MANTECH Publications

Journal of Android iOS Development and Testing
Volume 11, Issue 1, January-April 2026

Smart Phishing Detection Application

Ridhesh Chavarekar[1], Shraddha Kadam[2], Vaishnavi Patil[3]

Students

CSE (IOT&CSBT)

Annasaheb Dange College of Engineering and Technology, Ashta, Maharashtra, India

Email ID: *Ridheshrc7@gmail.com[1], Shraddhakadam859@gmail.com[2], Vaishanvipatil9030@gmail.com[3]*

DOI: *https//doi.org/10.5281/zenodo.19232302*

ABSTRACT

Phishing attacks remain a grave cybersecurity concern, especially on mobile operating systems where people are more vulnerable to imitated URLs. This paper introduces the Smart Phishing Detection Application, an intelligent Android- based application that detects phishing URLs in real-time using a hybrid mechanism. The mobile app cooperates well with a backend powered by an XGBoost machine learning powerhouse and threat intelligence APIs like VirusTotal and Google Safe Browsing. Scanning or entering suspicious URLs, which are checked and assigned a phishing threat score, can be done by users. Our technology combines machine learning-based predictions with third-party threat feeds to provide better detection while minimizing false positives. This application is designed to be easy to use, scalable, and efficient in providing mobile users timely phishing alerts, encouraging individual cybersecurity while in transit.

KEYWORDS: *Phishing Detection, Machine Learning, XGBoost, Android Security, Threat Intelligence, Real-time Analysis*

INTRODUCTION

The growth of the internet and mobile devices has led to significant increases in phishing attacks that exploit the users through deception by spoofing trust [7]. Phishing attacks are especially concerning for mobile users because the smaller screen size and interfaces on mobile devices make it harder to identify malicious Element Universal Resource Locator (URL) [6]. Traditional detection mechanisms typically utilize either static blacklists or signature-based

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

methods to detect phishing attacks. Unfortunately, these methods often fall short of keeping up with the ever-changing landscape of phishing [13].

To address these challenges to phishing detection, we develop the Smart Phishing Detection Application (SPDA), an intelligent Android application with a back-end system that incorporates machine learning and threat intelligence APIs. The application implements an Extreme Gradient Boosting classifier which we trained on numerous datasets that included both phishing and legitimate URLs to then classify lexical and structural features of hosted URLs [2], [5], [8]. The machine learning classifier works in parallel with independent queries made to multiple external threat intelligence services, creating a hybrid confidence score when assessing risk [1]. Our research aims to provide a real-time, accurate, and user- friendly detection mechanism for mobile users, facilitating informed decision-making processes to protect themselves from imminent cyber threats [4], [6]. This paper provides details on the system architecture, methods, dataset preparation, and evaluation, displaying it as a personal cyber security tool.

LITERATURE REVIEW

Zhang et al. [1] proposed a dynamic multi-perspective cascade graph framework to identify Ethereum-based phishing fraud in Web3 environments. Through examination of relationships between wallet addresses, domains, and actors, they effectively identified coordinated attack networks. This graph-based view motivated the addition of a Threat Graph Visualizer module to our system, which helps security teams effectively recognize sophisticated attack correlations.

Sadaf [2] suggested a sophisticated phishing website detection model based on XGBoost and CatBoost classifiers. The paper compared both ensemble tree-based models on different datasets and found that XGBoost yielded slightly better accuracy than CatBoost without hyperparameter optimization. This paper exhibits the power of gradient boosting techniques for identifying phishing URLs and justifies our application of similar machine learning models for real-time phishing detection.

Almomani et al. [3] created Phish Fighter, a self-updating machine-learning defense that examines HTML compositions of phishing kits. In contrast to static detection models, the

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

system automatically evolves to accommodate new types of attacks, improving long-term accuracy. Such a notion of autonomous evolution is very much in line with our project's focus on self-learning and smart model evolution.

Albladi et al. [4] have investigated phishing behavior during the COVID-19 pandemic, highlighting the ways in which psychological triggers like fear, urgency, and curiosity made phishing campaigns more successful. Their behavioral observations highlight the necessity for user-oriented alert systems, which we include in our reasoning engine in order to enhance user awareness and reaction.

Sahingoz et al. [5] targeted URL-based phishing identification using machine learning, isolating lexical, host-based, and content features to train the classifiers. Their conclusion demonstrated that multi-dimensional analysis of features greatly enhances detection accuracy over single-feature methods. This multi-layered approach supports our approach of integrating URL analysis with external threat intelligence feeds.

