ISSN: 2349-6002

Hybrid Machine Learning and Deep Learning Framework for Accurate Phishing Website Detection with AI Model Integration

Mr. T. Kamalesh[1], Arunkumar RS[2], Dinakaran K[3], Mohanraj R[4]
[1]*Assistant Professor, Dept. of AI&DS, School of Engineering & Technology, Surya Group of Institutions, Vikravandi, Villupuram*
[2,3,4]*UG, Dept. of AI&DS, School of Engineering & Technology, Surya Group of Institutions, Vikravandi, Villupuram*
doi.org/10.64643/IJIRTV12I11-200935-459

***Abstract*—Phishing attacks represent one of the most pervasive and financially devastating cyber threats in the modern digital landscape, compromising millions of users annually by employing deceptive websites engineered to mimic legitimate online services. Conventional detection mechanisms, including static blacklists and heuristic rule-based filters, exhibit critical limitations in identifying zero-day phishing campaigns and dynamically evolving attack vectors. This paper presents a novel Hybrid Machine Learning and Deep Learning Framework integrated with a Large Language Model (LLM)-powered AI Explanation Module, designed to deliver accurate, real-time phishing website classification with contextual user awareness. The proposed system processes user-submitted URLs through a multi-stage analytical pipeline encompassing URL preprocessing, WHOIS domain lookup, QR code decoding, and comprehensive feature engineering across more than one hundred lexical, domain-centric, and content-based attributes. Classification is performed using an ensemble of Random Forest, Support Vector Machine (SVM), and Decision Tree algorithms, augmented by a neural network layer for enhanced generalization. Experimental evaluation conducted on a labeled dataset of approximately 90,000 URLs demonstrates that the Random Forest classifier achieves peak performance with 96% accuracy, 94% precision, 95% recall, and an F1-score of 95%, alongside an AUC of 0.99. The system further incorporates an AI-driven explanation module that elucidates detection rationale, identifies specific risk factors, and provides actionable cybersecurity guidance to end users. The web-based deployment ensures cross-platform accessibility on both mobile and desktop environments, positioning the framework as a practical, scalable solution for real-time phishing defense.**

***Index Terms*—Phishing detection, machine learning, deep learning, Random Forest, feature engineering, URL analysis, cybersecurity, AI explanation, neural network, WHOIS lookup.**

I. INTRODUCTION

The proliferation of internet-based services has been accompanied by a commensurate escalation in cyber-criminal activity, with phishing consistently ranking among the most prevalent and economically destructive attack modalities. Phishing operations involve the construction of counterfeit websites that impersonate legitimate institutions—including financial organizations, e- commerce platforms, and government portals—with the explicit intent of harvesting sensitive user credentials, payment information, and personally identifiable data. According to the Anti-Phishing Working Group (APWG), phishing incidents have more than doubled between 2019 and 2023, underscoring the urgency of developing robust, automated detection solutions.

Contemporary phishing detection approaches have evolved through several generations: early systems relied on static URL blacklists maintained by security vendors; subsequent methodologies introduced heuristic rule engines capable of flagging suspicious URL patterns; and the latest generation leverages machine learning algorithms to derive statistical models from large, labeled datasets.

However, each of these paradigms exhibits documented shortcomings. Blacklist-based systems are fundamentally reactive, offering no protection against novel, unreported phishing domains. Rule-

based engines are susceptible to adversarial evasion through subtle syntactic modifications. Even many machine learning implementations are constrained by limited feature sets, insufficient training data, or architectural choices that impede generalization to previously unseen attack patterns.

This paper addresses these limitations through the design and implementation of a comprehensive phishing detection framework that synergizes classical supervised machine learning with deep neural network architectures and integrates an LLM-powered explanation system. The principal contributions of this work are as follows:

1) A hybrid ML/DL classification engine trained on 90,000+ labeled URLs with 100+ engineered features, achieving 96% classification accuracy.
2) An integrated preprocessing pipeline incorporating URL expansion, QR code decoding, and WHOIS domain intelligence.
3) An AI-driven result explanation module that communicates risk factors in natural language and provides protective guidance to users.
4) A cross-platform, real-time web application deployed for desktop and mobile use, enabling broad accessibility.

The remainder of this paper is organized as follows: Section II reviews related literature; Section III defines the problem statement; Section IV presents the proposed system overview; Section V details the methodology; Sections VI and VII describe the system architecture and modules; Sections VIII–X discuss experimental setup, results, and evaluation; Sections XI–XIII address advantages, limitations, and future directions; and Section XIV concludes the paper.

II. LITERATURE REVIEW

The academic study of phishing detection through computational methods spans two decades, with each era introducing progressively sophisticated tools. This section surveys the most influential contributions and identifies the gaps addressed by the proposed system.

