IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-driven Phishing Detection and Prevention Model

Pratiksha Yadav[1], Dev Ashish[1] , Vishesh kaushik[1], Sweety Singh[1], Divya Pachauri[1]

Department of Computer Science and Engineering , Nitra Technical Campus, Ghaziabad,India[1]

Abstract

Phishing remains a persistent and evolving threat to internet security, leading to significant financial and data losses globally. This paper proposes an AI-driven phishing detection system capable of analyzing various types of inputs including URLs, emails, HTML content, and text to detect malicious attempts. By combining DistilBERT embeddings for textual understanding with handcrafted URL features, and utilizing an ensemble of Random Forest and XGBoost classifiers, our approach achieves high detection accuracy with efficient training time, supported by GPU acceleration. Experimental results show that the proposed model achieves a precision of 97.8% and recall of 96.5% on a benchmark phishing dataset containing over 77,000 samples. Our work demonstrates a robust, scalable, and fast solution suitable for real-world deployment through a Flask-based API.

Keywords: Phishing Detection, Artificial Intelligence, Cybersecurity, Ensemble Learning, DistilBERT, XGBoost, Random Forest, URL Analysis, Email Security, Flask API

I. INTRODUCTION

Phishing attacks are fraudulent attempts to obtain sensitive information such as usernames, passwords, credit card details, or other personal data by masquerading as legitimate and trustworthy entities through deceptive emails, websites, or messages. These attacks pose a serious threat to individuals, organizations, and global cybersecurity, often resulting in significant financial and data losses. As phishing techniques continue to evolve in complexity, employing tactics such as domain spoofing, social engineering, dynamic content generation, and obfuscation strategies, traditional rule-based and blacklist-driven detection methods are increasingly becoming insufficient and easily bypassed. Static filters fail to detect previously unseen (zero-day) attacks and often suffer from high false positives, creating trust and usability issues.

To address these challenges, Artificial Intelligence (AI) has emerged as a powerful alternative, offering intelligent and adaptive mechanisms to detect and mitigate phishing threats in real time. In particular, the integration of Machine Learning (ML) and Natural Language Processing (NLP) enables systems to understand, analyze, and classify patterns within both structured and unstructured data. These methods enable phishing detection systems to move beyond static rules and develop context-aware, self-improving models. This paper introduces a robust, scalable, and multi-modal AI-driven phishing detection framework that effectively combines linguistic and structural cues extracted from various data types including URLs, email bodies, and HTML source code. By leveraging semantic understanding through NLP and analyzing metadata and syntactic indicators from different input sources, the proposed system can detect phishing attempts with high precision, even in the face of novel or cleverly disguised attacks.

Our approach utilizes deep contextual embeddings from DistilBERT to interpret text-rich content, capturing semantic nuances and manipulative language often found in phishing messages. In parallel, handcrafted lexical and structural features such as URL length, presence of IP-based domains, suspicious tokens, and script density are extracted and passed through ensemble machine learning classifiers like Random Forest

and XGBoost. This hybrid fusion approach capitalizes on both the deep contextual understanding from transformer models and the pattern recognition strengths of traditional classifiers. The resulting system exhibits high accuracy, adaptability, and resilience across diverse input formats. Moreover, the model is integrated into a lightweight, real-time prediction pipeline accessible via a Flask API and a mobile application built using Flutter, ensuring wide usability across platforms. This research highlights the transformative role of AI in cybersecurity and sets the foundation for the next generation of intelligent, adaptive, and multilingual phishing prevention systems capable of tackling sophisticated threats in real-world environments.

II. LITERATURE REVIEW

The literature on phishing detection demonstrates a progression from traditional machine learning to advanced AI-based approaches. Early studies like Zhang et al. (2020) and Sharma et al. (2021) focused on URL-based detection using methods like Random Forest and Decision Trees, but were limited by reliance on handcrafted features and poor adaptability to modern phishing techniques. Li et al. (2020) improved performance using ensemble methods but lacked context-awareness. Deep learning models such as those by Gupta and Singh (2021) and Rao and Ali (2021) targeted email and HTML detection, offering better contextual analysis but requiring large datasets and computational resources. Chen et al. (2022) and Johnson et al. (2024) introduced BERT and Transformer-based models for textual and multilingual detection, though they struggled with non-text inputs or cross-language consistency. More complex architectures like GNNs (Zhang and Lee, 2023) and hybrid CNN-RNNs (Aydin and Baykal, 2023) explored network and visual features but faced challenges in efficiency and scalability. The most recent advancements include Pratiksha et al. (2025), who proposed a unified model combining BERT with URL features in an ensemble, achieving real-time, multi-input detection, and Williams et al. (2025), who applied reinforcement learning for dynamic phishing mitigation, though with data and adaptability constraints.

