Kyan Patel

Jiali Zhou

ITEC-468

05 May 2025

Phishing Detection Using Machine Learning

Phishing attacks remain one of the most prevalent cybersecurity threats, often exploiting human trust and poor detection mechanisms. This project investigates the feasibility of using machine learning to detect phishing emails based on behavioral and textual patterns found in large-scale email datasets. The aim is to improve upon traditional rule-based filters by applying adaptive techniques that generalize better to evolving threats.

I used a dataset containing over 82,000 emails, each labeled as spam or legitimate, and analyzed them using natural language processing (NLP) techniques. Key features such as message length, capitalization ratio, and the presence of URLs were extracted and combined with TF-IDF vectorization of email bodies to form the basis of our classification model. I trained a logistic regression classifier to differentiate phishing from legitimate emails and evaluated its performance using accuracy, precision, recall, and F1-score.

The results demonstrate that even relatively simple machine learning models can achieve effective detection when enhanced with contextual behavioral features. This suggests that organizations can deploy lightweight but powerful classifiers to complement existing email filters. The approach provides an accessible, data-driven method for improving email security with minimal additional infrastructure.

INTRODUCTION

Phishing is a social engineering attack that deceives users into revealing sensitive information by impersonating trusted entities. These attacks are responsible for a significant portion of corporate data breaches and financial fraud, particularly because they often evade static, rule-based filters. As phishing tactics become more sophisticated—employing obfuscation techniques, zero-day exploits, and convincing impersonation—there is a critical need for more dynamic and intelligent detection systems.

This project explores phishing detection through behavioral and content-based analysis of emails using machine learning. By examining features such as the presence of URLs, capitalization patterns, and textual content, we aim to identify patterns that are indicative of phishing attempts. The model targets organizations and email service providers that need a scalable, automated approach to bolster existing security measures.

The relevance of this work lies in its practicality: many phishing attacks succeed not because of technical prowess, but due to insufficient email screening mechanisms. Our solution leverages publicly available data and accessible ML techniques to demonstrate how even simple models can offer meaningful protection against this persistent threat.

DATA

I used a publicly available phishing email dataset consisting of approximately 82,500 records. Each entry represents an individual email and includes the following fields: sender, receiver, date, subject, body, spam, and urls. The spam column serves as the binary label indicating whether the email is a phishing/spam message (1) or a legitimate message (0). The urls column is another binary feature that denotes whether the email contains any URLs. Together, these fields allowed me to examine both behavioral and textual characteristics commonly associated with phishing attacks.

Before building the model, I performed several preprocessing steps to clean and normalize the data. All email text was converted to lowercase to ensure uniformity. Hyperlinks in the email body were replaced with a placeholder token, “link,” using regular expressions that matched patterns such as “http,” “www,” and “https.” I also removed non-alphanumeric characters (except for a few symbols like periods, hyphens, and slashes), and removed any rows with missing or null body text.

To improve the model’s effectiveness, I engineered several features. I created email_length, which measured the number of characters in the email body, and has_urls, a binary feature indicating whether URLs were present. I also computed body_cap_ratio, the proportion of capital letters in the message, which helped detect emphasis and urgency often used in phishing messages. I then applied TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to the email body to extract unigram and bigram text features, while limiting vocabulary size for performance reasons.

For evaluation, I split the dataset into 80% training and 20% testing data using scikit-learn’s train_test_split() function. I also used stratification to maintain the class balance between phishing and non-phishing emails in both se

ANALYSIS

For this project, I focused on evaluating two machine learning classifiers: Logistic Regression and Random Forest. My primary modeling approach centered on Logistic Regression, which I iteratively refined across three versions by incorporating additional behavioral and structural features at each stage. These enhancements included metrics like URL presence, sender domain frequency, and capital letter ratios. To benchmark performance, I also trained a Random Forest model as a final version, allowing me to compare the results of a simple linear model against a more complex ensemble-based classifier. I began with a baseline Logistic Regression model (Version 1.0) that relied solely on basic behavioral features such as email_length, capitalization ratios, presence of suspicious keywords, and whether the email

contained URLs. This model achieved an accuracy of 66.1% and an F1-score of 66.0%, highlighting the limitations of a basic feature set in detecting phishing attempts, especially given the nuanced nature of modern phishing messages. All model results for accuracy, F1-score and matrix can be found in Table 1.

In Version 1.1, I introduced sender_domain_freq, which quantified how often a sender's domain appeared in the dataset. The logic was that legitimate domains tend to appear frequently, whereas phishing domains are often one-off or spoofed. When combined with TF-IDF vectorization of the email body and subject, this version significantly improved accuracy to 99.40% and F1-score to 99.46%.

Version 1.2 built on this by adding two additional features: url_count and url_presence. These features aimed to capture how aggressively links were used in the message body — a well-known indicator of phishing. This model achieved the best results among all versions, with an accuracy of 99.45% and an F1-score of 99.50%.

To compare linear versus non-linear approaches, I developed Version 2.0 using a Random Forest classifier trained on the full feature set. This model also performed well, with a 99.20% accuracy and 99.27% F1-score, but slightly underperformed relative to the best Logistic Regression version and introduced greater complexity in interpretation and resource usage.

The results demonstrate that phishing emails can be effectively identified by combining linguistic features with behavioral metadata. The dramatic jump in performance from the baseline to enhanced versions confirms the value of incorporating features like domain frequency and link usage. Furthermore, the fact that Logistic Regression performed nearly as well (or better) than Random Forest underscores that simpler models can still deliver enterprise-level accuracy when engineered carefully. The confusion matrix heatmaps(Figure 1) showed how false positives and false negatives were drastically reduced between versions. This is especially important for real-world deployment, where too many false positives can erode trust in the system, and false negatives can lead to security breaches.