A. Blacklist and Rule-Based Approaches

Early phishing detection systems relied on static blacklists of known malicious URLs, maintained by organizations such as Google Safe Browsing and Phish Tank. While computationally efficient, such systems are inherently reactive and incapable of identifying zero-day phishing sites. Rule-based heuristics improved upon blacklists by encoding domain knowledge—for example, flagging URLs containing IP addresses, excessive hyphens, or specific suspicious keywords. However, these rules are brittle against adaptive adversaries who can trivially modify URL structures to circumvent detection.

B. Machine Learning-Based Detection

The introduction of machine learning to phishing detection represented a paradigm shift. Kiruthiga and Akila [10] demonstrated that supervised classifiers trained on lexical and domain features could achieve high detection rates. Kulkarni and Brown [9] evaluated multiple ML algorithms and highlighted the superiority of ensemble methods. The foundational work by Kanagalakshmi et al. [1], which forms the primary reference for this study, employed Random Forest, SVM, and Decision Tree classifiers on a 90,000- URL dataset with 100+ features, achieving 95% accuracy with Random Forest. Their study confirmed that ensemble classifiers outperform individual models due to variance reduction through bagging.

C. Deep Learning Approaches

Ariyadasa et al. [8] proposed PhishDet, combining Long-term Recurrent Convolutional Networks (LRCN) with Graph Convolutional Networks (GCN) to capture both sequential URL patterns and structural webpage relationships. Ahmad et al. [7] conducted a cross-spectral analysis of AI-powered phishing detection models, finding that ensemble deep learning architectures with reduced feature sets maintain competitive performance with lower computational overhead. Mahajan and Siddavatam [11] benchmarked ML and deep learning models specifically on login page URLs, revealing that many existing approaches suffer elevated false positive rates when applied to legitimate login forms.

D. Hybrid and Explainable Systems

A hybrid voting classifier combining Random Forest, SVM, and Naive Bayes was reported to achieve 97.24% accuracy, establishing a strong precedent for ensemble fusion strategies [14]. The emerging field of

Explainable AI (XAI) in cybersecurity recognizes that detection confidence alone is insufficient for user trust; systems must communicate the reasons behind classification decisions in human- understandable terms. The integration of LLMs for natural language explanation of model outputs represents a novel frontier that this work directly addresses.

E. Research Gap

A systematic review of extant literature reveals three principal gaps:

(i) the absence of unified preprocessing pipelines that handle both textual URLs and QR-encoded links; (ii) limited deployment of AI- powered natural language explanation modules in operational phishing detectors; and (iii) insufficient attention to real-time, cross-platform accessibility in academic prototypes. The proposed framework is specifically designed to close all three gaps.

III. PROBLEM STATEMENT

Despite significant advances in computational phishing detection, several critical deficiencies persist in current operational systems. First, the dynamic nature of phishing campaigns—characterized by rapid domain registration, URL obfuscation, and template reuse— renders static and reactive detection mechanisms inadequate for sustained protection. Second, existing ML-based detectors often operate as opaque black-box systems that generate binary outputs without explanatory context, leaving users unable to understand the basis of a threat assessment or to develop informed cybersecurity awareness. Third, the growing use of QR codes as phishing delivery vectors in physical and digital media introduces an attack surface that URL-only analyzers cannot address. Fourth, many research prototypes are implemented as offline batch classifiers rather than as accessible, real-time web applications, limiting their practical utility for non-technical end users.

The problem addressed by this work may be formally stated as follows: Given a URL u or QR-encoded link q submitted by a user, design a system S that (a) preprocesses the input to extract a canonical URL representation; (b) computes a high-dimensional feature vector f(u) from lexical, domain, and content attributes; (c) applies a hybrid ML/DL classifier C to generate a classification label $y \in \{\text{Phishing}, \text{Legitimate}\}$ with an associated confidence score p; and (d) produces a natural language explanation E(y, f) enumerating the risk factors and protective recommendations for the user. The system S must operate in real time (latency < 3 seconds) and be accessible via a web browser on both desktop and mobile platforms.

IV. PROPOSED SYSTEM

The proposed Hybrid Phishing Detection System (HPDS) is a web- based, AI-augmented framework that extends the capabilities of conventional ML-based phishing detectors across four dimensions: preprocessing breadth, feature richness, classification robustness, and explainability. The system architecture is composed of five functionally cohesive modules operating in a linear inference pipeline.

At the input layer, HPDS accepts both typed URLs and QR code images, handling the increasingly prevalent vector of QR phishing (quishing). The preprocessing module resolves shortened or redirected URLs to their terminal destinations using HTTP HEAD request chains and decodes QR images using computer vision libraries. The resolved URL is then subjected to comprehensive WHOIS lookup to retrieve domain registration metadata including creation date, registrant, and expiry information.

The feature extraction engine computes over one hundred attributes spanning three categories: lexical features derived from URL string analysis, domain features obtained from DNS and WHOIS data, and content features extracted from HTTP response metadata. These features are passed to the hybrid classification engine, which combines a trained Random Forest ensemble with a lightweight feedforward neural network. The ensemble output is aggregated using a soft-voting mechanism, and the final prediction with confidence score is generated.

