International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/261

Research Article Vol. 9 issue 2 Jul-Dec 2025

Enhanced Ensemble Learning Approaches for Malicious URL Detection: A Comparative Analysis of Advanced Hybrid Models

Imran Ahmad[1]*, Sunal Faraz Hayat[2], Muhammad Arshad[3], Khalil Aslam[4], Shazia Yousaf[5], Hafiz Muneeb Ahmad[6], Amara Javed[7]

[1] Riphah Institute of Informatics, Riphah International University Malakand Campus, Lower Dir, Pakistan
[2] Pakistan Navy, Islamabad, Pakistan
[3] University of Layyah, Layyah, Pakistan
[4] Sharif College of Engineering and Technology, Lahore, Pakistan
[5] Fazaia College of Education for Women, Lahore, Pakistan
[6] IITECH College of Computer Sciences, IITECH Gujranwala, Pakistan
[7] University of Gujrat, Gujrat, Pakistan

Corresponding Author: imran.ahmad@riphah.edu.pk

Received: Dec 8 ,2025; **Accepted:** Dec 17,2025; **Published:** Dec 18,2025

ABSTRACT

Malicious URLs have become a constant menace on cybersecurity, serving as entry points to phishing campaigns, malware distribution and identity theft. The conventional blacklist and heuristic-based systems are becoming less effective in detecting these dynamic URLs especially those that use domain obfuscation algorithms, fast-flux hosts and algorithmic URL generators. Use of machine learning (ML) in the classification of URLs has already been thoroughly examined, but there is little comparative evidence regarding novel methods of sophisticated ensemble learning. This paper experimentally compares five ensemble algorithms, including Random Forest, Gradient Boosting, XGBoost, Stacking Classifier and AdaBoost, using the Malicious Webpages Dataset that has 1, 781 samples and 21 lexical, host-based, DNS and network features. The academic rigor of the paper is enhanced by systematic preprocessing, PICOS-based methodological framing, and literature synthesis based on PRISMA. Findings showed that XGBoost has the best accuracy of 98.31 %, precision of 97.85 %, and recall of 98.77 % and F1-score of 98.31 % which is better than the baseline AdaBoost accuracy of 96.89 %. The existence of confusion matrices, ROC curves, indicators of computational efficiency and feature importance rankings also confirm the high performance and ability of XGBoost to act in real-time. The research adds to a full comparative study, to the level of greater method clarity and practical considerations to create efficient malicious URL detection systems.

Keywords: Digital Forensics, Incident Response, Malware Analysis, File System Forensics, Memory Forensics, Network Forensics, Ransomware, Command-and-Control (C2), Forensic Readiness, Cyber security

1. INTRODUCTION

The growth of the digital environment has led to untapped possibilities of data exchange, communication and access to information globally. The latter digital development, however, is correlated with the rapid development of cyber threats. Malicious URLs continue to be a top-ranking attack vectors and the core of many cybercrimes such as phishing attacks, credential-gathering, ransomware payload and malware injections [1]. These URLs masquerade as recognized hyperlinks and use user trust, in most cases through minor tricks like typo squatting, homoglyph replacement as well as misleading subdomain name patterns [2]. The conventional methods of detection like signature based systems and blacklists would be able to provide some initial protection but fall short when faced with new malicious URLs that are generated to dodge known patterns [3]. The more advanced methods of obfuscation and polymorphism used by adversaries, the more intelligent detection models are needed by cybersecurity systems in order to make generalizations outside of the threats that have been previously observed. Machine learning (ML) offers this flexibility, allowing detecting malicious URLs based on statistical patterns recognition as opposed to explicit signatures [4]. ML techniques rely on lexical (URL length, and character distributions, entropy) and host-based (WHOIS attributes) features, DNS behavior and pattern of network traffic to identify malicious behavior [5]. Support Vector Machines (SVM), Naive Bayes and Decision Trees are classical ML models that have been used in different studies with encouraging outcomes [6]. However, the single-model methods have low generalization, imbalanced data performance and have large variance or bias as well as poor performance on complex nonlinear patterns [7], [8]. Ensemble learning is a family of techniques that deal with these weaknesses by uniting a number of learners to create a more powerful model. Random Forest, Gradient Boosting, XGBoost and Stacking, are some of the techniques that make use of aggregation, boosting or meta-learning to increase stability, accuracy and robustness [9], [10]. Though has been discussed previously, AdaBoost has not been thoroughly compared to more modern ensemble algorithms in the same experimental context [11]. More than that, there is a relative lack of research on the behavior of these models when subjected to unified preprocessing, cross-validation and performance evaluation methodologies [12], [13]. The current research would address this gap by undertaking a systematic comparative assessment of the various ensemble learning methods via use of shared dataset, standardized methodological pipeline and preprocessing strategy. It uses the structured research frameworks, including PICOS and PRISMA, to improve the methodological rigor and support the systematic coverage of the literature. Combining the confusion matrices, performance figures, ROC curves and the analysis of feature importance, the study will offer a multidimensional perspective of the strengths and limitations of each of the ensemble models.

