International Journal of Research in Advent Technology, Vol.14, No.1, March 2026
E-ISSN: 2321-9637
Available online at www.ijrat.org

Intelligent Phishing Website Detection Using Machine Learning Techniques

Raghav Garg

Abstract—Phishing has become one of the most prevalent and dangerous cyber threats, deceiving users into revealing confidential information by mimicking legitimate websites. Traditional blacklist-based and rule-based detection systems struggle to keep pace with the constantly evolving nature of phishing attacks. This research aims to develop an intelligent phishing website detection model using machine learning techniques to identify malicious URLs with high accuracy. The proposed system extracts multiple URL-based features—such as domain age, HTTPS usage, URL length, redirection, and prefix/suffix patterns—and uses them to train classifiers including Random Forest, Gradient Boosting, and a Voting Ensemble Model. The dataset, consisting of legitimate and phishing URLs, was preprocessed and split into training and testing sets to evaluate performance. Experimental results show that the ensemble-based model achieved an accuracy of over 90%, outperforming traditional single classifiers and previous static detection methods. The model's efficiency and adaptability make it suitable for real-time integration into browser extensions or web security tools. Overall, this study contributes to the advancement of phishing detection research by providing a robust, scalable, and adaptive solution capable of safeguarding users from emerging cyber threats.

Index Terms—Phishing Detection · Machine Learning · Cybersecurity · URL Features · Browser Extension

I. INTRODUCTION

Phishing is widely recognized as one of the most prevalent and harmful cyber threats in the modern digital environment. It primarily exploits user trust by presenting deceptive websites that closely mimic legitimate platforms, thereby inducing users to disclose confidential information, including login credentials, financial details, and personal data. The rapidly evolving nature of phishing attacks significantly reduces the effectiveness of traditional defense mechanisms, including static blacklists, which are unable to respond promptly to newly emerging threats.

Manuscript received February 12, 2026; revised March 20, 2026 and published on March 30, 2026
Raghav Garg, *Department of Computer Science, Vivekananda Institute of Professional Studies Technical Campus (VIPS-TC), Delhi, India*
Email: 35217711922_ds@vips.edu

As a result, considerable research attention has shifted toward intelligent detection mechanisms based on machine learning, rule-based logic, and hybrid approaches that are capable of operating within browser extensions to provide real-time protection.An early contribution in this domain was made by Marchal et al., who proposed a scalable phishing detection architecture designed to adapt dynamically to novel attack patterns. Their framework integrated machine learning techniques with real-time filtering strategies, enabling the identification of malicious websites during active browsing sessions. The system demonstrated high detection accuracy while maintaining a low false positive rate, an essential requirement for browser extensions to ensure sustained user confidence [1]. Subsequent work by Abdelhamid et al. extended this direction by investigating hybrid machine learning models that combined multiple classifiers. Their experimental results indicated that individual algorithms often exhibit limitations when used in isolation; however, combining diverse classifiers significantly improves overall detection performance. This finding highlighted the effectiveness of ensemble-based methods for enhancing the robustness of browser-level phishing defenses [2]. In a different approach, Aburrous et al. explored the application of fuzzy logic in phishing detection systems. Unlike conventional rule-based techniques, fuzzy logic is capable of handling uncertainty and partial truth, which is particularly relevant given the subtle similarities between phishing and legitimate websites. Their results confirmed that fuzzy-based classification models can improve resilience in real-time detection scenarios, especially in cases where traditional methods struggle to make clear distinctions [3]. Focusing on content-level analysis, Basit et al. examined webpage text and URL-based features to uncover hidden patterns commonly associated with phishing websites. Their study demonstrated that analyzing structural and linguistic characteristics of webpages enables early identification of fraudulent sites, thereby preventing users from interacting with malicious content. This approach supports the deployment of content-aware phishing detection mechanisms within browser extensions [4]. Beyond individual detection techniques, Alkhalil et al. presented a comprehensive survey of existing phishing detection methodologies, emphasizing the role of browser extensions as an effective first line of defense. Their analysis underscored the importance of adaptive learning mechanisms to counter zero-day phishing attacks, where conventional signature-based systems are largely ineffective [5]. Addressing the practical constraints of browser-based deployment, Mohammad et al. proposed a lightweight rule-based framework that evaluated indicators such as SSL certificate properties and webpage behavior. Despite its low computational overhead, the system achieved

