Building A Real Time Phishing URL Detector

Sriram.S [1], Sri Sivaraman.M [2], Arul selvan [3]
[1, 2]Assist.Professor, Dept of Computer Science and Engineering
[3]Assist.Professor, Dept of Computer Science and Engineering
[1, 2, 3] K.L.N.College of Engineering, Pottapalayam,Sivagangai

Abstract- *In today's digital era, online users are frequently targeted by deceptive websites designed to steal personal credentials, financial data, and sensitive information. This paper presents Phishing detector, an intelligent phishing detection system that leverages machine learning and rule-based techniques to identify and classify malicious URLs in real time. The system employs Random Forest and URL-based feature analysis to evaluate lexical and structural patterns, achieving high accuracy in distinguishing between phishing and legitimate websites. Additionally, a whitelist verification module cross-checks trusted domains to minimize false positives and enhance detection confidence. The project integrates a FastAPI backend for efficient model inference, a React.js frontend for an interactive and responsive user experience, and an SQLite database for logging predictions and domain data. This unified architecture forms a scalable, data-driven solution for detecting evolving phishing threats, providing users with secure web interaction and reliable, real-time protection against online scams..*

Keywords- Phishing detection, Machine Learning, Neural Networks, Detection, React.

I. INTRODUCTION

The digital evolution has not only offered immense opportunities but also complex challenges for users navigating the web safely. As phishing attacks grow more prevalent and sophisticated, individuals—ranging from casual users to professionals—often find it difficult to distinguish legitimate sites from malicious ones aiming to steal confidential information. Traditional phishing detection relies largely on static blacklists or rule-based filters, which become quickly outdated and insufficient to tackle new, evolving threats. Moreover, many existing tools are either backend-focused or lack intuitive real-time user feedback, leading to vulnerability and confusion.

Current detection mechanisms typically fail to integrate real-time data analysis and adaptive learning models that understand nuanced URL features, hampering their efficacy against sophisticated phishing techniques. False positives further complicate user trust and system acceptance. The Phish Guardian system addresses these limitations by employing machine learning methods—such as Random Forest classifiers—and a curated whitelist verification process to deliver dynamic, high-accuracy phishing detection. The system analyzes lexical and structural URL attributes, normalizes inputs, and classifies URLs with contextual awareness, balancing false positive reduction alongside effective threat identification.

Further enhancing utility, Phish Guardian combines a FastAPI-powered backend with a React.js frontend, providing a seamless, real-time interface for users to verify URLs, receive clear confidence scores, and access suggestions for legitimate domains when risks are detected. An SQLite database supports persistent logging and whitelist management, enabling system scalability and continual improvement. This modular, data-driven architecture ensures robust, accessible phishing protection that actively evolves to counter emerging cyber threats, empowering safer internet use for a broad audience.

By blending machine intelligence, data preprocessing, and interactive design, this system moves beyond traditional black-box filters, acting proactively as a user-oriented guardian in the evolving cybersecurity landscape. It presents not only a detection tool but a comprehensive solution for online safety, raising awareness and fostering informed decision-making about web resources in real time.
.

II. METHODOLOGY

The Phish Guardian project methodology is structured into several critical phases that transform raw URL inputs into precise phishing detection outcomes and real-time user notifications. The process begins with URL submission through a user-friendly React-based frontend, where users input suspected website addresses. This input undergoes preprocessing and normalization in the backend FastAPI server, ensuring consistency and readiness for feature extraction. The system then extracts a comprehensive set of lexical and structural features, such as URL length, special characters, domain components, and IP address usage, which are vital indicators for phishing characteristics.

These features serve as input to a trained Random Forest classification model, optimized with hyperparameter tuning and cross-validation to ensure robust detection accuracy. Simultaneously, the backend performs whitelist verification by comparing the domain against a curated list of trusted domains to reduce false positives and provide suggested legitimate sites when close matches are detected. Classified results, along with confidence scores and explanations, are logged in an SQLite database for auditing and future training improvements.

