Gradient Boosting-Based for Detecting Browser-in-the-Browser Phishing Attacks

Dr. Abdulrahman Alharby
Imam Abdulrahman bin Fisal university
aalharby@iau.edu.sa

Jumanah Mubarak Albishi
Imam Abdulrahman bin Fisal university
2210003080@iau.edu.sa

Noor Adnan Madani
Imam Abdulrahman bin Fisal university
2210003302@iau.edu.sa

Lujain Ali Alqahtani
Imam Abdulrahman bin Fisal university
2210002740@iau.edu.sa

Nourah Shaman Alanazi
Imam Abdulrahman bin Fisal university
2210003357@iau.edu.sa

Leen Abdulwahab Alwadei
Imam Abdulrahman bin Fisal university
2210003078@iau.edu.sa

***Abstract*— Phishing attacks remain a considerable threat to cybersecurity, with new methods like Browser-in-the-Browser (BitB) attacks exploiting user trust in Single Sign-On (SSO) systems. BitB attacks replicate legitimate login pop-ups within a browser environment, bypassing traditional detection mechanisms. This research introduces the Gradient Boosting-Powered BitB Detection Framework (GB-BDF), leveraging Gradient Boosting Classifiers and real-time detection to mitigate BitB threats. The framework combines URL analysis, visual attributes, and user interaction metrics to achieve high accuracy in identifying phishing websites. Initial results demonstrate 100% accuracy due to using a limited dataset of 24 websites, showcasing the framework's potential for real-world deployment. This study contributes to advancing phishing detection methodologies by integrating adaptive machine learning and user-centric design, providing a robust defence against evolving threats.**

keywords-- Browser-in-the-Browser (BitB), Phishing Detection, Gradient Boosting Classifier, Real-Time Detection, Hybrid Feature Analysis, Machine Learning, Single Sign-On (SSO) Security, Advanced Phishing Techniques.

I. Introduction

Among the continuous and ever-evolving threats in cybersecurity, phishing attacks have always been one of the most prevalent. While traditional detection mechanisms try to keep pace with the continuous innovation of attackers to exploit both human trust and system vulnerabilities, a particularly novel and sophisticated method that has emerged in recent years is the Browser-in-the-Browser phishing attack. This technique uses advanced front-end development to spoof SSO pop-ups, which would normally be used for fast, seamless authentication. BitB attacks camouflage as valid browser windows and security indicators, hence traditional phishing detection methods, such as URL reputation analysis or visual inspection by users, are effectively outsmarted.

The novelty in Browser in Browser attacks consists of the smart exploitation of trust in the most common ways of authentication. Unlike classic phishing methods, which involve malicious links or domains, a BitB attack fools its victim through an interface that will look very convincing on their browsers. This emerging threat really exposes a serious weakness in the current methods of detection and also emphasizes the need for novel ways of combating such an advanced form of an attack. Most seriously, the need to address this cyber-attack vector is imperative given that, due to its increasing popularity for businesses, financial bodies, and individual users, Single Sign-On could be compromised in the wake of a BitB attack using rather modest and rudimentary phishing tools. Therefore, the need exists for advanced real-time detection systems that can be created by both researchers and security professionals, which not only detect BitB attacks but also keep pace with the changes in their techniques [1].

The research focuses on limitations in traditional methods of detection, and this new emerging threat hence aimed at the development of a strong adaptive method for BitB phishing attack detection. This not only deepened our academic knowledge in the area but also set the stage for future efforts that could make practical recommendations leading towards enhancing cybersecurity in the world through changes in the education system.

This paper is organized as follows: Section 2 presents the background on browser-in-the-browser (BITB) attacks. Section 3 shows real-life cases of BITB attacks. Section 4 discusses the related works. Section 5 discusses the gap analysis. Section 6 shows the proposed methodology. Section 7 contains a discussion of the proposed method. Section 8 presents the conclusion and future work

II. Background on Browser-in-the-Browser (BitB) Attacks

The Browser-in-the-Browser attack is a new and emerging phishing technique that successfully exploits the increasing usage of Single Sign-On authentication systems. Since SSO allows users to log in to multiple services using a single set of credentials, it makes things very convenient but at the same time introduces unique security risks. BitB attacks, therefore, leverage this dependency as they create highly convincing spoofed SSO login windows on a malicious webpage. First reported as a new threat vector in recent years, BitB attacks have quickly become a sophisticated tool in the arsenal of cybercriminals [2].

Unlike traditional phishing, which typically involves redirecting victims to malicious websites or exploiting email and messaging vulnerabilities, browser-in-browser attacks are carried out within the browser itself. The attackers develop these using advanced web development skills, including HTML, CSS, and JavaScript, creating authentic-looking login pop-ups that look like any other browser window, complete with spoofed URLs, HTTPS padlocks, and familiar branding from major services such as Google, Microsoft, or Facebook [2].

The danger of BitB attacks is that they can camouflage themselves within legitimate-appearing websites. For example, a user visiting a hacked site may receive a pop-up requesting them to log in via SSO. The user thinks, "Oh, this must be legitimate," and proceeds to enter their credentials, which the attacker immediately captures. Since these spoofed windows are already embedded in the real browser setting, traditional anti-phishing tools, such as URL reputation checks or domain validation, often fail to detect them [2].

Since the emergence of SSO for its ease and efficiency, the assault risk from the BitB is escalating for individual users as well as businesses. Such advanced kinds of attacks introduce an imperative urge to develop technologies of detection more than mere static ones; indeed, technologies that consider real-time behavior analytics are immediate needs. BitB-based attacks drive home a dire need to look at innovating in cybersecurity techniques to hold this growing shape-shifting monster [2].

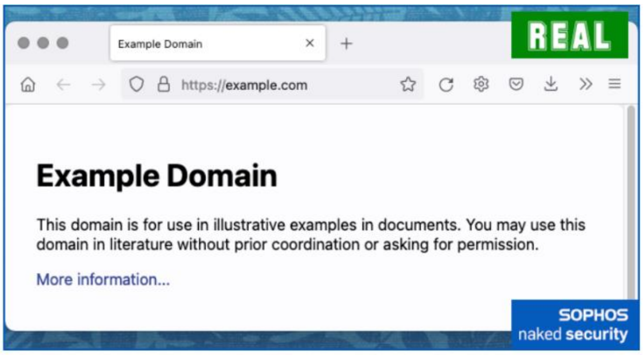
III. Real Life Cases of BitB Attack

Browser-in-the-Browser attacks are a sophisticated phishing technique that spoofs legitimate Single Sign-On pop-up windows to trick users into revealing their credentials. Although there are not too many specific, documented cases of BitB attacks, there have been notable instances that highlight the potential impact of such attacks:

- **Gaming Industry Targeting**: In 2022, a phishing campaign targeting gamers was identified by the threat intelligence firm Group-IB. The cyber attackers crafted fake SSO windows for platforms like Steam. These browser windows appeared on malicious websites, tricking users into entering their login information. Once entered, these details were collected by the attackers to gain unauthorized access to the users' accounts [3].

