European Journal of Computer Science and Information Technology, 13(52),53-63, 2025

Print ISSN: 2054-0957 (Print)

Online ISSN: 2054-0965 (Online)

Website: https://www.eajournals.org/

Publication of the European Centre for Research Training and Development -UK

A Visual Cryptographic Technique for Preventing Phishing Attacks in Online Banking Platforms

Victor Thomas Emmah[1] victor.emmah@ust.edu.ng

Kelvin Amadi, [2] kelvin.amadi@rsu.edu.ng

Fortune Baribesia Deedam [3] fortune.deedam@ust.edu.ng

[1,2,3]Department of Computer Science, Rivers State University, Port Harcourt, Nigeria

doi: https://doi.org/10.37745/ejcsit.2013/vol13n525363 Published November 10, 2025

Citation: Emmah V.T., Amadi K., Deedam F.B. (2025) A Visual Cryptographic Technique for Preventing Phishing Attacks in Online Banking Platforms, *European Journal of Computer Science and Information Technology*, 13(52),53-63

Abstract: *Phishing continues to be a prevalent threat to the integrity of online banking platforms, exploiting user trust through deceptive web interfaces and fraudulent URLs. These attacks compromise sensitive information such as login credentials and financial data. In response, this study was initiated to develop an enhanced security model that not only detects phishing attempts but also prevents unauthorized access using cryptographic authentication. This paper aims to secure online banking platforms using a dual-layered approach combining machine learning with Visual Cryptography. To achieve this, a hybrid phishing detection and prevention system was designed and successfully implemented. The system integrates two core modules: an intelligent phishing detection engine and a secure authentication mechanism. The phishing detection engine combines K-Nearest Neighbors (KNN) for analyzing URL-based features with a Convolutional Neural Network (CNN) for image-based classification of websites. For authentication, the system generates two Visual Cryptographic (VC) shares per user during registration. One share is emailed to the user, while the other is stored securely on the server, enabling share recombination at login to verify identity. The solution was integrated with WordPress via REST API endpoints and tested extensively using both browser-based interactions and Postman. The system achieved 94% accuracy with the KNN model and 84% with the CNN model. However, our dual-model approach improves robustness and reduces reliance on one detection path. The average response time for model predictions was approximately 0.136 seconds on Render-hosted API, demonstrating reasonable computational efficiency for real-time use.*

Keywords: phishing, security, visual cryptography, online banking, k-nearest neighbor

INTRODUCTION

The rapid advancement of digital banking has transformed the financial industry, offering users seamless and convenient access to financial transactions. However, with this technological evolution comes an

alarming rise in cyber threats, particularly phishing attacks. Phishing remains one of the most prevalent and sophisticated cybersecurity threats, accounting for a significant percentage of data breaches and financial fraud worldwide (Verizon Data Breach Report, 2021). Attackers impersonate legitimate financial institutions through fraudulent emails, fake websites, and social engineering tactics, deceiving unsuspecting users into divulging sensitive login credentials, financial information, or personal data (Hong, 2012). The growing number of cyberattack aimed at users of online banking has resulted in identity theft and significant financial losses, underscoring the urgent need for stronger security measures. While current security protocols include biometric authentication, One-Time Passwords (OTPs), although tokens provide some protection, they are nevertheless susceptible to advanced threats such as social engineering tactics, phishing websites, and Man-in-the-Middle (MitM) attacks (Gupta et al., 2018). The reactive nature of typical anti-phishing solutions is a major disadvantage; instead of stopping users from visiting fraudulent websites in the first place, they frequently identify and address phishing attempts only after an attack has been launched.

Visual or behavioral analysis of phishing, such as spoofed interfaces or deceptive user interface patterns. Storage limitations, Visual Cryptography generates large, random encrypted shares that are hard to compress, leading to high storage demands and High computational overhead, Visual cryptography, K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN) face significant computational challenges like image processing, high processing leading to high computational overhead.

