PhishFence: Integrating Explainable AI with Probabilistic Classifiers for Phishing Detection

Andrew Lee[1,4], Thomas Ha[2,4], Connor Lee[3,4], Eugene Pinsky[4]

[1] Yongsan International School of Seoul, Seoul, South Korea 04347; [2] Sharon High School, Sharon, MA 02067;
[3] Palos Verdes Peninsula High School, Rolling Hills Estates, CA 90274; [4] Boston University, Boston, MA 02215

Introduction

Phishing Attacks and Emails

- Phishing consists of approximately 25% of all internet crime complaints.
- Phishing attacks caused over $70 million in losses in 2024 alone.
- Nearly half of all cyber attacks involve some type of phishing attack.

Problem Statement

- Current models can accurately detect phishing emails but operate as black boxes.
- End-users and analysts receive minimal transparency regarding classification decisions, limiting interpretability and trust.

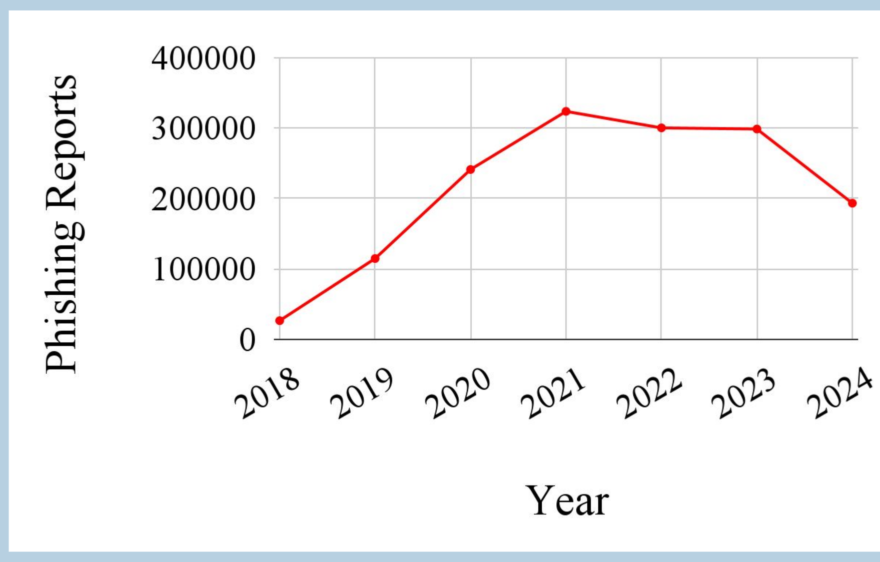

Fig. 1: While phishing rates have decreased, they are still many times higher than they were 7 years ago.

Goals

- Develop a phishing detection pipeline that matches the accuracy of current models while enhancing explainability.
- Produce a front-end interface for users to submit emails for classification, providing real-time feedback and recommendations.

Methods

Data Sources and Preprocessing

- Combine six partially-processed public datasets
- Remove extraneous columns, drop invalid rows, and reprocess for additional summary statistics (i.e., word count, URL count, etc.)
- Remove stopwords & special characters for natural language processing

Vectorization and Text Embedding Techniques

Text Frequency-Inverse Document Frequency (TF-IDF)

- Considers word frequency in a document and rarity across the dataset
- Efficient, but ignores word order and context; poor at semantics

Sentence-BERT (SBERT)

- Densely vectorizes documents using a pre-trained BERT model
- Strong contextualization, but computationally heavy
- Requires chunk-and-pool document embedding due to a 512 token limit

Evaluating Classification Models

- *MLP Classifier*: Neural network approach; accurate but slow.
- *Logistic Regression*: Simple mathematical approach; accurate and fast.
- *Random Forest*: Many decision trees; acceptable accuracy but slow.
- *Multinomial/Gaussian Naive Bayes*: Naive NLP approach; quick but inaccurate.
- *BERT Classifier*: Fine-tuned version of a pre-trained BERT model and tokenizer; captures textual nuance and context; slowest to train but highly accurate.

Results

Model (with TF-IDF)	Accuracy	Precision	Recall	F1-Score
MLP Classifier	98.2%	98.2%	98.4%	0.983
Logistic Regression	98.0%	97.9%	98.4%	0.981
Random Forest	97.5%	98.0%	97.3%	0.976
Multinomial Naive Bayes	93.1%	98.0%	88.6%	0.931

Table 1: The MLP model outperforms other models in all metrics.

