Final Paper

Abstract

Phishing emails remain one of the most prevalent cybersecurity threats, leading to financial losses and data breaches worldwide. Automated detection methods using machine learning have become critical in mitigating these risks. In this project, we explored the application of supervised machine learning models to detect phishing emails using the TREC_06 dataset, which contains approximately 15,000 labeled email records, sourced from Kaggle. Our goal was to identify the textual and structural characteristics most indicative of phishing and to develop an effective classification model. After thoroughly preprocessing and applying natural language processing (NLP) techniques, we trained and evaluated Logistic Regression, Random Forest Classifier, and Support Vector Machine (SVM) models enhanced with TF-IDF vectorization. The models were assessed on a range of success metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Our analysis revealed that logistic regression achieved the highest overall performance, attaining 97.7% accuracy, 98.3% precision, 92.3% recall, and an ROC-AUC score of 0.9982, with notable success in predicting phishing emails based on textual patterns and keywords. We also investigated feature importances to understand the distinguishing factors between phishing and legitimate emails. Our feature importance analysis revealed that legitimate emails were characterized by human elements like gratitude expressions ("thanks") and educational domain indicators ("edu"), while phishing emails frequently included unsecured URLs ("http"), generic terms ("company"), and numerical bait patterns ("000"). These insights not only enhanced model interpretability but also provided practical indicators for real-world phishing detection. Despite the strong results, limitations

remain, including the dataset's potential lack of diversity and the challenge of evolving phishing tactics. Future work could involve the application of deep learning approaches and retraining on newer datasets to maintain detection relevance. Overall, this work offers insight into automated phishing detection and highlights challenges and future directions for improving the model and adding adaptability.

Introduction

In an increasingly digital world, phishing attacks represent a major cybersecurity threat. These attacks exploit human vulnerabilities through deceptive emails designed to steal sensitive information, often bypassing traditional security filters. As phishing techniques evolve, the need for adaptive and intelligent detection systems that can identify subtle textual patterns and warning signs in malicious communications has become more urgent. Machine learning approaches offer promising solutions by identifying patterns and anomalies in email content that may indicate malicious intent. In this project, we applied supervised machine learning techniques to detect phishing emails using a labeled dataset known as TREC_06. This dataset, sourced from Kaggle, consists of approximately 15,000 emails labeled as spam (phishing) or not spam, and includes a range of features such as sender, subject line, body text, timestamps, and URLs. Its structure makes it ideal for training models that can classify emails based on their content.

Our primary objective was to investigate which characteristics most strongly indicate phishing attempts and to determine which classification model is most effective for this task. To answer this, we built and compared three classification models: Logistic Regression, Random Forest, and Support Vector Machine (SVM) combined with TF-IDF vectorization. Each model was selected to highlight different strengths: Logistic Regression as a fast, interpretable baseline,

Random Forest for handling structured features and non-linear interactions, and SVM with TF-IDF for identifying important textual signals in unstructured data.

The models were evaluated using several key metrics: accuracy, precision, recall, F1-score, and ROC-AUC. These metrics allow for a nuanced view of performance, especially in the context of phishing detection, where false negatives (missed phishing emails) pose greater risk than false positives. Logistic Regression ultimately emerged as the best-performing model, achieving 97.7% accuracy, 98.3% precision, and a ROC-AUC score of 0.9982. While SVM and Random Forest also performed well, Logistic Regression provided the best balance of simplicity, speed, and interpretability.

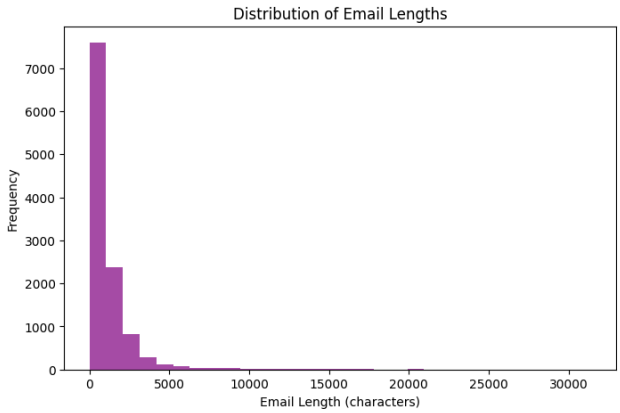
In addition to performance, we also examined feature importances to better understand what distinguishes phishing emails. Legitimate emails often included words like “thanks” or domain-specific tags like “.edu,” suggesting human interaction and trusted sources. In contrast, phishing emails were more likely to contain insecure URL formats (“http”), numeric bait (“000”), and vague references like “company”, words that hint at mass targeting and deception.