By addressing these key challenges, this paper proposes a practical, efficient, and user-friendly model that strengthens online banking security against phishing attacks while ensuring accessibility across different user devices. The reactive nature of typical anti-phishing solutions is a major disadvantage; instead of stopping users from visiting fraudulent websites in the first place, they frequently identify and address phishing attempts only after an attack has been launched. This paper is significant as it bridges the gap between usability and security by providing a phishing-resistant authentication system. Unlike traditional password-based authentication, which is easily compromised, Visual Cryptography ensures that the authentication process is immune to phishing attacks while remaining simple for end-users.

REVIEW OF RELATED WORKS

Almomani *et al* (2022) researched the efficacy of semantic features in detecting phishing websites using machine learning classifiers. The classifiers with the highest accuracy, almost 97%, were the Random Forest Classifier and Gradient Boosting Classifier. On the other hand, the accuracies of the Stochastic Gradient Descent and Gaussian Naïve Bayes classifiers were 81% and 84%, respectively. The study came to the conclusion that adding semantic information greatly improves machine learning models' ability to identify phishing websites.

Alshingit *et al*. (2023) developed a deep learning-based phishing detection system using Long Short-Term Memory (LSTMs) and Convolutional Neural Networks (CNNs). They trained their model on a dataset of 60,000 emails and incorporated email header analysis, lexical feature extraction, and behavioral analysis. The model achieved an accuracy of 97.2% and was resistant to obfuscation techniques used by modern phishing attacks.

Bilot *et al*. (2022) proposed an innovative approach to phishing detection by leveraging the inherent relational structure among Uniform Resource Locators (URLs). A graph-based detection technique that constructs a network of Uniform Resource Locators (URLs) and utilizes Graph Neural Networks (GNNs) to analyze the interconnections. Remarkably, their system achieved an accuracy of 99.7%, highlighting the strong potential of network-based detection methods for real world phishing prevention.

Chong *et al*. (2025) developed and evaluated a Uniform Resource Locator based phishing-attack detection system combining three classic machine-learning classifiers K-nearest neighbors (KNN), random forest (RF), and decision tree (DT) with genetic algorithms (GA) for hyperparameter optimization and K-fold cross-validation for performance validation. The results obtained was Random Forest & Decision Tree achieved perfect classification accuracy of 100% while K-Nearest Neighbors reached 99.87% accuracy. These outcomes underscore the high efficacy of ensemble and tree-based models in distinguishing phishing from legitimate Uniform Resource Locators (URLs).

Gao *et al* (2024) developed an image-based phishing detection system using Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs). The research focused on visually deceptive phishing websites that closely mimic legitimate brands. A pre-trained ResNet-50 model was fine-tuned for phishing detection, achieving an accuracy of 96.8%. When combined with Vision Transformers, accuracy improved to 98.1%. The study emphasized that visual similarity-based phishing detection complements traditional text-based methods, improving overall security effectiveness.

Huda *et al*. (2021) developed a highly accurate phishing detection system by leveraging deep neural networks (DNNs) that utilize Uniform Resource Locator (URL) features. The system was benchmarked against several conventional machine learning methods, including K-Nearest Neighbors (KNNs), which achieved a 97.69% accuracy, the deep neural network based approach, however, achieved a remarkable 98.3% accuracy in detecting phishing websites. This result demonstrates that deep learning methods can capture complex patterns in Uniform Resource Locator (URL) data more effectively than traditional classifiers.

Rao and Pais (2019) developed a deep learning-based phishing detection system using Long Short-Term Memory (LSTMs) and Convolutional Neural Networks (CNNs). The integrated deep learning model achieved an accuracy of 97.2% in classifying emails as phishing or legitimate. Notably, the system demonstrated robustness against various obfuscation techniques commonly employed by modern phishing attacks.

Verma and Das (2019) implemented a Convolutional Neural Networks-Long Short Term Memory (CNN-LSTM) hybrid model for phishing uniform resource locator detection, achieving an accuracy of 96.8% and a recall of 97.2%.