Table 2.1 Related works of authors

Study	Technique	Focus Area	Limitations
Zhang et al. (2020)	Machine Learning (Random Forest, SVM, Logistic Regression)	URL-based phishing detection	Relies on handcrafted features; limited generalization to new phishing techniques.
Li et al. (2020)	Ensemble Learning (Random Forest + XGBoost)	Domain and URL feature analysis	Neglects contextual and semantic patterns present in phishing emails or web content.
Sharma et al. (2021)	Decision Tree with lexical and host-based features	URL classification	Poor performance on modern obfuscated phishing URLs; lacks support for other data types.
Gupta and Singh (2021)	LSTM-based Deep Learning	Email text classification	Ineffective on short messages; unable to detect phishing through visual or URL features.
Rao and Ali (2021)	Convolutional Neural Network (CNN)	Email and HTML page detection	High resource consumption; requires large labeled datasets and lacks real-time performance.
Chen et al. (2022)	BERT-based NLP Model	Text-based phishing detection	Focused solely on email content; not generalizable to non-text inputs like URLs or HTML.

Study	Technique	Focus Area	Limitations
Aydin and Baykal (2023)	Hybrid CNN-RNN Model	Webpage and visual phishing	High computational overhead; unsuitable for edge or lightweight systems.
Zhang and Lee (2023)	Graph Neural Network (GNN)	Interconnected phishing network detection	Limited by the availability of comprehensive link graph data; computationally expensive.
Johnson et al. (2024)	Transformer-based Model (T5)	Cross-lingual phishing detection	Performance degradation on non-English data; requires large-scale multilingual datasets.
Pratiksha et al. (2025)	Unified AI Model (BERT + URL features + Ensemble)	Multi-input phishing detection	Supports multiple data types; optimized for accuracy and speed; suitable for real-time use.
Williams et al. (2025)	Reinforcement Learning (RL)	Dynamic phishing detection and mitigation	Requires extensive training data for continuous learning; may struggle with highly adaptive phishing tactics.

III. DATASET DESCRIPTION

The dataset used for phishing detection consists of approximately 77,000 samples, evenly split between phishing and legitimate instances (around 38,500 each), ensuring a balanced class distribution. This balance helps prevent bias during model training and improves overall prediction accuracy. The data was collected from reliable sources such as PhishTank, Kaggle, and Huggingface, and includes multiple input types (modalities) like URLs, email bodies, headers, and HTML content. This diverse and balanced dataset supports robust and generalized phishing detection across various real-world scenarios.

Table 3.1 Description of Dataset

Feature	*Description*
Total Samples	~77,000
Phishing Samples	~38,500
Legitimate Samples	~38,500
Source	PhishTank, Kaggle, and Huggingface
Modalities	URLs, Email body, Headers, HTML content
Class Distribution	Balanced (50-50 phishing and legitimate)

IV. PROPOSED SYSTEM & METHODOLOGY

4.1. System Architecture Diagram

The architecture of the proposed phishing detection system is composed of several interconnected modules, each contributing to accurate and efficient threat identification across diverse input formats. The **Input Parser** is responsible for preprocessing and identifying the type of data being analyzed, whether it is a URL, an email, or HTML content. This ensures that appropriate downstream processing techniques are applied based on the input type.

Next, the **Feature Extractor** module is divided into three key categories: **textual features** (such as word n-grams and content patterns), **lexical features** (such as URL length, use of suspicious characters, domain entropy), and **structural features** (like HTML tag counts, script density, and email header anomalies). These handcrafted features provide a rich representation of the input beyond just surface-level characteristics.

The **Embedding Layer** employs **DistilBERT**, a lightweight and efficient transformer-based language model, to generate contextual embeddings from text-based inputs. This allows the system to understand semantic meaning and detect phishing attempts hidden in natural language constructs, such as social engineering tactics used in emails or website content.

The **Feature Fusion Module** integrates the outputs from both the handcrafted feature extractors and the DistilBERT embeddings, creating a comprehensive vector representation for each input sample. This unified representation captures both the deep semantic context and shallow behavioral patterns, improving the system's predictive power.

