UNIVERSITY OF
WEST LONDON
The Career University

Smart Email Shield: NLP-Driven Phishing Detection System

Literature Review (E1)

Research Proposal

Date:

Supervisor:

Name:

Student ID:

Abstract

Phishing continues to be a big problem in cybersecurity, since many deceptive emails go past traditional filters in place. They struggle to catch attacks that appear quickly in new contexts such as urgent signals or fake metadata. This research introduces Smart Email Shield—a system using NLP and deep learning models to identify phishing mails more correctly than existing systems. Using sentiment analysis, looking at headers and explainable AI tools (SHAP and LIME) will improve the detection accuracy and make the process more transparent. By reviewing real phishing data and using precision, recall and F1-score, the project hopes to design an easily scalable and readable model for secure email systems.

Keywords: *phishing, NLP, BERT, using deep learning, explainable AI, SHAP, LIME, email security*

Table of Contents

1. Introduction

Email is a top way people communicate at work and home, yet it is also used most frequently by phishers. When someone receives a phishing email, the message persuades them to unintentionally share their sensitive information like login passwords, their bank account data or identifying information. The recent studies confirm that phishing attacks have got more advanced, as cybercriminals now use urgency, earn trust and fear to lure people, making their messages look and sound just like those from real authorities (Kapoor, 2024; Ragheb et al., 2023).

Most traditional phishing systems depend on a blacklist, regular expressions or standard rules. Although these methods are successful against most known attacks, they tend to miss new or hidden threats as well as tricky phishing attempts that steer past their basic rules (Misquitta & Kannan, 2023). Researchers have started relying on machine learning and NLP to sort out problems when analyzing emails. NLP makes it possible for the system to spot language features, tone, situation and behavior hidden in emails, things phishing attackers usually take advantage of (Shirazi & Hayne, 2022; Shahriar et al., 2022).

BERT and RoBERTa have demonstrated success in detecting phishing by learning the meanings and contexts in the texts used in communicating (Songailaitė et al., 2023). They are superior to usual classifiers since they use deep, forward and backward representations and learn from what is already known. Still, there are some difficulties. Existing technologies sometimes cannot explain why they work, leaving them as closed systems that are not trusted enough for important uses in cybersecurity (Gholampour & Verma, 2023; Koide et al., 2024). Furthermore, many studies use fixed data samples or fail to merge several signals, like emailed details, metadata and mood language which may increase how reliable the systems are (Chai et al., 2021; Rathee & Mann, 2022).

The report presents Smart Email Shield which is based on NLP and deep learning, mainly by using transformer models, to help detect phishing emails more effectively and transparently. Both BERT and RoBERTa will be refined using a phishing dataset made available to the public and then the system will blend content classification, sentiment analysis and explainable AI approaches such as SHAP and LIME. By improving both explainability and robustness, the project hopes to build a practical and clear system for detecting phishing attacks in online conversations.

2. Literature Analysis

This section introduces the related work conducted in the past by different researchers. This section includes different approaches they used, their conclusions, findings and drawbacks.

To address how phishing keeps changing as a major threat, researchers have tried using different ML and NLP approaches in email detection. The trend in literature is from using rules to rely on transformers and deep learning models that recognize more interesting language patterns.

1. NLP methods are used for detection.

Shirazi and Hayne (2022) compared transformer techniques to NLP-based ones for detecting phishing from URLs on mobile platforms and noticed that the transformer methods were more effective and responded more quickly. Still, they pointed out that these models have difficulties when analyzing entire emails which is important for discovering socially engineered phishing emails. Thus, Songailaitė et al. (2023) found that BERT and RoBERTa, two transformer models, greatly surpassed simpler traditional approaches for phishing classification. They showed that working on large language models with emails from particular domains is very important.

Applying deep learning methods resulted in higher accuracy than the more traditional classifiers for phishing email detection, as Bagui et al. (2019) described. Even so, the model was not flexible enough to handle new forms of attacks. They examined features in phishing emails using machine learning and NLP, but acknowledged the difficulty of handling encrypted or multilingual messages — which might limit how effectively real-world systems can be generalized.