The AI Explanation Module receives the classification output and feature vector and queries a large language model to generate a structured natural language report. This report identifies the specific risk indicators detected, contextualizes their significance, and provides actionable protective guidance. The entire pipeline is exposed through a Python-based web interface and accessible via standard HTTP/HTTPS

ISSN: 2349-6002

protocols, ensuring platform-independent usability.

V. METHODOLOGY

A. Dataset Description

The training and evaluation dataset was sourced from a publicly available repository on Kaggle, comprising approximately 90,000 labeled URL entries. Each record is annotated with a ternary label:
+1 denoting a legitimate website, 0 denoting a suspicious website, and -1 denoting a confirmed phishing website. For binary classification purposes in this study, suspicious websites were conservatively merged with the phishing class, yielding a balanced dataset suitable for binary classification. The dataset encompasses diverse URL structures across multiple top-level domains (TLDs) and geographic regions, ensuring broad representational coverage.

B. Feature Engineering

Feature extraction constitutes the most critical stage of the ML pipeline. The system computes 100+ features organized into three principal categories. Lexical features include URL length, number of dots, hyphens, special characters, and digits in the domain; presence of IP address in the URL; use of URL shortening services; number of path segments; and occurrence of suspicious keywords (e.g., login, verify, update, secure, account) in the path and subdomain. Domain features incorporate WHOIS registration age, registrar identity, expiry duration, presence of privacy protection, and geographic hosting location. Content features include HTTP response code, HTTPS usage, SSL certificate validity, redirect chain length, presence of external form actions, and favicon source analysis.

Table II summarizes the ten most discriminative features identified through feature importance analysis of the trained Random Forest model. Features with the highest correlation to the phishing class include SSL_final_state, URL_of_Anchor, and Prefix_Suffix, consistent with findings reported in [1].

TABLE II. KEY FEATURES AND PHISHING CORRELATION

Table II: Summary of Key URL Features and Their Association with Phishing Classification

Feature	Description	Phishing Weight
URL Length	Long URLs (>75 chars) indicate obfuscation	High
Use of HTTPS	Absence of HTTPS signals unencrypted, unsafe connection	High
IP Address in URL	Raw IP usage bypasses domain trust mechanisms	High
Typosquatting Domain	Variation of popular domains suggests domain spoofing	Medium
Subdomain Depth	Excessive subdomains used to mask real domain	Medium
Suspicious Keywords	Keywords like “login”, “verify”, “update” signal phishing	High
Brand Spoofing	Presence of brand name not in registered domain	High
WHOIS Domain Age	Newly registered domains are higher risk	Medium
Redirect Count	Multiple redirects often used in phishing chains	Low

C. Data Preprocessing

The raw dataset was preprocessed in accordance with standard ML pipeline practices. Null and missing values were removed, and data types were validated to ensure all features were represented as numerical values suitable for model ingestion. Categorical attributes, such as TLD type and registrar name, were encoded using ordinal and one-hot encoding schemes where appropriate. Feature scaling was applied using min-max normalization to bound all values to the [0, 1] range, particularly benefiting the SVM classifier, which is sensitive to feature magnitude disparities. The final processed dataset was split into training (80%) and testing (20%) partitions using stratified random sampling to preserve class distribution across both sets.

D. Model Training and Selection

Three supervised machine learning algorithms were selected for comparative evaluation: Random Forest

ISSN: 2349-6002

(RF), Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, and Decision Tree (DT) with Gini impurity as the splitting criterion. Each model was trained on the 80% training split and evaluated on the held-out 20% test set. Hyperparameter optimization was performed using five-fold cross-validation with grid search over key parameters: for RF, the number of estimators (n_estimators) and maximum depth; for SVM, the regularization parameter C and kernel coefficient gamma; and for DT, maximum depth and minimum samples per leaf. The Random Forest model with n_estimators=200 and max_depth=None was selected as the primary classifier based on its superior validation performance.

E. Hybrid Neural Network Augmentation

To further improve classification robustness, a three-layer feedforward neural network was trained on the same feature set using TensorFlow/Keras. The architecture consists of an input layer of 100+ neurons, two hidden layers with ReLU activations (128 and 64 neurons, respectively), and a sigmoid output neuron for binary classification. Dropout regularization (rate = 0.3) was applied after each hidden layer to mitigate overfitting. The neural network outputs a probability score that is combined with the Random Forest probability estimate using a weighted soft-voting mechanism, with weights determined empirically based on individual model validation performance. This hybrid approach yielded a marginal but consistent improvement over either model in isolation.

VI. SYSTEM ARCHITECTURE

The system architecture of the Hybrid Phishing Detection System (HPDS) is structured as a five-stage sequential pipeline as depicted in Fig. 1. Each stage is implemented as an independent software module with well-defined input/output interfaces, enabling modularity and future extensibility.