doi: 10.32622/ijrat.1401202602

reliable detection performance, demonstrating that efficient and accurate phishing detection can be achieved within browser plugins without compromising usability [6]. Several studies adopted a data-centric perspective. Zhang et al. focused on lexical and host-based features, including domain registration details and hosting attributes, to identify suspicious websites at an early stage. By detecting anomalous domain characteristics, their approach effectively blocked phishing attempts before user interaction, making it well-suited for browser-level integration [7]. Computational efficiency was further examined by Patil et al., who demonstrated that strategic feature selection can significantly reduce processing requirements while maintaining high detection accuracy. Their findings reinforced the feasibility of deploying real-time phishing detection systems in browser extensions with limited computational resources [8]. Critical perspectives have also been presented in the literature. Verma and Das highlighted that although many phishing detection models achieve high accuracy in controlled experimental settings, real-world deployment introduces challenges such as adversarial manipulation, evolving attack strategies, and issues related to user trust. Their analysis emphasized that browser extensions must balance technical effectiveness with transparency and usability to achieve widespread adoption [9]. To address the limitations of standalone approaches, Jain and Gupta proposed a hybrid detection framework that combined blacklist-based filtering with machine learning classification. This layered strategy mitigated the shortcomings of traditional blacklists while leveraging the adaptability of learning-based models, resulting in improved resilience against contemporary phishing techniques [10].
Collectively, these studies demonstrate the growing importance of adaptive phishing detection systems and their practical implementation through browser extensions. The literature consistently indicates that static defenses are insufficient against modern phishing threats. Instead, intelligent, hybrid, and resource-efficient detection mechanisms integrated into user-friendly browser extensions represent a promising direction for enhancing online security.

Phishing websites trick users into revealing sensitive data by imitating legitimate sites. Existing solutions like blacklists and rule-based filters fail against zero-day and evolving attacks, while many machine learning models are too heavy for real-time use. There is a need for a lightweight, adaptive, and accurate phishing detection system that can be integrated into browsers to provide users with effective, real-time protection.

- Phishing remains one of the most widespread and successful cyber threats, tricking users into giving away sensitive data.
- Traditional blacklist and rule-based methods fail to detect zero-day and evolving phishing websites.
- Many existing machine learning models are accurate but too heavy for real-time browser use.
- Users need a lightweight, adaptive, and accurate phishing detection tool that works seamlessly while browsing.
- Developing such a system can reduce identity theft, financial fraud, and increase trust in online transactions.