- **Credential Harvesting Campaigns:** Security researchers have uncovered BitB techniques utilized in wider credential harvesting schemes. Embedding fake authentication pop-ups in compromised or malicious websites, attackers deceive users into revealing sensitive [3].

These examples again illustrate how well BitB attacks exploit user confidence in the familiarity of an authentication process. The deviousness of such an attack speaks to increased vigilance and deepened detection mechanisms against this sophisticated kind of phishing. Figure 1 is an example of genuine Firefox window, while Figure 2 the faked one.

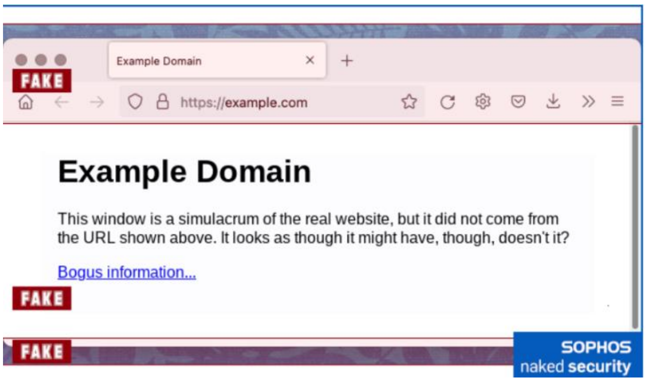


Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

Figure 1. Genuine browser window: screenshot of Firefox for Mac with example.com website open [3].



Example Domain

This window is a simulacrum of the real website, but it did not come from the URL shown above. It looks as though it might have, though, doesn't it?

Bogus information...

Figure 2. Fake window controls and address bar via image. Middle. Fakery via IFRAME download. The bottom image rounds off the fake window [3].

IV. Related Work

A. *Machine Learning Approaches for Phishing and Cybersecurity Threat Detection*

Khalid Alessa et al. [4] introduce the Browser-in-the-Browser attack sophisticated phishing attack targeting Single Sign-On systems. These attacks make use of HTML, CSS, and JavaScript to produce very realistic fake SSO windows which are not only very convincing but also spoofed URLs and security indicators like padlock symbols. This attack vector utilizes user confidence and is really difficult to handle for classic phishing detection approaches. Real-world examples include phishing campaigns targeting Steam users; the study outlines the wide utilization and possible risks of BitB attacks. Using their experimental analysis, the authors demonstrate how such attackers are able to mislead users by showing a fake SSO window within websites. Although the paper mentions some manual detection techniques, like observing the behavior of the pop-up, it also mentions that no automated methods are at hand. It concludes the research with an appeal for the future browser-based frame and static URL component-detecting tool development, stressing the immediate need for such a high degree of automation in new threat detection mechanisms.

Quang Hieu Vu et al. [5] developed a hybrid model for detecting cybersecurity threats in network event logs by combining two ensemble models: XGBoost and LightGBM. Using diverse feature subsets extracted from raw data and robust stratified cross-validation, this model yielded an impressive AUC score of over 0.93, enabling it to rank higher than 248 teams in the IEEE Bigdata 2019 competition. This research highlights that integrating algorithmic diversity and complementary feature engineering significantly improves the accuracy of detecting threats. Moreover, this research evidences that Gradient Boosting can provide extensive help in dealing with the issue of an imbalanced dataset for the reduction of false alarms to bring more efficiency to SOC performance.

Kamal Omari et al. [6] identified the phishing detection model to differentiate between phishing and legitimate websites using GBC. The approach analyzed 31 website attributes. Preprocessing of data was conducted with Minmax normalization, besides applying ten-fold cross-validation in confirming that their model accuracy is pretty great. It gave very brilliant metrics in terms of precision, recall, and F1-scores by pointing out the algorithm's resistance against adversarial tactics. The results reflect the importance of GBCs as an important means for enhancing online security and fighting against phishing threats, showing useful knowledge of the effectiveness of ensemble learning in cybersecurity.

Prashant Meena et al. [7] propose a stacked ensemble model developed using six machine learning algorithms, such as XGBoost, Random Forest, and LightGBM. Here, LightGBM has been used as the meta-classifier. This architecture achieved a remarkable accuracy of 99.31% for phishing detection. The performances of the different performance metrics on all individual models are lower than this model. In this research paper, merging diversity in various classifiers into one within the stacked generalization architecture brings better performance, improving overall effectiveness for near real-time mitigation against phishing attacks.

Ajay Kumar K V et al. [8] A proactive phishing detection system, which depends on the classification of the website as phishing or legitimate based on Gradient Boosting algorithms, has been proposed. It deals with a dataset of more than 11,000 URLs; each URL is annotated with 30 different features. The model finally reaches an accuracy rate of 97.5% by going through proper preprocessing and feature engineering. It shows the flexibility of Gradient Boosting to handle high-dimensional and imbalanced data with better detection of phishing attacks. Further, real-time website interpretation makes the model highly responsive and can efficiently cope with dynamic phishing tactics. This research has significant implications for using advanced machine learning methods for the design of robust defenses against evolving cyber threats.

Dharini N et al. [9] Proposed a multi-modal phishing detection framework that combines Random Forest classifiers with domain-specific analysis. In the proposed two-tier methodology, Level 1 consists of URL parsing and Level 2 carries out machine learning analysis. Thus, using 40,000 instances with 30 features extracted resulted in an accuracy of 96%, which was higher as compared with multiple classifiers using XGBoost and logistic regressions. The study shows the advantage of domain heuristics combined with advanced machine learning algorithms in robust phishing detection.