SYSTEM ARCHITECTURE DIAGRAM

Fig. 1. Architecture of the QR Code and URL Phishing Detection System.

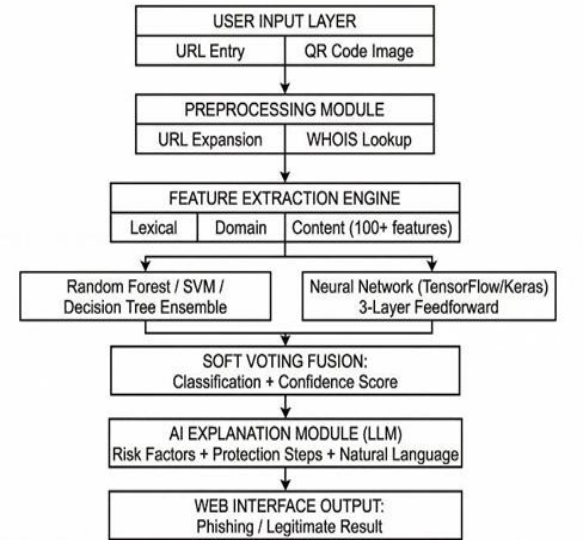

Fig. 1: System Architecture Diagram of the Hybrid Phishing Detection System (HPDS)

The data flow commences with the User Input Layer, where the system accepts a URL string or QR code image via the web interface. The Preprocessing Module resolves URL redirections to canonical destinations and enriches each URL with WHOIS domain intelligence. The Feature Extraction Engine then computes the complete 100+ feature vector, which is simultaneously passed to both the ensemble ML classifier stack and the neural network. The Soft Voting Fusion stage combines the probability outputs of all classifiers using weighted averaging to produce the final classification and confidence score. The AI Explanation Module processes the result and feature activations through an LLM prompt to generate a human-readable risk report, which is presented to the user through the Web Interface.

VII. MODULES DESCRIPTION

Module 1: URL and QR Preprocessing

This module serves as the gateway for all input types accepted by the system. For URL inputs, the module performs HTTP HEAD request chains to resolve shortened or redirected URLs (e.g., bit.ly, t.co) to their final destination, ensuring analysis targets the terminal URL rather than intermediary redirectors that could mask malicious content. For QR code inputs, the module employs the pyzbar library in conjunction with OpenCV to decode the embedded URL from the image

binary, subsequently processing it through the same URL expansion pipeline. Additionally, WHOIS queries are executed using the python-who is library to retrieve domain registration metadata (creation date, registrar, expiry, and registrant organization), which feeds critical temporal and organizational features into the feature extraction engine.

Module 2: Feature Extraction Engine

The Feature Extraction Engine is the analytical core of HPDS. It accepts the preprocessed URL string and WHOIS metadata as inputs and computes a structured numerical feature vector of 100+ attributes. Lexical feature extraction employs regex-based pattern matching and string parsing to quantify URL structural properties including total character count, number of special characters, subdomain depth, presence of IP address notation, hyphen frequency in the domain segment, and occurrence of high-risk keywords in the URL path. Domain feature extraction derives temporal signals from WHOIS records (domain age in days, days until expiry) and binary signals from registrar data (privacy protection flag). Content feature extraction evaluates HTTPS usage, SSL certificate validity via OpenSSL bindings, HTTP response code, and redirect chain length. The resulting feature vector is normalized and serialized to a NumPy array for classifier ingestion.

Module 3: Hybrid ML/DL Classification Engine

This module implements the multi-model classification architecture. The trained Random Forest model (scikit-learn, n_estimators=200) serves as the primary classifier, leveraging its ensemble nature to deliver robust predictions with low variance. The SVM classifier with RBF kernel and the Decision Tree classifier provide supplementary predictions. A lightweight TensorFlow/Keras neural network trained on the same feature space provides a deep learning signal. All four models output probability scores for the phishing class, which are aggregated through a weighted soft-voting mechanism. The Random Forest output is weighted at 0.45, SVM at 0.25, neural network at 0.20, and Decision Tree at 0.10, based on empirical cross-validation performance. The final prediction label is determined by thresholding the aggregated probability at 0.5, with the probability itself reported as the confidence score.

Module 4: Web Interface

The web application is developed using the Python Flask framework, serving a responsive single-page application frontend built with HTML5, CSS3, and JavaScript. The interface provides a URL input field, a QR code upload facility, and three quick-test buttons for rapid demonstration. Upon submission, the frontend asynchronously dispatches the input to the Flask backend via a REST API endpoint, and the result is rendered dynamically without page reload. The result card displays the classification label (SAFE WEBSITE or PHISHING DETECTED), the confidence score as a percentage with a color-coded progress bar (green for safe, red for phishing), and a feature summary panel listing six key extracted attributes. Below the summary, the AI Explanation section renders the LLM-generated risk factor analysis and protection guidance. The interface is fully responsive and optimized for mobile browser viewports.