Yang *et al* (2020) proposed a hybrid phishing detection system combining deep learning and image-based analysis. Their research aimed to detect visually deceptive phishing websites that mimic legitimate brands. They used Convolutional Neural Networks (CNNs) to analyze webpage screenshots, extracting features such as layout similarity, logo positioning, and font matching. The model achieved a detection accuracy of 95.8%. Their approach was compared with rule-based detection, and results showed that Convolutional

Neural Network based models significantly outperformed traditional text-based classifiers when identifying phishing pages based on visual structure.

Zhang *et al*. (2020) used a hybrid model combining transformers and Convolutional Neural Networks (CNNs) for real-time phishing website detection, achieving a 99% accuracy rate.

ARCHITECTURE OF THE PROPOSED SYSTEM

The architecture of the phishing detection system presented in figure 1 is structured around four core modules: User Interaction module, Prevention module, Detection module, System Intelligence, and Transaction module. Each module plays a unique role in ensuring effective protection against phishing attacks.
The User Interaction Module manages communication with users, while the Prevention Module works to block potential threats before they occur, the Detection Module actively identifies phishing attempts in real-time, and the System Intelligence and Transaction Module oversees decision-making and secure data handling using visual cryptography.

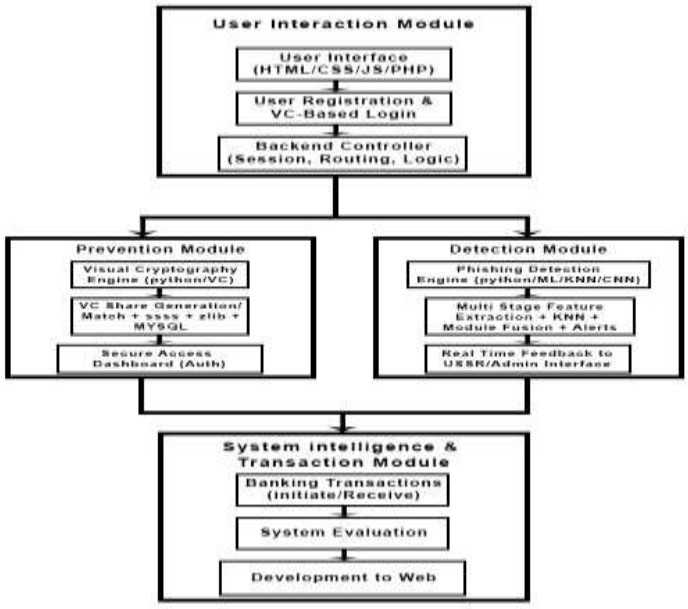

Figure 1. Architectural Design of the Proposed System

(i) User Interaction Module

The User Interaction Module serves as the entry point to the system, ensuring smooth and secure access for both new and returning users. Built with HTML, this module is responsible for rendering the interface

through which users interact with the platform. It supports vital functions such as user registration, login, and initiating requests like transaction activities.

A major component in this module is the User Registration and Visual Cryptography Based Login feature. During registration, users provide credentials and receive a unique visual cryptographic (VC) share which becomes part of their secure login process. When attempting to log in, users must upload their visual cryptographic (VC) share, which is verified before access is granted. This system ensures that credentials alone are not sufficient for authentication, thereby reducing the risk of phishing attacks. The Backend Controller complements the frontend interface by managing sessions, routing user requests, handling authentication logic, and ensuring secure communication between modules. It plays a critical role in maintaining system logic and tracking the user's session state throughout interactions with the system.

(ii) Prevention Module

The Prevention Module is designed to proactively secure the platform by preventing unauthorized access through a unique implementation of visual cryptography. This module is central to the system's innovation, as it integrates image-based two-factor authentication in a lightweight yet robust way.

At the heart of this module is the Visual Cryptography Engine, developed using Python and visual cryptography libraries. It processes the user-uploaded visual cryptographic (VC) share at login and merges it with the server-side stored share. If the combined image reconstructs the original, access is granted. This ensures that even if attackers steal credentials, they cannot log in without the matching visual cryptographic (VC) share.