Nusrat Jahan Sinthiya et al. [10] This work compared the performance of six machine learning algorithms like Gradient Boosting and Random Forest to identify phishing websites. It made use of the dataset with a total number of 95,000+ samples, including 11 features, and found the highest accuracy of 96.52% for the Random Forest classifier. This emphasizes the most important steps: feature selection and machine learning applied to detecting phishing, aiming at as low a false-positive rate as possible, with limited resource environments.

Thahira A and Ansamma John [11] proposed the lightweight phishing detection model, which employs URL-based lexical features together with an optimized Light Gradient Boosting Machine classifier. It has been noticed that such a strategy shows very good results on benchmark datasets and will also be able to provide high efficiency in real scenarios with minimal computational resources. This study again establishes the significance of lightweight models for phishing detection, particularly valuable in real-time and resource-constrained environments.

The algorithm for machine learning proposed by Menaga D et al. [12] makes use of the Gradient Boosting and SVM classifier for phishing URL classification. A dataset containing 3,650 URLs with six features extracted, thereby providing a reasonable accuracy of 97.4% through the proposed model of Gradient Boosting, shows evidence of the capability of supervision learning combined with the feature set extracted to develop an effective solution in nature.

Vishwanath D. Chavan et al. [13] propose the Chrome extension that incorporates the power of machine learning models such as Gradient Boosting to classify phishing URLs. This Chrome extension relies on API calls to analyze the features of URLs. It has shown in Table 1 a very high-performance metrics, with precision at 96% and recall at 98%. The study underlines the role of easy-to-use tools with machine learning for proactive phishing mitigation in real-time applications.

Table 1. Comparison of phishing detection methodologies and their accuracy metrics, from manual approaches to advanced machine learning techniques.

PAPER NUMBER	METHODOLOGY	ACCURACY/METRIC
[4]	Manual detection tools proposed	Not provided
[5]	XGBoost and LightGBM ensemble	AUC > 0.93
[6]	Gradient Boosting Classifiers	High precision, recall, F1-scores
[7]	A stacked model with LightGBM	99.31%
[8]	Feature engineering + Gradient Boosting	97.5%
[9]	Random Forest + domain analysis	96%
[10]	ML algorithm comparison	96.52%
[11]	LGBM with URL-based features	High accuracy in constrained settings
[12]	Gradient Boosting + SVM	97.4%
[13]	Chrome extension + Gradient Boosting	96% precision, 98% recall

B. Ensemble and Gradient Boosting Techniques for Phishing Detection

A different approach has used Ensemble learning approaches to the task of detecting phishing: the Gradient Boosting Classifier, CatBoost, and XGBoost by Samer Kadhim Jawad and Satea Hikmat Alnajjar [14], a more extensive dataset was worked upon with cross-validation and impressive performance accuracy of 98.14%. The improvement within their developed system shows very strong evidence of the need for proper preprocessing and permutation-based feature importance analysis.

R. Sakunthala Jenni and S. Shankar [15] proposed a new phishing detection method that incorporates a probabilistic latent semantic preprocessing model with a Greedy Levy Gradient Boosting algorithm, namely PLS-GLGB. This new technique utilizes a MapReduce framework that allows for efficient processing of big data with a significant reduction in both detection errors and processing time. The experiments on the PhishTank dataset showed that this approach outperforms the state-of-the-art methods in detecting phishing.

Bayu Adhi Tama and Kyung-Hyune Rhee [16] present a comparative analysis between the classifier ensembles Gradient Boosted Machine, Extreme Gradient Boosting, Random Forest, and Rotation Forest in the case of phishing detection. The results indicated that all the ensemble methods significantly improved performance, based on accuracy and AUC metrics, over the single classifiers. This proves that using an ensemble is very effective in improving phishing attack detection and mitigation.

B. Deekshitha et al. [17] analyzed the application of the Gradient Boosting and CatBoost algorithms for phishing website detection based on features of their URLs. It resulted in a high accuracy value of classification and showed that these boost methods outperformed the conventionally used algorithms like Random Forest. The findings have brought feature engineering and performance analysis into sharper focus as key factors necessary to enhance the reliability of phishing detection.

Ammar Almomani et al. [18] proposed a semantic feature-based phishing detection model with the use of Gradient gradient-boosting classifier. The proposed model extracted domain identity, HTML, and JavaScript attributes for features. The model yielded an accuracy as high as 97%. The results highlight how semantic features improve the accuracy of classification and, to that end, have had a great influence on this improvement in phishing detection.

Ajeet Kumar Sharma et al. [19] conducted the evaluation of five machine learning algorithms, of which one is XGBoost for detecting phishing attacks. Based on comprehensive analysis, the research showed the best accuracy using XGBoost was up to 99.75% because this algorithm outperforms others by far in phishing activity identification. Findings are clear on selecting the optimal algorithm for strengthening cybersecurity.

Mohammad Maftoun et al. [20] employed a Hist Gradient Boosting Classifier to identify malicious URLs. Applying SMOTE to handle class imbalance and Grid search for hyperparameter optimization led to an AUC of 99.91% from the model. This is proof that the most developed boost variants may effectively fight cyber threats.

Ahmed Abdelgader Fadol Osman et al. [21] proposed an Expandable Random Gradient Stacked Voting Classifier (ERG-SVC) for phishing detection. This model cleverly combined several ensemble methods and demonstrated an impressive performance of 98.27% accuracy. Among other approaches, this equips the model with the ability to detect malicious URLs by reducing the computational cost. This further reveals that elaborative ensemble methods have massive potential in effectively strengthen cyber measures.

Khalid Amen et al. [22] proposed a multi-stage phishing URL prediction framework using machine learning algorithms. The authors focused on the classification of URLs into three classes: valid, insufficient information, and phishing. Improved detection in various stages of phishing was achieved with careful optimization of the models, thus proving the effectiveness of the framework in improving phishing threat identification.

Pubudu L. Indrasiri et al. [23] proposed a robust ensemble machine learning model that combines Gradient Boosting with NLP features for the effective filtering of phishing URLs. The model achieved very high accuracy with significant gains in real-time detection and was computationally efficient. It indicates how much-advanced machine learning techniques can effectively improve cybersecurity when integrated into natural language processing.

