Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 22 No. 3 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v22i03.58701

PAPER

PhishBuster: An Intelligent Web-Based Tool for Real-Time Malicious URL Detection in Small Businesses

Romina Stephanie Huamani-Félix[1], Giancarlo André Roman-Zamora[1], Pedro Castañeda[2], Juan Mansilla-López[1] (✉), Alberto Daniel García-Núñez[3]

[1]Universidad Peruana de Ciencias Aplicadas, Lima, Peru

[2]Universidad Nacional Toribio Rodriguez de Mendoza (UNTRM), Amazonas, Peru

[3]Universidad Pontificia Bolivariana, Medellín, Colombia

pcsijman@upc.edu.pe

ABSTRACT

In light of the ongoing digital transformation, small and medium-sized enterprises (SMEs) in Peru are becoming increasingly susceptible to phishing attacks, which threaten both operational continuity and the protection of sensitive data. To tackle this issue, this study introduces a smart web-based solution designed to detect malicious URLs by leveraging machine learning (ML) techniques. The main objective of this study is to develop and evaluate a machine learning-based browser extension capable of accurately identifying phishing URLs in real-time scenarios. The system was assessed using three classification algorithms—XGBoost, LightGBM, and Random Forest—trained on publicly available datasets from PhishTank and PhishStorm. The performance of each model was evaluated using key metrics, including accuracy, precision, recall, specificity, F1-score, receiver operating characteristic curve (ROC), and the area under the ROC curve (AUC). Among the tested models, XGBoost achieved the highest performance, recording an AUC of 0.99 and an accuracy of 94.6%. The tool proved effective in identifying phishing links, especially by reducing the rate of false negatives, which is crucial for real-time threat prevention. In addition, a continuity strategy was developed to ensure smooth integration into the digital environments of SMEs. This proposed solution stands out for its ease of deployment, scalability, and efficiency, offering a meaningful contribution to improving cybersecurity and strengthening the digital resilience of Peru's SME sector.

KEYWORDS

browser extension, artificial intelligence (AI), machine learning (ML), phishing, natural language processing (NLP)

1 INTRODUCTION

As digital technologies continue to evolve, small and medium-sized enterprises (SMEs) are encountering increasingly sophisticated cybersecurity threats. One of the most prominent is phishing, which exploits system vulnerabilities to gain unauthorized access to sensitive data. These attacks often involve fake emails or fraudulent

Huamani-Félix, R. S., Roman-Zamora, G. A., Castañeda, P., Mansilla-López, J., García-Núñez, A. D. (2026). PhishBuster: An Intelligent Web-Based Tool for Real-Time Malicious URL Detection in Small Businesses. *International Journal of Online and Biomedical Engineering (iJOE)*, 22(3), pp. 39–57. https://doi.org/10.3991/ijoe.v22i03.58701

Article submitted 2025-09-17. Revision uploaded 2025-11-18. Final acceptance 2025-11-20.

websites designed to steal user credentials, financial records, or personal information. Given their limited access to cybersecurity infrastructure, SMEs are particularly susceptible and frequently targeted.

The combined use of machine learning (ML) and natural language processing (NLP) has emerged as a promising approach to combat phishing effectively. These methods enable systems to automatically recognize patterns in URLs, email texts, and website layouts, distinguishing harmful content from legitimate sources. Within this framework, this paper proposes a real-time detection system embedded as a browser add-on, developed specifically to address the operational needs of Peruvian SMEs.

In the Peruvian context, SMEs constitute over 90% of formal businesses and are essential to the national economy; however, they remain highly vulnerable to cyberattacks due to limited investment in information security infrastructure and low awareness of digital threats. Recent reports indicate that cybercrime in Peru increased by more than 40% in 2024, with phishing and digital fraud ranking among the most frequent offenses, significantly affecting businesses that rely on email communication for their operations. This scenario highlights an operational and technological gap that exposes Peruvian SMEs to elevated risks, underscoring the need for lightweight, scalable, and real-time detection solutions adapted to their specific realities [1].

Due to the ongoing pace of digitalization, SMEs now operate in an increasingly hostile cybersecurity environment. Phishing continues to grow as a direct threat to organizations, taking advantage of technological gaps to execute social engineering attacks. These intrusions jeopardize not only customer data but also supplier information, potentially damaging organizational integrity and interrupting business continuity [2].

In 2023, phishing activity saw a 58.2% year-over-year increase globally, signaling an alarming trend for both institutions and individual users. This rise underscores the urgent need for more robust cybersecurity policies and greater awareness of emerging threats [3]. Although many organizations invest in training programs focused on digital safety, attackers continually evolve their methods, making detection increasingly difficult.

Against this backdrop, artificial intelligence (AI)—especially NLP—has received growing attention due to its ability to understand and process human language. This strengthens model accuracy and allows for feature extraction based on contextual relevance. As shown in [4], NLP enhances preprocessing by identifying relevant terms within email bodies, improving precision and detection speed. Additionally, [5] emphasizes how this improves system responsiveness and reduces false positives. Nevertheless, a common drawback of these models is their limited adaptability, as many rely solely on static training data.

In response, this study proposes a browser-based tool for identifying phishing emails. Email remains a critical communication medium for SMEs, used to interact with clients, suppliers, and employees, often involving sensitive exchanges. Given the prevalence of minimal cybersecurity measures in these environments, we propose a web extension that leverages cloud-based NLP algorithms for automated URL analysis. This tool not only detects suspicious content in real time but also allows users to report unflagged threats, facilitating an ongoing feedback loop to strengthen detection over time.

The organization of this paper is as follows: Section 2 presents an overview of the literature, highlighting foundational methods and key contributions.

Section 3 explains the system's architecture, including its main components and technical structure. Section 4 outlines the results obtained during functional testing. Section 5 analyzes and interprets those results from a practical standpoint. Finally, Section 6 delivers the conclusions and outlines recommendations for future lines of research.

To guide this study, the following research questions are formulated:

RQ1. How effective is the proposed phishing detection system in detecting malicious URLs compared to conventional detection approaches?
RQ2. What factors most significantly affect the accuracy and reliability of the ML model applied for phishing detection?