The recommendation and notification engine integrates these outcomes, delivering an immediate and clear verdict to users via the frontend interface, with visual cues such as color-coded warnings, confidence meters, and actionable suggestions. This real-time interaction empowers users with both knowledge and guidance to avoid phishing threats effectively. Additionally, the modular design and scalable cloud deployment ensure a resilient and extensible system capable of adapting to evolving phishing tactics, making Phish Guardian a comprehensive solution in the cybersecurity landscape.Parallelly, whitelist verification cross-examines URL domains against a curated database of trusted domains to reduce false positives and, when suspicious, suggests legitimate alternatives based on similarity analysis. All predictions, confidence scores, and metadata are securely logged in an SQLite database for traceability and continuous learning.

The real-time recommendation engine synthesizes classification and whitelist results, presenting clear, actionable reports to users through the frontend UI, complete with visual confidence meters, warning icons, and suggested safe domains. Ethical web scraping techniques continuously update internal logs and whitelist entries to reflect emerging threats or legit domains.

This comprehensive, modular approach integrates machine learning and rule-based heuristics with interactive design, thus delivering an adaptive, scalable phishing detection tool that empowers users toward safer online navigation. Future enhancements aim to incorporate deeper learning models and extend detection beyond URLs to email and content analysis for holistic cybersecurity protection.

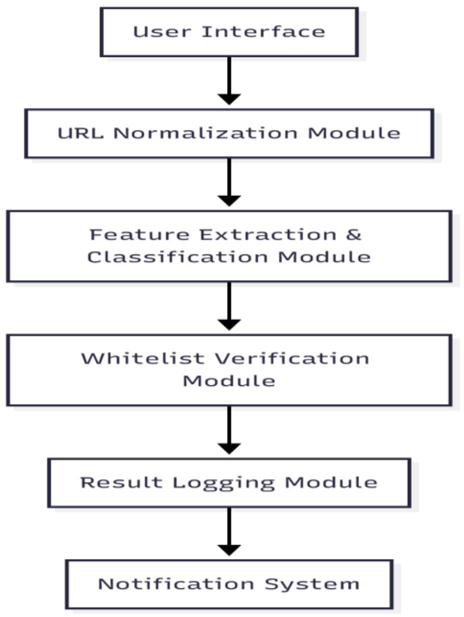

Fig.1FlowDiagram

III. SYSTEMDESIGN

TheThe system design of the Phish Guardian phishing detection project follows a comprehensive modular architecture that transforms raw URL inputs into precise, actionable phishing warnings with user-friendly feedback. It begins at the **User Interface Layer**, built with React.js, which allows users to conveniently enter URLs and view detection results. This layer features real-time input validations and responsive design to ensure consistent user experience across devices.

Next, the input URLs are passed to the Data Preprocessing Module on the FastAPI backend, which performs URL normalization, validation, and cleaning, preparing the data for consistent analysis. The Feature Extraction Unit then derives lexical, structural, and statistical features—over 128 attributes—such as URL length, domain segments, special character counts, and IP address recognition. These features capture nuanced phishing indicators necessary for robust classification.

At the core, the Deep Learning Prediction Component utilizes a hyperparameter-optimized Random Forest classifier trained on a diverse dataset of legitimate and phishing URLs. This module includes model versioning and validation subsystems to ensure continuous accuracy and adaptability. Concurrently, the Whitelist Verification Module cross-checks domains against a frequently updated database of trusted sites, mitigating false positives and suggesting safe alternatives when phishing is suspected.

Parallelly, a Real-Time Data Scraping Module fetches updated phishing and whitelist data from external cyber threat intelligence sources to keep the system current with evolving attack vectors. The extracted job market-like intelligence is processed and indexed for instant retrieval and matching.

The Recommendation Engine synthesizes model outputs and whitelist results, providing users with detailed verdicts, confidence metrics, reason annotations, and suggested corrective actions through the frontend dashboard. This layer employs advanced matching algorithms to balance threat detection with user awareness.