Model (with SBERT)	Accuracy	Precision	Recall	F1-Score
MLP Classifier	98.1%	98.4%	97.9%	0.982
Logistic Regression	95.5%	97.5%	93.8%	0.956
Random Forest	95.1%	95.3%	95.3%	0.953
Gaussian Naive Bayes	89.0%	92.1%	86.4%	0.892

Table 2: Model performance generally decreases with SBERT text encodings.

Model	Accuracy	Precision	Recall	F1-Score
`bert-base-uncased`	99.27%	99.35%	99.26%	0.993
`bert-large-uncased`	99.31%	99.27%	99.40%	0.993

Table 3: The `bert-large-uncased` and `bert-base-uncased` models have similar performance.

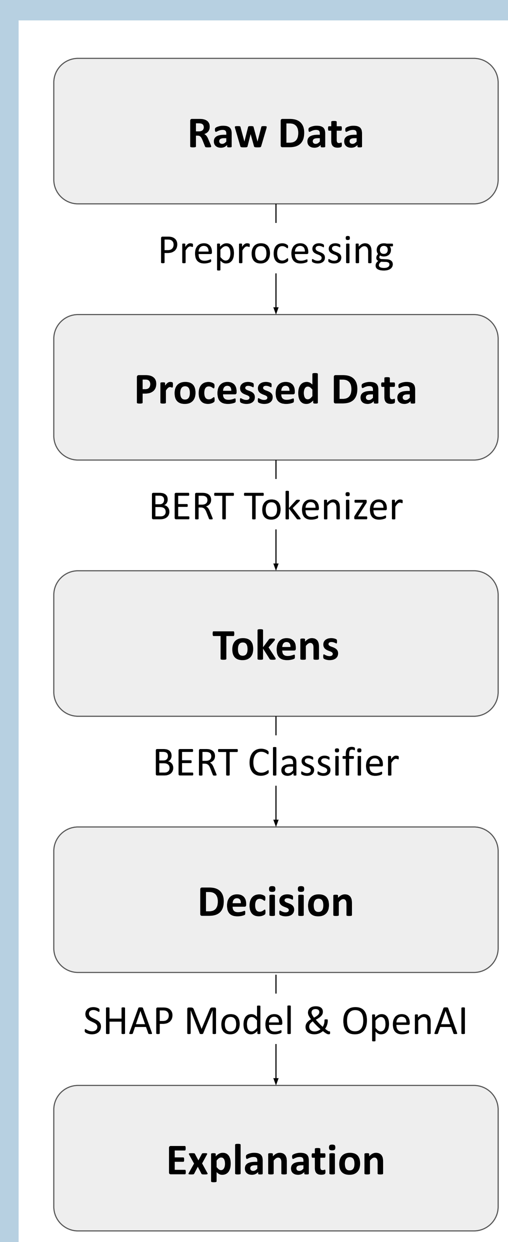

Fig. 2: PhishFence pipeline.