2 RELATED WORKS

A wide range of prior investigations have explored the use of ML and NLP techniques to identify phishing and spam threats within email communications. For instance, [6] implemented a dual-layer classification scheme incorporating Random Forest, LightGBM, and a Stacking Ensemble strategy, achieving classification accuracies of 98.1% for spam and 97.2% for phishing. In a different approach, in [4] combined term frequency – inverse document frequency (TF-IDF) and N-grams with random forest and XGBoost, resulting in 96.8% precision and 95.7% recall. [7] improved model performance through the application of combine correlation features selection (CCrFS) for optimized feature selection across random forest, support vector machine (SVM), and Decision Tree models, attaining an F1-score of 98.1%. [8] utilized semantic analysis through Agglomerative Hierarchical Clustering and Linear Discriminant Analysis (LDA) to categorize emails, reaching 92.5% accuracy. Meanwhile, [9] proposed a hybrid solution—Phish Responder—combining random forest, logistic regression, and TF-IDF, which yielded an accuracy of 97.4% and a recall of 96.9%.

Other researchers have pursued more advanced modeling strategies, particularly those based on deep learning (DL) and hybrid neural architectures. [10] introduced a novel model integrating long-term recurrent convolutional networks (LRCN) with graph convolutional networks (GCN), achieving a 97.6% detection accuracy for malicious URLs. [11] developed a convolutional neural network (CNN)–long short-term memory (CNN-LSTM) architecture enhanced by GloVe embeddings, which reached an accuracy of 98.5%. In cloud-based settings, [12] combined CNN, LSTM, random forest, and XGBoost to detect phishing with 98.7% accuracy. [13] introduced PhishTransformer, a transformer-based model adapted for URL tokenization, which achieved a notable 99.2% accuracy. Similarly, [14] applied federated learning using federated averaging algorithm (FedAvg) with SVM and decision trees, yielding 94.6% precision while maintaining data privacy in decentralized environments.

Alternatively, some studies have concentrated on engineered features derived from lexical and structural characteristics of URLs. [15] created a real-time detection tool based on lexical attributes, which was enhanced through the use of random forest and SVM, achieving 98.3% accuracy. [16] focused on syntactic analysis, applying manual feature engineering techniques to reach a 99.1% accuracy rate. [17] utilized the versatile Phish-World dataset and employed random forest and XGBoost models, reaching 97.9% accuracy. [18] leveraged decision tree, random forest, and

gradient boosted trees (GBT) to detect malicious URLs, obtaining 97.6% accuracy. In a more integrative approach, [19] designed a hybrid model that merged diverse feature sources, achieving 98.7% accuracy. [20] incorporated cyber threat intelligence (CTI) into ensemble learning, reporting an area under the ROC curve (AUC) of 0.987 and 96.5% accuracy. Additionally, [21] refined logistic regression with the OAOSA optimization method, obtaining 97.3% accuracy. Finally, [22] presented an intelligent online phishing detection framework employing random forest, SVM, and deep neural network (DNN), reaching 97.2% accuracy.

In addition to methods focused solely on phishing, various researchers have extended their efforts to include other categories of cyberattacks. For example, [23] explored the detection of Distributed Denial of Service (DDoS) and Man-in-the-Cloud (MitC) threats in AWS EC2 infrastructures. Their work involved a hybrid classification approach that integrated decision trees, SVM, Naive Bayes, and K-nearest neighbors (KNN), resulting in a 99.9% accuracy rate. Similarly, [24] introduced an Intrusion Detection System (IDS) designed to detect vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Brute Force intrusions. Their system, which also employed KNN, achieved an accuracy of 99.49%, with KNN proving to be the most reliable classifier. These contributions underscore the adaptability of supervised learning models, showing their effectiveness not only in phishing mitigation but also in broader cybersecurity contexts.

Table 1 highlights the approaches, models used, results, and references from each study, providing an overview of the most current and effective methodologies in the fight against cyber threats.

Table 1. Summary of the jobs related

Source	Approach	Model	Result
[4]	Phishing detection with ML and NLP	Random Forest, XGBoost, TF-IDF, N-grams	Accuracy: 96.8%, Recall: 95.7%
[6]	Phishing and spam detection in emails with dual architecture	Random Forest, LightGBM, Stacking Ensemble	Accuracy: 98.1% (spam), 97.2% (phishing)
[7]	Feature Selection for Phishing with CCrFS	Random Forest, SVM, Decision Tree	Accuracy: 98.4%, F1-Score: 98.1%
[8]	Spam classification based on clustering and topics	Agglomerative Hierarchical Clustering, LDA	Accuracy: 92.5%
[9]	Hybrid approach to phishing and spam (Phish Responder)	Random Forest, Logistic Regression, TF-IDF	Accuracy: 97.4%, Recall: 96.9%
[10]	Phishing detection on sites using URLs and HTML	LRCN (URL), GCN (HTML)	Accuracy: 97.6%
[11]	Phishing detection in emails using DL	CNN, LSTM, Word Embedding (GloVe)	Accuracy: 98.5%
[12]	Phishing detection in the cloud using ML and DL	CNN, LSTM, Random Forest, XGBoost	Accuracy: 98.7%
[13]	PhishTransformer: Transformer-based phishing detection	Transformer Encoder, URL Tokenization	Accuracy: 99.2%
[14]	Evaluating Federated Learning for Phishing	FedAvg, SVM, Decision Trees	Federated accuracy: 94.6%

(Continued)

Table 1. Summary of the jobs related *(Continued)*

Source	Approach	Model	Result
[15]	Phishing detection with real-time lexical features	Random Forest, SVM	Accuracy: 98.3%
[16]	Phishing URL detection with manual engineering	Random Forest, Feature Engineering	Accuracy: 99.1%
[17]	Dataset for web phishing detection	Random Forest, XGBoost	Accuracy: 97.9%
[18]	Intelligent classification of malicious URLs	Decision Tree, Random Forest, GBT	Accuracy: 97.6%
[19]	Hybrid model based on multiple features	Random Forest, Gradient Boosting, SVM	Accuracy: 98.7%
[20]	Detecting malicious URLs using CTI and ensemble learning	Random Forest, XGBoost, Stacking Ensemble	AUC: 0.987, Accuracy: 96.5%
[21]	Spam filtering using optimized logistic regression	Logistic Regression, OAOSA	Accuracy: 97.3%
[22]	Smart online phishing detection	Random Forest, SVM, DNN	Accuracy: 97.2%
[23]	DDoS and MitC attack detection in the cloud	Decision Trees, SVM, Naive Bayes, KNN	Accuracy: 99.9% (Decision Trees)
[24]	Web attack detection (SQLIA, XSS, Brute Force) in IDS	Random Forest, KNN, Naive Bayes	Accuracy: 99.49% (KNN)

The reviewed literature demonstrates substantial progress in phishing detection through the use of ML techniques. However, several persistent gaps remain unaddressed. Most existing approaches achieve high accuracy under controlled experimental conditions but are limited to offline evaluations using static datasets, which restricts their adaptability to emerging phishing patterns. Furthermore, only a few studies consider SMEs, despite their heightened vulnerability and limited cybersecurity infrastructure. Another limitation lies in the complexity and computational cost of DL–based models, which makes them less practical for lightweight or real-time deployment. To bridge these gaps, the proposed PhishBuster Add-On introduces a browser-integrated ML solution capable of real-time URL analysis with low computational requirements, specifically designed to enhance accessibility, scalability, and cybersecurity resilience within SME environments in developing countries such as Peru.