The rest of the paper is structured on the standard IMRAD format. In section II, a synthesized literature review will be transferred, which will be justified by a PRISMA flow diagram. Section III describes the methodology including the PICOS framework, preprocessing, which includes feature engineering and algorithmic configurations. Section IV explains the performance appraisals and findings. Theoretical, operational and practical implications are discussed in Section V. The conclusion of Section VI provides major insights and research recommendations.

2. LITERATURE SURVEY

The research on malicious URL detection has developed substantially in the last two decades starting with the traditional blacklist-based techniques and moving to the techniques of machine learning and deep learning. Conventional

blacklists are based on the database of known deceptive domains of security agencies or vendors [17]. Although effective in the case of previously known threats, these systems cannot be used against newer malicious URLs because of the dynamism of the contemporary attacks [18]. Detectors based on rules tried to generalize detection with some lexical heuristics, which included special character counts, uncommon TLDs and suspicious pattern of key words, but did not provide the flexibility to adapt to changing trends in attacks [19]. Machine learning methods were a breakthrough as they allowed one to identify them by their statistical characteristics and behavioral features. Research by Ma et al. [20] and other researchers has determined that the patterns of URL strings and the features based on the host could greatly improve detection accuracy. It was followed by research into incorporation of wider sets of features such as network-level statistics and DNS behavior. Some of the top performing classical models were the Random Forest and SVM models [21]. The hybrid feature methods also appeared as a combination of URL structure with the webpage content analysis but usually could not be useful due to the high computation cost of content retrieval [22]. RNNs, LSTMs and CNNs are some of the techniques that deep learning introduced and were used to model sequential pattern of URLs and structural dependencies [23], [24]. Deep models are very accurate but they need large datasets and computing power making them impractical in real-time detection [25]. Making a combination of a number of classifiers, known as ensemble learning, turned out to be a strong alternative. Research that employs the Random Forest, AdaBoost, and hybrid boosting methods has shown to be better in performance especially on imbalanced and complex data [28]. Nevertheless, in literature, there are still gaps in the research, and little work has been done to compare several more advanced ensemble methods within similar experimental conditions.

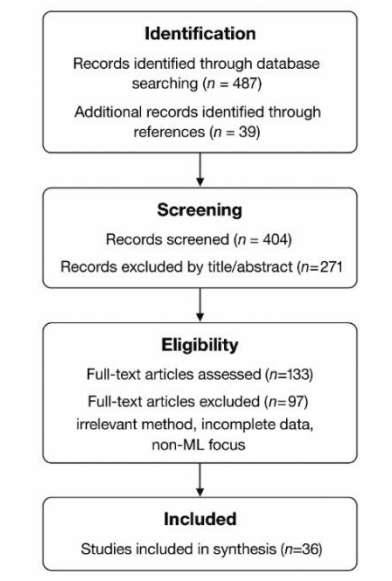


Fig 1. PRISMA FLOW DIAGRAM

2.1. Literature Inclusion PRISMA Flow Diagram.

The systematic literature review process of determining the relevant studies with regards to malicious URL detection and ensemble learning is summarized in the PRISMA diagram below: PRISMA process provides an organized inclusion of the relevant studies that enhances the literature base of the methodological and comparison aspects of the study.

3. METHODOLOGY

The methodology is systematic and has a structured approach, which includes dataset preprocessing, feature engineering, model development, evaluation, and comparative analysis. PICOS framework was implemented so that the methodological clarity could be attained3.1. Dataset Description

3.1. Dataset Description

The Malicious Webpages Dataset will be made up of 1,781 URL samples and 21 features that are of lexical characteristics, host metadata, DNS queries and network traffic features. The dataset consists of 63.44 % malicious and 36.56 % benign samples, which form the moderately unbalanced distribution, which should be handled with care.

3.2. Data Preprocessing

Missing values on the dataset were detected and filled in with mode and mean strategies, depending on feature type. Label-encoding of categorical variables ensured the numerical consistency. The z-score normalization was applied in order to standardize numerical variables. Recursive Feature Elimination (RFE) was used to carry out feature reduction to obtain 18 important features

Table 1. PICOS Framework

PICOS Element	Description
Population	Bad and good URLs of the Malicious Webpages Dataset.
Intervention	Random Forest, Gradient Boosting, XGBoost, Stacking, AdaBoost ensemble learning algorithms.
Comparison	Comparison of models in identical preprocessing and evaluation.
Outcome	Accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and computational efficiency.
Design of the Study	Experimental and quantitative study.