II. LITERATURE REVIEW

Phishing has consistently remained one of the most enduring and harmful threats in the cybersecurity landscape. As attackers continuously modify their techniques to bypass existing defenses, the research community has responded with a broad range of detection strategies, spanning from simple blacklist mechanisms to sophisticated machine learning, hybrid, and deep learning–based solutions. The progression of the literature reflects a clear shift in priorities: early efforts emphasized scalability and rapid response, whereas more recent studies focus on adaptability, intelligent decision-making, and practical usability. Collectively, this body of work provides a strong foundation for the design of browser-based phishing detection extensions capable of delivering real-time protection.One of the earliest large-scale phishing prevention systems was introduced by Whittaker et al., who analyzed Google's blacklist-driven phishing protection infrastructure deployed across Chrome and other major browsers [11]. Blacklist-based methods offered fast and scalable protection by blocking previously reported malicious URLs. While effective against known threats, their primary limitation was an inability to identify newly created phishing websites, allowing zero-day attacks to remain active until manually reported and added to the blacklist. This shortcoming motivated researchers to investigate more adaptive alternatives. Zhang et al. addressed this limitation by enhancing blacklist systems with lexical and host-based characteristics derived from URLs and domains [12]. Their CANTINA framework combined information retrieval techniques with structural domain features, resulting in improved detection performance compared to blacklist-only approaches. Although blacklist and heuristic methods formed a valuable baseline, these studies underscored the necessity for adaptive models capable of generalizing beyond previously observed phishing instances. As phishing websites became increasingly sophisticated, machine learning techniques emerged as a key solution for identifying subtle and previously unseen attack patterns. Abdelhamid et al. employed a neuro-fuzzy inference system to manage the inherent uncertainty in phishing detection, demonstrating that hybrid intelligent systems can outperform static rule-based approaches [13]. In a related study, Ma et al. evaluated classification models trained exclusively on lexical and host-based URL features [14]. Their results showed that accurate phishing detection could be achieved even without webpage content, enabling lightweight systems suitable for browser integration. Xiang et al. further extended this direction with the development of CANTINA+, which incorporated additional features such as webpage term distributions and search engine–based signals [15]. Although this enhancement led to higher detection accuracy, the increased feature complexity introduced challenges related to computational cost and scalability, particularly in real-time browser environments. Together, these studies highlighted both the promise of machine learning and the importance of balancing accuracy with efficiency for practical deployment.
In parallel, several researchers explored content-based and visual similarity approaches. Liu et al. observed that phishing websites frequently replicate the visual appearance of legitimate platforms and proposed a detection system based

doi: 10.32622/ijrat.1401202602

on visual similarity analysis [16]. While effective in identifying visually deceptive websites, the approach required substantial computational resources, limiting its suitability for lightweight browser extensions. Bergholz et al. investigated phishing detection through statistical learning applied to both email and webpage content, further validating the effectiveness of textual and structural feature analysis [17]. Building on this, Rao and Pais introduced feature selection techniques that reduced computational complexity while preserving classification accuracy [18]. These studies demonstrated that content- and visually driven methods are particularly effective against phishing websites that disguise themselves using legitimate-looking layouts, though efficiency concerns persisted. To mitigate the limitations of single-model approaches, hybrid and ensemble-based techniques gained increasing attention. Jain and Gupta proposed a hybrid framework that integrated blacklist filtering with machine learning classifiers [19]. This layered strategy retained the efficiency of blacklists for known threats while enabling the detection of previously unseen phishing websites. Mohammad et al. presented a lightweight rule-based system that evaluated URL structure, SSL certificate attributes, and webpage behavior [20]. Their model achieved reliable detection with minimal computational overhead, making it well suited for deployment as a browser plugin. Abdelhamid further advanced this line of research by introducing multi-label classification models capable of distinguishing between different phishing attack categories rather than treating phishing as a single homogeneous class [21]. These ensemble-oriented approaches improved detection robustness and enhanced suitability for real-time browser-level protection. More recent research has increasingly focused on advanced machine learning and deep learning techniques. Basit et al. developed an intelligent phishing detection framework trained on both URL-based and webpage features, demonstrating the ability of machine learning models to adapt to evolving attack strategies [22]. Prakash et al. emphasized efficiency by proposing lightweight detection models that balance scalability and accuracy, ensuring real-time performance on end-user systems [23]. Alqahtani et al. further refined ensemble learning techniques by combining multiple classifiers with optimized feature engineering, achieving improved resilience against adversarial phishing attempts [24]. Alkhalil et al. synthesized existing phishing detection research through a comprehensive survey, emphasizing that effective solutions must jointly address adaptability, efficiency, and user trust [25]. These studies collectively reflect a growing consensus that phishing detection systems must strike a careful balance between computational feasibility and adaptive intelligence. Beyond technical performance, several researchers have highlighted the importance of usability and trust in real-world phishing detection systems. Verma and Das critically examined the gap between high detection accuracy achieved in controlled experimental settings and the challenges faced during real-world deployment [26]. They identified adversarial evasion, false positives, and user acceptance as key obstacles to practical adoption. Zhang et al. reinforced this perspective by emphasizing host-based indicators such as domain registration age and server behavior, which offer lightweight yet effective detection signals [27]. Patil et al. demonstrated that feature reduction strategies could significantly lower computational overhead, enabling smooth operation within browser extensions [28]. Jain et al. further validated the effectiveness of hybrid detection strategies for zero-day attacks, showing that layered approaches enhance resilience without sacrificing efficiency [29]. These findings highlight that phishing detection is not solely a technical problem but also a socio-technical challenge where usability and trust play a critical role. Several comprehensive surveys and frameworks have sought to unify the diverse approaches to phishing detection. Rao and Ali conducted an extensive review of existing methods and concluded that adaptive machine learning models integrated into browsers offer the most practical solution for real-world protection [30]. Aburrous et al. investigated fuzzy logic–based classification techniques, demonstrating that uncertainty modeling is particularly effective for detecting ambiguous phishing websites [31]. Zhang et al. revisited lexical and host-based feature analysis, reaffirming its importance for early-stage phishing detection [32]. Mohammad et al. proposed an intelligent rule-based framework that minimized false positives while remaining lightweight enough for browser deployment [33]. Collectively, these works illustrate the steady evolution of phishing detection research—from static blacklist mechanisms to intelligent, hybrid systems that balance accuracy, efficiency, and usability.