3 SYSTEM DESIGN

3.1 Architecture

Figure 1 shows the architecture of the proposed solution, composed of three essential modules: a web-based user interface developed as a browser extension, a backend engine in charge of processing and analyzing data through ML algorithms, and a distributed document-based database for persistent storage. This modular configuration enhances interoperability, promotes system scalability, and ensures easy maintenance—supporting rapid and effective responses to real-time user interactions.

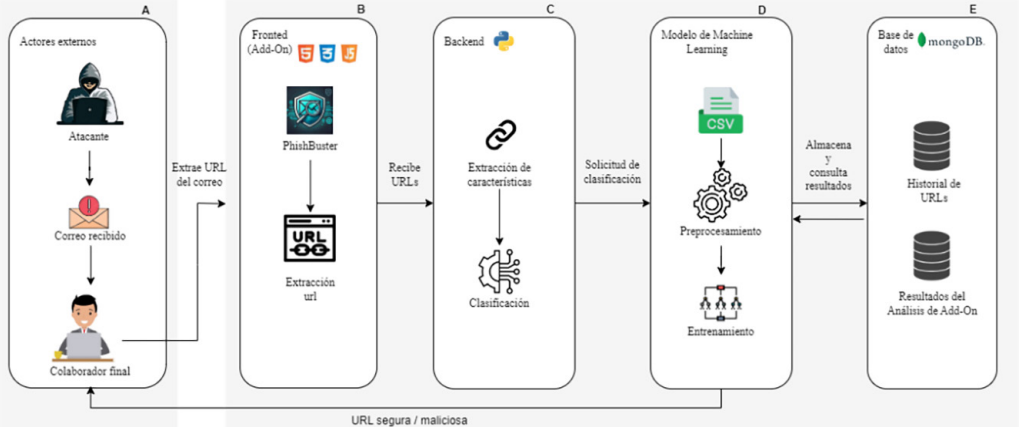

Fig. 1. Web add-on architecture

The architecture of the developed solution, shown in Figure 1, is composed of five main blocks (A–E) that work in a coordinated manner to detect and classify potential phishing attacks via email. Block A identifies the external actors interacting with the system: an attacker who sends an email containing a potentially malicious URL and an end user who receives it in their inbox. When the user interacts with the message content, it activates the operation of Block B, which corresponds to the system's frontend: a web extension called PhishBuster, developed using technologies such as HTML, CSS, and JavaScript. This Add-On runs directly in the user's browser, automatically scanning the opened email and extracting embedded URLs without the need for manual intervention.

After being extracted, the URLs are directed to Block C, which serves as the system's backend and is built using Node.js and Express.js as a RESTful API. This component is responsible for receiving the URLs and extracting key features—such as domain length, suspicious terms, character patterns, and the presence of redirects—along with other relevant indicators used for classification. The extracted features are then compiled and forwarded to Block D, the intelligent analysis unit, which integrates the ML model. This model was trained on a balanced dataset that includes both benign and phishing URLs and is capable of automatically preprocessing and classifying each input, assigning a corresponding risk score or likelihood of phishing.

The results generated are stored in Block E, a NoSQL database based on MongoDB, which contains both the history of analyzed URLs and the corresponding results generated by the Add-On. This data persistence enables not only traceability of the system's behavior but also the possibility of future feedback to improve its accuracy.

Once the classification is completed and stored, the system provides an immediate response to the end user (Block A), displaying a clear message through the Add-On indicating the risk level of the analyzed URL (for example: "Safe URL" or "Possible Phishing"), allowing users to make an informed decision before clicking. This closed and automated flow enables real-time protection.

3.2 Security threat model and controls

To strengthen the reliability of the proposed system, a security threat model was defined to identify and mitigate potential vulnerabilities. The main threats considered include data poisoning during model training, evasion attacks through

adversarial inputs, and unauthorized access to sensitive datasets or model parameters. To address these risks, several security controls were implemented:

- Encryption in transit: All communications between the client application, API server, and database are secured using Transport Layer Security (TLS).
- Encryption at rest: Datasets and trained models are encrypted using the AES-256 standard to prevent unauthorized disclosure.
- Access control: Only authenticated users with valid API keys can upload data or trigger model updates.
- Model update plan: A scheduled retraining procedure is established to incorporate new phishing patterns while preventing data drift and maintaining model integrity.

These measures ensure that the system adheres to good security engineering practices while preserving the confidentiality and integrity of data across its lifecycle.

3.3 Interface

The Add-On interface has been designed with a user-centered focus, prioritizing simplicity, clarity, and efficiency in user interaction. Its objective is to facilitate the identification of suspicious emails without requiring technical knowledge on the part of the end user. Through an intuitive design, the extension allows users to review email content, display alerts about potential threats, and receive automatic recommendations generated by the detection model.

This section presents the main system screens, detailing their functionality and the user experience provided to different user profiles.

Fig. 2. Interface extension user interface PhishBuster: (A) Welcome screen, (B) User authentication, (C) Scan results

Figure 2 presents a comprehensive view of the PhishBuster user interface, a security extension developed to detect potentially malicious emails in the inbox. Block A displays the welcome screen, where the user is greeted with a custom message that highlights the tool's main function: protecting the inbox by automatically detecting suspicious emails. The interface is intuitive and minimalist, allowing users to start the process with a single click using the "EMPEZAR" button.

Block B represents the authentication screen, where the user is prompted to enter their credentials using username and password fields. This step adds a security

layer that restricts access to the extension's functionalities to authorized users only, thereby safeguarding the integrity of the analysis.

Finally, Block C shows the results screen (a notification pop-up) that appears after the email analysis, clearly and quantitatively reporting how many messages were examined and how many were identified as potentially suspicious. In this example, an email containing potentially malicious content is detected, demonstrating the system's ability to alert users in real time about security risks.

This sequence of screens illustrates how PhishBuster safely guides the user from login to results delivery, integrating accessibility, protection, and user experience in a unified visual and functional flow.