Further enhancing this process is the visual cryptographic (VC) Share Generation and Matching System, which uses Shamir's Secret Sharing Scheme (SSSS) to divide and store image shares securely. These shares are then compressed using zlib before being saved in a MySQL database, optimizing storage without compromising integrity. The Prevention Module culminates in providing access to a Secure Dashboard, where only verified users can perform sensitive operations like initiating transactions.

(iii) Detection Module

The Detection Module is a critical part of the system's defense mechanism. It uses a combination of Machine Learning (ML) and Deep Learning (DL) to detect phishing threats in real time. This module continuously scans and analyzes incoming data such as Uniform Resource Locators (URLs), metadata, and page layouts, especially during transactions or other user actions that might attract phishing attempts.

The module's core is the Phishing Detection Engine, powered by Python and employing models like K-Nearest Neighbors (KNN) and a lightweight Convolutional Neural Network (CNN). The K-Nearest Neighbors (KNN) model classifies phishing attempts based on extracted features like domain age, Secure Socket Layer (SSL) usage, and Uniform Resource Locator (URL) structure, while the Convolutional Neural Network (CNN) model analyzes visual elements such as website layout or email screenshots to detect visual spoofing.

Supporting this is a Multi-Stage Feature Extraction Unit, which gathers both static and dynamic features. The output from various models is then fused through a model ensemble mechanism to enhance accuracy and minimize false positives. Once an analysis is complete, the system generates real-time alerts that are

routed to the appropriate user and admin interfaces. This ensures quick awareness and response to phishing threats before any damage can occur.

(iv) System Intelligence & Transaction Module

This final module handles the core business logic, including secure financial transactions, system evaluations, and deployment. It represents the bank's operational engine, ensuring that users can carry out transactions safely and that the system continuously improves. The Banking Transactions component allows authenticated users to initiate or receive transfers, check balances, and review transaction histories. Before any operation is executed, it is passed through the phishing detection engine for validation, adding another layer of security.

To ensure performance, the System Evaluation process continuously monitors functionality, user feedback, phishing detection effectiveness, and performance benchmarks. Based on these insights, updates or optimizations are suggested or applied.

Finally, the Deployment to Web component encapsulates the readiness of the system to go live. Once all modules are verified and secure, the platform is deployed onto a web server (such as Namecheap shared hosting), where it becomes publicly accessible to users.

EXPERIMENTAL SETUP

A variety of modern design tools were seamlessly integrated to build a secure online banking system that leverages visual cryptography and advanced phishing detection. The development process begins with the creation of a responsive and intuitive user interface, crafted with HTML. To ensure that the interface adapts gracefully across a range of devices from desktops to mobile phones, the Bootstrap framework was employed, providing a robust foundation for a clean, modern design that enhances usability without compromising functionality. On the server side, PHP was selected as the primary programming language due to its broad support across web hosting environments and its proven effectiveness in handling dynamic content. PHP facilitates the processing of user authentication requests and the management of secure sessions. Complementing this, MySQL was used as the database management system, offering a reliable platform for storing essential data such as user credentials, cryptographic shares, and detailed logs of authentication events. This backend infrastructure is designed to be both robust and scalable, ensuring that user data is managed securely and efficiently. The heart of the security mechanism in this system lies in its implementation of visual cryptography, which was developed using Python. Python was chosen not only for its straightforward syntax and readability but also for the extensive ecosystem of libraries that support image processing and numerical computation. The Pillow library will be instrumental in managing and manipulating images, enabling the efficient splitting of a critical authentication image into multiple shares. Meanwhile, NumPy provided the necessary tools for high-performance, pixel-level operations, ensuring that the reconstruction of images is both accurate and swift. In addition to these core technologies, the system also incorporates OpenSSL to safeguard communications between the client and the server, ensuring that data transmitted over the network is properly encrypted and secure. For the machine learning components specifically, for developing a phishing detection mechanism the scikit-learn library in Python was utilized. This enabled the implementation of a K-Nearest Neighbors (KNN) model, which is used to

analyze and classify Uniform Resource Locators (URLs) in real time, effectively distinguishing between legitimate and phishing sites.