Jain et al. [6] introduced client-side phishing detection based on real-time machine learning classifiers. Their method showed improved end-user security and influenced the mobile-centric design of our phishing detection framework.

Gupta et al. [7] developed a detailed taxonomy of anti- phishing methods, emphasizing current shortcomings and limitations in adaptability. Their work instigated our incorporating multi-source threat intelligence and adaptive detection capabilities to address the issues of static models.

Rao et al. [8] presented Phish-URL, a supervised learning system for phishing URL detection via feature extraction. Their findings demonstrated impressive performance improvement over blacklist-based systems, and justified the effectiveness of supervised learning in phishing protection.

Marchal et al. [9] suggested sophisticated methods for detection of phishing websites and the corresponding legitimate brands they target using content analysis and logo recognition. This contextualization directed our incorporation of sandbox previews and investigation modules for a more in- depth phishing examination.

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Basnet et al. [10] also suggested rule-based phishing detection using pre-defined signatures like suspicious domains and deceptive HTML structures. While it was effective against known attacks, the approach was not adaptive, highlighting the need for machine learning-based models like our work.

Basnet et al. [11] have also carried out one of the first experiments using machine learning classifiers like SVMs and Decision Trees for the detection of phishing. Although their method was more effective than blacklist systems, it had a problem with zero-day attacks — which our adaptive model is designed to overcome.

Bergholz et al. [12] improved phishing detection with the use of statistical and model-based features, showing that well- designed feature sets can enhance classification accuracy. Their result reinforces our emphasis on advanced feature engineering for URL analysis.

Lastly, Zhang et al. [13] tested anti-phishing tools in realistic settings, exposing their shortcomings against adaptive attacks. Their research remains seminal and encourages further research into smart, adaptive phishing detection systems like ours.

PROPOSED METHODOLOGY

This section outlines the holistic approach taken in order to create the Smart Phishing Detection Application through the use of machine learning together with threat intelligence APIs to provide precise and real-time phishing detection for Android devices[6],[7].

A. Dataset Collection and Feature Extraction

A balanced dataset was created from publicly available and proprietary sources, which included thousands of labeled URLs split equally between phishing and legitimate categories. URLs were gathered from PhishTank, OpenPhish, and Alexa's most popular sites to make the dataset diverse and representative[8].

Every URL was preprocessed in order to derive meaningful features that reflect phishing behavior. The major features are domain length, HTTPS usage[3], frequency of special symbols, URL entropy, presence of suspicious keywords, and number of subdomains. These features play an important role in distinguishing between phishing URLs and benign URLs, serving as the basis of the machine learning classifier[5],[8].

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Table 1: Features Extracted from URLs for Phishing Detection

Feature Name	Description	Range/Type
Domain Length	Number of characters in domain	Numeric (int)
Presence of HTTPS	Whether URL uses HTTPS protocol (1 or 0)	Binary
Special Symbol Count	Count of suspicious symbols ('@', '-', '_', etc.)	Numeric (int)
URL Entropy	Shannon entropy of URL characters	Numeric (float)
Suspicious Keywords	Count of phishing-related keywords	Numeric (int)
Subdomain Count	Number of subdomains	Numeric (int)

B. Machine Learning Model Training

The Extreme Gradient Boosting (XGBoost) algorithm was selected for classification due to its ability to efficiently process tabular data and provide robust generalization[2]. The extracted features were normalized and vectorized before training.

Hyperparameters such as learning rate, tree depth, and number of estimators were optimized using grid search and cross-validation techniques[12]. The trained XGBoost model outputs a phishing probability score which classifies URLs into three classes based on a normalized threat scale: Safe, Suspicious, and Phishing.

C. Multi-API Threat Intelligence Integration and Fusion

To enhance detection confidence and reduce false positives, the system integrates real-time results from multiple external threat intelligence APIs: VirusTotal, Google Safe Browsing, and AbuseIPDB. Each API returns a normalized threat score corresponding to URL safety.

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

The final phishing threat score is derived via a weighted fusion formula that aggregates all confidence outputs [1][7]:

$$Y_{(ensemble)} = w_1 \cdot Y_{(XGBoost)} + w_2 \cdot Y_{(VirusTotal)} + w_3 \cdot Y_{(GoogleSafeBrowsing)} + w_4 \cdot Y_{(AbuseIPDB)}$$

Where wi represent the adaptive weights assigned to each source, summing to 1 ($\sum$i = 14wi = 1$\sum i = 14wi = 1$).

These weights prioritize data sources based on their reliability and current response quality.

