Grenze International Journal of Engineering and Technology, June Issue

PhishNet: Smart Website Security with Conversational AI

Anvit Magadum[1], Avishkar Aher[2], Anirudh Abhisheki[3], Atharva Deshmukh[4], Nilesh Shewale[5] and Priyanka Kadam[6]
[1-6]Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, India
Email: anvit.magadum24@vit.edu, avishkar.aher24@vit.edu, anirudh.abhisheki24@vit.edu, atharva.deshmukh24@vit.edu, nilesh.22310141@vit.edu, priyanka.kadam@vit.edu

***Abstract*— Phishing remains a significant cybersecurity threat, enabling adversaries to impersonate legitimate digital platforms and compromise sensitive user information. Traditional blacklist-based detection mechanisms are inherently reactive and often fail to identify newly generated phishing domains in real time. This study presents PhishNet, a hybrid phishing detection framework integrating blacklist intelligence, heuristic URL analysis, and conversational AI within a browser-based architecture. The proposed system employs structural and lexical feature evaluation combined with external threat verification to classify URLs as safe, suspicious, or malicious. Experimental evaluation on a dataset of 6,000 labeled URLs achieved 97.8% accuracy, 97.1% precision, and 98.2% recall, with an average latency of 265 ms. The results demonstrate that the hybrid approach effectively balances detection accuracy, computational efficiency, and user interpretability.**

***Index Terms*— Phishing detection, hybrid security framework, heuristic URL analysis, blacklist verification, conversational AI, browser security.**

I. INTRODUCTION

Phishing attacks have evolved into sophisticated cyber threats that exploit both human trust and technical vulnerabilities. Modern campaigns employ techniques such as domain spoofing, homograph substitution, and HTTPS-based manipulation to bypass conventional security mechanisms. Although modern browsers rely on blacklist-based systems to detect malicious domains, these approaches remain reactive and provide limited protection against zero-day attacks. The rapid proliferation of dynamically generated phishing domains necessitates proactive detection mechanisms capable of identifying structural anomalies in real time. Furthermore, traditional browser warnings often lack contextual clarity, thereby reducing user comprehension and trust. To address these challenges, this study proposes PhishNet, a hybrid phishing detection framework designed for real-time browser environments.

II. CONTRIBUTIONS

The main contributions of this study are summarized as follows:

- Hybrid Detection Framework: A real-time phishing detection architecture integrating blacklist intelligence with URL-based anomaly analysis to enhance robustness against zero-day attacks.

Grenze ID: 01.GIJET.12.2.240_24

- Weighted Risk Scoring Model: Development of a risk-scoring mechanism that evaluates lexical and structural URL features for accurate classification.
- Explainable Conversational AI Integration: Incorporation of a conversational AI module that provides interpretable threat explanations, thereby enhancing user awareness and trust.
- Low-Latency Browser Deployment: Implementation of a lightweight Chrome browser extension capable of real-time execution with minimal computational overhead.
- Comprehensive Experimental Evaluation: Validation using a balanced dataset of 6,000 URLs with confusion matrix analysis and performance metrics.

III. Literature Review

Safi and Langarib [1] reviewed phishing detection techniques and highlighted the limitations of relying exclusively on blacklist-based methods. Their study emphasized the necessity of hybrid detection models that integrate structural feature analysis with external threat intelligence sources.

Alshingiti et al. [2] demonstrated the effectiveness of deep learning architectures, particularly CNN and LSTM models, for phishing detection. Although their approach achieved high predictive accuracy, it required significant computational resources, limiting its suitability for lightweight deployment environments.

Atawneh [3] proposed deep learning techniques for phishing email detection and improved classification performance through sequence modelling. However, the study did not extensively evaluate real-time deployment constraints.

Ozcan [4] presented a hybrid DNN–LSTM model for classifying phishing URLs and observed increased inference complexity associated with the integrated architecture.

Guptta et al. [5] proposed a hybrid phishing detection approach combining lexical and hyperlink-based features, demonstrating improved detection performance compared to traditional classifiers.