While our results are promising, we also acknowledge key limitations. The TREC_06 dataset, while widely used, may not fully capture the diversity of modern phishing attacks. Furthermore, phishing tactics evolve quickly, and models trained on historical data may lose effectiveness over time without regular updates. Additionally, some legitimate-looking emails may still contain malicious links, and our model’s recall, while strong, is not perfect. These limitations suggest a need for further research into deep learning models that can better capture complex sentence structures and contextual clues.

Overall, this project demonstrates the potential of machine learning in phishing detection and highlights how simple, interpretable models can yield powerful results when paired with thoughtful feature engineering. In the following sections, we describe the dataset in greater detail, walk through our modeling pipeline, present evaluation results, and conclude with recommendations for future work and potential real-world applications.

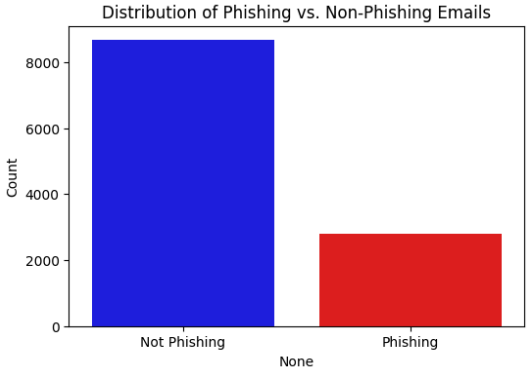
Data

We decided to look at data from Kaggle, and discovered a dataset that included information from phishing emails, and around 15,000 messages. We decided to focus our energy on TREC_06, a recently updated but smaller size file of information. The variables included are “sender”, “receiver”, “subject”, “body”, “date”, “label”, and “urls”. Sender and receiver refer to metadata used to keep track of information sent, subject and body are the main email content, date refers to when the email was sent, label highlights whether the email was flagged as spam, and urls contains information if the email contained any.




```
Emails Containing URLs:
 0    7246
1    4225
```

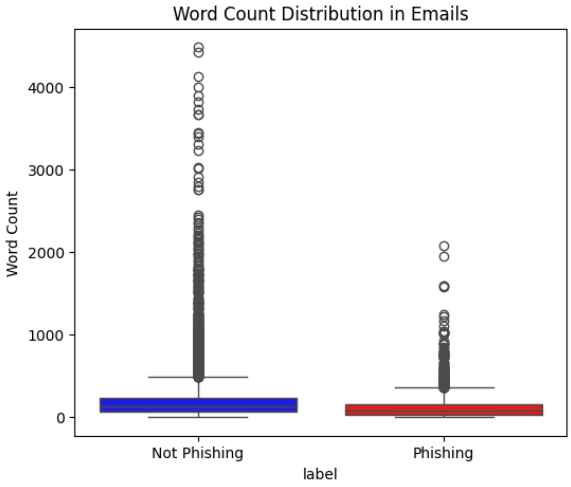
Our main goal with this data is to eventually create a predictive model that can successfully categorize phishing emails. These Text Retrieval Conference (TREC) Spam Track datasets were actually developed for this exact



purpose; to simulate incoming emails that required filters to determine if they were spam or not spam. Since we will have the label variable available for us, we will know whether an email was flagged or not, so the model can successfully train and be tested without this category. Based on initial statistical analyses, we know that there are 8,668 non-spam and 2,803 spam emails present.