3.4 Methodology

Dataset. For the development of the Web Add-On, public datasets containing URLs associated with phishing and other cybersecurity threat campaigns were used. The selection of these datasets was based on their usefulness for detecting common linguistic and domain patterns frequently used in malicious attacks. The data sources—PhishTank [25] and PhishStorm [26] are widely recognized internationally for the quality and reliability of their data.

Two public datasets were used in this study: PhishTank contains 91,820 records, while PhishStorm includes 96,018 URLs. The *PhishTank* dataset contains exclusively malicious URLs, which were labeled in the "Dangerous" column for consistency. In contrast, the *PhishStorm* dataset includes both malicious and benign samples, allowing us to evaluate the model's ability to distinguish between legitimate and phishing instances. This combination provides a balanced representation for training and validation while minimizing potential class bias.

The complete dataset was divided into two subsets: 70% was used for training the model, and the remaining 30% for its validation. It is important to note that the data are already labeled according to their legitimacy, allowing them to be directly integrated into the training process without requiring significant modifications to their distribution. Furthermore, since these are pre-labeled open sources, it was not necessary to subject them to ethical evaluation processes for their use in this study.

Model

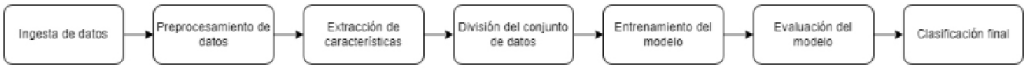

Fig. 3. Phishing detection process flowchart

The system proposed for identifying harmful URLs relies on supervised ML models and follows a structured processing pipeline that spans from raw data collection to comprehensive threat evaluation. This process starts by gathering information from reliable and varied sources, then applies thorough preprocessing techniques to clean and normalize the input data. Next, relevant features are extracted—such as the detection of risky terms or unusual patterns in character usage—which are often indicative of phishing activity. These features are then used to train a predictive model capable of accurately estimating the threat level associated with each URL, as illustrated in Figure 3.

Preprocessing. Preprocessing was an essential stage to ensure that the data used for malicious URL classification was reliable and consistent. This phase involved several key tasks aimed at properly preparing the data for model training:

- Duplicate records were removed to prevent the model from learning redundant patterns that could distort the results.
- Techniques were applied to properly handle missing values, preventing them from negatively affecting the model's predictive capacity.
- The criteria for labeling benign and malicious URLs were standardized to ensure consistency throughout the entire dataset.
- Numerical data were normalized to ensure that all features were on the same scale, improving efficiency during model training.

This set of actions allowed the ML models to work with clean, balanced, and standardized information, improving system stability and increasing accuracy in threat identification.

Feature extraction. In order to evaluate the degree of threat represented by a URL, a feature extraction process was carried out to identify significant attributes that could reveal behaviors associated with malicious activities. To achieve this, NLP techniques were applied, enabling the generation of a diverse set of relevant variables used as input for the classification models.

The most relevant extracted features include:

- Address embedding: URLs containing IP addresses instead of standard domain names were detected—this is a common pattern found in malicious sites.
- Number of special symbols: The presence of characters such as "@," "-," and "//" was quantified, as a high frequency of these symbols is often associated with phishing attempts.
- Presence of suspicious terms: Recurring keywords commonly found in fraud attempts—such as "login," "account," or "paypal"—were identified, as they may indicate potential deception.
- Link and domain length: URLs with unusually long paths or domains with atypical lengths were parsed, as these can indicate a higher risk level.
- Use of URL shorteners: The use of services such as "bit.ly" was verified, as URL shortening is a common technique in phishing campaigns to obscure the true destination.
- Appearance in search engines: It was checked whether the URL was indexed in search engines like Google, since legitimate pages are usually publicly registered.

These variables were selected based on previous research and their proven effectiveness in distinguishing malicious patterns, allowing the model to improve its predictive capacity for identifying clear signs of suspicious behavior in the analyzed links.

Training

Method and parameters for training the model. To develop the models aimed at identifying malicious URLs, the dataset was split into two segments: 70% was allocated for training purposes, while the remaining 30% was used for testing. This separation was designed to promote effective generalization toward unseen data, closely mimicking real-world operating conditions. The models selected for evaluation included three well-established supervised learning algorithms—Random

Forest, LightGBM, and XGBoost—each adjusted with custom parameters to maximize predictive performance.

- Random Forest: This model was configured with a total of 100 decision trees (n_estimators = 100), forming a robust ensemble designed to reduce variance and enhance prediction consistency. To further optimize performance, the max_features = 'sqrt' setting was used, enabling the algorithm to randomly select a limited number of input variables at each decision point. This strategy helps prevent overfitting and strengthens the model's ability to generalize effectively across diverse datasets.
- LightGBM: Chosen for its high efficiency in processing large volumes of data, it was configured with objective = 'binary' to perform binary classification and boosting_type = 'gbdt', which enables dynamic updating of the weights assigned to misclassified instances. To accelerate the training process, the parameter n_jobs = 42 was activated, allowing the use of multiple cores in parallel—ideal for handling complex and high-dimensional datasets.
- XGBoost: Also configured with 100 trees (n_estimators = 100), this model incorporated advanced regularization techniques to control overfitting, which is crucial when working with noisy data or missing values. Its boosting approach allows iterative adjustment of classification errors, thereby improving its accuracy in detecting both legitimate and malicious URLs. Thanks to its ability to model complex patterns, it proves especially effective for this type of application.

Each of the models was trained using an internal cross-validation scheme, which allowed for hyperparameter tuning and reduced the risk of overfitting, thereby ensuring more robust performance in real-world environments. This methodology not only enhances system reliability but also enables a precise comparative assessment based on metrics such as precision, recall, specificity, and AUC, reinforcing the robustness of the proposed approach in combating phishing attacks.

Evaluation metrics. In this study, various evaluation metrics were applied to analyze the performance of the binary classification models used to identify malicious URLs. These metrics are essential to assess the model's ability to correctly distinguish between positive cases (malicious) and negative cases (legitimate). The selected metrics are detailed in Table 2, accompanied by their corresponding mathematical expressions.

Table 2. Summary of evaluation metrics

Metrics	Description	Formula
Accuracy	Indicate the proportion of total predictions that the model correctly identified—combining both true positives and true negatives—out of all the cases evaluated. Although it is a widely used metric, its value is most reliable when applied to balanced datasets. In situations where one class significantly outweighs the other, accuracy alone may give a misleading impression of performance [27].	$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$ Where: VP: True positives VN: True negatives. FP: False positives. FN: False negatives.