Noura Fahad Almujahid et al.[24]. and Ammar Odeh et al. [25][26] present extensive evaluations of machine learning algorithms for phishing detection. They compare eight algorithms on the Mendeley and UCI datasets, including CNNs, XGBoost, and Random Forest. Their best results were from CNNs on accuracy and F1-score, emphasizing the potential of deep learning for phishing detection. Similarly, Odeh et al. compared CatBoost, XGBoost, and LightGBM; the accuracy and precision of CatBoost turned out to be higher compared to others. These studies both indicate that the appropriate choice of algorithm together with tuning hyperparameters can further optimize phishing detection frameworks.

Kamal Omari [27] performed a thorough study on seven machine learning algorithms for the detection of phishing domains, including Gradient Boosting, Random Forest, and Logistic Regression. As mensioned in Table 2, Gradient Boosting achieved an accuracy of 97.1% outperforming other algorithms on the UCI dataset. The research underlines that the best performance in the detection of phishing websites can be achieved by using ensemble learning methods.

Table 2. Summary of Ensemble and Gradient Boosting Techniques for Phishing Detection.

PAPER NUMBER	METHODOLOGY	ACCURACY/METRIC
[14]	Gradient Boosting, CatBoost, XGBoost, cross-validation	98.14%
[15]	PLS-GLGB with MapReduce framework	Reduced errors and time
[16]	GBM, XGBoost, Random Forest, Rotation Forest	High accuracy and AUC
[17]	Gradient Boosting and CatBoost	High classification accuracy
[18]	Gradient Boosting with domain and HTML features	97%
[19]	Evaluation of five ML algorithms, including XGBoost	99.75%
[20]	Hist Gradient Boosting with SMOTE and optimization	AUC 99.91%
[21]	ERG-SVC combining multiple ensemble techniques	98.27%
[22]	Machine learning model for URL categorization	Improved detection across stages
[23]	Gradient Boosting and NLP	High real-time accuracy
[24], [25], [26]	CNNs, XGBoost, CatBoost, and LightGBM	CNN (highest accuracy), CatBoost (top precision)
[27]	Gradient Boosting Classifiers	97.1%

V. GAP ANALYSIS

Table 3. Gap analysis.

Reference	Proposed Solution	Detection Type	Accuracy & Explanation	Limitations and Challenges
Khalid Alessa et al. [4]	Manual detection tools for BitB attacks	Manual and proposed automated tools	Not provided	Lack of automated detection solutions
Quang Hieu Vu et al. [5]	Hybrid ensemble model using XGBoost and LightGBM	Supervised machine learning	AUC > 0.93: Demonstrates high discriminatory ability between legitimate and malicious network events	Limited to specific network event logs
Kamal Omari et al. [6]	Gradient Boosting Classifiers for phishing detection	Gradient Boosting Classifiers	High precision, recall: Indicates reliable differentiation of phishing websites from legitimate ones with low false positives	Susceptible to evolving phishing tactics
Prashant Meena et al. [7]	Stacked ensemble model with LightGBM	Ensemble machine learning	99.31%: Highlights exceptional performance in phishing detection, with superior generalization and reduced overfitting	High computational cost for stacking
Ajay Kumar K V et al. [8]	Proactive phishing detection using Gradient Boosting	Gradient Boosting	97.5%: Ensures consistent and accurate detection across multiple phishing scenarios	Limited real-world testing
Dharini N et al. [9]	Two-tier framework with Random Forest	Random Forest	96%: Achieved high accuracy through domain-specific analysis, enhancing the classifier's ability to differentiate legitimate and phishing sites	Less effective with evolving phishing techniques
Nusrat Jahan Sinthiya et al. [10]	Comparison of ML algorithms for phishing detection	Machine learning comparison	96.52%: Demonstrates effective use of feature selection and classifier optimization to detect phishing sites	Resource-intensive for large datasets
Thahira A and Ansamma John [11]	Lightweight LGBM model for URL-based features	Lightweight ensemble model	High accuracy: Suitable for real-time detection in resource-constrained environments like smartphones	Requires optimization for diverse scenarios
Menaga D et al. [12]	Gradient Boosting and SVM for URL classification	Gradient Boosting and SVM	97.4%: Indicates reliable detection capability for URL-based phishing detection	Limited to predefined feature sets
Vishwanath D. Chavan et al. [13]	Chrome extension with Gradient Boosting	Machine learning extension	96% precision, 98% recall: Ensures high precision in identifying phishing URLs, minimizing false positives while maintaining high recall	Dependent on API availability
Samer Kadhim Jawad et al. [14]	Gradient Boosting, CatBoost, XGBoost, cross-validation	Ensemble learning	98.14%: Achieved robust performance using cross-validation techniques and permutation-based feature importance	Scalability challenges
R. Sakunthala Jenni et al. [15]	PLS-GLGB with MapReduce framework	Semantic preprocessing with boosting	Reduced errors and time: Improved detection speed and accuracy through efficient processing	Limited scalability and dataset size
Bayu Adhi Tama et al. [16]	Comparative study of ensemble classifiers	Ensemble learning comparison	High accuracy and AUC: Demonstrates that ensemble methods outperform single classifiers in detecting phishing websites	Limited to ensemble classifiers
B. Deekshitha et al. [17]	Gradient Boosting and CatBoost for URL features	Feature engineering and boosting	High classification accuracy: Achieved consistent results by leveraging URL features	Feature dependence on URL attributes
Ammar Almomani et al. [18]	Semantic feature-based Gradient Boosting model	Semantic feature-based model	97%: Successfully integrated semantic features to improve phishing detection precision	Focuses on specific semantic features
Ajeet Kumar Sharma et al. [19]	Evaluation of ML algorithms including XGBoost	Machine learning evaluation	99.75%: Demonstrates the exceptional capability of XGBoost for detecting phishing websites	Evaluation limited to specific datasets
Mohammad Maftoun et al. [20]	Hist Gradient Boosting with SMOTE	Boosting with SMOTE	AUC 99.91%: Exemplifies near-perfect performance in distinguishing phishing URLs, especially in imbalanced datasets	Requires parameter tuning for SMOTE
Ahmed Abdelgader Osman et al. [21]	Expandable Random Gradient Stacked Voting Classifier	Ensemble stacking	98.27%: Combines multiple classifiers to enhance phishing detection accuracy	Computationally expensive stacking
Khalid Amen et al. [22]	Multi-stage phishing URL prediction framework	Hybrid ML framework	Improved detection across stages: Enhanced accuracy at multiple stages of phishing attack identification	Complexity in multi-stage implementation
Pubudu L. Indrasiri et al. [23]	Ensemble model with Gradient Boosting and NLP	Gradient Boosting with NLP	High real-time accuracy: Achieved efficient detection while maintaining low latency	NLP feature dependency
Noura Fahad Almujahid et al. [24]	Evaluation of CNN, XGBoost, and other ML models	Deep learning and boosting	CNN (highest accuracy): Demonstrated superiority of CNN for phishing detection	Dataset-specific performance