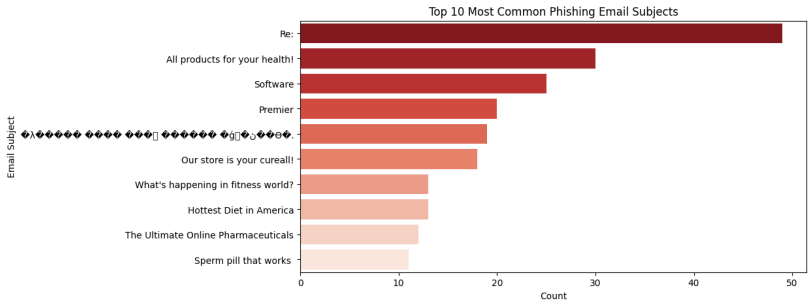
```
Phishing Email Counts:
 0    8668
1    2803
```

There are a few different aspects of the data that we might predict the model will learn to use in its distinctions. Phishing emails seem to include less words, often include “A0” and demanding language such as “will”, “need”, “now”, “want”, and are likely to include “money” as well. The subject line might also be incredibly telling, with distinct lines including “Re:”, long strings of special characters, exclamation points and question marks, and other eye-catching titles.



The main issue we expect to deal with is the sheer size of the dataset. At 15.5 MB, processing with python will be tedious, especially if we attempt to filter it for specific entries. Unfortunately, there is not much that can be done about the processing delays we will likely experience, writing the model code and testing it will take a lot of time, so we will need to carefully plan ahead. Furthermore, as of right now we are planning on splitting the data into

training and non-training for model accuracy testing, however we are unsure of exactly how we will want to split it percentage wise to avoid overfitting or underfitting our model.



Methods

What characteristics make an email most likely to be a phishing attempt?

We use the TREC_06 dataset from Kaggle, which consists of approximately 15,000 emails, each labeled as either phishing (1) or not (0). In this study, each observation is a single email, represented as a row containing features such as body text, subject line, sender, presence of URLs, and timestamp. Since our target variable is binary, this study will use supervised learning for classification rather than regression.

To classify phishing emails, we will experiment with models including Logistic Regression, Random Forest Classifier, and SVM with TF-IDF Vectorization--each of which serves a different purpose. Logistic Regression is a simple classification approach that will estable a baseline for prediction performance. Random Forest Classifier can handle structured features like presence of URLs, email length, and sender frequency while simultaneously reducing overfitting. SVM with TF-IDF Vectorization allows us to extract textural patterns from email bodies using Term Frequency-Inverse Document Frequency (TF-IDF) to convert text into numerical features, like the email body, subject line, and URLs, and then use SVM to analyze

those textural features. This aids in capturing the importance of specific words within the context of the email while reducing the impact of more common words including "and", "in", and "the".

Given that phishing emails rely heavily on textual deception, Natural Language Processing (NLP) techniques will be employed to refine the dataset. Specifically, we will experiment with different NLP techniques such as stemming, lemmatization, and n-grams. Furthermore, the data may require additional cleaning to fix inconsistent timestamps, missing values, or encoding errors in emails.

Success is defined by a model that achieves high predictive accuracy without significant overfitting. Success of the model was measured through accuracy (overall classification correctness), precision and recall (missed phishing email classifications), F1 score (balance between precision and recall), and ROC-AUC score (measure of overall classifier performance). Precision and recall is particularly important for phishing detection since false negatives are riskier than false positives, as these emails can contain potentially dangerous links that steal pertinent receiver information. Visualizations such as confusion matrices and feature importance plots will provide deeper insights into model performance.

Feature engineering may require multiple iterations. We will experiment with different NLP techniques to improve classification accuracy. Additionally, inconsistent timestamps, missing values, and encoding errors may arise. We will preprocess data to address these issues before model training. Finally, regularization techniques and cross-validation will be used to ensure models generalize well to unseen data. If our approach does not achieve the desired performance, we may gain insights into identifying additional phishing indicators. We may find ourselves exploring deep learning models like LSTMs or BERT, or understanding evolving

tactics used in phishing campaigns. By iterating on these approaches, we aim to develop an interpretable model for phishing email detection.