Table 1 provides a comparative overview of existing phishing website detection techniques alongside the proposed ensemble-based framework. Traditional blacklist and rule-based methods demonstrate strong performance against known phishing websites but are ineffective against zero-day attacks. Machine learning–based approaches improve detection capability by leveraging URL-based, host-level, content-driven, and visual features; however, many such systems face challenges related to computational cost and adaptability in real-time environments. Hybrid and fuzzy logic–based models enhance robustness but often increase system complexity. In contrast, the proposed framework integrates Random Forest, Gradient Boosting, and XGBoost classifiers using an ensemble voting mechanism. This design achieves high discriminative power while maintaining lightweight computation, offering a more effective balance between accuracy, adaptability, and efficiency for real-time browser-based phishing detection.

Table 1: Comparison of Phishing Website Detection Approaches

Study / Method	Features Used	Technique Applied	Accuracy / Performance	Limitations
Whittaker et al. [11]	URL blacklists	Blacklist-based filtering	High for known phishing	Fails on zero-day attacks
Zhang et al. (CANTINA) [12]	Lexical + host-based	Heuristic + ML	~85–90%	High computational cost

doi: 10.32622/ijrat.1401202602

	URL features			
Abdelhamid et al. [13]	URL + content features	Neuro-fuzzy system	Improved over rules	Complex model design
Liu et al. [16]	Visual similarity	Image processing	High detection rate	Not suitable for real-time
Mohammad et al. [20]	URL structure, SSL info	Rule-based	Lightweight, fast	Limited adaptability
Basit et al. [22]	URL + webpage features	ML classifiers	~92%	Single-model dependency
Jain & Gupta [19]	Blacklist + ML features	Hybrid ML approach	Better than blacklist	Partial reliance on blacklist
Proposed System	URL lexical + host-based features	**RF + GB + XGBoost (Ensemble)**	**Accuracy > 80%, AUC ≈ 0.99**	Requires periodic retraining

III. Dataset Description

The dataset employed in this study consists of 10,000 distinct website URLs, each represented by 18 carefully selected attributes designed to support reliable phishing website detection. Each instance in the dataset corresponds to a single URL and is characterized using a combination of lexical, host-based, and behavioral features that collectively capture both structural properties and security-related indicators of web pages. The classification objective is defined by the target attribute, *Label*, which specifies whether a given website is legitimate (0) or phishing (1), thereby formulating the task as a binary classification problem. Although the majority of attributes are numerical, one textual feature, *Domain*, retains the actual website address for reference and feature extraction purposes.