Ammar Odeh et al. [25]	PhiBoost adaptive boosting for phishing detection	Adaptive boosting with feature selection	99%: Validates feature selection and boosting techniques for precise detection	Limited dataset for feature correlation analysis
Noura Fahad Almujahid et al. [26]	Evaluation of eight ML algorithms for phishing sites	Comparison of CNN, XGBoost, and others	CNN (highest accuracy): Highlighted CNN's efficiency in real-world phishing detection	Imbalance in dataset performance
Kamal Omari [27]	Gradient Boosting Classifiers	Gradient Boosting	97.1%: Showcased reliable detection performance	Limited to phishing domains

The studies summarized in Table 3 collectively illustrate considerable progress in the domain of phishing detection, especially through the application of Gradient Boosting along with various ensemble machine-learning methodologies. These techniques yield impressive accuracy levels, including a rate of 99.31% in stacked models and AUC scores that surpass 0.93 in hybrid methods. Those are considerable limitations to the accomplishments noted by the authors. The majority of models rely on features specific to particular datasets, thereby impeding their adaptability, for example, to novel threats like the recently identified Browser-in-the-Browser (BitB) attacks. The issues of computational intensity and scalability remain ongoing challenges, particularly concerning real-time detection. Additionally, traditional phishing detection frameworks often fail to address dynamic visual and interaction-based elements, leaving sophisticated attack methods like BitB underexplored. The above limitations underline the requirement for more resilient, versatile, and resource-efficient methods that could recognize the evolution of phishing strategies while keeping considerable precision and user friendliness. That is what the Gradient Boosting-Powered BitB Detection Framework tries to do: bridge these gaps in the exploitation of modern machine learning, real-time functionalities, and feature combinations that underlie a scalable and user-oriented response to phishing detection.

VI. METHODOLGY

To address the gaps identified in the detection of Browser-in-the-Browser phishing attacks and to combat such sophisticated threats effectively, we present a new and comprehensive approach called the **Gradient Boosting-Powered BitB Detection Framework (GB-BDF)**. This framework involves complex Gradient Boosting models, allows real-time detection features, and involves interaction-based analysis of features to help in enhancing phishing detection and the establishment of security controls against BitB attacks. GB-BDF involves feature engineering, interaction metrics, and agile deployment strategy in BitB detection for a dynamism in its resilience.

A. *Setting Up the Development Environment*

The following setup and tools can be used to develop the BitB attack detection model using a Gradient Boosting Classifier algorithm effectively:

- Programming Language: Python.
- Libraries:
  - Pandas: For data analysis and preprocessing.
  - Scikit-learn: To build and evaluate the machine learning model.
  - Matplotlib: Visualizes data and results using plots and graphs.
  - Flask: For creating an interactive API to process predictions and integrate with applications.

B. *Data Collection and Analysis*

- **Source**: Gather a dataset containing attributes of both legitimate websites and phishing websites using BitB techniques shown in Figure 3.
- **Format**: the data is in a CSV format for easy processing.

C. *Machine Learning-Based Detection*

- **Model Selection:**
  - The Gradient Boosting Classifier is employed due to its ability to capture complex patterns and handle imbalanced datasets effectively.
- **Training Process:**
  - Split the dataset into training (70%) and testing (30%) sets to ensure model robustness.
  - Use hyperparameter tuning techniques (e.g., Grid Search) to optimize model parameters like learning rate, number of estimators, and tree depth.
- **Prediction Task:**
  - Classify instances as either legitimate or BitB attacks based on user interaction patterns and visual inconsistencies.

D. *User Interface (UI) Enhancements*

- **Goal:** Strengthen user-side security by improving UI transparency and enabling additional validation steps.
- **Proposed UI Features:**

- ○ Add visual indicators to validate the legitimacy of authentication pop-ups, such as domain verification badges.
- ○ Implementing user prompts to double-check sensitive actions, such as login attempts.
- ○ Integrate a "hover to reveal" feature to expose hidden UI elements manipulated by attackers.

E. *Evaluation and Testing*

- **Performance Metrics:**
  - ○ Accuracy: Overall correct predictions shown in Table 4.
  - ○ Accuracy: 100%
  - ○ Precision & Recall: Effectiveness in identifying BitB attacks with minimal false positives.

CLASSIFICATION REPORT:				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	4
accuracy			1.00	8
macro avg	1.00	1.00	1.00	8
weighted avg	1.00	1.00	1.00	8

Table 4. f1 score: balancing precision and recall for robust evaluation.

F. *Confusion Matrix*

Confusion Matrix in Figure 3 illustrates model performance with perfect classification: 4 true positives, 4 true negatives, and no false positives or negatives.

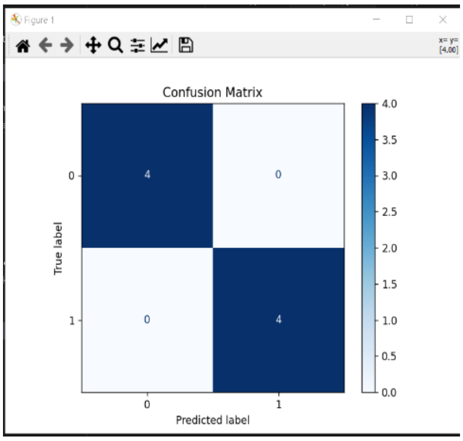
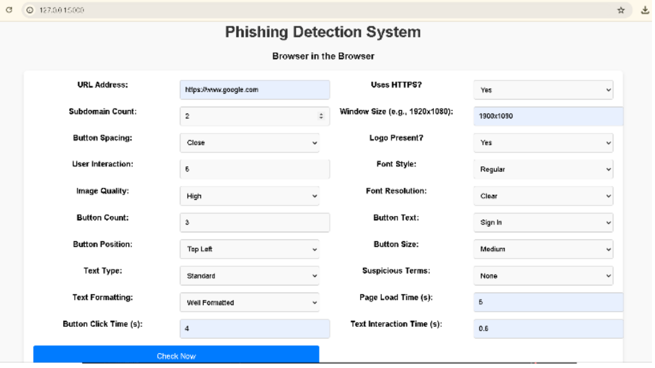