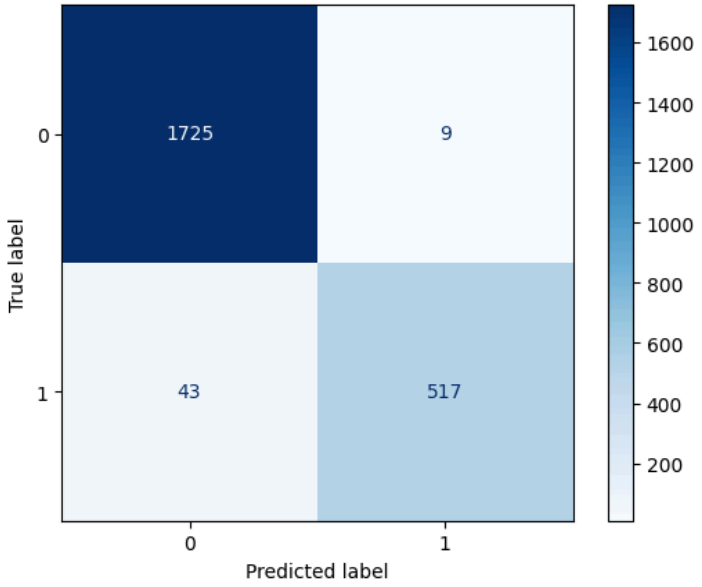
Once the models are trained and evaluated, we will present our findings through various visualizations and performance metrics. We will use confusion matrices to analyze misclassification patterns and evaluate false positives and false negatives, which are critical in phishing detection. Similarly, we will have a summary table comparing accuracy, precision, recall, F1-score, and ROC-AUC for each model to determine the best-performing classifier. Additionally, we will look into the importance of given features for the best performing classification algorithm. Through these results, we aim to answer which characteristics most strongly indicate phishing emails and determine which model is best suited for practical phishing detection applications.

Results

Before delving into full analyses, we wanted to determine which of the three models performed the best. These models included support vector machine (SVM) and TF-IDF vectorization, random forest, and logistic regression. Of these, the logistic regression had the highest accuracy of 97.69% and the highest precision of 98.29%. Additionally, this model had a recall percentage of 0.9232, meaning it was able to successfully identify positive phishing emails 92.32% of the time. Finally, logistic regression had the second highest F1 score of 0.9521 and the highest ROC AUC score of 0.9982, meaning this model was incredibly successful in predicting what emails were and were not phishing.

	Logistic Regression	Random Forest	SVM/TF-IDF Vectorization
Accuracy	0.9773	0.9769	0.9647
Precision	0.9829	0.9828	0.9650
Recall	0.9232	0.9210	0.9647
F1 Score	0.9521	0.9509	0.9648
ROC AUC Score	0.9982	0.9982	0.9871

To better visualize the accuracy of this logistic regression model we settled on, we decided to create a confusion matrix to analyze the relationship between predicted and true labels, either phishing or not. We discovered that out of 1,734 non-phishing emails, our model correctly labeled 1,725 of them, with only 9 being misidentified. Similarly, out of 560 phishing emails, our model correctly labeled 517 and only incorrectly labeled 43. Though the accuracy was higher in determining non-phishing emails, we can still say that this logistic regression model is very effective overall.



Knowing this, we decided to use logistic regression to determine which characteristics are indicative of a phishing email. This is the most important factor in being able to determine, and predict, whether an email is phishing or not. We decided to look at feature importances, in which a positive importance value means an increased chance the email is phishing, and a negative importance value means an increased chance the email is not phishing. Interestingly, we found that characters like “thanks” and “edu” are two of the most important features present in non-phishing emails. This is likely due to the fact that legitimate correspondence have human touches like saying thanks or including any educational links. Furthermore, “http”, “com”, “000”, and “company” were highly indicative of phishing emails. Many of these emails include links to external and insecure websites indicated by the “http” instead of “https” and likely less reputable domains such as “com” instead of “edu” or “gov”. The presence of triple zeros is likely due to the presence of large numbers with multiple zeros present at the end, used to grab the recipient's attention, either malicious saying they owe money or trying to pull at their greed by stating they could be endowed with money if they respond. Finally, “company” is a very general term, and if a specific company were to be contacting someone they would be much more likely to utilize their actual name instead.