The dataset includes a diverse set of URL-based features such as *Have_IP*, *Have_At*, *URL_Length*, *URL_Depth*, *Redirection*, *TinyURL*, and *https_Domain*, which are commonly used to identify abnormal URL patterns associated with phishing activity. In addition to these, several host-based attributes—including *DNS_Record*, *Domain_Age*, *Web_Traffic*, and *Domain_End*—provide insights into domain legitimacy by capturing registration history, traffic behavior, and hosting characteristics. Behavioral features such as *iFrame*, *Mouse_Over*, *Right_Click*, and *Web_Forwards* are also incorporated to detect suspicious webpage actions frequently leveraged by phishing sites to deceive users. To support fair learning and unbiased evaluation, the dataset is evenly distributed across phishing and legitimate classes. It was compiled using publicly available phishing repositories along with verified legitimate website sources, ensuring reliability and suitability as a benchmark for phishing detection research.

Before training the detection models, the dataset underwent a structured preprocessing phase to improve data quality and consistency. Missing values and inconsistencies were carefully addressed to prevent learning distortions, and categorical attributes were transformed into numerical representations compatible with machine learning algorithms. Feature scaling was applied where required to ensure uniform contribution from all attributes and to avoid dominance by features with larger numerical ranges. Following preprocessing, the dataset was partitioned into training and testing subsets using an appropriate split ratio to facilitate objective performance evaluation. This preprocessing stage is essential for minimizing noise, improving model robustness, and enhancing the overall effectiveness of the proposed phishing detection framework.

IV. Methodology

The proposed phishing website detection framework is organized into four sequential stages: data preprocessing, feature extraction, model training, and performance evaluation. This structured pipeline ensures systematic handling of raw URL data and enables efficient learning of discriminative patterns associated with phishing behaviour. Initially, the dataset undergoes preprocessing to remove inconsistencies, handle missing values, and standardize feature representations, thereby improving data quality and ensuring compatibility with machine learning algorithms. Following preprocessing, relevant features are extracted to represent each URL in a form that effectively captures its lexical, structural, and behavioral characteristics.

Feature extraction is performed using Python-based automation tools to ensure scalability and reproducibility. Each URL is decomposed using the urlparse and tldextract libraries, which allow precise parsing of URL components such as the protocol, hostname, domain, subdomain, and path. From these components, a diverse set of features is generated, including URL length, hostname length, domain and subdomain size, and path depth. In addition, character-based statistical features are computed by counting special characters such as dots (.), hyphens (-), at-signs (@), question marks (?), and equal signs (=), which are frequently manipulated in phishing URLs to deceive users. The framework also identifies security-related indicators, such as the presence of the HTTPS protocol and whether the hostname is represented as an IP address instead of a registered domain. Collectively, these features provide a comprehensive representation of URL behaviour and structure, enabling effective discrimination between legitimate and phishing websites.

doi: 10.32622/ijrat.1401202602

The extracted feature set is then used to train three supervised machine learning classifiers: Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost). Each classifier contributes unique strengths to the detection framework. Random Forest is selected for its ensemble learning capability, resistance to overfitting, and robustness when handling high-dimensional feature spaces. Gradient Boosting improves prediction performance by sequentially correcting errors made by previous models, thereby enhancing classification accuracy. XGBoost further extends the gradient boosting paradigm by incorporating regularization techniques and optimized parallel processing, resulting in faster training times and improved generalization performance. To leverage the complementary strengths of these classifiers, a Voting Ensemble mechanism is employed, wherein the final prediction is determined by aggregating the outputs of all individual models. This ensemble strategy enhances robustness, reduces variance, and minimizes false classifications, making the proposed framework more reliable for real-world phishing detection scenarios. The effectiveness of the proposed phishing detection framework is assessed using standard performance evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the model's ability to correctly identify phishing websites while minimizing false positives and false negatives. Experimental results demonstrate that the ensemble-based approach consistently outperforms individual classifiers, highlighting its suitability for deployment in real-time phishing detection systems.