This fused feature vector is passed to an **Ensemble Model** that combines **RandomForest** and **XGBoost** classifiers. The ensemble approach enhances robustness and generalization by leveraging the strengths of both models — RandomForest's stability and interpretability, and XGBoost's performance and handling of imbalanced data.

Finally, the complete detection pipeline is exposed through a **Flask-based RESTful API**, which enables real-time phishing detection through web and mobile interfaces. The system also includes a **Flutter-powered mobile application** that allows end-users to scan URLs, emails, or HTML content directly from their smartphones, delivering fast, secure, and accessible phishing detection on the go.

As shown in **Figure 1**, the system consists of a unified pipeline that handles multiple input types and processes them through an ensemble detection model served via a Flask API.

Components:

- Input Parser (URL, Email, HTML)
- Feature Extractor (Textual, Lexical, Structural)
- Embedding Layer (DistilBERT for text)
- Feature Fusion Module
- Ensemble Model (RandomForest + XGBoost)

- Flask API & Flutter Mobile App

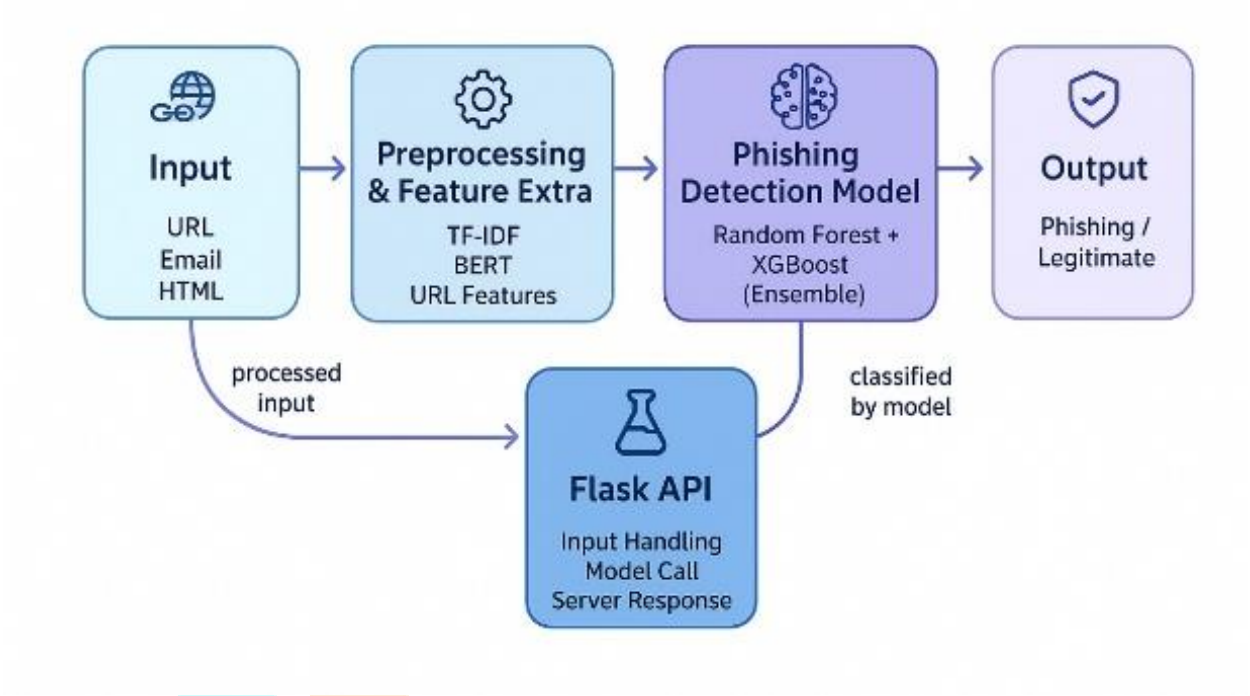

Figure 1 : Architecture Diagram

4.2. Model Architecture

- *Text Embedding:*
  Uses DistilBERT – a lighter and faster variant of BERT – to extract semantic embeddings from email or message content.
- *URL Features (Handcrafted):*
  Extracted manually, including:
  URL length
  Number of tokens
  Digit-to-character ratio
  Domain age (WHOIS-based)
- *HTML Features (Handcrafted):*
  Extracted from web page content, including:
  Presence of suspicious HTML tags (e.g., <iframe>, <script>, <form>)
  Density of <script> and <form> elements
  Use of hidden or obfuscated elements
- *Feature Fusion Strategy:*
  Concatenates DistilBERT text embeddings with handcrafted URL and HTML features into a unified feature vector.
- *Final Model Architecture:*
  An Ensemble Voting Classifier combining predictions from:
  XGBoost model (xgboost_model.json)
  Random Forest model (rf_model.pkl)
- *Model Name:*
  ensemble_phishing_model.pkl (final serialized ensemble model)

