Phishing Email Detection Using Machine Learning

CIS-619 | Professor: Dr. Naureen Houque
Team: Spandana, Vijay, Sathwik

1. Introduction

Phishing emails represent one of the most pervasive cybersecurity threats, responsible for over 90% of successful data breaches and causing billions of dollars in losses annually. Traditional rule-based spam filters struggle to adapt to evolving phishing tactics, creating an urgent need for intelligent, adaptive detection systems. This project addresses the challenge of automatically distinguishing phishing emails from legitimate correspondence using machine learning techniques applied to the CEAS_08 email dataset. Our objective is to develop and compare classification models that accurately identify phishing attempts based on email content, metadata, and structural features, ultimately providing an automated defense mechanism that can protect users from email-based cyber threats.

2.Dataset Source and Structure

The CEAS_08 Phishing Email Curated dataset was obtained from the Zenodo repository, a trusted open-science platform for research data. This dataset contains real-world email samples collected in 2008, providing authentic examples of both phishing attempts and legitimate correspondence. The dataset comprises 39,153 email records with 7 attributes per record, provided in CSV format with each row representing a single email message. The binary classification task involves distinguishing spam/phishing emails (label = 1) from legitimate emails (label = 0). The class distribution shows 21,842 phishing emails (55.79%) and 17,311 legitimate emails (44.21%), indicating a relatively balanced dataset suitable for classification modeling.

Attributes and Selection

The dataset includes the following key variables: *Sender* (email address of the message originator, useful for identifying suspicious or spoofed sources), *Receiver* (email address of the recipient, contains 462 missing values), *Date* (timestamp indicating when the email was sent, enabling temporal trend analysis), *Subject* (subject line text with 28 missing values, often containing phishing indicators), *Body* (complete email message content where phishing tactics are most evident), *URLs* (numeric count of hyperlinks present in the email body), and *Label* (binary target variable where 0 = legitimate/ham and 1 = spam/phishing). For model training, we selected Subject, Body, Sender, and URLs as primary predictive features, with Label as the target variable. The Subject and Body attributes capture linguistic patterns and persuasion tactics characteristic of phishing attempts. The Sender attribute enables domain-based analysis to identify suspicious email sources. The URLs count indicates potential malicious link presence. The Date attribute was retained for exploratory temporal analysis but excluded from the primary prediction pipeline. The Receiver attribute was not prioritized as it lacks strong predictive power and introduces unnecessary noise with its missing values.

3.Preprocessing & Data Cleaning

Initial preprocessing involved loading the CSV file using Python's pandas library with error-handling for malformed rows. The Date column was converted to datetime format with UTC timezone standardization to enable temporal analysis. Text normalization was applied to both Subject and Body fields, converting all characters to lowercase and removing extraneous whitespace and escape characters to ensure consistency in text analysis.

Missing Value Handling

The dataset contained 462 missing values in the Receiver field and 28 missing values in the Subject field. Since the Receiver attribute was not included in the core feature set, its missing values were left unmodified. For the 28 missing Subject entries, we applied a conservative imputation strategy by replacing null values with the placeholder string 'no subject', preserving the information that these emails lacked subject lines rather than artificially generating content. No missing values were present in critical predictive attributes (Body, Label, URLs), ensuring the integrity of our core features.

Train-Test Split Strategy

We employed an 80-20 stratified train-test split with a fixed random seed (random_state=42) for reproducibility. Stratification ensured that both training and test sets maintained the same class distribution as the original dataset, preventing bias in model evaluation. This resulted in 31,322 training samples and 7,831 test samples, with class proportions preserved across both sets.

4.Feature Engineering and TF-IDF

Beyond the raw attributes, we created four derived features to capture additional predictive signals. *email_length* represents the character count of the email body, capturing content volume patterns as phishing emails often exhibit extreme length patterns - either very brief urgent messages or excessively long persuasive texts. *subject_length* counts the characters in the subject line, as phishing attempts frequently use abnormally long subject lines with multiple exclamation marks or all-caps text. *sender_domain* extracts the domain portion from the Sender email address, enabling identification of suspicious or spoofed domains. *text* concatenates Subject and Body fields, providing complete contextual information. For text vectorization, we applied TF-IDF with English stop word removal and a 30,000-feature maximum, balancing model expressiveness with computational efficiency. TF-IDF weights terms by their frequency within individual emails relative to corpus prevalence.