D. Backend and Mobile Application Implementation

A Python Flask backend runs the XGBoost model and handles API integration. RESTful APIs are used between the backend and the Android frontend application, which is built using Jetpack Compose and Kotlin. The application uses OkHttp for network requests and Jetpack DataStore for secure storage of API keys and configuration preferences [6].

Users may enter manually or scan suspicious URLs through barcode or QR code scanning. The application forwards these URLs to the backend for analysis and presents results in real time. The detection results include the hybrid threat score, categorical safety label, and an optional base64- encoded webpage preview for user visualization[9].

E. Evaluation Metrics

The performance of the system is measured on a new unseen test set with standard classification metrics [5],[8]:

1. Accuracy = TP+TN/FP+FN+TP+TN
2. Precision = TP / (TP + FP)
3. Recall = TP / (TP + FN)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

Where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and false Negatives, respectively.

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

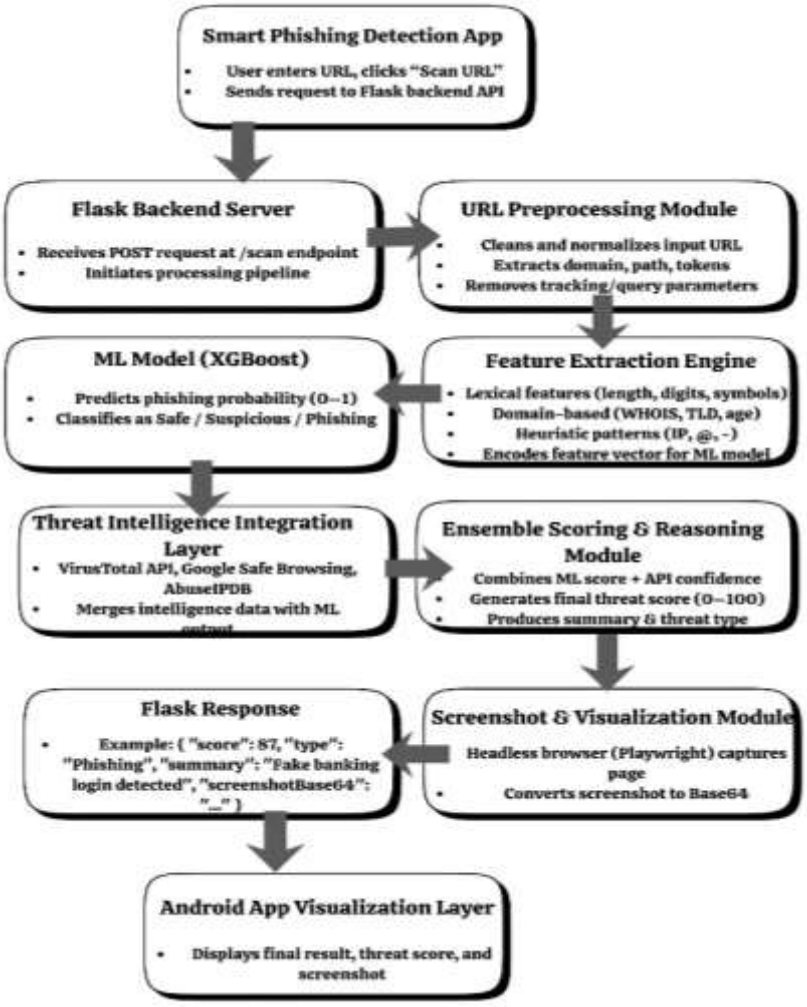

Figure 1: System flow diagram of the proposed application, illustrating the stepwise processing pipeline from user input to phishing detection and result visualization.

RESULTS

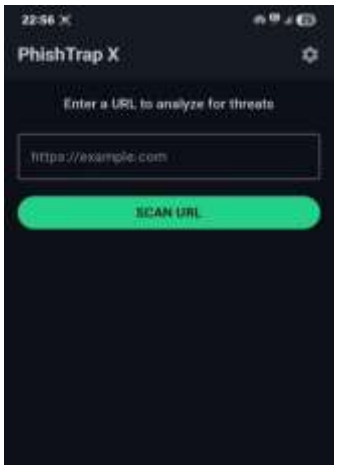

Figure 2: Main user interface for URL threat analysis

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved



Figure 3: Phishing result screen showing a detected phishing URL, threat score, and summary.



Figure 4: Detailed API analysis showing VirusTotal, Google Safe Browsing, and AbuseIPDB threat intelligence reports for a scanned URL



Figure 5: Phishing result screen showing a detected phishing URL, threat score, and summary.