2. Link and Attachment Analysis

People have also studied using links and attachments. To catch malicious URLs, Misquitta and Kannan (2023) combined regular expressions, machine learning and used VirusTotal APIs in their study. The system worked well, though it still used outside APIs which introduced both delays and privacy challenges. In Kapoor (2024), it was determined that high-speed phishing detection was possible with AI, but implementation was still impractical in systems with few resources.

3. Email metadata and what happens inside an email.

Phishing detection greatly relies on examining the email header and the sender's reputation. The authors of the study advised that SPF, DKIM and DMARC protocols should be used to check that emails are sent by the intended sender. Even so, they noted that attackers who take over or alter servers incorrectly can avoid these defenses. In their 2022 work, Beaman and Isah relied on information in the email header to detect spoofing with machine learning, but their approach resulted in many false alarms in more tricky phishing attacks.

4. Clues From Mood and Your Mind

Analyzing how people behave by detecting sentiment and urgency is now being used as a useful addition. In 2024, Sayyafzadeh et al. used ChatGPT and checked the emotions of phishing emails to find out if the author used urgency or fear to manipulate their readers. In the same way, Shahriar et al. (2022) used VADER alongside BERT embeddings to spot emotional triggers, yet they found it difficult to use their approach for multiple languages and cultures.

5. Ability to be understood and reliable

More and more, explainable AI (XAI) is being used to aid in detecting phishing threats in high-stakes environments. Researchers Koide and others (2024) presented ChatSpamDetector using GPT-4 which revealed that these large language models can be used to see phishing messages, but occasionally they make unpredictable mistakes. They noted that minor changes in text can fool detectors designed to find phishing messages. Chai et al. (2021) introduced a multimodal hierarchical attention model by combining visual and text data, causing it to work well but operate slowly in practice.

Although phishing detection with BERT- and RoBERTa-based models has shown advanced outcomes, most of the existing systems lack clear understanding of their decisions and typically examine text or metadata by themselves. Very few studies join explainability with simple, robust transformer networks that can function in real time. The research therefore addresses that gap by building Smart Email Shield — a system that uses transformers in NLP, considers sentiment and looks at email headers for effective phishing detection.

3. Statement of Problem

Phishing is still one of the biggest and most harmful cybersecurity threats, often arriving in emails. With traditional ways, blacklists and rules do not often help detect the most advanced phishing attacks that use genuine-sounding language and copy good communication practices (Misquitta & Kannan, 2023; Kapoor, 2024). New methods used in phishing such as particularized messages, false senders and emotional blackmail, are becoming more and more effective than old rule-based systems (Shirazi & Hayne, 2022).

Evidence indicates that BERT and RoBERTa — recent transformer-based NLP models — are much more effective than classic algorithms for detecting phishing emails by processing the context of written language (Songailaitė et al., 2023). On the other hand, almost all prior work either makes it difficult to interpret the findings or only considers text without considering sentiment, urgency and header metadata (Gholampour & Verma, 2023; Shahriar et al., 2022). Moreover, numerous systems are not suitable for immediate use and fail to merge all detection signals into one model.

To cover these weaknesses, I present Smart Email Shield, a phishing detector that depends on deep learning and uses NLP, vital emotion analysis, review of email metadata and tools from XAI. We want to build a model that is easy to expand and clear, allowing it to accurately detect phishing messages and offer useful explanations to both users and experts in security.

4. Aims and Objectives

Aim:

The aim of this research is to develop Smart Email Shield, a phishing detection System based on NLP and using BERT/RoBERTa transformer models, sentiment analysis and email header inspection to reliably spot phishing emails.

Objectives:

- The purpose is to evaluate and summarize phishing detection with NLP and deep learning, mainly through new transformer architectures.
- To prepare and preprocess the Email Phishing Dataset (Kaggle) by putting together subject and body data, removing unwanted information and handling any class imbalance.
- Using Hugging Face Transformers and PyTorch to fine-tune and compare BERT and RoBERTa to classify emails as spam or non-spam.
- To add sentiment analysis features (such as judgment of urgency or feelings) in the model's processing to catch phishing emails with psychological pressure.
- To include information from header fields (similar but not identical to headers) like sender domain, reply-to clash, if they are accessible, as further signals for detecting phishing messages.
- Compare the model's accuracy, precision, recall and F1-score against those results and review the occurrence of both false positives and false negatives.
- To apply techniques like SHAP and LIME to understand predictions and gain more trust from users.