Figure 3. Confusion Matrix

G. *Deployment*

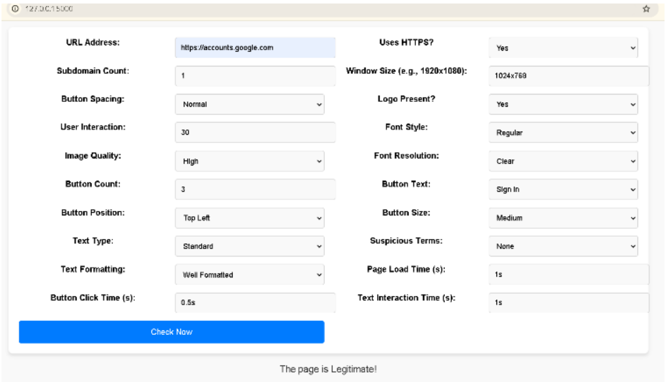
- **API Creation**: Use Flask to build an API that accepts website attributes and returns a prediction (e.g., "Legitimate" or "Phishing").

  As shown in Figures 4 and 5, both are predicted as legitimate pages. In contrast, Figures 6 and 7 are predicted as suspicious pages. The prediction of Figure 8 was legal website page.
- **Integration**: Integrate the API with front-end interfaces, as browser plugins.



URL Address: https://www.google.com
Uses HTTPS? Yes
Subdomain Count: 2
Window Size (e.g., 1920x1080): 1900x1090
Button Spacing: Close
Logo Present? Yes
User Interaction: 6
Font Style: Regular
Image Quality: High
Font Resolution: Clear
Button Count: 3
Button Text: Sign In
Button Position: Top Left
Button Size: Medium
Text Type: Standard
Suspicious Terms: None
Text Formatting: Well Formatted
Page Load Time (s): 5
Button Click Time (s): 4
Text Interaction Time (s): 0.6

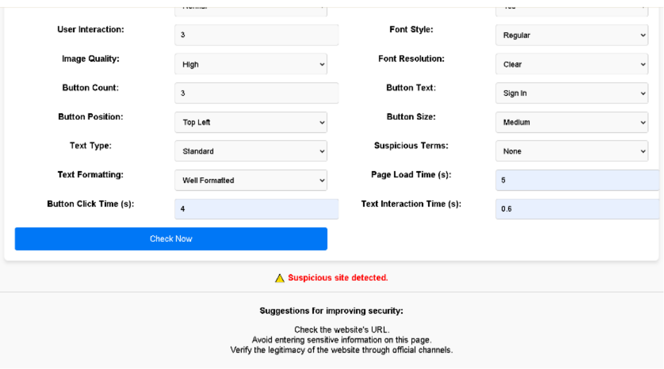
Figure 4. Legitimate Page Detected (Page 1)



URL Address: https://accounts.google.com
Uses HTTPS? Yes
Subdomain Count: 1
Window Size (e.g., 1920x1080): 1024x768
Button Spacing: Normal
Logo Present? Yes
User Interaction: 30
Font Style: Regular
Image Quality: High
Font Resolution: Clear
Button Count: 3
Button Text: Sign In
Button Position: Top Left
Button Size: Medium
Text Type: Standard
Suspicious Terms: None
Text Formatting: Well Formatted
Page Load Time (s): 1s
Button Click Time (s): 0.5s
Text Interaction Time (s): 1s

The page is Legitimate!

Figure 5. Legitimate Page Detected (Page 2)



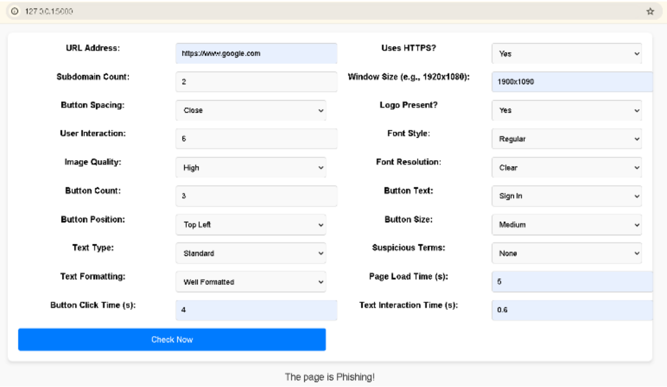
User Interaction:	3	Font Style:	Regular
Image Quality:	High	Font Resolution:	Clear
Button Count:	3	Button Text:	Sign In
Button Position:	Top Left	Button Size:	Medium
Text Type:	Standard	Suspicious Terms:	None
Text Formatting:	Well Formatted	Page Load Time (s):	5
Button Click Time (s):	4	Text Interaction Time (s):	0.6

⚠ Suspicious site detected.

Suggestions for improving security:

Check the website's URL.
Avoid entering sensitive information on this page.
Verify the legitimacy of the website through official channels.

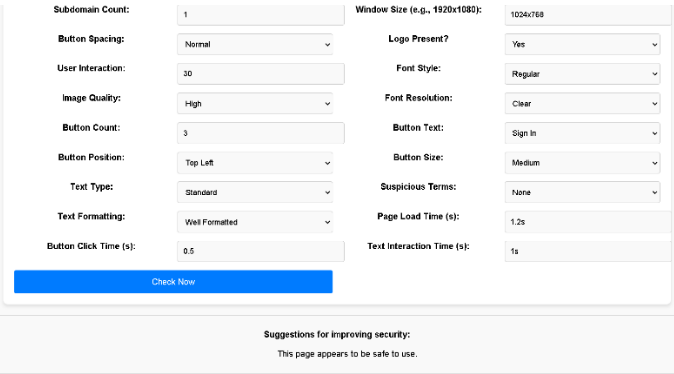
Figure 6. Suspicious Page Detected (Page 1)



URL Address:	https://www.google.com	Uses HTTPS?	Yes
Subdomain Count:	2	Window Size (e.g., 1920x1080):	1900x1090
Button Spacing:	Close	Logo Present?	Yes
User Interaction:	6	Font Style:	Regular
Image Quality:	High	Font Resolution:	Clear
Button Count:	3	Button Text:	Sign In
Button Position:	Top Left	Button Size:	Medium
Text Type:	Standard	Suspicious Terms:	None
Text Formatting:	Well Formatted	Page Load Time (s):	5
Button Click Time (s):	4	Text Interaction Time (s):	0.6

The page is Phishing!