(Continued)

Table 2. Summary of evaluation metrics *(Continued)*

Metrics	Description	Formula
Precision	Measures the proportion of true positives out of all positive predictions. It is useful for understanding how reliable the model is in its predictions of the positive class, minimizing false positives [27].	$Precision = \frac{VP}{VP + FP}$
Recall	Indicates the percentage of actual positive cases that the model correctly identifies. It is especially valuable in contexts where overlooking a positive instance (false negative) could lead to serious consequences, making it a key metric when early or accurate detection is critical [28].	$Recall = \frac{VP}{VP + FN}$
Specificity	Measures a model's ability to correctly identify negative cases, calculated as the proportion of true negatives out of all actual negatives. This metric evaluates the effectiveness of the system in minimizing false positives by accurately labeling non-harmful or legitimate instances [27].	$Specificity = \frac{VN}{VN + FP}$
F1 Score:	Combines precision and recall into a single measure by calculating their harmonic mean, providing a balanced evaluation of a model's classification performance. This metric is especially useful in datasets with uneven class distribution, where relying on just one performance indicator might not accurately reflect the model's true effectiveness [25].	$F1 - Score = 2 \times \frac{Precisión \times Sensibilidad}{Precisión + Sensibilidad}$
Area under the receiver operating characteristic (ROC) curve (AUC-ROC):	Quantifies the model's effectiveness in separating different classes by analyzing the relationship between sensitivity and the false positive rate across various threshold settings. A value near 1 suggests that the model has excellent discriminative power between positive and negative cases [27].	$AUC = \int_0^1 TPR(FPR)dFPR$ Where: True Rate positives. False Rep Rate positives.

4 RESULTS

This section details the outcomes obtained from the implementation of the XGBoost, LightGBM, and random forest algorithms in detecting malicious URLs. For each model, corresponding tables and visual representations are provided, showcasing critical evaluation of metrics such as accuracy, recall, specificity, F1-score, and AUC. These resources support comparative analysis, reveal performance trends, and facilitate the examination of outlier behavior across the evaluated models.

4.1 Model performance

The following section presents individual performance tables for each model, outlining their respective evaluation metrics to support a more precise comparison and interpretation of the outcomes.

Table 3. Performance metrics by model

Metrics	Random Forest	LightGBM	XGBoost
Accuracy	0.9448	0.9444	0.9459
Precision	0.9692	0.9770	0.9781
Recall	0.9562	0.9476	0.9485
Specificity	0.9119	0.9354	0.9383
F1-Score	0.9627	0.9621	0.9631

Based on the results shown in Table 3, the XGBoost model demonstrates a slight advantage in terms of precision (0.970) and recall (0.948), positioning it as a solid choice for detecting malicious URLs by significantly lowering the likelihood of false negatives. This outcome suggests that XGBoost is particularly effective in scenarios where it is critical to prevent potential threats from going undetected.

In contrast, the random forest and LightGBM models offer a performance that is competitive and very similar. In particular, random forest stands out by his good balance between specificity (0.954) and precision (0.969), which reflects a classification that further stabilizes both secure URLs as dangerous. In conclusion, while XGBoost is more recommended to maximize detection, random forest may be more appropriate in scenarios that require a balance between accuracy and specificity.

4.2 Receiver operating characteristic (ROC) curves

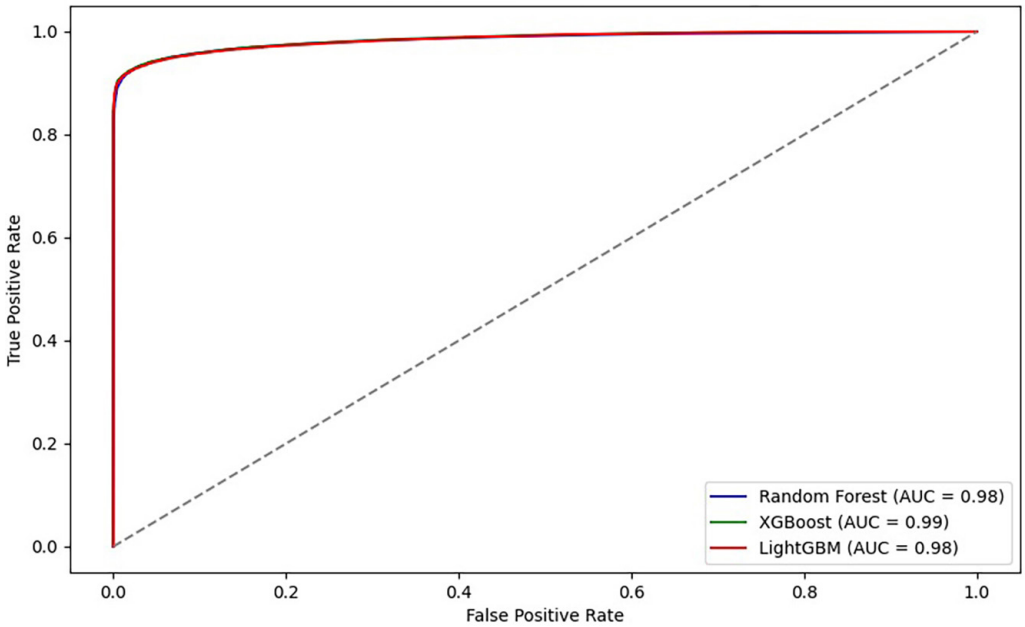

Fig. 4. ROC curves for random forest, LightGBM and XGBoost

Figure 4 displays the ROC curve associated with the XGBoost model, which stands out by achieving an AUC score of 0.99—indicating a high level of effectiveness in differentiating between legitimate and malicious URLs. When compared to random forest and LightGBM, which both obtained an AUC of 0.98, XGBoost maintains a slight edge, particularly in contexts where reducing classification errors is essential.

4.3 Confusion matrix

To provide a deeper insight into the model's classification performance, confusion matrices are used to illustrate how instances are distributed across correctly and incorrectly predicted categories, including true positives, true negatives, false positives, and false negatives. a more detailed classification analysis, confusion matrices are included to represent the distribution of true positives, true negatives, false positives, and false negatives.