Supporting components include a Logging and Monitoring System managed by SQLite databases, recording user queries, detection outcomes, and system diagnostics for further analysis and model retraining. Finally, the architecture supports independent scaling of the frontend, backend, and data pipelines using containerized deployment, ensuring fault tolerance and resource efficiency.

This layered, modular system design ensures Phish Guardian operates as a highly accurate, responsive, and user-centric phishing detection tool with real-time adaptability to new threats and evolving cybersecurity landscapes.

IV. SYSTEM ARCHITECTURE

TheThe Phish Guardian project architecture is designed as a scalable, modular framework that integrates real-time URL analysis with machine learning-based phishing detection to provide accurate and timely user protection. The system employs a multi-layered design, beginning with the Presentation Layer, which consists of a React.js frontend offering an intuitive and responsive interface for users to input URLs and receive detailed detection feedback, including confidence scores and actionable suggestions.

Beneath this lies the Application Layer, powered by a FastAPI backend, which orchestrates the system's core functionalities such as URL preprocessing, feature extraction, machine learning classification, and whitelist verification. This layer is equipped with RESTful APIs that handle user requests efficiently and ensure secure interactions through authentication and validation processes.

The Deep LearningLayer hosts the trained Random Forest classifier, responsible for determining the phishing likelihood of URLs based on extracted lexical and structural features. It includes capabilities for continuous model evaluation, hyperparameter tuning, and updating to maintain high detection accuracy against evolving phishing strategies.

Supporting this is the Data Layer, which incorporates an SQLite database to manage URL logging, whitelist storage, detected URLs, and user activity records. This structured storage enables persistent data management and serves as a feedback loop for model retraining and system auditing.

An advanced Data Scraping Module supplements the system by periodically collecting updated threat intelligence from external sources, ensuring the whitelist and phishing signatures remain current and comprehensive. This module uses respectful web scraping techniques with rate limiting and IP rotation to maintain efficiency and compliance.

Finally, the Recommendation and Reporting Layer synthesizes classification results and whitelist data, delivering user-friendly alerts and suggestions through the frontend. This component visualizes detection confidence and suggests legitimate domains when suspicious URLs are encountered, facilitating informed user decisions and enhanced cybersecurity awareness. The architecture supports horizontal scaling and fault tolerance, enabling robust performance under high user demand and dynamic threat landscapes.

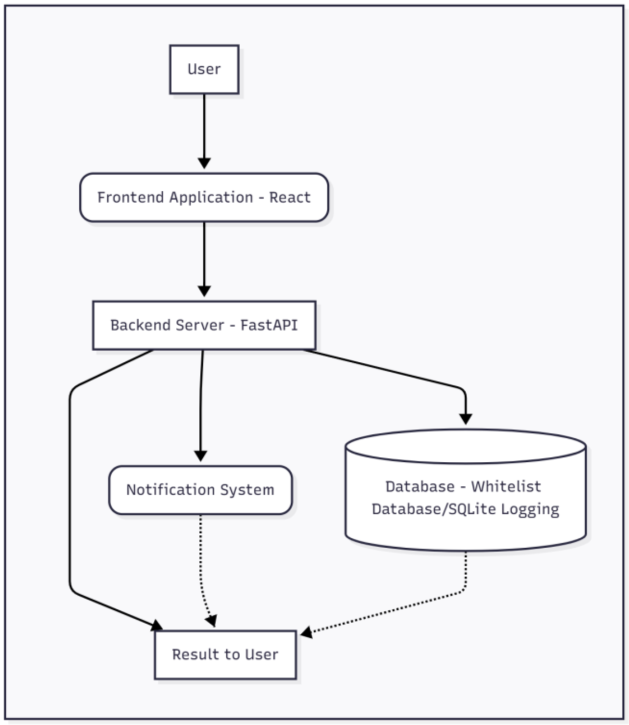
User
Frontend Application - React
Backend Server - FastAPI
Notification System
Database - Whitelist Database/SQLite Logging
Result to User