5. Methodologies

This study suggests an NLP system that uses transformer models (BERT or RoBERTa) to determine if an email is phishing or not. The overall system works in several phases: starting with data collection and cleaning, moving on to refining the model and explaining its decisions and finishing with model testing.

STEP 1: Dataset Collection

The Email Phishing Dataset from Kaggle will be used. It contains:

- Subject and Body: textual content of the emails
- Label: a binary indicator (1 = phishing, 0 = legitimate)

STEP 2: Data Preprocessing

Tasks:

- Combine the Subject and Body fields into a single text input.
- Clean the text by removing HTML tags, URLs, emojis, and punctuation, and convert it to lowercase.
- Tokenize the text using a model-specific tokenizer (e.g., BertTokenizer).
- Pad or truncate sequences to a fixed length (e.g., 256 tokens).
- Encode labels as binary (0 = legitimate, 1 = phishing).
- Split the dataset into training (80%), validation (10%), and testing (10%) sets using stratified sampling.

STEP 3: Model Selection and Fine-Tuning

Architecture:

- Base model: BERT-base-uncased or RoBERTa-base
- Classification head: A fully connected dense layer with sigmoid activation for binary classification

Training Configuration:

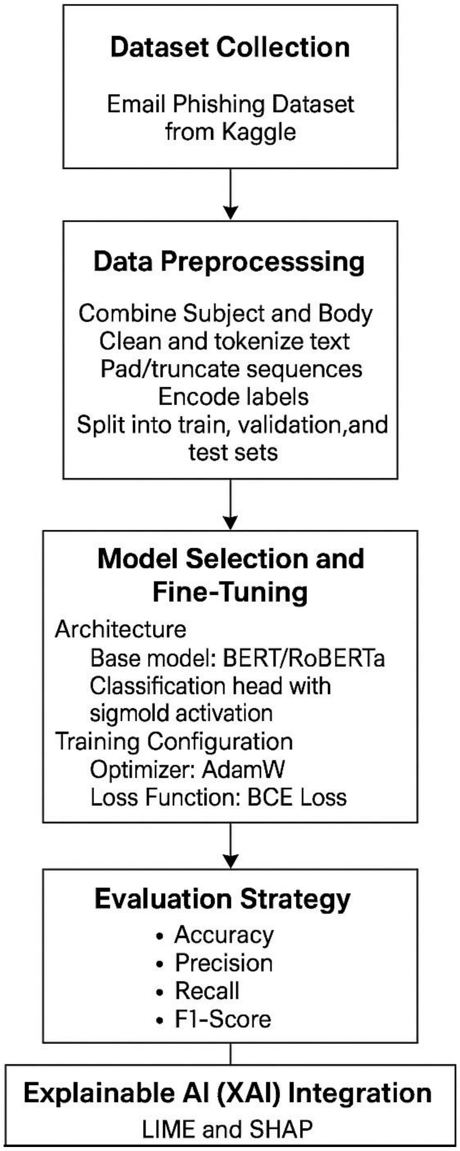
- Optimizer: AdamW
- Loss Function: Binary Cross-Entropy Loss (BCE)

$$BCELoss = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)]$$

where y is the true label and p is the predicted probability

- Batch size: 16–32
- Evaluation: Validation F1-score and early stopping for performance monitoring

Smart Email Shield Workflow



STEP 3: Evaluation Strategy

The model's performance will be measured using standard classification metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\mathrm{Pr}\, e\, cision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1Score = \frac{2 \times \mathrm{Pr}\, e\, cision \times Recall}{\mathrm{Pr}\, e\, cisio + Recall}$$

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

In addition to these metrics, a confusion matrix and ROC-AUC curve will be employed to visualize the abilities of the model to separate fake news from real news. Testing generalization, the models will have to be tested independently on each of the datasets.