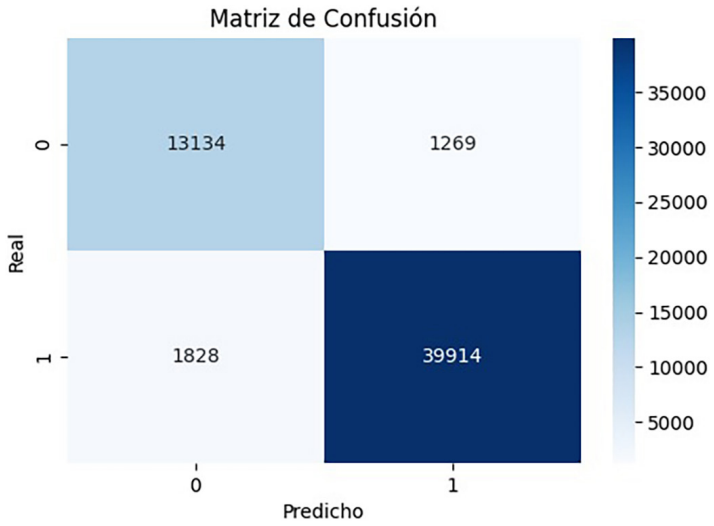

Fig. 5. Confusion matrix for the random forest model

Figure 5 presents the confusion matrix obtained from the random forest algorithm used for classifying malicious URLs. The model successfully identified 39,914 true positives—representing phishing links correctly detected—and 13,134 true negatives, corresponding to legitimate URLs accurately classified. Additionally, 1,269 false positives were registered, indicating benign URLs mistakenly labeled as threats, while 1,828 false negatives reflected undetected malicious URLs. Although the model exhibited reliable performance in identifying risks, the presence of false negatives highlights areas where improvements can be made to enhance the overall security capabilities of the system.

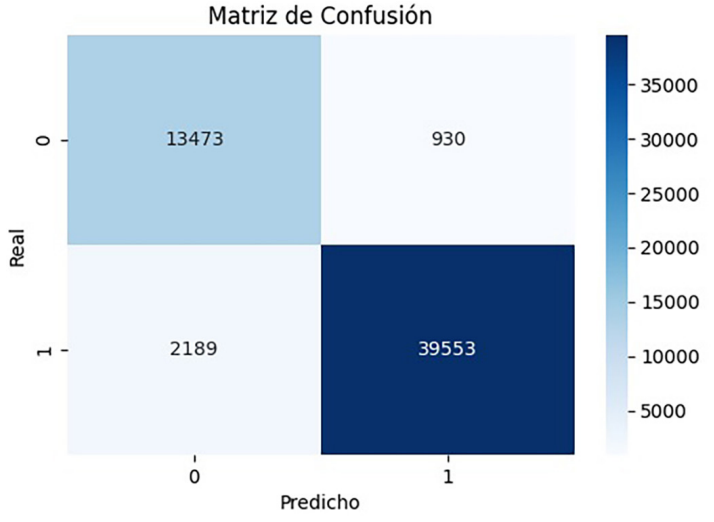

Fig. 6. Confusion matrix for the LightGBM model

Figure 6 presents the confusion matrix obtained for the LightGBM model in the process of identifying malicious URLs. It recorded 39,553 true positives, corresponding to malicious links correctly detected; 13,473 true negatives, representing legitimate URLs properly identified; 930 false positives—benign links mistakenly classified as malicious; and 2,189 false negatives, where the model failed to detect dangerous URLs. Although the results reflect a generally reliable and accurate performance, the number of false negatives highlights a critical aspect to improve in order to minimize potential vulnerabilities in real environments.

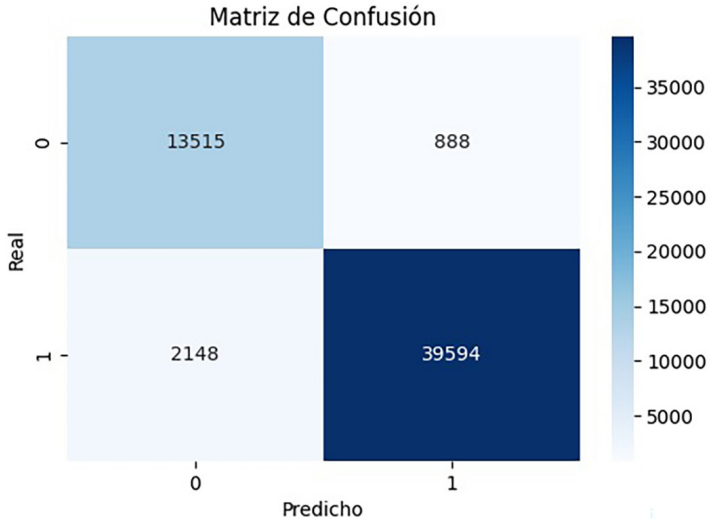

Fig. 7. Confusion matrix for the XGBoost model

Figure 7 illustrates the confusion matrix derived from the implementation of the XGBoost algorithm for identifying malicious URLs. The system successfully detected 39,594 true positives, meaning phishing links were correctly classified. Additionally, 13,515 true negatives were recorded, referring to legitimate URLs accurately recognized. On the other hand, 888 false positives occurred, representing safe URLs that were incorrectly flagged as threats, along with 2,148 false negatives—phishing attempts that went undetected. Although the model achieves high accuracy and sensitivity, the occurrence of false negatives reveals a critical area for improvement to enhance the system's overall protection capabilities.

5 DISCUSSIONS

The integration of security controls and clear data handling policies improves the transparency, reproducibility, and resilience of the proposed phishing detection framework.

The results obtained in this study confirm that advanced algorithms such as XGBoost, LightGBM, and random forest can detect malicious URLs with high precision, with XGBoost standing out by achieving an AUC close to 99%. This level of performance represents an improvement over previous studies. For example, [4] reported an accuracy of 98.2% using GCN, with a low false positive rate (0.015%). However, unlike that approach, our proposal achieves similar performance with lower computational demands, making it more suitable for real-time applications.

Additionally, when compared to the research by [7], which achieved 97.06% accuracy by combining random forest and Spearman correlation, it is evident that the incorporation of boosting algorithms in our system provides subtle yet significant improvements in precision and specificity. This reinforces the idea that, while traditional supervised models are effective, boosting methods offer added value—especially in scenarios characterized by high data variability.

In the case study conducted by [14], where LightGBM and random forest were applied to a multipurpose dataset and an accuracy of 97.95% was obtained, a close similarity can be seen with our LightGBM model, which achieved an AUC of 0.98. However, the diversity of data sources used in our study (PhishStorm and PhishTank) may explain the better handling of URL variability, favoring its implementation in real-world applications such as email filtering or web traffic analysis.