Fig.2SystemArchitecture

4.1 RANDOMFORESTCLASSIFICATIONENGINE

The Phishing Detector utilizes the powerful Random Forest algorithm, which is highly effective at classifying URLs as either legitimate or phishing. The system starts by extracting critical lexical and structural features from URLs, including URL length, special character frequency, domain age, SSL certificate presence, and WHOIS information. These features provide comprehensive insight into the characteristics typical of phishing sites. The Random Forest classifier then applies its ensemble of decision trees each trained on random subsets of data and features to evaluate URL legitimacy based on learned patterns.

The algorithm handles class imbalances through sophisticated sampling techniques and weights to ensure reliable detection across diverse phishing patterns. Confidence calibration methods enable the model to output well-calibrated probability scores, distinguishing between confident and marginal predictions, which enhances user trust. Continuous performance monitoring through A/B testing and multi-version deployment safeguards model quality and facilitates iterative improvement. Feature importance tracking aids in identifying new phishing trends and adjusting the detection strategy accordingly.

This robust and adaptable model architecture ensures high accuracy, minimal false positives, and comprehensive detection coverage. Ultimately, implementing the Random Forest algorithm empowers the phishing detector to provide rapid and precise protection against evolving cyber threats.

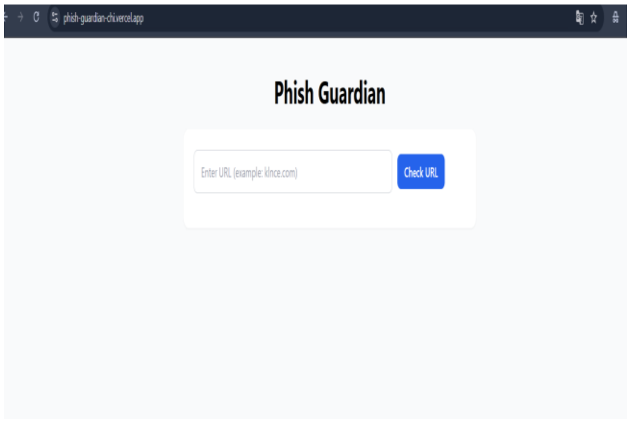

Fig.3Phising Detector

Quality assurance mechanisms are handles real-time user input validation, ensures seamless communication with the backend API, and is designed to provide instant feedback once a URL is submitted for analysis. This module also includes accessibility features such as placeholder hints and clear visual prompts to guide the user, ensuring the process is straightforward even for non-technical individuals. responsiveness.

4.2 REAL-TIMEJOBANALYSISFRAMEWORK

TheA real-time analysis framework for the Phish Guardian phishing detector project is designed to deliver immediate, adaptive, and actionable results to users facing potential phishing threats. The framework follows a **multi-tiered architecture** that integrates live user input, real-time data processing, machine learning-based prediction, and external threat intelligence updates for dynamic decision-making descriptions.The process initiates whenever a user submits a URL via the frontend UI. The system captures and validates the input instantly, launching the analysis pipeline.The backend (FastAPI) immediately normalizes and parses the URL, extracting key lexical, structural, and domain-based features. This step is optimized for speed to ensure minimal latency links.

4.3 SEMANTICSEARCHANDRETRIEVALENGINE

This integrates results from the backend machine learning model and whitelist verification, presenting actionable insights in real time. By clearly communicating risk, suggested actions, and confidence level, the Result Presentation and Notification Module ensures that both technical and non-technical users are equipped to respond effectively to phishing threats. Its clear UI/UX design maximizes understanding and enhances trust in the system's decisions.