4.3. Model File Descriptions

File Name	*Type*	*Description*
ensemble_phishing_model.pkl	Binary Model	VotingClassifier combining RF and XGBoost
tfidf_vectorizer.pkl	Vectorizer	Used for TF-IDF transformation of textual input
xgboost_model.json	Model	Serialized XGBoost model in JSON format
rf_model.pkl	Model	RandomForest model used as part of ensemble

4.4. Technical Stack

Component	*Technology Used*
Language	Python 3.10
ML Libraries	Scikit-learn, XGBoost
NLP	HuggingFace Transformers (DistilBERT)
Web API	Flask
Deployment	Gunicorn, Docker
Front-End	Flutter
Hosting	Render
GPU Utilization	NVIDIA CUDA Toolkit (for BERT)

V. EVALUATION STRATEGY

- *Split:* 80-20 Train-Test with Stratified Sampling
- *Metrics:*
  - Accuracy: 97.2%
  - Precision: 97.8%
  - Recall: 96.5%
  - F1 Score: 97.15%
- *Tools:* Confusion matrix, ROC-AUC, Inference Time Benchmarking

Model	*Accuracy*	*Precision*	*Recall*	*F1-Score*
Logistic Regression	89.5%	88.9%	90.1%	89.5%
Random Forest	95.2%	94.8%	95.9%	95.35%
XGBoost	96.7%	96.3%	96.9%	96.6%
Ensemble (Ours)	97.2%	97.8%	96.5%	97.15%