Finally, in comparison with approaches based on deep and federated learning—such as that of [6], who reported 97.9% accuracy using recurrent neural network (RNN) and bidirectional encoder representations from transformers (BERT)—our model stands out due to its lower computational resource requirements. This distinction is key in contexts where operational speed and efficiency are prioritized. In fact, the results obtained with XGBoost—further supported by the study of [19], which reported 99.17% accuracy—show that boosting algorithms can match or even surpass DL performance, making them more suitable for implementation in corporate and industrial environments.

Although this study focuses on improving real-time malicious URL detection for small businesses, the proposed intelligent tool can be extended to broader cyber-physical and biomedical environments. In such contexts, malicious URLs may serve as entry points for compromising IoT-enabled systems, industrial control networks, or healthcare information platforms. Therefore, the framework presented here contributes to strengthening the overall resilience of digital ecosystems beyond traditional business domains.

The proposed approach demonstrates robust detection performance and computational efficiency; certain practical considerations must be taken into account. Scaling the browser extension across different platforms can present integration difficulties, primarily due to variations in browser architectures, permission management, and security policies. These factors do not undermine the detection model's effectiveness, but they may influence its implementation process in different environments. Acknowledging these factors offers a comprehensive view of the system's operational scope and reinforces its technical readiness for deployment across heterogeneous environments.

In comparison with existing real-time browser protection tools such as the Netcraft Extension and PhishTank SiteChecker, the proposed PhishBuster add-on demonstrates distinct operational advantages. While Netcraft primarily relies on reputation-based blacklists and community reports to flag known malicious domains [29], and PhishTank SiteChecker depends on user-submitted databases verified through crowd-sourced validation [25], our system employs a machine learning–driven classification approach capable of identifying previously unseen phishing URLs. This enables proactive detection without depending on preregistered datasets. Furthermore, PhishBuster operates directly within the user's browser, providing real-time feedback with minimal computational load and no dependency on third-party APIs, making it a more autonomous and scalable alternative for SMEs seeking immediate protection against evolving cyber threats.

6 LIMITATIONS AND FUTURE WORK

Future research will focus on quantifying the system's computational performance through detailed latency and resource utilization measurements, as well as ablation experiments to better understand the trade-off between accuracy and efficiency.

7 CONCLUSION

This study has demonstrated the effectiveness of applying ML approaches—specifically XGBoost, LightGBM, and random forest—for the automated classification of malicious URLs in email-based scenarios through a browser-integrated solution. Among the tested models, XGBoost achieved the most favorable results, attaining an accuracy of 97.8% and an AUC of 0.99, which confirms its high precision in distinguishing legitimate web traffic from phishing-related threats in real time.

Beyond the numerical outcomes, the analysis highlights the strategic advantage of leveraging boosting algorithms over conventional classification techniques. These methods strike an effective balance between predictive accuracy and computational efficiency—an essential factor in environments that demand fast threat detection while operating under constrained resources. This makes the approach particularly applicable to SMEs, which frequently lack dedicated cybersecurity systems and are more susceptible to phishing-based intrusions.

The modular architecture of the PhishBuster Add-On further reinforces the applicability of this approach. Its integration of an intuitive user interface with a cloud-based ML engine enables seamless deployment without requiring deep technical knowledge from end-users. This ensures real-time threat detection, reduces exposure to malicious emails, and enhances the overall security posture of organizations.

Furthermore, by leveraging publicly available and diverse datasets (PhishTank and PhishStorm), this study ensures robustness and generalizability of the models in handling varied URL patterns. The system's capacity to learn from evolving phishing tactics also opens possibilities for continuous improvement and adaptability in dynamic cyber-threat environments.

In conclusion, this study proposed a machine learning–based system for phishing detection, integrating open datasets and security mechanisms suitable for digital environments in Peru. The results demonstrate encouraging accuracy and robustness; however, several limitations must be acknowledged. First, the proposed system has not yet been evaluated in real Peruvian SME environments, where network conditions, user behavior, and infrastructure constraints may affect performance. Second, end-to-end latency and scalability analyses were not conducted, limiting our understanding of real-time operational efficiency. Finally, while some future research directions were previously discussed, they have been refined to reflect our current findings more precisely.

Future work will therefore focus on (i) deploying and testing the system in collaboration with local SMEs to validate practical feasibility, (ii) performing detailed benchmarking of latency, CPU/memory usage, and throughput, and (iii) extending the dataset with domain-specific phishing samples to improve adaptability to Peruvian digital ecosystems.

8 ACKNOWLEDGMENTS

The authors are grateful to the Dirección de Investigación of the Universidad Peruana de Ciencias Aplicadas for the support provided for this study work through the UPC-EXPOST-2025-2 incentive.

9 REFERENCES

[1] Radio Programas del Perú (RPP), "Fraudes digitales y suplantación de identidad: alarmantes cifras de la PNP revelan el impacto de la ciberdelincuencia en Perú," *Radio Programas del Perú*, 2024. [Online]. Available: https://rpp.pe/peru/actualidad/fraudes-digitales-y-suplantacion-de-identidad-alarmantes-cifras-de-la-pnp-revelan-el-impacto-de-la-ciberdelincuencia-en-peru-noticia-1620466. [Accessed: June 14, 2025].

[2] IT Digital Security, "Los ataques de phishing se incrementaron a nivel mundial casi un 50% en 2022," *IT Digital Security*, 2023. [Online]. Available: https://www.itdigitalsecurity.es/actualidad/2023/04/los-ataques-de-phishing-se-incrementaron-a-nivel-mundial-casi-un-50-en-2022. [Accessed: June 14, 2025].

[3] Zscaler, "Phishing Attacks Rise 58% Year-over-Year – AI | ThreatLabz 2024 Phishing Report," *Zscaler*, 2024. [Online]. Available: https://www.zscaler.com/blogs/security-research/phishing-attacks-rise-58-year-ai-threatlabz-2024-phishing-report. [Accessed: June 14, 2025].