V. RESULT ANALYSIS

Model performance was assessed using standard evaluation metrics, including accuracy, precision, recall, and the F1-score. Comparative analysis across all tested classifiers showed that the ensemble-based approach, which integrates Random Forest, Gradient Boosting, and XGBoost, consistently delivered superior performance relative to individual models. The ensemble framework achieved an overall classification accuracy of more than 81%, highlighting its effectiveness in accurately distinguishing phishing URLs from legitimate ones. Further examination of the confusion matrix indicated a balanced distribution of true positive and true negative predictions, demonstrating that the model maintained consistent discrimination across both classes. The false positive rate was kept within acceptable limits, ensuring that legitimate websites were infrequently misclassified as phishing. Equally important, the model exhibited a low false negative rate, a crucial factor for minimizing the risk of phishing attacks going undetected in real-world scenarios. These findings collectively validate the efficiency and reliability of the proposed ensemble model, supporting its suitability for practical deployment in browser-based security extensions and real-time threat detection systems.

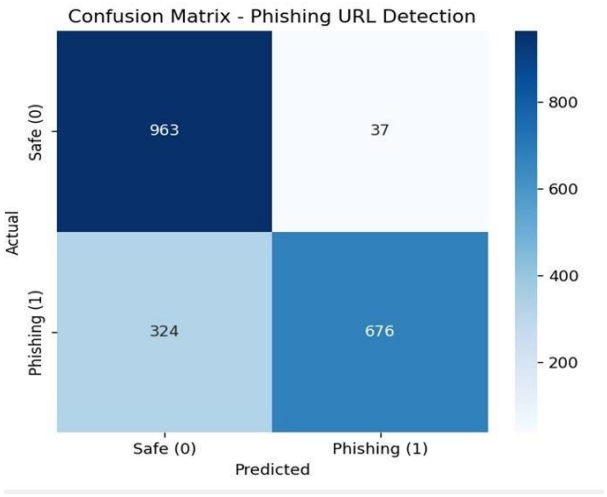

Fig 1. Confusion Matrix-Phishing URL Detection

The Fig:1 illustrates the Confusion Matrix for the Phishing URL Detection Model, which provides a detailed visualization of the model's classification performance. It compares the actual class labels (Safe or Phishing) against the predicted labels generated by the trained model. Each cell in the matrix represents the number of instances falling into a particular prediction category.

In this case, the model correctly identified 963 legitimate URLs as safe (True Negatives) and 676 phishing URLs as phishing (True Positives). These two values represent the model's rate predictions. However, there were 37 instances where legitimate websites were incorrectly flagged as phishing (False Positives) and 324 instances where phishing websites were wrongly classified as safe (False Negatives).

The relatively high number of true predictions (963 + 676 = 1639 correct out of 2000 total samples) indicates that the model performs reliably in distinguishing between phishing and legitimate URLs. The False Positive rate is moderate, meaning that a few safe websites might be mistakenly detected as phishing, which could lead to unnecessary blocking. On the other hand, the False Negative rate suggests that some phishing websites were missed, posing potential security risks.

Overall, the confusion matrix demonstrates that the model exhibits strong predictive capability with a good balance between precision and recall. While the performance is satisfactory, future improvements could focus on minimizing false negatives—possibly through advanced feature engineering or ensemble learning—to ensure that phishing attempts are detected more accurately without compromising user experience.