Final Feature Set

The modeling pipeline utilizes the Body text as the primary input, processed through TF-IDF vectorization to generate a 30,000-dimensional sparse feature matrix. The engineered length and domain features were created for exploratory analysis but not included in the final classification pipeline, which focuses on content-based detection.

5.Model Training and Evaluation

We trained and compared two classification algorithms. Logistic Regression was selected for its interpretability, computational efficiency, and strong performance on high-dimensional text data, with maximum iterations set to 200 to ensure convergence on the large feature space. Random Forest was chosen for its ability to capture non-linear patterns, configured with 100 trees, balanced class weighting to address potential class imbalance, and minimum samples split/leaf parameters of 0.01 to prevent overfitting while maintaining expressiveness.

Evaluation Metrics

We assessed model performance using accuracy, precision, recall, and F1-score. Accuracy provides overall correctness, while precision measures the proportion of phishing predictions that were correct, minimizing false alarms. Recall quantifies the proportion of actual phishing emails successfully detected, minimizing missed threats. F1-score balances precision and recall, providing a single metric for model comparison.

6.Results

Table 1: Model performance comparison (phishing class metrics)

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	99.48%	99.54%	99.52%	99.53%
Random Forest	95.88%	95.00%	98.00%	96.00%

Logistic Regression achieved the highest overall performance with 99.48% accuracy on the held-out test set. The model demonstrated excellent precision of 99.54%, indicating that emails flagged as phishing were highly likely to be genuine threats, and excellent recall of 99.52%, successfully detecting the vast majority of actual phishing attempts. Random Forest achieved 95.88% accuracy with 95.00% precision and 98.00% recall, representing a 3.6 percentage point accuracy disadvantage compared to Logistic Regression. The high F1-scores for both models indicate effective balancing of precision and recall, crucial for practical deployment where both false positives (legitimate emails incorrectly blocked) and false negatives (phishing emails reaching users) carry significant costs. The superior performance of Logistic Regression suggests that the relationship between TF-IDF features and phishing classification is largely linear, making the simpler model more effective than the ensemble approach.

7.Dataset Limitations

The CEAS_08 dataset, while valuable, originates from 2008 and may not fully represent contemporary phishing tactics, which have evolved significantly with advances in social engineering and technical sophistication. The dataset size of approximately 39,000 emails, while substantial, is modest compared to industrial-scale email filtering systems that process millions of messages. The moderately imbalanced class distribution (55.79% phishing vs. 44.21% legitimate) could impact model generalization. Missing values in the Receiver and Subject fields, though handled conservatively, represent information loss.

Model Limitations

Our models rely exclusively on content-based features and do not incorporate email header analysis (SPF, DKIM, DMARC authentication), sender reputation scoring, or behavioral patterns that production systems typically employ. The TF-IDF approach, while effective, does not capture semantic meaning or context as deeply as modern transformer-based language models would. The models were trained on 2008 data and may exhibit performance degradation on recent phishing attempts using novel techniques.

Scope Limitations

This project focuses specifically on English-language emails and binary classification (phishing vs. legitimate), not addressing multilingual detection or fine-grained categorization of phishing types. We did not evaluate model robustness against adversarial examples or deliberate evasion attempts. The evaluation is limited to offline metrics on a single held-out test set rather than longitudinal performance monitoring in a production environment.

8.Conclusion

This project successfully developed machine learning models achieving exceptional phishing detection accuracy. Logistic Regression achieved 99.48% accuracy, effectively distinguishing phishing from legitimate emails through content analysis. The TF-IDF vectorization approach proved highly effective at capturing linguistic phishing patterns, validating content-based detection as a viable strategy.

The final model demonstrates practical applicability for email security infrastructure, though deployment requires integration with authentication mechanisms and continuous retraining. Future improvements should incorporate transformer models (BERT), expand feature engineering to include URL and header analysis, evaluate on contemporary datasets, implement ensemble methods, develop explainability tools, and conduct performance monitoring.