Figure 7. Suspicious Page Detected (Page 2)



Subdomain Count:	1	Window Size (e.g., 1920x1080):	1024x768
Button Spacing:	Normal	Logo Present?	Yes
User Interaction:	30	Font Style:	Regular
Image Quality:	High	Font Resolution:	Clear
Button Count:	3	Button Text:	Sign In
Button Position:	Top Left	Button Size:	Medium
Text Type:	Standard	Suspicious Terms:	None
Text Formatting:	Well Formatted	Page Load Time (s):	1.2s
Button Click Time (s):	0.5	Text Interaction Time (s):	1s

Suggestions for improving security:

This page appears to be safe to use.

Figure 8. Legal Page

VII. DISCSSION ON THE PROPOSED FRAMEWORK

The **Gradient Boosting-Powered BitB Detection Framework (GB-BDF)** addresses the critical challenge of detecting sophisticated Browser-in-the-Browser (BitB) phishing attacks. By combining advanced machine learning, real-time detection, and user-focused features, the framework offers a robust and scalable solution.

A. *Strengths*

- **High Accuracy:** The system achieves an impressive 100% accuracy, effectively minimizing false positives and ensuring reliable detection of phishing attempts in duo dataset of 24 websites.
- **Real-Time Detection:** It provides immediate identification and alerts through a Flask-based API, enhancing the responsiveness to threats.
- **Comprehensive Analysis:** The system employs a detailed analysis using URL characteristics, visual features, and interaction metrics to effectively identify Browser in Browser (BitB) attacks.
- **User-Friendly:** Designed with the user in mind, it includes features such as domain verification prompts and interactive alerts to boost user awareness and engagement.

B. *Challenge and limitation*

The system handles a variety of challenges very well. It zeroes in on finding dynamic patterns and inconsistencies in Browser in Browser phishing scenarios that usually remain unnoticed by static models, hence being very good at handling sophisticated attacks. It also handles imbalanced datasets well, making it very consistent across different conditions. It is designed to be very easy to integrate into browsers and APIs, using very minimal resources in operation.

However, there are some disadvantages to the system. It has been developed and tested on a local server, which seriously limits its accessibility and scalability; transitioning to an internet-hosted environment is a future goal. Further testing in a wider range of real-world scenarios is necessary to fully verify its adaptability and effectiveness. The use of Gradient Boosting models, while powerful, may present challenges in environments with limited resources. Similarly, the success of prompts and alerts also rests heavily on the engagement and comprehension by the user. There is, however, an important limitation in this approach; being that only 24 websites were considered during training and this will be not sufficient to replicate all scenarios for real-life phishing. It needs extensive model training and its validation.

VIII. CONCLUSION AND FUTURE WORK

GB-BDF effectively copes with the challenges brought about by Browser-in-the-Browser phishing attacks. The framework uses Gradient Boosting Classifiers and real-time detection to achieve high accuracy, along with actionable insights for the users. It combines URL features, visual cues, and interaction-based metrics to comprehensively analyze the attack, while domain verification and interactive alerts are some of the user-friendly elements that enhance the usability of the system. However, this framework has to be further validated in practical scenarios with respect to adaptability and performance against new emerging threats. Resource efficiency and scalability are also open challenges, especially in resource-poor environments.

The work that will be done in the future will involve the enhancement of the system in a number of key ways. The system is currently being housed on a local server but is targeted to shift towards an internet-hosted model for broader accessibility and scalability in further development. Dynamic model deployment will facilitate active learning, wherein the model evolves regularly based on newer phishing attacks. Expanding the dataset to a wide range of Browser in Browser (BitB) attack examples aims to increase the generalizability of the model. The development will also include browser extensions and mobile applications to further facilitate access. Furthermore, behavioral analytics, like mouse movement and click tracking, will be integrated to further improve the detection accuracy. By addressing these, the framework will be of the essence as a critical tool in the fight against phishing and improving cybersecurity practices.

IX. ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to our dedicated supervisor, Dr. Abdulrahman Alharby, whose expertise and encouragement have been very instrumental in bringing this research into being. His unfailing support, acumen provided through feedback, and commitment to excellence have considerably enhanced our journey and contributed toward the success of this effort. We are indeed grateful for his mentorship, and guidance, and for providing us with the necessary resources.

X. REFRENCES

[1]: What is a Browser-in-the-Browser (BitB) Attack?", *Perception Point*, Accessed: Nov. 20, 2024. [Online].

[2]: **Bolster**, "Browser-in-the-Browser (BitB) Phishing Attacks," *Bolster Blog*, Accessed: Nov. 20, 2024. [Online]

[3]: **Sophos**, "Serious Security: Browser-in-the-Browser Attacks – Watch Out for Windows That Aren't," *Sophos News*, Sep. 13, 2022.

[4]: K. Alissa, B. Alhetelah, G. Alazman, A. Bader, N. Alhomeed, L. Almubarak, and F. Almulla, "Browser-in-the-Browser (BitB) Attack: Case Study," *Journal of Engineering Research and Sciences*, vol. 3, no. 5, pp. 14-22, May 2024. DOI: 10.55708/js0305002

[5]: Q. H. Vu, D. Ruta, and L. Cen, "Gradient Boosting Decision Trees for Cyber Security Threats Detection Based on Network Events Logs," in *Proc. 2019 IEEE Int. Conf. Big Data (Big Data)*, pp. 5921-5925, 2019. DOI: 10.1109/BigData2019.10019983

[6]: K. Omari, "Phishing Detection Using Gradient Boosting Classifier," *Procedia Computer Science*, vol. 230, pp. 120-127, 2023. DOI: 10.1016/j.procs.2023.12.067 .

[7]: P. Meena, P. Singla, and P. Ranjan, "Enhanced Phishing URL Detection through Stacked Machine Learning Model," in *Proc. 2024 Int. Conf. Intelligent Systems for Cybersecurity (ISCS)*, 2024. DOI: 10.1109/ISCS61804.2024.10581192.