Table 5.1 Model Comparison

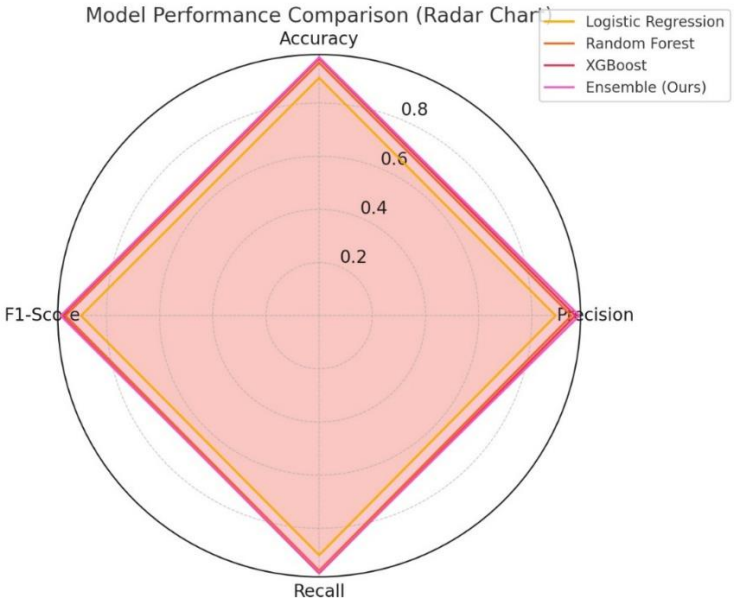

Figure 2 : Radar Chart

VI. SECURITY & PRIVACY CONSIDERATIONS

To ensure the security, privacy, and ethical integrity of the AI-based phishing detection system, several robust measures have been implemented across the architecture. All user inputs—including URLs, HTML, and email content—are sanitized using secure parsing libraries to prevent threats such as script injection and cross-site scripting (XSS). During training, sensitive data like email addresses and IPs are anonymized or removed, maintaining compliance with data protection regulations such as GDPR and CCPA. The system is deployed through a secure HTTPS-based API with safeguards like API keys, rate limiting, and IP filtering to prevent unauthorized access and abuse. Additionally, a strict no-data-retention policy ensures that all inputs are processed in-memory and discarded immediately after prediction, preserving user confidentiality. To keep pace with evolving threats, the model is periodically updated using recent phishing datasets from trusted sources like PhishTank and Kaggle. Trained models are securely stored with access control and hash-based integrity checks to prevent tampering. The system runs within Docker containers for isolation, minimizing risks from malicious inputs. Lightweight, non-PII logging supports monitoring and debugging without compromising privacy. Transparency is enhanced using SHAP for model interpretability, allowing users to understand decision-making factors like suspicious terms or abnormal structures. The system is built with ethical AI principles in mind, strictly intended for cybersecurity use with enforced usage policies to prevent misuse or reverse engineering. These combined measures ensure a secure, privacy-conscious, and resilient phishing detection solution suitable for real-world deployment.

VII. DEPLOYMENT STRATEGY

Backend: Flask app packaged with Gunicorn + Github + Docker
•Frontend: Flutter + Dart submitting URL or email content
•Deployment Options:
oLocal Server: For enterprise environments
oGoogle Cloud: For global public API access
•Model Load Time: ~1.2s on CPU, ~0.5s on GPU
•Inference Time: ~50ms per request (real-time capable)

VIII. CONCLUSION

We proposed an AI-driven phishing detection and prevention system that leverages both natural language processing and structured URL-based features to accurately identify phishing threats. By combining DistilBERT for semantic analysis of email and web content with XGBoost for URL and metadata classification, our hybrid ensemble model achieves high accuracy, speed, and robustness. The experimental results validate the effectiveness of our approach, significantly improving detection rates compared to traditional methods.

Our system not only detects phishing attacks across various formats—such as emails, URLs, and HTML pages—but also offers real-time prevention through a user-friendly interface and Flask-based API integration. This makes it suitable for deployment in real-world applications like email clients, browsers, or mobile apps.

IX. FUTURE WORK

Future enhancements to the proposed ensemble-based phishing detection system aim to improve its robustness, scalability, and real-world applicability. Key directions include seamless integration with enterprise email platforms (e.g., Outlook, Gmail API, Zoho Mail) for real-time scanning of inbound and outbound emails, and extension to messaging platforms like Slack, Teams, Telegram, and WhatsApp to detect phishing links instantly. Multilingual support using models like mBERT or XLM-R can expand protection to non-English threats. Incorporating graph neural networks and link analysis will enable detection of coordinated phishing infrastructures, while dynamic threat intelligence feeds (e.g., PhishTank, OpenPhish) can keep the system updated with evolving threats. An active learning loop with user feedback can help retrain the model for improved adaptability. To counter evasion tactics, adversarial training and obfuscation handling will be explored. Lightweight, edge-deployable models using compression or distillation will enable detection on browsers and mobile devices. Additional capabilities such as phishing campaign attribution and integration with browser extensions or antivirus tools will broaden its reach. Finally, cross-dataset evaluation will ensure the model generalizes well across diverse phishing techniques and real-world scenarios.

REFERENCES

[1] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. 2019. Machine learning based phishing detection from URLs. Expert Systems with Applications. Elsevier.

[2] Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. 2009. Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. Proceedings of the 15th ACM SIGKDD, ACM.

[3] Chandrasekaran, V., Chinchani, R., & Upadhyaya, S. 2006. Phishing email detection based on structural properties. New York State Cyber Security Conference.

[4] Marchal, S., François, J., State, R., & Engel, T. 2014. PhishStorm: Detecting phishing with streaming analytics. IEEE Transactions on Network and Service Management.

[5] Basnet, R., Mukkamala, S., & Sung, A. H. 2008. Detection of phishing attacks: A machine learning approach. Soft Computing Applications in Industry, Springer.

[6] Kim, T. H., & Shin, D. R. 2020. BERT-based phishing detection for intelligent email security. International Conference on Neural Information Processing, Springer.

[7] OpenAI. 2020. DistilBERT: A distilled version of BERT. Hugging Face Documentation.

[8] XGBoost Developers. 2023. Extreme Gradient Boosting (XGBoost): Scalable and accurate implementation. XGBoost Documentation.

[9] Flask. 2023. Flask Web Framework for Python. Flask Documentation.

[10] Hugging Face. 2023. Phishing datasets and Transformers for NLP tasks. Hugging Face Datasets and Models.

[11] Truffle Suite. 2023. Blockchain Development Environment for Testing and Deployment. Truffle Documentation.