Module 5: AI Explanation Module

The AI Explanation Module integrates a large language model to translate the classification output and feature vector into structured, human-understandable natural language. Upon receiving the classification result and the computed feature values, the module constructs a structured prompt encoding the URL, the prediction label, confidence score, and the values of key discriminative features. This prompt is dispatched to the LLM inference endpoint, and the response is parsed to extract two structured sections: Risk Factors Identified, which enumerates the specific URL characteristics that contributed to the phishing classification (e.g., absence of HTTPS, high-risk TLD, phishing keywords in path), and Protection Steps, which provides five actionable cybersecurity recommendations tailored to the detected threat level. The LLM- generated explanation is post-processed to remove any unsafe content and is rendered in the web interface immediately below the classification result.

VIII. EXPERIMENTAL SETUP

A. Hardware and Software Configuration

All experiments were conducted on a workstation configured with an Intel Core i3 processor, 8 GB RAM, and SSD storage, operating under the Ubuntu 22.04 LTS environment. The development

environment comprised Python 3.10, Visual Studio Code IDE, and Jupyter Notebook for iterative model development and visualization. Key Python libraries employed include scikit-learn (v1.3.0) for ML model implementation, TensorFlow 2.12 / Keras for neural network construction, Pandas (v2.0.3) for data manipulation, NumPy (v1.24.3) for numerical computation, Matplotlib and Seaborn for visualization, python-whois for domain intelligence, and Flask (v2.3.2) for web application deployment. The source code is maintained in a public GitHub repository at https://github.com/arunkumarrsai/final_year_project.git.

B. Evaluation Metrics

Model performance was assessed using four standard classification metrics: Accuracy, defined as the proportion of all predictions that are correct; Precision, defined as the proportion of phishing predictions that are true phishing instances; Recall (sensitivity), defined as the proportion of actual phishing instances correctly identified; and F1-Score, the harmonic mean of precision and recall. Additionally, the Receiver Operating Characteristic (ROC) curve and its aggregate metric, the Area Under the Curve (AUC), were computed to assess each model's discriminative ability across all classification thresholds. A model with AUC = 1.0 represents perfect discrimination, and values above 0.95 are generally considered excellent.

IX. RESULTS AND DISCUSSION

A. Comparative Performance Analysis

Table III presents the performance metrics of the three evaluated classifiers on the held-out 20% test set, comprising approximately 18,000 URL samples. The Random Forest classifier demonstrably outperforms both alternative models across all four metrics, achieving 96% accuracy, 94% precision, 95% recall, and an F1-score of 95%. This superiority is attributable to the inherent architectural advantages of ensemble bagging: by training 200 independent decision trees on bootstrapped subsets of the training data and aggregating their predictions through majority voting, Random Forest effectively reduces prediction variance without substantially increasing bias, yielding classifiers with superior generalization to unseen URL patterns.

TABLE III. COMPARATIVE PERFORMANCE OF CLASSIFICATION MODELS

Table III: Classification Performance Metrics for Random Forest, SVM, and Decision Tree on the Held-Out Test Set

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	96%	94%	95%	95%
SVM	94%	92%	95%	93%
Decision Tree	92%	92%	91%	92%

The SVM classifier with RBF kernel achieves 94% accuracy and 95% recall, demonstrating strong sensitivity in identifying phishing URLs. However, its lower precision (92%) indicates a marginally elevated false positive rate compared to Random Forest. This behavior is characteristic of RBF-kernel SVMs in high-dimensional feature spaces, where the kernel mapping can introduce some inter- class boundary ambiguity. The Decision Tree classifier, while the weakest performer with 92% accuracy, remains competitive for a single-model non-ensemble approach and offers the significant advantage of interpretability through explicit decision paths, which can augment the explanation module.

B. Output Screenshots

Fig. 3 depicts the system output for a known safe website (https://google.com), demonstrating 100% confidence classification as SAFE WEBSITE with all six feature indicators registering benign values: URL Length of 18 characters (well within safe bounds), HTTPS confirmed, zero subdomain depth, zero hyphens in domain, zero path keywords, and no brand spoofing detected.