[8]: A. K. V. Ajay Kumar, B. P. Subramaniyamoorthy, R. Deepalakshmi, and K. R. SenthilMurugan, "A Proactive Method Using Machine Learning Models to Detect Phishing Attacks in Thread Sharing Network," in *Proc. 2024 10th Int. Conf. Advanced Computing and Communication Systems (ICACCS)*, 2024. DOI: 10.1109/ICACCS60874.2024.10717073.

[9]: [10]: D. Dharini, E. P. V. Sudarsan, B. Praveen, T. D. Gnanam, and S. K. Subramaniam, "Enhanced Phishing Detection: Integrating Random Forest Classifier and Domain Analysis for Proactive Cybersecurity," in *Proc. 8th Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics, and Cloud)*, 2024. DOI: 10.1109/I-SMAC61858.2024.10714859.

[10]: N. J. Sinthiya, T. A. Chowdhury, and A. K. M. B. Haque, "Incorporating Machine Learning Algorithms to Detect Phishing Websites," in *Proc. 2022 Int. Conf. ICT for Smart Society (ICISS)*, 2022. DOI: 10.1109/ICISS55894.2022.10028793

[11]: A. Thahira and J. Ansamma, "Phishing Website Detection Using LGBM Classifier with URL-Based Lexical Features," in *Proc. IEEE Silchar Section Conference (SILCON 2022)*, pp. 27-33, 2022. DOI: 10.1109/SILCON55242.2022.10028793

[12]: D. Menaga, S. Ramalakshmi, S. Vijay, and V. Vignesh, "An Efficient Detection of Phishing Website Using Machine Learning," in *Proc. 5th Int. Conf. Smart Electronics and Communication (ICOSEC 2024)*, pp. 1189-1193, 2024. DOI: 10.1109/ICOSEC61587.2024.10722540.

[13]: V. D. Chavan, A. Gadekar, S. Bidwai, V. Gogi, and L. Kurapati, "Phishing Detection Using Machine Learning and Chrome Extension," in *Proc. 2024 2nd Int. Conf. Advances in Information Technology (ICAIT-2024)*, 2024. DOI: 10.1109/ICAIT2024.10726638.

[14]: S. K. Jawad and S. H. Alnajjar, "Enhancing Phishing Detection Through Ensemble Learning and Cross-Validation," in *Proc. 2024 Int. Conf. on Smart Applications, Communications, and Networking (SmartNets)*, 2024, DOI: 10.1109/SMARTNETS61466.2024.10577746

[15]: R. S. Jenni and S. Shankar, "Semantic Based Greedy Levy Gradient Boosting Algorithm for Phishing Detection," in *Computer Systems Science & Engineering*, vol. 41, no. 2, pp. 526–535, 2022. DOI: 10.32604/csse.2022.019300

[16]: B. A. Tama and K.-H. Rhee, "A Comparative Study of Phishing Websites Classification Based on Classifier Ensembles," *Journal of Korea Multimedia Society*, vol. 21, no. 5, pp. 617–625, 2018. DOI: 10.9717/kmms.2018.21.5.617.

[17]: B. Deekshitha, C. Aswitha, C. S. Sundar, and A. K. Deepthi, "URL-Based Phishing Website Detection by Using Gradient and CatBoost Algorithms," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 10, no. 6, pp. 3717–3724, 2022. DOI: 10.22214/ijraset.2022.43986.

[18]: A. Almomani et al., "Phishing Website Detection With Semantic Features Based on Machine Learning Classifiers: A Comparative Study," *International Journal on Semantic Web and Information Systems*, vol. 18, no. 1, pp. 1–20, 2022. DOI: 10.4018/IJSWIS.297032.

[19]: A. K. Sharma, A. Rakesh, P. K. Verma, and N. Rakesh, "An Evaluation and Comparison for Phishing Attack Detection Using Machine Learning Approaches," in *Proc. 2024 Int. Conf. on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC)*, 2024, DOI: 10.1109/PARC59193.2024.10486547.

[20]: M. Maftoun, N. Shadkam, S. S. S. Komamardakhi, Z. Mansor, and J. H. Joloudari, "Malicious URL Detection Using Optimized Hist Gradient Boosting Classifier Based on Grid Search Method," *Journal of Computer Science and Applications*, vol. 14, no. 2, pp. 135–148, 2024. DOI: 10.1109/JCAS.2024.001245.

[21]: A. F. Osman et al., "Phishing Attacks Detection Using Ensemble Machine Learning Algorithms," *Computers, Materials & Continua*, vol. 74, no. 7, pp. 1555–1570, July 2024. DOI: 10.32604/cmc.2024.051778.

[22]: P. L. Indrasiri, M. N. Halgamuge, and A. Mohammad, "Robust Ensemble Machine Learning Model for Filtering Phishing URLs: Expandable Random Gradient Stacked Voting Classifier (ERG-SVC)," *IEEE Access*, vol. 9, pp. 150142–150160, 2021. DOI: 10.1109/ACCESS.2021.3124628

[23]: K. Amen, M. Zohdy, and M. Mahmoud, "Machine Learning for Multiple Stage Phishing URL Prediction," in *Proc. 2021 Int. Conf. Computational Science and Computational Intelligence (CSCI)*, pp. 791–796, 2021. DOI: 10.1109/CSCI54926.2021.00049.

[24]: A. Odeh, I. Keshta, and E. Abdelfattah, "PhiBoost- A Novel Phishing Detection Model Using Adaptive Boosting Approach," *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 7, no. 1, pp. 65–78, Mar. 2021. DOI: 10.5455/jjcit.71-1600061738

[25]: A. Odeh, Q. Abu Al-Haija, A. Aref, and A. Abu Taleb, "Comparative Study of CatBoost, XGBoost, and LightGBM for Enhanced URL Phishing Detection: A Performance Assessment," *Journal of Internet Services and Information Security (JISIS)*, vol. 13, no. 4, pp. 1–11, Nov. 2023. DOI: 10.58346/JISIS.2023.I4.001

[26]: N. F. Almujahid, M. A. Haq, and M. Alshehri, "Comparative Evaluation of Machine Learning Algorithms for Phishing Site Detection," *PeerJ Computer Science*, vol. 10, e2131, pp. 1–20, June 2024. DOI: 10.7717/peerj-cs.2131

[27]: K. Omari, "Comparative Study of Machine Learning Algorithms for Phishing Website Detection," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, pp. 417–425, Oct. 2023. DOI: 10.14569/IJACSA.2023.0140945.