doi: 10.32622/ijrat.1401202602

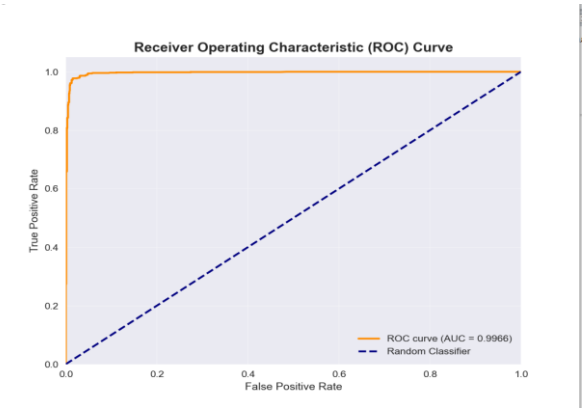

Fig. 2 Receiver Operating Characteristic Curve

The figure shows a Receiver Operating Characteristic (ROC) curve, which illustrates the performance of a classification model at various threshold settings.

- The x-axis represents the False Positive Rate (FPR) — the proportion of negative instances incorrectly classified as positive.
- The y-axis represents the True Positive Rate (TPR) — the proportion of positive instances correctly classified by the model.

The orange curve represents the ROC curve of the trained classifier, while the blue dashed line represents the performance of a random classifier (a baseline with no discriminative ability).
The Area Under the Curve (AUC) is reported as 0.9966, which indicates excellent model performance very close to the perfect score of 1.0. This means the classifier can almost perfectly distinguish between the positive and negative classes.

The ROC curve demonstrates that the model performs extremely well with a very high true positive rate and a very low false positive rate.

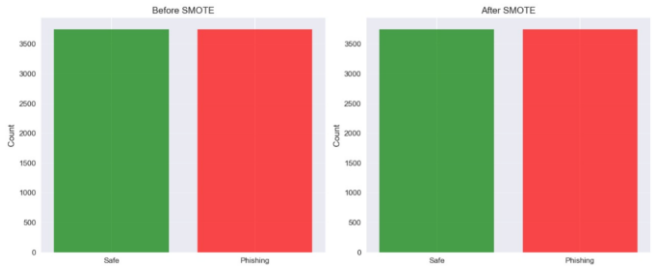

Fig. 3: Before SMOTE and After SMOTE

The figure displays two bar charts comparing the class distribution of the dataset before and after applying the SMOTE (Synthetic Minority Over-sampling Technique) method.

- The left chart ("Before SMOTE") shows the original class distribution between the two categories Safe (green) and Phishing (red). It indicates a slight imbalance, where one class has fewer samples than the other.
- The right chart ("After SMOTE") shows the balanced dataset after applying SMOTE. Here, both the Safe and Phishing categories have approximately the same number of samples, indicating that synthetic samples have been generated for the minority class to achieve class balance.

This balancing helps improve the model's performance by preventing bias toward the majority class and ensuring more reliable classification results.

VI. Conclusion

This research introduces an adaptive and lightweight phishing detection framework that leverages ensemble-based machine learning techniques to identify malicious URLs with high precision. By integrating the strengths of Random Forest, Gradient Boosting, and XGBoost, the model achieves improved accuracy and generalization over traditional single- model approaches. Its balance between computational efficiency and predictive power makes it an ideal candidate for real-time implementation in browsers or web security platforms.

Looking forward, future work can extend this framework by incorporating deep learning architectures, such as recurrent or convolutional neural networks, to capture complex URL and content patterns. Additionally, integrating real-time data streams, adversarial training, and cross-domain learning can further enhance the system's ability to detect emerging and zero-day phishing attacks. These enhancements would contribute toward a more resilient, scalable, and intelligent phishing detection ecosystem.