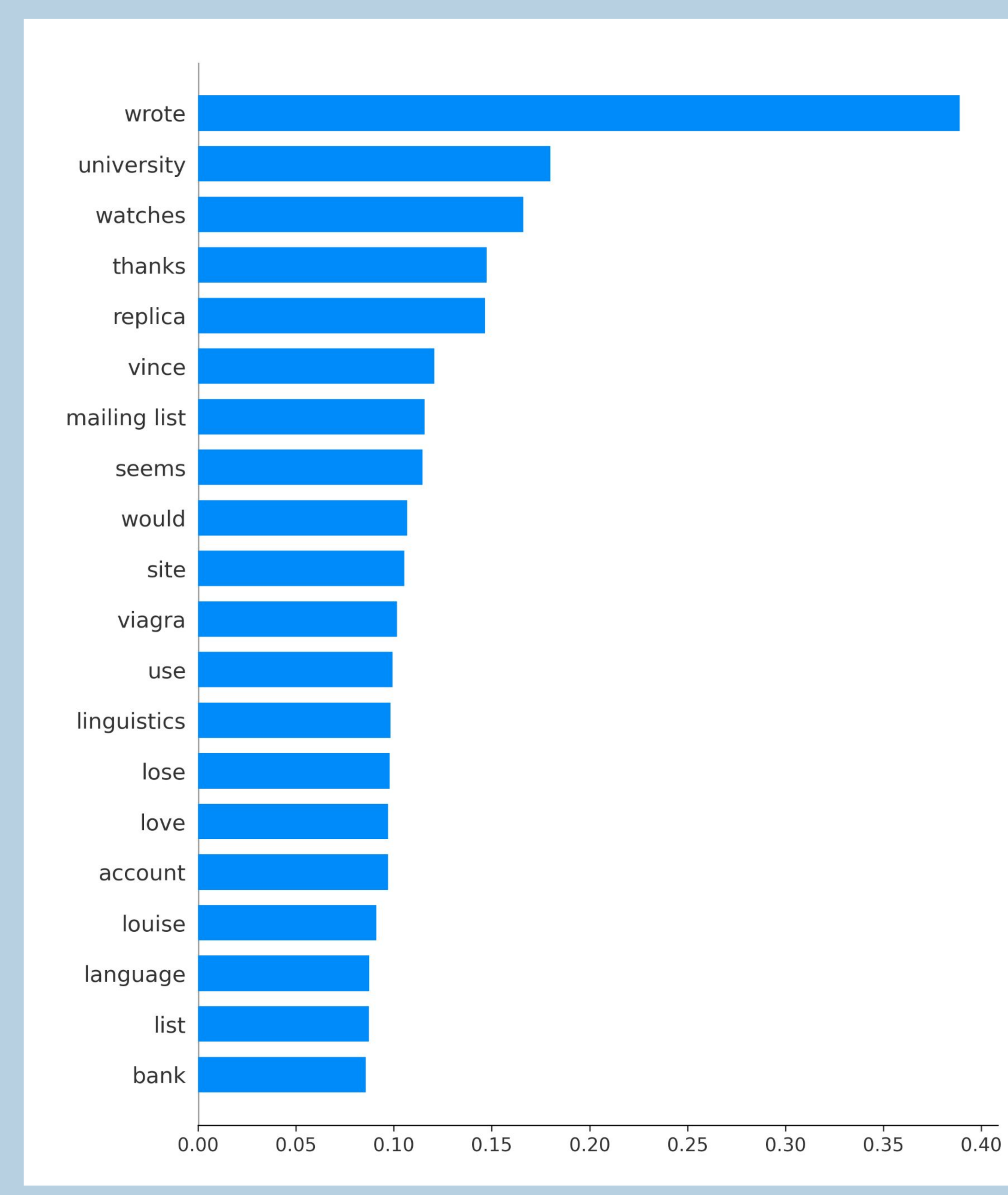

Fig. 3: Aggregated SHAP results for a random sample of emails.

Discussion

Findings

- TF-IDF outperforms SBERT vectorization with simpler models.
- BERT outperforms simpler models regardless of vectorization techniques.
- SHAP combined with LLMs can be effectively used to create approachable, high-level explanations.

Limitations

- *BERT token limit*: Inputs longer than 512 tokens must be chunked, leading to a loss of contextual continuity and reduced classification accuracy.
- *Language Bias*: Classification in languages other than English is not supported.
- *Static Data*: Outdated training data may not reflect modern phishing tactics.

Future Work

- *Broader Datasets*: Current datasets are skewed toward specific message types (e.g., emails or SMS), limiting the model's generalizability. Future efforts should aim to collect more diverse, cross-platform phishing data.
- *Model Diversification and Optimization*: Future experiments could explore alternative NLP architectures, such as RoBERTa, DistilBERT, or domain-specific transformers, to improve accuracy, speed, or resource efficiency.
- *Deployment*: The usability and impact of PhishFence can be further enhanced by developing a browser extension, mobile app, or API. Broadening public access would strengthen its role in combating real-world threats.

External Links

References

Appendices

Source Code

Acknowledgements

We would like to thank our mentors, **Patrick Bloniasz**, **Dr. Eugene Pinsky**, **Tharunya Katikireddy**, **Tejovan Parker**, **Zhengyang Shan**, and **Kevin Quinn**, for their support and contributions to our project. We would also like to extend our gratitude to **Boston University** and the **RISE** program for this opportunity.