Paper 1

Phishy? Detecting Phishing Emails Using Machine Learning and Natural Language Processing

https://www2.cose.isu.edu/~minhazzibran/resources/MyPapers/Phishing_Springer.pdf

Methodologies

Dataset Preparation:

Cleaning and Preprocessing: Removing HTML tags, special characters, and irrelevant metadata to ensure clean data for analysis.

Feature Extraction: Extracting textual features, metadata, and hyperlink information from email bodies, subject lines, and headers.

Feature Extraction:

Textual Features: Using NLP techniques to extract features such as word frequency, TF-IDF (term frequency-inverse document frequency), n-grams, and specific keywords commonly associated with phishing (e.g., "urgent", "verify", "account").

Metadata Features: Analyzing features from email headers such as sender’s email address, domain reputation, and IP address.

Hyperlink Analysis: Examining URLs embedded in emails for suspicious patterns, such as mismatched domains or the use of URL shorteners.

Machine Learning Models:

LR KNN AB MNB GB RF（3.2）

Logistic Regression (LR): LR uses a linear combination to combine the input features before running them through a sigmoid function to create a probability. A binary prediction can be made by thresholding this probability. To reduce the binary cross-entropy loss, the model learns its parameters from training data using gradient descent. K-Nearest Neighbors (KNN): KNN identifies related samples in the training dataset and applies a class label to the input sample. To determine the majority class label, the algorithm calculates the distance to the K nearest neighbors from the input data. KNN is simple to use and effective at managing high-dimensional and noisy data. AdaBoost (AB): AdaBoost combines a number of weak classifiers to create a strong one. Weak classifiers are

learned using weighted examples after each example has been given a weight during training. To create the final strong classifier, these weak classifiers are then combined. Multinomial Naive Bayes (MNB): MNB uses the Bayes theorem to calculate the likelihood that a new email might fall into each class depending on its attributes. It determines key features, such as important words or phrases, and then calculates the likelihood that an email falls into each class. Gradient Boosting (GB): GB integrates the predictions from various models to enhance performance. Using a gradient descent technique to the residual errors, the algorithm trains distinct models that correct the faults generated by the prior model. The weighted average of all the predictions from each model is the final prediction. Random Forest (RF): RF combines the predictions of multiple decision trees to produce a more accurate final prediction. To avoid overfitting and enhance model performance, each decision tree is trained using a random subset of the data. The algorithm can find the most crucial factors in the data by concentrating on the most important characteristics of a phishing email and disregarding unimportant ones.

Model Training and Evaluation:

Cross-Validation: Dividing the dataset into training and testing sets to validate model performance. Cross-validation techniques ensure robustness and prevent overfitting.

Handling Imbalanced Datasets: Applying techniques like oversampling phishing emails or undersampling legitimate emails to balance the dataset.

Continuous Learning:

Regular Updates: Implementing a pipeline for regularly collecting new email data, retraining the models, and deploying updates to maintain detection efficacy against evolving phishing tactics.

Challenges and Strategies

Challenge: Phishers use sophisticated methods to make emails appear legitimate.

Strategy: Implementing advanced NLP techniques to extract subtle textual features and combining them with metadata analysis to improve detection accuracy

Dynamic Nature of Phishing Attacks:

Challenge: Phishing tactics evolve rapidly, rendering static models ineffective.

Strategy: Developing adaptive models that continuously learn from new data and implement real-time updates to stay current with emerging phishing tactics

Imbalanced Datasets:

Challenge: Phishing emails are much less frequent than legitimate ones, causing class imbalance.

Strategy: Using data balancing techniques such as oversampling the minority class (phishing emails) and undersampling the majority class (legitimate emails) to create a balanced training set. Synthetic data generation can also be used to augment the dataset

Feature Engineering Challenges:

Challenge: Effective feature extraction from email content, metadata, and hyperlinks is complex.

Strategy: Combining multiple feature extraction techniques, such as NLP for textual analysis and metadata analysis for sender and hyperlink verification, to capture comprehensive features that indicate phishing

Paper 2

Improving Phishing Email Detection Using a Hybrid Machine Learning Approach

https://www.mdpi.com/2076-3417/13/24/13269

Methodologies

Dataset Preparation:

Data Collection: The dataset consists of a mix of phishing and legitimate emails. Emails are pre-processed to remove noise such as HTML tags and special characters.

Feature Selection: Identifying relevant features from email content and metadata, focusing on those that best differentiate phishing from legitimate emails.

Feature Extraction:

Textual Features: Extracting features such as word frequency, TF-IDF (term frequency-inverse document frequency), n-grams, and specific phishing-related keywords (e.g., "urgent", "account", "verify").

Metadata Features: Analyzing sender's email address, domain reputation, and other header information.

Hyperlink Analysis: Examining embedded URLs for suspicious patterns, such as mismatched domains or the use of URL shorteners.

Machine Learning Models:

Logistic Regression: Used for its simplicity and efficiency in binary classification tasks.

Support Vector Machine (SVM): Effective in high-dimensional spaces, useful for distinguishing subtle differences in email features.

Random Forest: An ensemble method that builds multiple decision trees to improve classification accuracy and reduce overfitting.

Hybrid Approach: Combining the strengths of different models to enhance detection accuracy.

Model Training and Evaluation:

Cross-Validation: The dataset is split into training and testing sets, and cross-validation techniques are used to ensure model robustness and prevent overfitting.

Evaluation Metrics: Models are evaluated using accuracy, precision, recall, F1-score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve).

Continuous Learning:

Adaptive Models: Emphasizing the importance of models that continuously learn from new data, incorporating real-time updates to adapt to emerging phishing tactics.

Challenges and Strategies

Sophistication and Evasion Techniques:

Challenge: Attackers use sophisticated methods to make phishing emails appear legitimate.

Strategy: Implementing comprehensive feature extraction techniques to detect subtle cues in email content and metadata that indicate phishing attempts.

Dynamic Nature of Phishing Attacks:

Challenge: Phishing tactics evolve rapidly, rendering static models ineffective.

Strategy: Developing adaptive models that continuously learn from new data and implementing real-time updates to remain current with emerging phishing techniques.

Imbalanced Datasets:

Challenge: Phishing emails are much less frequent than legitimate ones, leading to class imbalance.

Strategy: Using data balancing techniques such as oversampling the minority class (phishing emails) and undersampling the majority class (legitimate emails) to create a balanced training set.

Feature Engineering Challenges:

Challenge: Effective feature extraction from email content, metadata, and hyperlinks is complex.

Strategy: Combining multiple feature extraction techniques, such as NLP for textual analysis and metadata analysis for sender and hyperlink verification, to capture comprehensive features indicative of phishing.