Visual Cryptography is implemented to enhance login security by requiring users to present a unique visual token, referred to as Share 1, during the authentication process. The steps in Visual Cryptography integration are Image splitting into two shares, the visual cryptography scheme implemented follows a (2, 2) secret sharing scheme, where the original image (e.g., a bank logo or authentication symbol) is divided into two separate shares. Each share appears as a random noise image when viewed individually, making it impossible to reconstruct the original without both parts. The splitting process involves converting the image into a binary format (black & white) for easier processing and generating two random-looking shares using pixel expansion, where each pixel is split into two subpixels across the two shares.

Mathematically, if $I\ (x, y)$ represent the pixels at position (x, y) in the original image, then the share $S_1\ (x, y)$ and $S_2\ (x, y)$ are generated using the following pixel expansion rules:

$$S_1\ (x, y),\ S_2\ (x, y) = \begin{cases} Random\ Pair\ A & if\ I\ (x, y) = 0 \quad \text{(black pixel)} \\ Random\ Pair\ B & if\ I\ (x, y) = 1 \quad \text{(white pixel)} \end{cases}$$

Where: Random Pair A ensures that overlaying S_1 and S_2 produces a black pixel and Random Pair B ensures that overlaying S_1 and S_2 produces a white pixel.

RESULTS

The process of Visual Cryptography (VC) share generation in the system begins immediately after a user completes the registration form on the platform. As part of this mechanism, the username entered by the user is programmatically converted into a simple black-and-white image using PHP's GD library. This image serves as the foundational input for the cryptographic splitting process. The transformation of the textual username into a black-and-white image ensures a consistent, binary structure required for pixel-level operations during share creation. Once the username image is generated, it is sent to a Flask-based API endpoint named /generate-vc-shares. The backend receives this image and executes a visual cryptography algorithm that divides it into two separate images (Share 1 and Share 2). The visual cryptography algorithm utilizes randomized pixel mapping and duplication techniques to ensure that neither of the two shares reveals any useful information on its own. Only when both shares are superimposed can the original content be visually reconstructed. The successful generation of these shares is confirmed by converting them into Base64-encoded strings and then either storing or delivering them as appropriate. The validation algorithm then evaluates the degree of pixel-level alignment between the two shares. Specifically, it calculates the percentage of black pixels in the combined image to determine the integrity of the visual match. A threshold value of 0.5 (i.e., 50% black pixel density) is used as the cutoff point. If the combined image meets or exceeds this threshold, the system considers the visual cryptography shares valid and the login process proceeds, otherwise, access is denied, and an error is displayed to the user indicating that the cryptographic share verification failed. In figure 2, the black and white dot image provided to the user as Share1.png is a direct result of the Visual Cryptography algorithm.

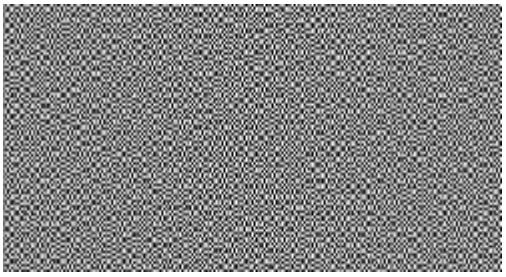

Figure 2. Sample Image of a Share1.png

After the shares are being generated, the K-Nearest Neighbors (KNN) phishing detection model analyzes structured website features, such as domain age, presence of HTTPS, Uniform resource locator (URL) length, and similar characteristics to classify a website as either 'phishing' or 'legitimate'.