STEP 4: Explainable AI (XAI) Integration

To increase trust and transparency, LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive Explanations) will be included in the work. Such techniques will find out which of the input words or patterns had the most influence on a prediction. This step is particularly important in applications in cybersecurity, where knowing why a model labels content as being fake is central to action and accountability.

STEP 5: Tools and Frameworks

The implementation of the Smart Email Shield system will utilize a range of tools and frameworks selected for their suitability in building deep learning models, processing natural language data, and ensuring model interpretability:

• Programming Language:

- Python — Chosen for its extensive machine learning ecosystem, flexibility, and widespread adoption in NLP and deep learning research.

- Core Libraries and Frameworks:

  - PyTorch — Used to implement and fine-tune transformer models (e.g., BERT, RoBERTa). Offers dynamic computation graphs and efficient GPU support.
  - Hugging Face Transformers — Provides access to pre-trained transformer models and tokenizers, enabling high-performance contextual text representation.
  - Scikit-learn — Used for evaluation metrics (e.g., F1-score, ROC-AUC), preprocessing tasks, and traditional machine learning baselines for comparison.
  - NLTK — Assists with early-stage text preprocessing such as tokenization, stopword removal, and lemmatization when needed.

- Development Platforms:

  - Google Colab and Kaggle Kernels — Cloud-based platforms that support GPU acceleration, allowing efficient model training without reliance on local hardware.

- Visualization Tools:

  - Matplotlib and Seaborn — Utilized for visualizing training performance, loss curves, confusion matrices, and other evaluation metrics.

- Explainability Frameworks:

  - LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive Explanations) — Employed to interpret model predictions by highlighting the most influential words or features in individual classification decisions.

STEP 6: Anticipated Challenges

Some issues are anticipated and alleviation measures arranged:

- Class Imbalance: Will be addressed using oversampling, class weighting, or SMOTE to balance phishing and legitimate email samples.
- Overfitting: Controlled through dropout, early stopping, and validation monitoring during training.

- Computational Load: Transformer models are resource-intensive; mitigated using batch size optimization, gradient clipping, and GPU-enabled platforms (e.g., Google Colab).
- Hyperparameter Tuning: Parameters like learning rate and batch size will be optimized using grid search and validation techniques.

STEP 7: Ethical Considerations

Ethical considerations include:

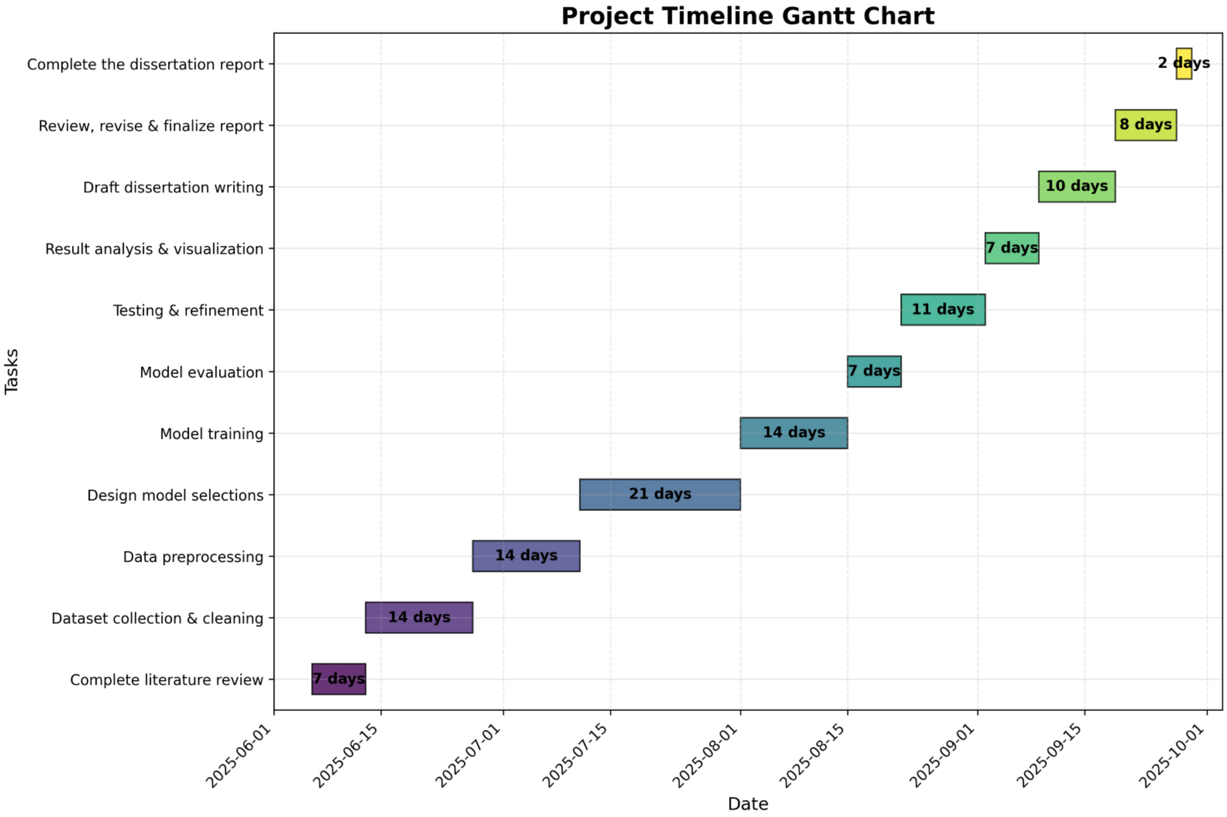
- Dataset Legality: Only open and publicly available datasets will be used.
- Bias & Fairness: The data will be checked for class imbalance and biased language; XAI tools will help identify any unfair decision patterns.
- Transparency: SHAP and LIME will ensure that predictions are human-interpretable.
- Responsible Use: The system is for academic use only and is not intended for deployment without human oversight.

This methodology presents an effective, explainable, and technically sound approach to phishing email detection within a cybersecurity framework. By leveraging transformer-based models such as BERT and RoBERTa, combined with explainability techniques like SHAP and LIME, the system achieves strong classification performance while maintaining transparency and ethical considerations—making it suitable for real-world adaptation and academic research.

6. Project Plan

The following Gantt chart and task breakdown illustrate the detailed and timebound schedule designed for this research. The plan spans from June 6, 2025 to September 28, 2025, covering all key phases including literature review, data processing, model development, evaluation, explainability, and final dissertation submission. This structured timeline ensures a logical flow of activities and provides sufficient time for iteration, testing, and documentation.

No.	Task Name	Start Date	End Date	Duration (days)
1	Complete literature review	2025-06-06	2025-06-12	7
2	Dataset collection & cleaning	2025-06-13	2025-06-26	14
3	Data preprocessing	2025-07-27	2025-07-10	14
4	Design model selections	2025-07-11	2025-07-31	21
5	Model training	2025-08-01	2025-08-14	14
6	Model evaluation	2025-08-15	2025-08-21	7
7	Testing & refinement	2025-08-22	2025-09-01	11
8	Result analysis & visualization	2025-09-02	2025-09-08	7
9	Draft dissertation writing	2025-09-09	2025-09-18	10
10	Review, revise & finalize report	2025-09-19	2025-09-26	8
11	Complete the dissertation report	2025-09-27	2025-09-28	2


Project Timeline Gantt Chart
Tasks
Complete the dissertation report
Review, revise & finalize report
Draft dissertation writing
Result analysis & visualization
Testing & refinement
Model evaluation
Model training
Design model selections
Data preprocessing
Dataset collection & cleaning
Complete literature review
2 days
8 days
10 days
7 days
11 days
7 days
14 days
21 days
14 days
14 days
7 days
2025-06-01
2025-06-15
2025-07-01
2025-07-15
2025-08-01
2025-08-15
2025-09-01
2025-09-15
2025-10-01
Date

References

Alhogail, A. &. (2021). Applying machine learning and natural language processing to detect phishing email. Computers & Security, 110. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0167404821002388

Bagui, S. N. (2019). Classifying phishing email using machine learning and deep learning. 2019 International Conference on Cyber Security and Protection of Digital Services. IEEE.