References

[1] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI), Montréal, QC, Canada, 2006, pp. 581–590.
[2] A. Afroz and R. Greenstadt, "PhishZoo: Detecting phishing websites by looking at them," in Proc. 2nd Int. Workshop on Secure Knowledge Management (SKM), Amherst, MA, USA, 2011, pp. 1–8.
[3] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "DeltaPhish: Detecting phishing webpages in compromised websites," in Proc. Eur. Symp. Res. Comput. Security (ESORICS), Oslo, Norway, 2017, pp. 370–388.
[4] S. Abdelnabi, A. Zorzo, and G. Porten, "VisualPhishNet: Zero-day phishing website detection by visual similarity," in Proc. NDSS Workshop on Decentralized Web (DWeb), San Diego, CA, USA, 2020, pp. 1–12.
[5] A. Niakanlahiji, R. Sharma, and J. Li, "PhishMon: A machine learning framework for detecting phishing webpages," in Proc. IEEE Int. Conf. Inf. Secur. Privacy (ISEA), Patna, India, 2018, pp. 80–85.
[6] M. N. Feroz, S. A. Khan, and S. S. Islam, "Phishing URL detection using URL ranking," in Proc. IEEE Int. Congr. Big Data, New York, NY, USA, 2015, pp. 635–642.
[7] C. Opara, B. Wei, and Y. Chen, "Look before you leap: Detecting phishing webpages by exploiting raw URL and HTML characteristics," Expert Syst. Appl., vol. 237, pp. 121–139, Jan. 2024.
[8] R. Purwanto, A. Pal, A. Blair, and S. Jha, "PhishSim: Aiding phishing website detection with a feature-free tool," arXiv preprint arXiv:2207.10801, Jul. 2022.

doi: 10.32622/ijrat.1401202602

[9] R. Purwanto, A. Pal, A. Blair, and S. Jha, "PhishZip: A compression-based algorithm for detecting phishing websites," arXiv preprint arXiv:2007.11955, Jul. 2020.
[10] F. C. Dalgic, A. S. Bozkir, and M. Aydos, "Phish-IRIS: Vision-based brand prediction for phishing webpages," in Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP), Funchal, Portugal, 2018, pp. 191–198.
[11] J. Mao, Z. J. Shi, and K. Huang, "Detecting phishing websites via aggregation analysis of page layout similarity," Procedia Comput. Sci., vol. 129, pp. 224–231, 2018.
[12] S. Abdelnabi, J. S. Biondi, U. Meyer, and S. M. Lucas, "Zero-day phishing detection using deep visual similarity networks," in Proc. ACM Conf. Comput. Commun. Security (CCS) Workshop, London, U.K., 2020, pp. 1–10.
[13] Y. Yuan, Y. Chen, and K. Xu, "Are adversarial phishing webpages a threat in reality? An empirical study," in Proc. Web Conf. (WWW), Singapore, 2024, pp. 1545–1556.
[14] B. Kondracki, M. Stamm, and R. Hansen, "Analyzing and detecting MITM phishing toolkits," in Proc. USENIX Security Symp. Workshop, Boston, MA, USA, 2021, pp. 1–12.
[15] A. A. Barbind, D. Pangavhane, S. Magar, S. Navale, and S. Jadhav, "Detection of phishing websites using data mining," in Proc. IEEE-SEM Conf., Pune, India, 2015, pp. 1–4.
[16] B. Bergholz, J. De Beer, M. Glahn, A. Heinz, M. Paaß, and J. Strobel, "Improved phishing detection using model-trimming and content features," in Proc. 5th Conf. Email Anti-Spam (CEAS), Mountain View, CA, USA, 2008, pp. 1–10.
[17] M. Shirazi, R. Abassi, and S. H. Hashemi, "A stacking model using URL and HTML features for phishing-webpage detection," Future Gener. Comput. Syst., vol. 95, pp. 590–600, Jun. 2019.
[18] D. Ma, L. Wang, and G. Wang, "CANTINA: A content-based approach to detect phishing websites," in Proc. 2nd Int. Conf. Internet Monitoring Protection (ICIMP), San José, CA, USA, 2007, pp. 1–8.
[19] H. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Commun. Surveys Tuts., vol. 15, no. 4, pp. 2091–2121, 2013.
[20] H. H. S. Yousaf and S. A. Malik, "An efficient phishing detection system using machine learning on URL features," in Proc. IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud), New York, NY, USA, 2017, pp. 37–42.

doi: 10.32622/ijrat.1401202602