Said et al. [6] enhanced phishing detection performance using CNN architectures with self-attention mechanisms. While this approach improved contextual feature representation, it also increased computational overhead.

Senouci [7] explored phishing detection within cloud-based environments using RNN–LSTM architectures, emphasizing scalability and distributed processing capabilities.

Choudhary [8] evaluated traditional machine learning methods, including Support Vector Machines and Random Forest classifiers, for phishing detection tasks.

Jayaraj et al. [9] investigated the integration of phishing detection mechanisms into intrusion detection systems to strengthen comprehensive security architectures.

Tang and Mahmoud [10] conducted a comprehensive survey of machine learning techniques for phishing detection and identified persistent challenges such as feature selection complexity, dataset imbalance, interpretability limitations, and real-time deployment constraints.

Although deep learning approaches achieve strong predictive accuracy, their computational overhead often restricts browser-level applicability. Lightweight hybrid frameworks that balance detection performance with real-time deployability remain comparatively underexplored. The proposed PhishNet framework addresses this gap by integrating heuristic risk scoring, blacklist intelligence, and explainable conversational mechanisms within a browser-based real-time environment.

IV. Methodology

A. System Architecture

The proposed PhishNet framework adopts a hybrid browser-based detection architecture consisting of the following components:

- Chrome browser extension (frontend interface)
- Flask-based backend server
- Blacklist verification module
- Heuristic risk scoring engine
- Conversational AI explanation module

The browser extension intercepts URL requests during browsing sessions and forwards them to the backend server for evaluation. Upon receiving a URL, the backend performs parallel verification using external threat intelligence services, including Google Safe Browsing and PhishTank APIs. If the URL is identified as malicious, it is immediately classified as phishing.

In cases where blacklist verification does not detect malicious activity, heuristic feature extraction is performed. The system evaluates structural and lexical URL attributes, including domain length (L), subdomain depth (S), special character ratio (C), HTTPS presence indicator (H), and domain similarity index (D).
Fig. 1 illustrates the proposed architecture of the PhishNet framework.

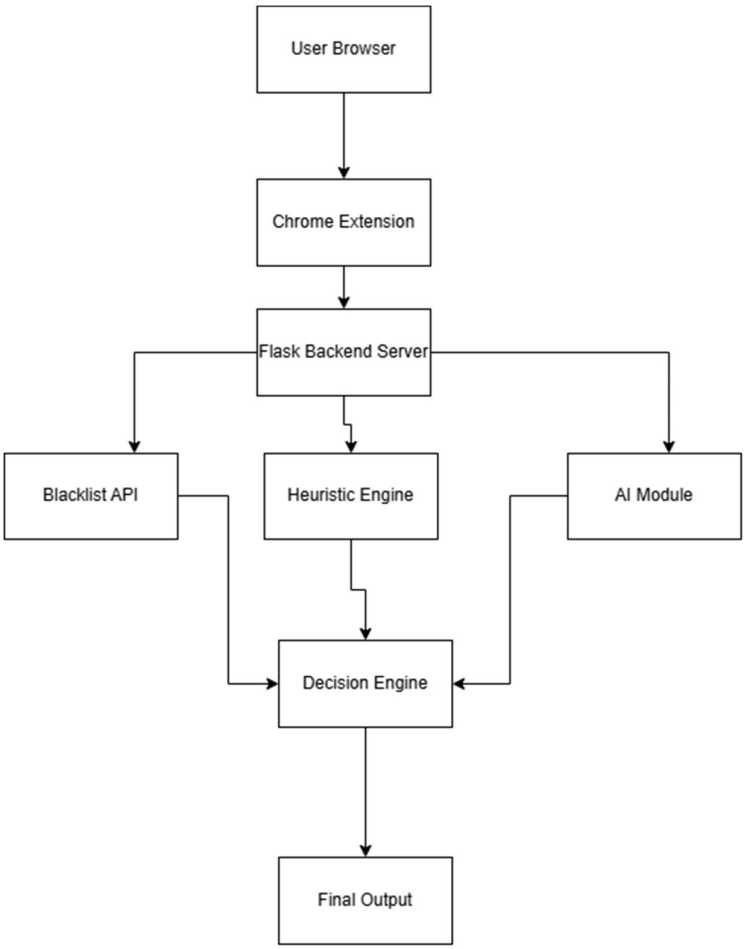

Fig. 1. illustrates the system architecture of the proposed system.