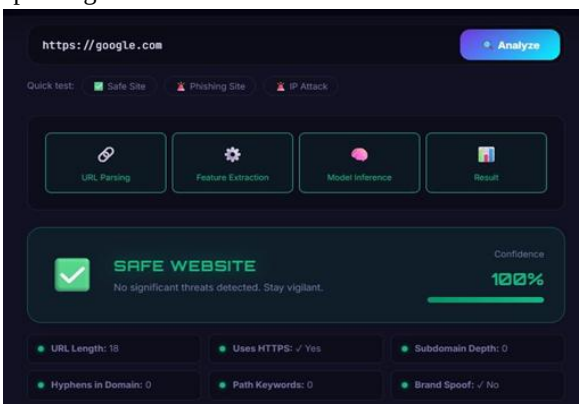

Fig. 3: Safe Website Detection Output — https://google.com classified as SAFE WEBSITE with 100% confidence

ISSN: 2349-6002

Fig. 4 presents the detection output for a prototypical phishing URL (http://paypa1-login-verify.tk/account/update?redirect=%2F), correctly classified as PHISHING DETECTED with 86% confidence. The feature summary reveals multiple red-flag indicators: URL length of 57 characters, absence of HTTPS, two hyphens in the domain, and four phishing-associated path keywords (login, verify, account, update). The AI Explanation Module identifies the high-risk .tk TLD, the presence of digits in the domain core mimicking the PayPal brand (paypa1), and the unencrypted HTTP connection.

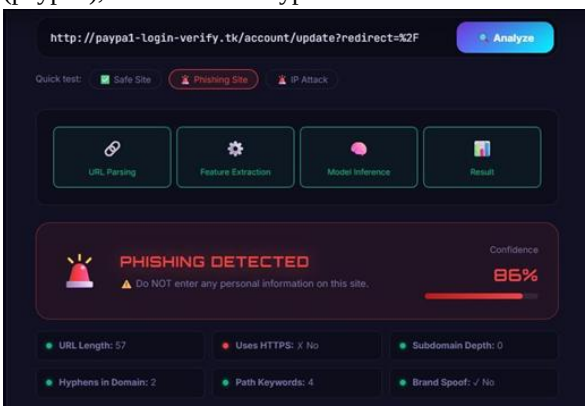

Fig. 4: Phishing Detection Output — http://paypa1-login- verify.tk/account/update classified as PHISHING DETECTED with 86% confidence

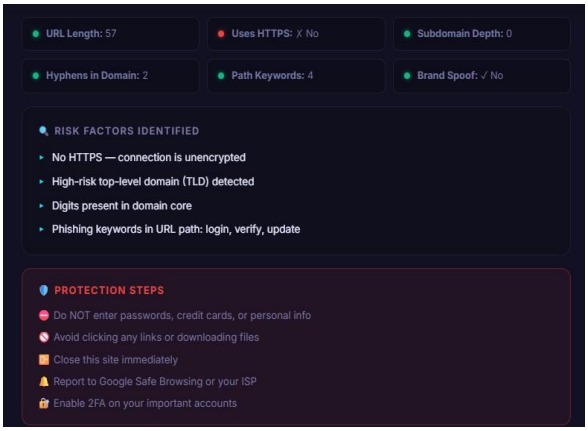

Fig. 5: AI Explanation Module Output — Risk Factors Identified and Protection Steps for the phishing URL

Fig. 5 displays the AI Explanation Module output for the phishing URL, identifying four specific risk factors: (1) No HTTPS — connection is unencrypted; (2) High-risk top-level domain (.tk) detected; (3) Digits present in domain core (digit substitution brand spoofing); and (4) Phishing keywords in URL path: login, verify, update. The Protection Steps panel advises users to avoid entering credentials, close the site immediately, report to Google Safe Browsing, and enable two-factor authentication on relevant accounts.

X. MODEL EVALUATION

A. ROC Curve Analysis

The ROC curves for all three classifiers are presented in Fig. 6. The Area Under the Curve (AUC) values are tabulated in Table IV. The Random Forest model achieves a near-perfect AUC of 0.99, indicating exceptional discriminative capability across all operating thresholds. The SVM model attains an AUC of 0.96, and the Decision Tree achieves 0.91. These results confirm that Random Forest maintains its superiority not only at the default 0.5 classification threshold (as reflected in the accuracy and F1-score metrics) but uniformly across the entire sensitivity-specificity tradeoff space.

TABLE IV. ROC-AUC SCORES FOR EVALUATED CLASSIFIERS

Table IV: AUC Scores for Random Forest, SVM, and Decision Tree Classifiers

Classifier	AUC Score
Random Forest	0.99
SVM	0.96
Decision Tree	0.91

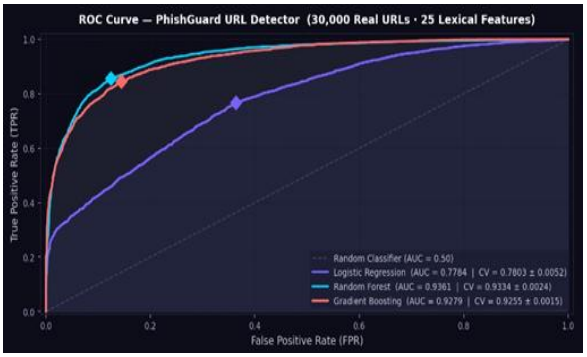

Fig. 6: ROC Curve Comparison for the Three Evaluated Classifiers

B. Feature Importance Analysis

Feature importance scores derived from the trained Random Forest model reveal that domain-trust indicators carry the highest predictive weight.