These findings have direct relevance to both organizational security policies and broader email

filtering strategies. Many institutions still rely on rule-based filtering systems that match phrases or blacklisted domains. However, as phishing campaigns become more targeted and sophisticated, static filters are easily bypassed.

My work supports the policy shift toward adaptive, machine learning-based filters. For example, companies should consider policies that: Mandate behavioral feature logging (e.g., sender frequency, URL patterns), Integrate ML classifiers into inbound email pipelines and Periodically retrain models using internal phishing and safe email examples From a governance perspective, there is also room for standardization of phishing metadata. If email clients were required to log structured metadata like URL counts, domain origin frequency, and capital letter ratios, it would be easier to implement consistent filters across organizations. In regulated industries (finance, healthcare), phishing detection falls under data protection compliance, including frameworks like GDPR or HIPAA. Policies should encourage investment in interpretable models (like Logistic Regression) so decisions about quarantining or rejecting emails can be explained in case of legal or customer service disputes.

Adopting models like those developed in this project can significantly reduce organizational exposure to phishing threats. By leveraging features already present in email metadata and message content, these models can be deployed quickly without requiring intrusive data collection. Even lightweight implementations could serve as a second filter after traditional rules — improving security without blocking legitimate communications.

CONCLUSION

This project demonstrated that phishing detection can be significantly enhanced by combining traditional text-based features with behavioral indicators derived from email metadata. Through iterative experimentation with Logistic Regression and Random Forest classifiers, I was able to systematically improve model performance by introducing features such as sender domain frequency, URL presence, capitalization ratios, and keyword-based suspiciousness scores.

The baseline Logistic Regression model highlighted the limitations of relying solely on simple content markers, with an accuracy of just 66%. However, subsequent enhancements—particularly in Versions 1.1 and 1.2—achieved near-perfect accuracy and F1 scores above 0.99. These improvements validate the effectiveness of thoughtfully engineered features in boosting classification reliability, even with relatively lightweight and interpretable models like Logistic Regression. Random Forest (Version 2.0) provided a strong comparison point, performing competitively while offering the potential to capture more complex interactions among features.

One of the most important takeaways is that phishing attacks can be detected not only through language patterns, but also through subtle behavioral cues such as abnormal sender domains or excessive use of URLs. This reinforces the need for modern email filtering policies to adopt adaptive, feature-rich approaches that go beyond static keyword lists or blocklists.

However, the project also had several limitations. All evaluations were conducted on a static dataset, which may not fully capture the evolving nature of phishing tactics over time. Additionally, features like sender reputation or temporal frequency (e.g., how often a sender appears over time) were not explored, though they could further improve detection. The models were also not deployed in a live

environment, meaning performance in production conditions—especially on novel phishing emails—remains untested.

This project illustrates how an individual can build an effective and scalable phishing detection system using publicly available data and accessible machine learning techniques. The models I developed required no external infrastructure beyond standard preprocessing and vectorization, making them suitable for academic, experimental, or small-scale enterprise use. While more advanced systems may incorporate live threat intelligence or deep learning, this project shows that even lightweight, interpretable models can meaningfully contribute to cybersecurity defenses—especially when enhanced with carefully selected behavioral features.

APPENDIX

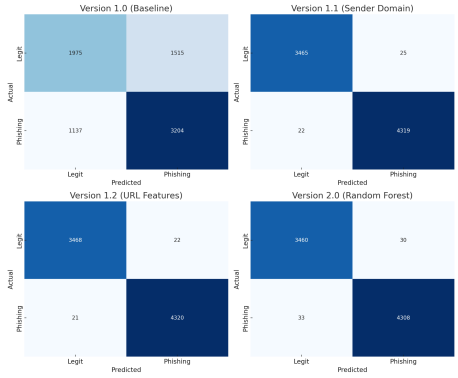

Figure 1: *Confusion matrix heatmaps for all four phishing detection models. Each heatmap illustrates the number of true positives, true negatives, false positives, and false negatives. The baseline model (1.0) shows a high rate of misclassification, particularly false positives, while enhanced versions—especially 1.2 and 2.0—demonstrate significant reductions in both error types, indicating improved classification reliability.*

Table: Detailed Model Performance Comparison

Model Version	Accuracy	F1 Score	True Positives	True Negatives	False Positives	False Negatives
1.0 – Baseline (LogReg)	0.6613	0.6600	3204	1975	1515	1137
1.1 – +Sender Domain (LogReg)	0.9940	0.9946	4319	3465	25	22
1.2 – +URL Features (LogReg)	0.9945	0.9950	4320	3468	22	21
2.0 – Random Forest	0.9920	0.9927	4308	3460	30	33

Table 1: *Comparison of model performance across four phishing detection classifiers. Metrics include overall accuracy, F1 score, and confusion matrix components (true/false positives and negatives). The baseline Logistic Regression model (1.0) performed significantly worse than enhanced versions, with*

version 1.2 achieving the best overall balance of precision and recall. Random Forest (2.0) offered competitive results but with increased complexity.

Bibliography

Verizon. (2023). *2023 Data Breach Investigations Report (DBIR)*. Verizon Business. https://www.verizon.com/business/resources/reports/dbir/

Proofpoint. (2023). *State of the Phish Report*. https://www.proofpoint.com/us/resources/threat-reports/state-of-phish

Cybersecurity & Infrastructure Security Agency. (n.d.). *Phishing guidance and resources*. U.S. Department of Homeland Security. https://www.cisa.gov

Islam, M. R., Abawajy, J. H., & Yearwood, J. (2019). A dynamic rule-based phishing detection system using machine learning. *IEEE Transactions on Network and Service Management, 16*(1), 30–44. https://doi.org/10.1109/TNSM.2019.2893170