B. Risk Scoring Model

The overall risk score is computed as:

$$\text{RiskScore} = w_1L+w_2S+w_3C+w_4(1-H)+w_5D \qquad (1)$$

Where wi represents empirically determined feature weights. The computed score is compared against predefined classification thresholds to categorize URLs as safe, suspicious, or phishing.
The score computed in (1) is used to classify URLs as safe, suspicious, or phishing.

C. Detection Workflow

The detection process follows these steps:

- URL capture through browser extension.
- Blacklist verification using threat intelligence APIs.
- Heuristic feature extraction.
- Risk score computation.
- URL classification.
- Conversational AI-based explanation generation.

The detection procedure is summarized in Fig. 2.

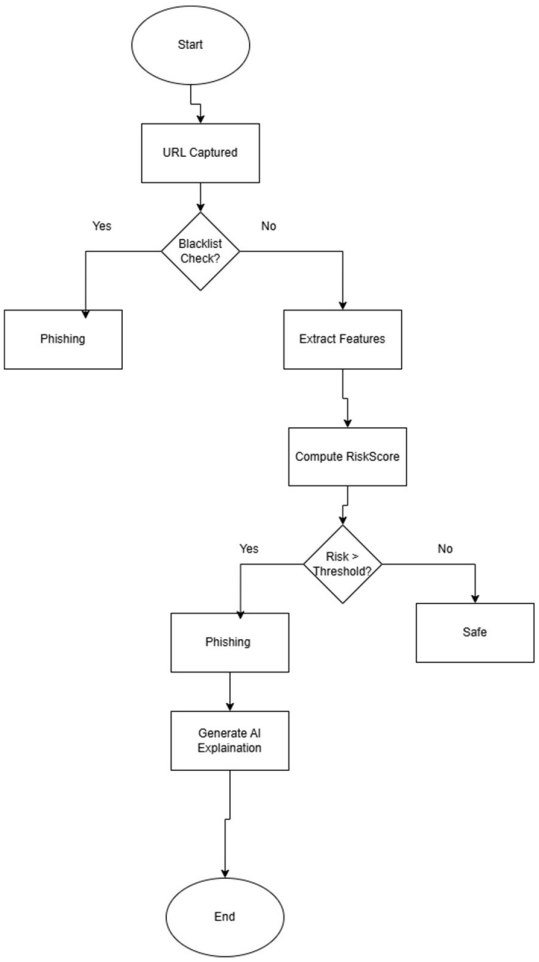

Fig. 2. Workflow of the real-time phishing URL detection and classification process.

V. Experimental Setup

A. Dataset Description

The experimental evaluation was conducted using a balanced dataset comprising 6,000 labelled URLs, including 3,000 verified phishing URLs and 3,000 legitimate URLs. Phishing samples were collected from publicly available threat intelligence repositories such as PhishTank, while legitimate URLs were obtained from trusted domain listings.

Prior to experimentation, duplicate entries, inactive links, and malformed URLs were removed to ensure dataset integrity. The dataset was randomly shuffled to minimize bias.

B. Experimental Environment

All experiments were performed in a controlled environment using a Chrome browser extension integrated with a Flask-based backend server. Stable network conditions were maintained to ensure consistent latency measurements.

C. Performance Evaluation Metrics

System performance was evaluated using standard classification metrics including Accuracy, Precision, Recall, and F1-score. A confusion matrix was constructed to analyze classification outcomes in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

System performance was evaluated using the following metrics:

The evaluation metrics are calculated using (2)–(5).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)} \quad (5)$$

VI. Results and Discussion

A. Confusion Matrix Analysis

Table I presents the confusion matrix obtained during experimental evaluation.