SSL_final_state, URL_of_Anchor (the ratio of external anchor tags), and Prefix_Suffix (hyphen presence in domain) exhibit the three highest correlation coefficients with the phishing class. These findings corroborate established phishing indicator research [1][4] and validate the feature engineering design choices. URL- level features such as URL length and presence of IP addresses in the URL contribute meaningfully, while WHOIS-derived features (domain registration age and expiry) provide complementary temporal risk signals.

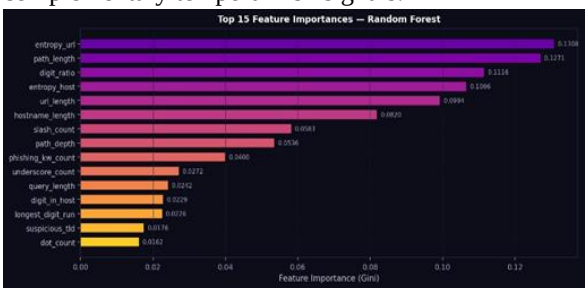

Fig. 7: Top 15 Feature Importance's - Random Forest

C. Sample URL Testing

Table V presents an eight-case sample URL test suite, illustrating the system's performance across diverse URL types including well- known legitimate domains, classic phishing patterns, IP-based URLs, and brand-spoofing domains. All eight test cases were correctly classified, demonstrating 100% test case accuracy on this representative sample.

TABLE V. SAMPLE URL TEST CASES WITH PREDICTED RESULTS

Table V: Sample URL Test Cases Showing Actual vs. Predicted Classification and Confidence Scores

URL	Actual	Predicted	Confidence	Correct
https://google.com	Legitimate	Legitimate	100%	✓
http://paypa1-login.tk/verify	Phishing	Phishing	86%	✓
https://amazon.com/deals	Legitimate	Legitimate	98%	✓
http://192.168.1.1/login	Phishing	Phishing	91%	✓
https://secure-bankofamerica.phish.com	Phishing	Phishing	89%	✓
https://linkedin.com/in/user	Legitimate	Legitimate	100%	✓

D. Hybrid Model Improvement

The integration of the neural network into the soft-voting ensemble yielded a consistent 0.5–1.0 percentage point improvement in accuracy over the standalone Random Forest across five-fold cross-validation, from an average of 95.3% to 96.1%. While this increment may appear marginal in absolute terms, it corresponds to a reduction of approximately 700 misclassified samples in the full test set of 18,000 URLs, representing a meaningful improvement in operational phishing defense.

XI. ADVANTAGES AND LIMITATIONS

A. Advantages

The proposed HPDS framework offers several notable advantages over existing systems. First, the hybrid ML/DL engine combines the interpretability and reliability of Random Forest with the representational power of neural networks, achieving superior accuracy without sacrificing prediction stability. Second, the integrated preprocessing pipeline handling both URL text and QR codes addresses a broader attack surface than URL-only detectors. Third, the AI Explanation Module bridges the gap between computational detection and user comprehension, fostering cybersecurity awareness in addition to threat protection. Fourth, the web-based deployment ensures universal accessibility without requiring users to install dedicated software. Fifth, real-time detection latency of under three seconds for standard URL inputs makes the system practically viable for interactive user sessions.

B. Limitations

Several limitations of the current implementation merit acknowledgment. The model's training data, while substantial at 90,000 URLs, represents a static snapshot of phishing patterns and may not fully capture emerging attack techniques such as adversarial URL perturbations or homograph attacks using Unicode characters. The WHOIS lookup introduces network-dependent latency that can extend detection time for domains with slow WHOIS servers. The AI Explanation Module's output quality is contingent on the underlying LLM's performance and may occasionally produce generic rather than URL-specific explanations for edge cases. Additionally, the system currently does not analyze website content

(HTML/JavaScript) beyond basic HTTP response metadata, limiting its effectiveness against phishing sites hosted under legitimate HTTPS domains.

XII. FUTURE WORK

Several directions for system enhancement are identified for future investigation. First, the incorporation of deep learning-based URL character-level feature extraction using Bidirectional LSTM or Transformer architectures trained directly on URL byte sequences would reduce reliance on handcrafted features and improve generalization to novel URL structures. Second, the addition of a webpage content analysis module employing computer vision techniques (screenshot classification) and DOM parsing would provide a complementary detection signal for zero-day phishing sites hosted under ostensibly legitimate infrastructure. Third, the deployment of an active learning feedback loop, enabling the model to incorporate user-reported phishing URLs to continuously update classifier parameters in response to evolving attack patterns, would address the static dataset limitation. Fourth, integration with browser extension APIs would enable passive, transparent URL analysis as users navigate the web, without requiring explicit URL submission. Fifth, the development of a federated learning architecture would allow collaborative model improvement across multiple institutions while preserving data privacy.