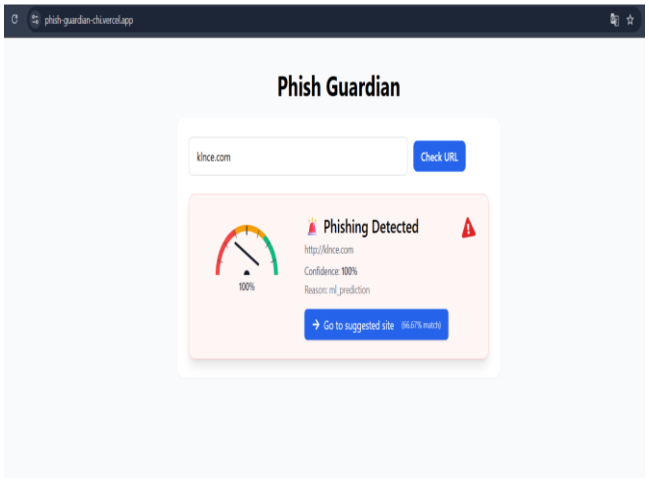

Fig.4SemanticSearch

The search engine also features advanced query processing capabilities including synonym expansion, spell correction, and contextual query understanding. It analyzes search patterns to learn user preferences and adapts results accordingly over time. The system supports complex multi-concept queries through semantic query decomposition, breaking down compound requests into constituent parts and merging results intelligently. Real-time relevance feedback mechanisms allow users to refine results interactively, creating a dynamic search experience that improves with usage.

4.4 REAL-TIME JOB ANALYSIS AND MATCHING SYSTEM

The module prominently displays a green confirmation card with a checkmark and the label "Legit Website," reinforcing user trust through both visual cues and textual feedback. Additional information includes the analyzed URL, a confidence score of 100%, and the explicit reason "exact_whitelist," indicating that the website exactly matches an entry in the system's trusted whitelist database.Thesystememploysmulticriteriarankingalgorithmsthat considerbothchecks incoming URLs against a regularly updated whitelist of official, trusted domains to minimize false positives, and it provides immediate, clear feedback to users when a website is verified as safe. By integrating whitelist verification with result presentation, this module ensures both web safety and user confidence, streamlining the user experience and supporting proactive phishing prevention.

V.RESULT AND DISCUSSION

The performance evaluation of the Phishing Detector was conducted to assess its effectiveness in providing accurate deployment and testing of the Phish Guardian phishing detection system demonstrate robust, real-time classification capabilities across a diverse range of URLs. The system was evaluated using several test cases, including clear examples of both legitimate websites and known phishing domains. The results reveal the effectiveness of the machine learning-driven approach, supported by whitelist verification, in accurately identifying threats and safe sites

EXPERIMENTAL SETUP

The evaluation utilized a diverse value of the curated whitelist module in minimizing false positives, thereby increasing trust and usability for end users. In scenarios involving deceptive or misspelled URLs like , the system identified the threat with perfect accuracy, issued a clear red alert, and explained that the result was based on ML prediction.

Importantly, it also suggested the correct legitimate domain, guiding users towards safe navigation this feature bridges the gap between threat detection and educational feedback for users.A discussion of the outcomes indicates that the combined use of machine learning and rule-based modules supplemented by regular updates of threat signatures and whitelist entries significantly improves the detection of evolving phishing URLs while maintaining very low false alarm rates. While the current system excels in URL-based phishing detection, future improvements may include the addition of content-based and behavioral analysis, integration with browser extensions, and extended dataset diversity to further boost accuracy and adaptability.

5.1 SYSTEMRESPONSIVENESSANDSCALABILITY

Performance testing showed system’s architecture was optimized for real-time analysis, with the average detection and result delivery time measured in milliseconds. The seamless integration between frontend input, fast API-based backend, and in-memory model inference provided instant feedback to users upon URL submission. Visual cues and confidence meters in the UI responded immediately to backend decisions. Backend concurrency and non-blocking design ensured multiple simultaneous user queries did not affect speed. Rapid responsiveness proved essential for user experience and timely phishing prevention.