TABLE I: CONFUSION MATRIX ANALYSIS

	Predicted Phishing	Predicted Legitimate
Actual Phishing	2946 (TP)	54 (FN)
Actual Legitimate	72 (FP)	2928 (TN)

B. Performance Evaluation

Based on the confusion matrix, the system achieved an accuracy of 97.8%, precision of 97.1%, recall of 98.2%, and F1-score of 97.6%. The false-positive rate was 2.4%, and the false-negative rate was 1.8%. The average detection latency was measured at 265 ms.

C. Discussion

The high recall value demonstrates strong capability in identifying phishing URLs, which is critical in cybersecurity applications. The hybrid model reduces false negatives compared to blacklist-only approaches while maintaining acceptable latency for real-time use.

The integration of heuristic analysis enables detection of previously unseen phishing URLs, improving system robustness. Additionally, the conversational AI component enhances user interpretability and trust.

VII. Future Scope

Although the proposed PhishNet framework demonstrates strong detection performance and real-time feasibility, several avenues for further enhancement remain. Future work may explore cross-browser compatibility and mobile platform deployment to broaden accessibility and practical applicability. Incorporating adaptive machine learning mechanisms for dynamic weight optimization could further improve detection robustness as phishing strategies continue to evolve.

Additional investigation into adversarial resilience, including homograph manipulation and domain obfuscation attacks, may strengthen system robustness against emerging threat patterns. Expanding the evaluation dataset to include multilingual and region-specific phishing domains could improve generalization capability across diverse operational contexts. Furthermore, integrating behavioural and contextual webpage analysis may enhance detection precision while preserving real-time performance constraints.

VIII. Conclusion

This study presents PhishNet, a hybrid phishing detection framework integrating blacklist intelligence, heuristic risk modelling, and conversational AI within a browser-based architecture. Experimental validation confirms high detection accuracy, low false-negative rates, and minimal latency, demonstrating practical viability for real-time deployment.

By combining structural anomaly detection with external threat verification and user-centred interpretability, the framework addresses both technical and usability limitations of conventional blacklist-only approaches. The results indicate that lightweight hybrid architectures can achieve a balanced trade-off between detection reliability, computational efficiency, and scalability.

Overall, the proposed system contributes toward enhancing proactive phishing mitigation strategies suitable for modern browser environments.

REFERENCES

[1] A. Safi and N. Langarib, "Phishing detection techniques: A review," Data Science: Journal of Computing and Applied Informatics, vol. 9, no. 1, pp. 32–46, 2025.

[2] Z. Alshingiti, A. Alassaf, and M. H. Al-Zubaidi, "A deep learning-based phishing detection system using LSTM, CNN and hybrid models," Electronics, vol. 12, no. 1, pp. 232, 2023.

[3] S. Atawneh, "Phishing email detection model using deep learning," Electronics, vol. 12, no. 20, pp. 4261, 2023.

[4] A. Ozcan, "A hybrid DNN–LSTM model for detecting phishing URLs," Neural Computing and Applications, 2023, doi: 10.1007/s00521-021-06401-z.

[5] S. D. Guptta et al., "Modeling hybrid feature-based phishing websites detection," Applied Sciences, vol. 12, no. 9, pp. 4123, 2022.

[6] Y. Said et al., "Detecting phishing websites through improved CNN with self-attention mechanism," Journal of Information Security and Applications, vol. 80, 2024.

[7] O. Senouci, "Enhancing phishing detection in cloud environments using RNN-LSTM models," Journal of Telecommunications and Information Technology, 2025.

[8] T. Choudhary, "A machine learning approach for phishing attack detection," Journal of Artificial Intelligence and Technology, 2023.

[9] R. Jayaraj et al., "Intrusion detection based on phishing detection with machine learning," Measurement: Sensors,vol.31,2024,doi:10.1016/j.measen.2023.101003.

[10] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," Machine Learning and Knowledge Extraction, vol. 3, no. 3, pp. 672–694, 2021.