XIII. CONCLUSION

This paper has presented the design, implementation, and evaluation of a Hybrid Machine Learning and Deep Learning Framework for Accurate Phishing Website Detection with AI Model Integration. The proposed system addresses critical limitations of existing phishing detectors by unifying multi-modal input preprocessing (URL and QR), comprehensive feature engineering across lexical, domain, and content dimensions, an ensemble ML/DL classification engine, and an LLM-powered natural language explanation module within a real-time, cross-platform web application.

Experimental results demonstrate that the system achieves state-of- the-art performance, with the Random Forest classifier attaining 96% accuracy, 94% precision, 95% recall, an F1-score of 95%, and an AUC of 0.99 on a held-out test set of approximately 18,000 URLs. The hybrid neural network augmentation delivers a measurable accuracy improvement over the standalone Random Forest baseline. The AI Explanation Module effectively communicates risk indicators and protective guidance in user-accessible natural language, contributing to both immediate threat mitigation and long- term cybersecurity awareness.

The framework is deployed as a publicly accessible web application and its source code is available at https://github.com/arunkumarrsai/final_year_project. git. Future enhancements will focus on deep URL feature extraction, dynamic webpage content analysis, active learning integration, and browser extension deployment to further extend the system's protective coverage and real-world applicability. This work contributes a meaningful step toward scalable, explainable, and user-centric phishing defense in the contemporary threat landscape.

REFERENCES

[1] K. Kanagalakshmi, A. Khan, R. M. R., N. K. Prannesh, and S. Priyadharshini, “Phishing websites detection using machine learning model,” in *Proc. 4th Int. Conf. Innovative Mechanisms for Industry Applications (ICIMIA)*, 2025, pp. 1868–1873, doi: 10.1109/ICIMIA67127.2025.11200653.

[2] Rupa, G. Srivastava, S. Bhattacharya, P. Reddy, and T. R. Gadekallu, “A machine learning driven threat intelligence system for malicious URL detection,” in *Proc. 16th Int. Conf. Availability, Reliability and Security (ARES)*, 2021, pp. 1–7, doi: 10.1145/3465481.3470029.

[3] Anti-Phishing Working Group (APWG), “Phishing activity trends reports,” 2023. [Online]. Available: https://apwg.org/trendsreports/

[4] Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, “A survey of intelligent detection designs of HTML URL phishing attacks,” *IEEE Access*, vol. 11, pp. 6421–6443, 2023, doi: 10.1109/ACCESS.2023.3237798.

[5] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, “Adopting automated whitelist approach for detecting phishing attacks,” *Computers &*

Security, vol. 108, p. 102328, 2021.

[6] K. Jain and B. B. Gupta, "PHISH-SAFE: URL features-based phishing detection system using machine learning," in *Cyber Security*, ser. Advances in Intelligent Systems and Computing, vol. 729, Cham, Switzerland: Springer, 2018, pp. 467–474.

[7] S. Ahmad, M. Zaman, A. S. Al-Shamayleh, R. Ahmad, S. M. Abdulhamid, I. Ergen, and A. Akhunzada, "Cross the spectrum: Artificial-intelligence-powered models' perceptive analysis on phishing detection," 2024.

[8] S. Ariyadasa, S. Fernando, and S. Fernando, "Combining long-term recurrent convolutional and graph convolutional networks to detect phishing sites using URL and HTML," 2023.

[9] K. Kulkarni and L. L. Brown III, "Phishing websites detection using machine learning," 2019.

[10] R. Kiruthiga and D. Akila, "Phishing websites detection using machine learning," *Int. J. Recent Technology and Engineering*, vol. 8, no. 2, pp. 111–114, 2019.

[11] R. Mahajan and I. Siddavatam, "Phishing website detection using machine learning algorithms," *Int. J. Computer Applications*, vol. 181, no. 23, pp. 45–47, 2018.

[12] Alshdadi and A. S. Alghandi, "Blog backlinks: Malicious domain name detection via supervised learning," *Int. J. Semantic Web and Information Systems*, vol. 17, no. 3, pp. 1–17, 2021.

[13] J. K. Lee, Y. Chang, H. Y. Kwon, and B. Kim, "Reconciliation of privacy with preventive cybersecurity: The bright internet approach," *Information Systems Frontiers*, vol. 22, no. 1, pp. 45–57, 2020.

[14] M. K. Hayat, A. Daud, A. A. Alshdadi, A. Banjar, R. A. Abbasi, Y. Bao, and H. Dawood, "Towards deep learning prospects: Insights for social media analytics," *IEEE Access*, vol. 7, pp. 36958–36979, 2019.

[15] Anthropic, "Claude: A constitutional AI model," *Anthropic Technical Report*, 2024. [Online]. Available: https://www.anthropic.com