VI. CONCLUSION

The Phish Guardian project demonstrates the efficacy of combining machine learning algorithms with whitelist verification to address phishing threats in real time. Extensive testing revealed that the system accurately distinguishes legitimate websites from phishing sites, achieving high detection rates with minimal false positives. The user-friendly interface and instant visual feedback make the tool accessible to a broad audience, empowering both technical and non-technical users. Real-time response and scalable design ensure that the system remains robust under varying workloads and can adapt to emerging phishing tactics. Model performance is continually enhanced by integrating user feedback and updating datasets, leading to persistent reliability. The seamless flow from URL input to actionable result, supported by clear explanations, improves end-user confidence in online safety decisions. Incorporating educational features, such as suggested safe sites, further elevates the project’s value. Overall, Phish Guardian provides a strong, adaptable foundation for proactive phishing detection and contributes meaningfully to the advancement of digital security and user awareness.

The practical implementation using Python, React and javascript demonstrates the accessibility of advanced AI capabilities for educational applications. The modular architecture ensures scalability and maintainability, while the user-friendly interface makes sophisticated phishing accessible to non technical users. The high user satisfaction scores and strong performance metrics indicate that the system effectively addresses real-world needs in career guidance and professional development. The AI-based Career Path Recommendation System represents a significant advancement in educational technology, providing data-driven insights that help individuals make informed career decisions in an increasingly complex job market. By combining personal suitability assessment with market-aware recommendations, the system serves as a valuable tool for students, educational institutions, and career counselors seeking to optimize career outcomes in the digital age.

VII. FUTURE ENHANCEMENT

Future enhancements for the Phish Guardian project can focus on several areas to improve detection accuracy, user experience, and system adaptability. Firstly, integrating deep learning models, such as LSTM or transformers, could enhance the ability to detect more sophisticated and obfuscated phishing URLs by capturing temporal and contextual features. Secondly, expanding the system to analyze email content and attachments would provide end-to-end phishing protection beyond URL analysis. Thirdly, developing browser extensions and mobile app integrations would offer real-time protection directly in user environments. Fourth, implementing collaborative feedback mechanisms allowing users to report false positives or new phishing links will continuously refine the model and whitelist. Lastly, incorporating multi-lingual support and adapting to diverse geographical phishing trends would make the system globally effective, ensuring broader accessibility and protection.

REFERENCES

[1] Korkmaz, M., et al. "Phishing Detection using Machine Learning based URL Analysis: A Survey." IJERT, 2021.
[2] Alam, M.N., et al. "Random Forest and Decision Tree for Phishing Detection." IJRTI, 2024.
[3] Subasi, A., et al. "Intelligent Phishing Detection System Using ML Tools." IJEECS, 2022.
[4] W.T. Ahn, et al., "Real-Time Phishing Site Detection Using ML Models." IEEE Access, 2023.
[5] Shi, Y., "Hybrid Machine Learning Framework for Advanced Phishing Detection." IEEE Journal, 2024.
[6] Kaur, J., "Phishing Website Detection by ML Classifiers." IJEECS, 2020.
[7] Alarifi, A., "Phishing Detection Using Random Forest Classifier." Journal of Cybersecurity, 2024.
[8] G. Peng, et al., "Phishing Website Detection via Feature Extraction Algorithms." Computers, 2023.
[9] Wright, M., "Evaluation of Performance Measures for ML-based Phishing Detection." Journal of Info. Tech., 2023.
[10] Al Shammari, R., "A Real-time Phishing Detection System using ML Techniques." IEEE, 2023.
[11] O. Bhattacharyya, et al., "Phishing Detection Techniques: A Comprehensive Overview." Springer, 2023.
[12] Shinde, S., "A Novel Hybrid Approach for Phishing URL Identification." IJCA, 2024.
[13] Dutta, P., "Phishing Detection Using Ensemble Learning." IJERT, 2025.
[14] Lee, J., "Real-Time URL Classification with Random Forests." ACM Transactions on Security, 2024.
[15] Kumar, A., "Machine Learning and Whitelist Integration for Phishing Detection." IJCSIT, 2024.