0–9) in the URL is indicated by NumNumericChars, and this quantity can reveal patterns or strategies related to phishing.

Furthermore, a binary indicator Phishing: 0 or 1 is appended to every data point in the dataset to indicate if the associated URL is phishing or not. It is significant to highlight that our dataset includes extra features that have been carefully selected to ensure its accuracy and applicability in the field of phishing detection, in addition to being a consolidation of pre-existing datasets. We will create and assess strong machine learning models and algorithms for phishing attack detection and mitigation by utilizing this large dataset. This will improve cybersecurity protocols and shield users from online threats.

# Methodology/Algorithm

The phishing detection methodology and algorithm implementation involve a methodical approach that incorporates data preparation, algorithm selection, model training, assessment, and visualization of results. To understand the properties of the dataset and get it ready for modeling, we start the procedure with data exploration and preparation. To comprehend the distribution and relationships between features, correlation analysis, visualization tools, and descriptive statistics are used. To maintain the integrity and quality of the data, duplicate entries and missing values are also addressed.   
  
After preparing the data, we use train\_test\_split function. The train\_test\_split function from scikit-learn divides the dataset into training and testing sets. Stratified sampling is used to preserve the target variable's distribution of phishing or non-phishing. Next, categorical variables are kept unaltered and numerical predictors are chosen for feature scaling utilizing StandardScaler. Categorical features are left untouched while the scaling transformation is applied exclusively to numerical characteristics using the ColumnTransformer.   
  
We, then use the training data to instantiate and train a variety of classification algorithms. XGBoost, LightGBM, Gradient Boosting, Random Forest, Logistic Regression, and CatBoost classifiers are some of these algorithms. Regarding computational efficiency, interpretability, and performance, every algorithm has its own advantages. Performance indicators, including the F2 score, are calculated for every trained model across the training and testing sets. Accurately detecting true positives is critical in imbalanced classification tasks such as phishing detection, where the F2 score which prioritizes recall over precision becomes more important.

The model's performance on the training and testing sets is further broken down with the generation of classification reports and confusion matrices. These visual aids make it easier to evaluate the model's performance in accurately classifying phishing and non-phishing URLs and pinpoint areas in need of development. To see the trade-off between recall and precision at various probability thresholds, we show it with precision-recall curve. This curve can be used to improve the model's decision threshold according to requirements and provides insights into the model's performance across different sensitivity levels.

# Software Packages Used -

Several software packages and libraries are used in the approach and algorithm implementation for phishing detection to help with data preprocessing, algorithm selection, model training, evaluation, and result visualization. Among them are:   
  
**NumPy:** For manipulating arrays and performing numerical computations.   
  
**Matplotlib:** Used to create heatmaps, scatter plots, and histograms, among other visualizations, to examine the properties of the dataset.   
  
**Seaborn:** This tool is used to plot correlation matrices and distribution plots, among other statistical data visualization tasks.   
  
**Plotly Express:** Pie charts and three-dimensional plots are examples of dynamic, interactive visualizations that were made with Plotly Express.   
  
**Sklearn :** This is an extensive machine learning package with capabilities for selecting, evaluating, and preparing data. For data splitting, the train\_test\_split module is utilized; for feature scaling, StandardScaler is employed; and other classifiers, like LogisticRegression, RandomForestClassifier, and XGBClassifier, are used.

**Polars:** Large datasets can be loaded and handled quickly and effectively with the help of the data manipulation package Polars.   
  
**Statsmodels:** Offers tests and statistical models for analyzing data, including QQ plots to determine normality.   
  
**TQDM:** A library of progress bars for tracking the development of iterative procedures, including model training.   
  
**CatBoost:** The CatBoostClassifier is trained using a gradient boosting library made especially for categorical features.