The K-Nearest Neighbors (KNN) algorithm serves as the first line of defense in the system's phishing detection module, focusing specifically on analyzing the structural, lexical, and behavioral features of a uniform resources locator (URL) to classify it as either phishing or legitimate. Upon invoking the K-Nearest Neighbors detection process through the "Verify" button on the dashboard, the system internally computes thirty distinct features from the submitted uniform resource locator. These features include values such as the presence of HTTPS, the use of IP address instead of a domain name, the number of dots or subdomains, the presence of special characters like '@' and '-', and keyword-based analysis for terms like "secure", "login", or "bank". Additionally, it calculates metrics like uniform resources locator length, domain age, domain expiration period, redirection count, and iframe usage, among others. The prediction result is returned as either "phishing" or "legitimate". Figure 3 shows the KNN-Based detection interface. Figure 4 and figure 5 show the prediction result returned as legitimate in one instance and another prediction result returned as phishing respectively

Figure 3. KNN-Based Detection

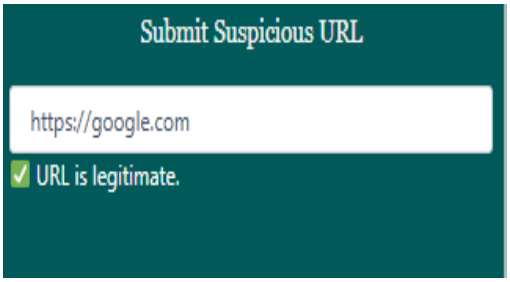

Figure 4. KNN-Based Detection for a Legitimate URL

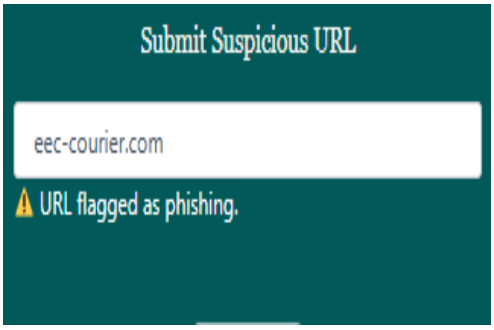

Figure 5. KNN-Based Detection for a Phishing URL

DISCUSSION OF RESULT

The implementation and testing phases of the system produced a comprehensive set of results that reflect both the functional and security capabilities of the platform. Each component of the system was evaluated individually to assess its effectiveness, reliability, and alignment with the project's overarching goals.

To complement the feature-based approach used by K-Nearest Neighbors, the system integrates a convolutional neural network (CNN) model that classifies websites based on their visual appearance. The process begins when a user either clicks the "Initiate Transaction" or "Run Manual Security Check" button on the dashboard. These actions trigger the backend system to automatically launch a Selenium-powered headless Chrome browser, which visits the target uniform resource locator and captures a full-page screenshot without opening a physical browser window. The use of a headless browser ensures that the entire rendering is handled efficiently, in a secure environment, and without affecting user performance.

To provide a more comprehensive and resilient phishing detection mechanism, the system employs a hybrid approach that combines both structural and visual analysis. This is achieved through the /predict-combined endpoint, which simultaneously utilizes the K-Nearest Neighbors model for feature-based detection and the Convolutional Neural Network model for visual classification based on screenshots. This combined approach enhances detection accuracy by mitigating the limitations of relying on a single detection method.

When a user clicks on either the "Initiate Transaction" or "Run Manual Security Check" button on the dashboard, the application sends a request to the /predict-combined endpoint. On the backend, this endpoint performs two major operations. First, it extracts all 30 relevant features from the submitted uniform resource locator and passes them into the K-Nearest Neighbors model. These features may include the presence of IP addresses, uniform resource locator length, number of special characters, and redirection count, among others. Simultaneously, the system uses a headless Chrome browser (via Selenium) to take a screenshot of the webpage rendered by the given uniform resource locator. This screenshot is then processed and classified by the Convolutional Neural Network model. Each model returns its independent prediction, either "phishing" or "legitimate." The system is designed to return "legitimate" only if both K-Nearest Neighbors and Convolutional Neural Network independently classify the input as safe. If either of the models predicts "phishing," the outcome is flagged as "phishing".