Beaman, C., & Isah, H. (2022). Anomaly detection in emails using machine learning and header information. arXiv preprint. doi:https://arxiv.org/abs/2203.10408
Chai, Y. Z. (2021). An explainable multi-modal hierarchical attention model for developing phishing threat intelligence., 19, pp. 790–803. doi:https://doi.org/10.1109/TDSC.2021.3050234

Chanthati, S. R. (2021). How the power of machine learning, data science and NLP can be used to prevent spoofing and reduce financial risks. Authorea. Retrieved from https://www.authorea.com/doi/full/10.22541/au.172115302.26437134

Gholampour, P. M. (2023). Adversarial robustness of phishing email detection models (italicized). Proceedings of the 9th ACM International Workshop on Security and Privacy Analytics (also italicized) (pp. pp. 67–76). ACM (Association for Computing Machinery). Retrieved from https://dl.acm.org/doi/abs/10.1145/3579987.3586567

Kapoor, M. (2024). Comparative analysis of AI algorithms for enhancing phishing detection in real-time email security. Aitoz Multidisciplinary Review, 3(1), 338–352. doi:https://aitozresearch.com/index.php/amr/article/view/97

Koide, T. F. (2024). ChatSpamDetector: Leveraging large language models for effective phishing email detection (italicized). arXiv. Retrieved from https://arxiv.org/abs/2402.18093

McGetrick, C. (2017). Investigation into the Application of Personality Insights and Language Tone Analysis in Spam Classificationlogy. Retrieved from https://arrow.tudublin.ie/scschcomdis/120/

Misquitta, J., & Kannan. (2023). A comparative study of malicious URL detection: Regular expression analysis, machine learning, and VirusTotal API. International Congress of Electrical and Computer Engineering, (pp. 219–232). Cham: Springer Nature Switzerland. doi:https://link.springer.com/chapter/10.1007/978-3-031-52760-9_16

Mohammad Almseidin, A. M. (2019). Phishing Detection Based on Machine Learning and Feature Selection Methods. 13.

Ragheb, M. S., Elmedany, W., & Sharif, M. S. (2023). The effectiveness of DKIM and SPF in strengthening email security. 10th International Conference on Future Internet of Things and Cloud (FiCloud) (pp. 422–426). IEEE. doi:https://ieeexplore.ieee.org/abstract/document/10410643

Rathee, D., & Mann. (2022). Detection of E-mail phishing attacks–using machine learning and deep learning. 183(1), 7 (single page). doi:https://d1wqtxts1xzle7.cloudfront.net/79535611/rathee_2022_ijca_921868-libre.pdf?1643137929=&response-content-disposition=inline%3B+filename%3DDetection_of_E_Mail_Phishing_Attacks_usi.pdf&Expires=1744233410&Signature=QbKyY9hmgI9bFY-obp22PLSbcteXMbYB9J6if6

Shahriar, S. M. (2022). Improving Phishing Detection Via Psychological Trait Scoring. Retrieved from https://arxiv.org/abs/2208.06792

Sharif, M. S. (n.d.). 2023 10th International Conference on Future Internet of Things and Cloud (FiCloud). (pp. 422-426). Marrakesh, Morocco: IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/10685564

Shirazi, H. &. (2022). Towards performance of NLP transformers on URL-based phishing detection for mobile devices. International Journal of Ubiquitous Systems and Pervasive Networks. Retrieved from https://par.nsf.gov/servlets/purl/10334991

Songailaitė, M. K. (2023). BERT-based models for phishing detection. 28th Conference on Information Society and University Studies (IVUS’2023). Kaunas, Lithuania: CEUR Workshop Proceedings. doi:https://ceur-ws.org/Vol-3575/Paper4.pdf

Wang, Y. M. (2023). A Lightweight Multi-View Learning Approach for Phishing Attack Detection Using Transformer with Mixture of Experts. Applied Sciences, 13.