	feature	importance	abs_importance
4532	thanks	-5.324070	5.324070
2295	http	4.873127	4.873127
1640	edu	-4.822734	4.822734
3685	ra	4.269415	4.269415
4749	use	-3.732315	3.732315
776	board	-3.682443	3.682443
4973	wrote	-3.506038	3.506038
1547	does	-3.302036	3.302036
2717	list	-3.300271	3.300271
1087	com	3.201612	3.201612
2181	handyboard	-3.195584	3.195584
1	000	3.081886	3.081886
4756	using	-2.964023	2.964023
1111	company	2.876386	2.876386
3923	robot	-2.836361	2.836361
2202	hb	-2.817252	2.817252
4660	try	2.795099	2.795099
4715	university	-2.794609	2.794609
3562	problem	-2.633563	2.633563
2294	html	-2.592719	2.592719

We also determined the prevalence of different categories of special characters present in phishing and non-phishing emails. Overall, we really only found a higher occurrence of uppercase letters in phishing emails, which is likely a tactic used to better grab the recipient's attention and persuade them to think the email contains pertinent and time-sensitive information.

Conclusion

This project successfully demonstrated the feasibility of building an accurate phishing detection model using basic machine learning techniques and natural language processing. Logistic Regression, despite its simplicity, emerged as the top-performing model, achieving the best accuracy, precision, and recall metrics. This suggests that even simple models, when combined with thoughtful feature engineering, can be highly effective in cybersecurity applications.

Our investigation found that linguistic features play a critical role in phishing detection. Phrases expressing gratitude ("thanks"), educational domain indicators ("edu"), and proper company identifiers are strong indicators of legitimate communications. In contrast, the frequent presence of generic terms ("company"), unsecured URLs ("http"), numerical bait ("000"), and excessive capitalization pointed strongly toward phishing attempts. These findings highlight the subtle yet exploitable differences between legitimate and malicious emails.

However, despite these successes, several limitations must be acknowledged. First, our model was trained on a single dataset representing a specific collection of emails, potentially limiting its generalizability to emails from other institutions or phishing campaigns employing new strategies. Phishing tactics evolve rapidly, and attackers continuously adjust their methods to

evade detection. A static model trained on historical data may lose effectiveness over time without regular retraining or enhancement.

Additionally, while the model's recall rate was high, it was not perfect, meaning some phishing emails could still slip through. In the real world, even a small number of missed phishing emails can have serious consequences underlining the need for continuous improvement. Future directions could include implementing deep learning models like LSTM networks that better capture sequence patterns in email text or fine-tuning transformer models like BERT that can understand deeper contextual relationships within emails.

Finally, a promising extension of this work would be to apply the model to more organization-specific datasets, such as phishing emails received by UVA students and staff. training on local phishing campaigns could allow the model to develop a better understanding of threats specific to particular demographics or institutions, resulting in a much more customized defense mechanism. In conclusion, our project not only created an effective model for detecting phishing emails, but also highlighted important features distinguishing legitimate emails from malicious ones. It reaffirmed the potential of data-driven methods in cybersecurity that can better protect users in an increasingly digital world.

Furthermore, the insights gained from this study highlight important directions for future research and application. By building on these findings, researchers can develop more refined strategies that address current limitations and explore new opportunities. Continued investigation will not only deepen our understanding of the topic but also contribute to broader advancements within the field. Ultimately, sustained commitment to this area of inquiry will help translate theoretical knowledge into practical, real-world benefits.

References

1. A. I. Champa, M. F. Rabbi, and M. F. Zibran, “Why phishing emails escape detection: A closer look at the failure points,” in *12th Interna- tional Symposium on Digital Forensics and Security (ISDFS)*, 2024, pp. 1–6.

2. A. I. Champa, M. F. Rabbi, and M. F. Zibran, “Curated datasets and feature analysis for phishing email detection with machine learning,” in 3rd IEEE International Conference on Computing and Machine Intelligence (ICMI), 2024, pp. 1–7.