Literature Review

Phisher-Zero

By:
Zehra Jabeen Mirza 22K-4781
Muhammad Shehryar Zubair Khan 22k-4736
Anas Ghazi 21L-5081

1. Introduction

Phishing remains one of the most widespread cybersecurity threats, with attackers exploiting deceptive emails, malicious URLs, and fraudulent websites to steal sensitive information. According to the 2023 Verizon Data Breach Investigations Report, phishing was responsible for over 36% of breaches, making it the single largest threat category [1]. Attackers employ social engineering, urgency cues, and obfuscated URLs to bypass traditional filters.

Traditional blacklist and heuristic-based methods have proven inadequate, as phishing domains can be generated dynamically and often remain undetected until reported [2]. This has motivated the adoption of machine learning (ML), natural language processing (NLP), and hybrid ensemble models to achieve real-time, adaptive phishing detection [3].

2. Existing Approaches to Phishing Detection

2.1 Rule-based and Heuristic Approaches

Early detection relied on URL blacklists, signature-based filters, and header analysis [4]. Although efficient, these methods exhibit poor adaptability against zero-day phishing domains.

2.2 Machine Learning Approaches

Supervised ML models such as Random Forests, Decision Trees, and Support Vector Machines (SVM) have shown effectiveness in distinguishing phishing from benign URLs using lexical and domain-based features [5], [6]. For example, URL classifiers trained on PhishTank data achieved accuracies above 93% [7]. However, adversarial manipulation (e.g., homoglyph attacks, cloaking) can reduce their robustness.

2.3 Deep Learning and Transformer-based Models

Recent work emphasizes NLP-driven models. Transformers like BERT and DistilBERT capture semantic and contextual cues, enabling better detection of fraudulent intent in phishing emails [8]. Proprietary large language models (LLMs), such as OpenAI GPT and Google Gemini, extend this ability by providing explainability and contextual analysis at scale [9].

2.4 Hybrid and Ensemble Approaches

Combining URL-based and NLP-based detection improves resilience. Ensemble systems aggregate multiple classifiers, reducing false positives and increasing generalization [10]. Explainable AI methods have also been integrated into phishing detection frameworks to foster user trust [11].

2.5 Browser-integrated Defenses

Browser-based approaches, such as Google Safe Browsing and Microsoft SmartScreen, provide endpoint-level protection [12]. Academic studies have also proposed Chrome extensions for phishing detection, but most lack interpretability or adaptability [13].

3. Related Work on Email and URL-based Phishing Detection

The **PhishTank** and **OpenPhish** repositories provide updated malicious URL feeds and have been widely used for ML training [7]. The **Enron Email Dataset** has served as a benchmark for NLP-based phishing detection research [14]. Past studies report high detection rates when using ensemble learning and deep models, but challenges persist in real-world latency, scalability, and explainability [3], [8].

These limitations highlight the need for real-time, browser-integrated, explainable phishing detection solutions.

4. Technology Stack for the Proposed System

The technology stack outlined is tentative and may evolve over the course of the project as new requirements emerge, optimizations are identified, or better-suited tools become available

4.1 Frontend (Chrome Extension)

The extension is built with **React + Vite**, using **Manifest V3 APIs** for content scripts and service workers. **TailwindCSS** and **ShadCN/UI** provide lightweight, modern components for usability [15].

4.2 Backend / API Layer

A backend powered by **FastAPI (Python)** handles ML inference with serverless deployment options such as **Google Cloud Run** and **AWS Lambda** [16]. Node.js/Express is considered as an alternative for JavaScript-heavy integration.

4.3 Agents

- **NLP Agent**: Uses **Gemini API** for semantic detection of phishing intent, with **DistilBERT** as fallback [8], [9].
- **URL Agent**: Trains scikit-learn classifiers (Random Forest, Gradient Boosting) on **PhishTank + OpenPhish datasets** [7].
- **Ensembler Agent**: Aggregates NLP + URL predictions into a final phishing score [10].
- **Explainer Agent**: Uses Gemini API to generate human-readable justifications [11].

4.4 Data Sources

- **Email ingestion**: Gmail API [17].
- **Datasets**: PhishTank, OpenPhish, Enron email corpus.

4.5 Storage

- PostgreSQL/SQLite for metadata.
- MinIO / AWS S3 for logs and dataset storage.

4.6 Deployment

- Docker for containerization.
- Kubernetes/Google Cloud run for scalability.
- GitHub Actions for CI/CD automation [16].

4.7 Security & Compliance

- Requests tokenized before transmission.

- Only embeddings/features sent to Gemini API, preserving privacy [9].
- TLS/HTTPS enforced across all calls.

5. Gap Analysis

While prior systems achieve strong accuracy, most lack three critical aspects:

1. **Real-time browser-level integration**.
2. **Explainability for end-users**.
3. **Hybrid analysis (linguistic + structural)**.

The proposed system addresses these gaps by combining NLP and URL-based detection in a **multi-agent ensemble** integrated into a **Chrome/browser extension**, with **explainable justifications** for phishing flags.

6. Summary

The reviewed literature shows a clear trend from static blacklist methods toward hybrid, explainable AI approaches. By leveraging Gemini, scikit-learn, and browser integration, the proposed system builds on this trajectory to offer real-time phishing detection that is accurate, explainable, and user-friendly.

References

[1] Verizon, *2023 Data Breach Investigations Report*. Verizon Enterprise, 2023.
[2] T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94–100, 2007.
[3] M. Basit, M. Zafar, and H. Malik, "Phishing detection using ensemble learning," *IEEE Access*, vol. 9, pp. 147497–147511, 2021.
[4] A. Herzberg and A. Gbara, "Protecting naive web users from phishing," *ACM Computing Research Repository (CoRR)*, 2004.
[5] I. Almomani, B. Gupta, and S. Atawneh, "Phishing detection based on machine learning and feature selection," *Security and Communication Networks*, vol. 2019, pp. 1–12, 2019.

[6] R. Mohammad, F. Thabtah, and L. McCluskey, “An assessment of features related to phishing websites using an automated technique,” in *Proceedings of the International Conference on Internet Technology and Secured Transactions*, 2012.
[7] PhishTank, “PhishTank data feed,” 2023. [Online]. Available: https://www.phishtank.com
[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT 2019*.
[9] Google, “Gemini API documentation,” 2024. [Online]. Available: https://ai.google.dev
[10] H. Xiao, H. Xu, and S. Xu, “A novel ensemble learning approach for phishing detection,” *Computers & Security*, vol. 114, pp. 102–122, 2022.
[11] S. Raji, Y. Jiang, and R. Xu, “Explainable AI for phishing detection: A survey,” *ACM Computing Surveys*, vol. 55, no. 7, pp. 1–37, 2023.
[12] Google, “Safe Browsing: Protecting users from phishing,” 2023. [Online]. Available: https://safebrowsing.google.com
[13] A. Jain and B. Gupta, “Phishing detection in Chrome extensions: A comparative study,” *Future Generation Computer Systems*, vol. 125, pp. 456–468, 2021.
[14] W. Cohen, “Enron email dataset,” Carnegie Mellon University, 2009.
[15] Tailwind Labs, “TailwindCSS documentation,” 2023. [Online]. Available: https://tailwindcss.com
[16] FastAPI, “FastAPI framework documentation,” 2023. [Online]. Available: https://fastapi.tiangolo.com
[17] Google, “Gmail API overview,” 2023. [Online]. Available: https://developers.google.com/gmail/api