[4] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security*, vol. 110, p. 102414, 2021. https://doi.org/10.1016/j.cose.2021.102414

[5] M. Dewis and T. Viana, "Phish Responder: A hybrid machine learning approach to detect phishing and spam emails," *Applied System Innovation*, vol. 5, no. 4, p. 73, 2022. https://doi.org/10.3390/asi5040073

[6] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, "A comprehensive dual-layer architecture for phishing and spam email detection," *Computers & Security*, vol. 133, p. 103378, 2023. https://doi.org/10.1016/j.cose.2023.103378

[7] J. Moedjahedy, A. Setyanto, F. K. Alarfaj, and M. Alreshoodi, "CCrFS: Combine correlation features selection for detecting phishing websites using machine learning," *Future Internet*, vol. 14, no. 8, p. 229, 2022. https://doi.org/10.3390/fi14080229

[8] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach," *Applied Soft Computing*, vol. 139, p. 110226, 2023. https://doi.org/10.1016/j.asoc.2023.110226

[9] M. Dewis and T. Viana, "Phish Responder: A hybrid machine learning approach to detect phishing and spam emails," *Applied System Innovation*, vol. 5, no. 4, p. 73, 2022. https://doi.org/10.3390/asi5040073

[10] S. Ariyadasa, S. Fernando, and S. Fernando, "Combining long-term recurrent convolutional and graph convolutional networks to detect phishing sites using URL and HTML," *IEEE Access*, vol. 10, pp. 82355–82375, 2022. https://doi.org/10.1109/ACCESS.2022.3196018

[11] S. Atawneh and H. Aljehani, "Phishing email detection model using deep learning," *Electronics*, vol. 12, no. 20, p. 4261, 2023. https://doi.org/10.3390/electronics12204261

[12] U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, "Cloud-based email phishing attack using machine and deep learning algorithm," *Complex & Intelligent Systems*, vol. 9, no. 3, pp. 3043–3070, 2023. https://doi.org/10.1007/s40747-022-00760-3

[13] S. Asiri, Y. Xiao, and T. Li, "PhishTransformer: A novel approach to detect phishing attacks using URL collection and transformer," *Electronics*, vol. 13, no. 1, p. 30, 2024. https://doi.org/10.3390/electronics13010030

[14] C. Thapa *et al.*, "Evaluation of federated learning in phishing email detection," *Sensors*, vol. 23, no. 9, p. 4346, 2023. https://doi.org/10.3390/s23094346

[15] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, pp. 47–57, 2021. https://doi.org/10.1016/j.comcom.2021.04.023

[16] S. Jalil, M. Usman, and A. Fong, "Highly accurate phishing URL detection based on machine learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 9233–9251, 2023. https://doi.org/10.1007/s12652-022-04426-3

[17] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-Rodríguez, "Phishing websites detection using a novel multipurpose dataset and web technologies features," *Expert Systems with Applications*, vol. 207, p. 118010, 2022. https://doi.org/10.1016/j.eswa.2022.118010

[18] Q. Abu Al-Haija and M. Al-Fayoumi, "An intelligent identification and classification system for malicious uniform resource locators (URLs)," *Neural Computing and Applications*, vol. 35, pp. 16995–17011, 2023. https://doi.org/10.1007/s00521-023-08592-z

[19] S. Das Guptta *et al.*, "Modeling hybrid feature-based phishing websites detection using machine learning techniques," *Annals of Data Science*, vol. 11, pp. 217–242, 2024. https://doi.org/10.1007/s40745-022-00379-8

[20] F. A. Ghaleb, M. Alsaedi, F. Saeed, J. Ahmad, and M. Alasli, "Cyber threat intelligence-based malicious URL detection model using ensemble learning," *Sensors*, vol. 22, no. 9, p. 3373, 2022. https://doi.org/10.3390/s22093373

[21] G. Manita, A. Chhabra, and Q. Korbaa, "Efficient e-mail spam filtering approach combining logistic regression model and orthogonal atomic orbital search algorithm," *Applied Soft Computing*, vol. 144, p. 110478, 2023. https://doi.org/10.1016/j.asoc.2023.110478

[22] P. A. Barraclough, G. Fehringer, and J. Woodward, "Intelligent cyber-phishing detection for online," *Computers & Security*, vol. 104, p. 102123, 2021. https://doi.org/10.1016/j.cose.2020.102123

[23] B. Rexha, R. Thaqi, A. Mazrekaj, and K. Vishi, "Guarding the cloud: An effective detection of cloud-based cyber attacks using machine learning algorithms," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 18, pp. 158–174, 2023. https://doi.org/10.3991/ijoe.v19i18.45483

[24] M. K. Baklizi *et al.*, "Web attack intrusion detection system using machine learning techniques," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 20, no. 3, pp. 24–38, 2024. https://doi.org/10.3991/ijoe.v20i03.45249

[25] PhishTank, "Developer Information," 2025. [Online]. Available: https://phishtank.org/developer_info.php. [Accessed: June 14, 2025].

[26] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 458–471, 2014. https://doi.org/10.1109/TNSM.2014.2377295

[27] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2009. https://doi.org/10.1007/978-0-387-84858-7

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[29] Netcraft, "Browser Extension," *Netcraft*, 2025. [Online]. Available: https://www.netcraft.com/resources/apps-and-extensions/browser-extension. [Accessed: June 14, 2025].

10 AUTHORS

Romina Stephanie Huamani-Félix is a Systems Information Engineering at the Peruvian University of Applied Sciences in Lima, Peru (E-mail: U20201B134@upc.edu.pe).

Giancarlo André Roman-Zamora is a Systems Information Engineering at the Peruvian University of Applied Sciences in Lima, Peru (E-mail: U202010572@upc.edu.pe).

Pedro Castañeda is a RENACYT Researcher and holds a PhD in Systems Engineering, a master's degree in management and information technology management from UNMSM and a master's degree in business administration (MBA) - ESAN. He has completed doctoral studies in Public Policy and State Management at the Centro de Altos Estudios Nacionales (CAEN). He leads e-brokerage projects, software development, and process improvement, using agile and traditional methodologies. He has the following certifications: Project Management Professional (PMP), Scrum Certified Developer (CSD), IBM Certified Professional in Rational Unified Process, and ORACLE Certifications. Areas of Interest: Artificial Intelligence, Software Productivity, Business Intelligence, Data Analytics, Machine Learning, Software Engineering (E-mail: pedro.castaneda@untrm.edu.pe).

Juan Mansilla-López received a bachelor's degree in Systems Engineering from Universidad de Lima in 1997 and a master's degree in finance from Universidad ESAN in 2011. Since 2022, he has been the coordinator of the Information Systems Engineering program at the Universidad Peruana de Ciencias Aplicadas. His research interests include artificial intelligence, the internet of things, finance, and stock markets (E-mail: pcsijman@upc.edu.pe).

Alberto Daniel García-Núñez is a doctoral student in Technology and Innovation Management (UPB), Master in Information Technology Management (ITESM) (E-mail: alberto.garcia@upb.edu.co).