CONCLUSION

This paper presented the design, development, and successful deployment of an enhanced security model to secure online banking against phishing attacks. The entire system was anchored on a WordPress front-end interface while relying on a Python-based Flask API hosted on Render for executing core backend logic, including machine learning predictions and secure share validation. The seamless connection between these two platforms showcased the power of hybrid systems in modern cybersecurity applications. At the core of the authentication mechanism lies the Visual Cryptography (VC) share system, which provides a more secure alternative to conventional login procedures. Upon registration, each user is issued a unique Share 1 an image composed of black and white dots derived from their username, which is sent to their registered email address. Simultaneously, Share 2 is stored securely in the backend. During login, the user is prompted to upload their Share 1, and a pixel-wise validation is performed against the stored Share 2. This physical possession model introduces an additional layer of security, making it significantly harder for intruders to access the platform even if passwords are compromised. The system's ability to generate and validate these shares accurately was rigorously tested and confirmed.

Parallel to this, a second major component of the project focused on AI-based phishing detection, leveraging both lexical and visual signals. Two distinct machine learning models were developed and integrated. The first, a K-Nearest Neighbor (KNN) model, used 30 structural and syntactic features extracted from uniform resource locators to classify them as phishing or legitimate. The second, a Convolutional Neural Network (CNN), processed screenshots of websites to detect phishing threats based on visual mimicry. These models were deployed within the Flask API and exposed through dedicated endpoints, one for K-Nearest Neighbor (KNN), another for Convolutional Neural Network (CNN), and a combined endpoint that aggregated predictions from both models to deliver a final verdict. This structure enabled the system to detect phishing attempts with high accuracy, even in the face of new or disguised threats.

This paper stands out for its unique integration of WordPress and Flask, merging the flexibility and accessibility of a content management system with the intelligence and extensibility of a Python-based AI backend. The architectural design not only demonstrates the feasibility of hybrid systems for real-time security tasks but also sets a precedent for how low-cost, scalable tools can be employed in the development of privacy-preserving, intelligent web systems.

REFERENCES

Almomani, A., Alauthman, M., Shatnawi, M. T., Alweshah, M., Alrosan, A., Alomoush, W., & Gupta, B. B. (2022). Phishing website detection with semantic features based on machine learning classifiers: A comparative study. *International Journal on Semantic Web and Information Systems, 18*(1), 1–24.

Alshingiti, Z., Alaqel, R., Al Muhtadi, J., Haq, Q. E. U., Saleem, K., & Faheem, M. H. (2023). A deep learning based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics, 12*(1), 232.

Bilot, T., Geis, G., & Hammi, B. (2022). PhishGNN: A phishing website detection framework using graph neural networks. In *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT)* (pp. 428–435).

Chong, J. C., Sim, N. Y., Khoh, C. W., & Law, T. Y. (2019). Phishing attack detection on URLs using KNN, RF, DT with GA and K-fold cross validation approach. *International Journal of Research and Innovation in Social Science*, 9(1), 1623–1641.

Gao, Z., & Mogos, G. (2024). Phishing recognition software based on machine learning. In *Proceedings of the 8th International Conference on Communication and Information Systems (ICCIS 2024)* (pp. 134–139). Institute of Electrical and Electronics Engineers.

Gupta, B. B., Arachchilage, N. A. G., & Psannis, K. E. (2018a). A comprehensive survey on phishing attacks: Techniques, trends, and countermeasures. *Expert Systems with Applications, 117*, 345–357.

Hong, J. I. (2012). The state of phishing attacks. *Communications of the ACM, 55*(1), 74–81.

Huda, S., Abawajy, J., Islam, R., Alghamdi, M., & Yearwood, J. (2021). A malicious threat detection model for cloud infrastructure using concise features. *Future Generation Computer Systems, 111*, 47–57.

Rao, R. S., & Pais, A. R. (2019a). Detection of phishing websites using machine learning algorithms. In *2019 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 945–949).

Verma, R., & Das, M. L. (2019). Visual cryptography-based phishing detection techniques: A survey. *Computers & Security, 85*, 368–384.

Yang, R., Zheng, K., Wu, B., Wu, C., & Wang, X. (2020). Phishing website detection based on deep convolutional neural network and Random Forest ensemble learning. *Sensors, 21*(24), 8281.

Zhang, Q., Yang, L., & Zheng, Y. (2020). Exploiting edge-oriented features for salient object detection. *IEEE Transactions on Image Processing, 29*, 3748–3761.