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

CONCLUSION

This document presents the concept and development of the Smart Phishing Detection Application, a mobile security system that is smart and effective in detecting phishing URLs quickly and intelligently by utilizing machine learning and multi-source threat intelligence. The application leverages an XGBoost based classification model, and integrates real-time threat information from trusted external APIs (VirusTotal, Google Safe Browsing, and AbuseIPDB) to enhance detection quality and reduce false positives, thus providing better detection than traditional detection methods.

The application's architecture further supports the communication between a simple client side Android application and a lightweight Flask back end server, providing an informative experience by delivering users with a risk score and preview of a webpage in a timely manner. The hybrid detection framework captures the predictive capability of supervised learning, while the real-time threat intelligence external to the application technologies allows for detection of new emergent phishing URLs.

While taking the usability aspect of the application into account, the system serves the individual user, organization, or educational institution, with analysis that would enhance their cyber posture and improve their cyber situational awareness to sophisticated cyber threats. Future work may be to expand the technology to include deep learning methodologies, fortify and future-proof our technology stack when adding any additional new threat intelligence sources, as well as deliver mobile application for cross platform accessibility.

ACKNOWLEDGMENT

We are grateful to Dr. Sunita.S.Shinde and Dr. Tahseen A. Mulla Head of Department for their constant support, insightful input, and advice, which considerably aided the completion of this study.

REFERENCES

1. X. Zho[1] H. Zhang, Z. Zheng, Y. Wang, and D. Wang, "Unraveling the Deception of Web3 Phishing Scams: Dynamic Multiperspective Cascade Graph Approach for Ethereum Phishing Detection," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 5252– 5265, 2023.

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

2. K. Sadaf, “Phishing Website Detection using XGBoost and Catboost Classifiers,” in Proc. IEEE Int. Conf. on Software, Computing and Applications (ICSCA), pp. 1–6, Feb. 2023, doi: 10.1109/ICSCA57840.2023.10087829.
3. M. Almomani, "Phish Fighter: Self-Updating Machine Learning Shield Against Phishing Kits Based on HTML Code Analysis," IEEE Access, vol. 10, pp. 107651–107663, 2022.
4. M. N. Albladi and G. J. Moran, "COVID-19 and Phishing: Effects of Human Emotions, Behavior, and Demographics on the Success of Phishing Attempts During the Pandemic," Computers & Security, vol. 112, 102499, 2021.
5. O. K. Sahingoz, E. Buber, O. Demir and B. Diri, "Machine Learning Based Phishing Detection from URLs," Expert Systems with Applications, vol. 117, pp. 345–357, 2019.
6. S. Jain and P. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems, vol. 68, no. 4, pp. 687–700, 2018.
7. S. Gupta, B. Arachchilage, and K. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," Telematics and Informatics, vol. 35, no. 6, pp. 1473–1493, 2018.
8. S. Rao and S. Ali, "Phish-URL: A novel feature-based machine learning framework for phishing URL detection," Journal of Information Security and Applications, vol. 40, pp. 92–100, 2018.
9. S. Marchal, N. Singh, and N. Asokan, "Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets," in Proc. IEEE 41st Conference on Local Computer Networks (LCN), pp. 254– 262, 2016.
10. R. B. Basnet, A. H. Sung, and Q. Liu, "Rule-Based Phishing Attack Detection," in Recent Advances in Information Assurance and Security, Springer, pp. 160–170, 2012.
11. R. Basnet, S. Mukkamala and A. H. Sung, "Detection of Phishing Attacks: A Machine Learning Approach," in Studies in Fuzziness and Soft Computing, vol. 226, Springer, Berlin, Heidelberg, pp. 373–383, 2008.
12. Bergholz, J.-H. Chang, G. Paass, F. Reichartz, and S. Strobel, "Improved Phishing Detection using Model-Based Features," in Proc. Conference on Email and Anti-Spam (CEAS), 2008.
13. Y. Zhang, J. Hong, and L. Cranor, "Phinding Phish: Evaluating Anti- Phishing Tools," in Proc. Network and Distributed System Security Symposium (NDSS), 2007

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Cite as:

Ridhesh Chavarekar, Shraddha Kadam, Vaishnavi Patil (2026). Smart Phishing Detection Application. Journal of Android iOS Development and Testing, 11(1), 1-11. https//doi.org/10.5281/zenodo.19232302

Page 1-11 © MANTECH PUBLICATIONS 2026. All Rights Reserved