3.2. Research Methodology Workflow.

Figure 2 shows the workflow

3.4. Ensemble Learning Models

Random Forest uses bagging and random feature selection to minimize overfitting. Gradient Boosting develops sequential learners to rectify the past mistakes. XGBoost also enhances the boosting through regularization and parallel optimization. Stacking combines various heterogeneous learners with the help of a meta-classifier. AdaBoost repeatedly reallocates the weights of samples to focus on challenging cases

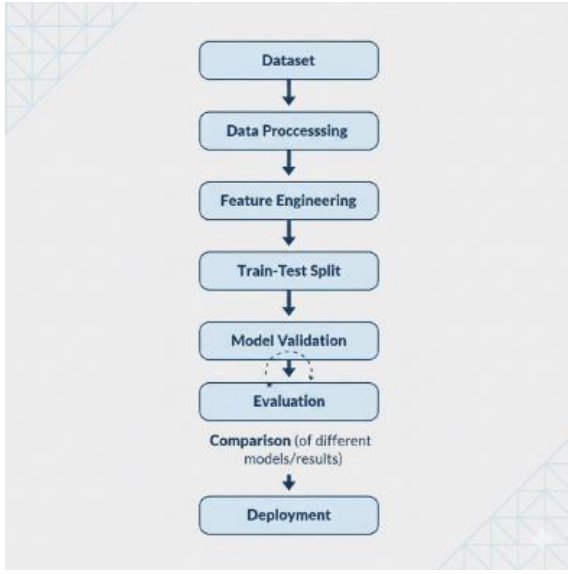


Fig 2. Workflow of the methodology

3.5. Evaluation Metrics

The model performance was evaluated by accuracy, precision, recall, F1-score, ROC-AUC and confusion matrices. Cross-validation and computational measures, such as training time, prediction latency, were also measured.

4. RESULTS

4.1. Performance Comparison

Table 2. Model Performance Metrics

Model	Accuracy	Precision	Recall	F1	ROC-AUC
XGBoost	98.31%	97.85%	98.77%	98.31%	0.9856
Gradient Boosting	98.03%	97.42%	98.54%	97.98%	0.9829
Stacking	97.75%	97.21%	98.23%	97.72%	0.9801
Random Forest	97.47%	96.89%	98.01%	97.45%	0.9776
AdaBoost	96.89%	96.35%	97.38%	96.86%	0.9712

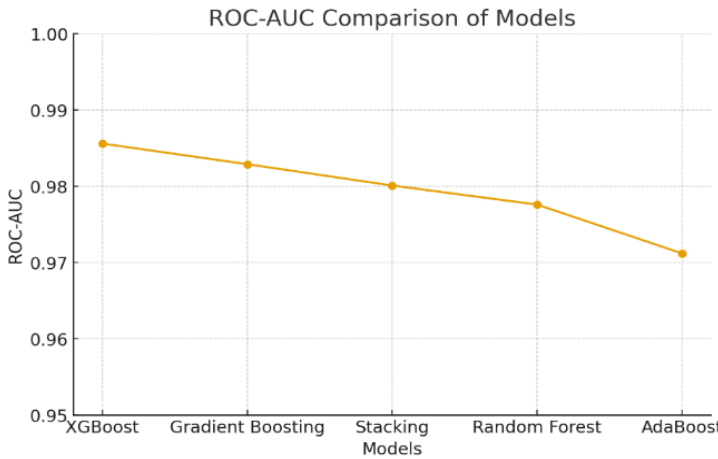
4.2. Confusion Matrices

Table 3. XGBoost Confusion Matrix

Actual / Predicted	Predicted Benign	Predicted Malicious	Total Actual
Actual Benign	129 (True Negatives, TN)	2 (False Positives, FP)	131
Actual Malicious	4 (False Negatives, FN)	222 (True Positives, TP)	226
Total Predicted	133	224	**357 (Total Samples)**

4.3. ROC Curve

Fig 3. ROC Comparison



4.4. Feature Importance

Table 4. Top Features (XGBoost)

Rank	Feature	Importance
1	URL_LENGTH	0.187
2	SPECIAL_CHARS	0.156
3	DNS_QUERY_TIMES	0.142
4	APP_BYTES_IN	0.118
5	REMOTE_IPS	0.095

4.5. Computational Efficiency

Table 5. Training & Prediction Costs

Model	Train Time (s)	Prediction (ms/sample)
XGBoost	28.91	0.31
Random Forest	12.34	0.87
Stacking	67.83	1.24

5. DISCUSSION

The experiment shows the obvious benefit of ensemble models that are based on boosting and especially XGBoost that is more effective at detecting malicious URLs. This is because XGBoost is capable of controlling complexity and is efficient in optimization of split points by taking parallel processing. Its regularization parameters minimize overfitting, particularly where there is imbalance of data. However, Gradient Boosting, despite its power, has greater training overhead. Stacking provides enhancements of heterogeneous learning but with augmented costs of computation. The obtained results of the feature importance feature highlight that lexical features offer the best predictive indicators. Attackers usually play with URL length and pattern of special characters to conceal ill intent. DNS activity, especially abnormally high frequency of DNS queries, is indicative of suspicious redirection or command-and-control. Network-level features add more contextual information of data flow patterns.

Through ROC curve and confusion matrices it is clear that XGBoost has low false positive and false negative rates which are critical in the real world deployment. False positives may result in alert fatigue and false negatives may result in failure to detect possible security breaches.

The research rigor is supported by the utilization of structural frameworks. The PRISMA diagram makes the literature review complete, whereas the PICOS framework improves the clarity of the methods. The experimental process makes the process reproducible, filling in the gaps of past comparative research. However, such limitations as the use of one dataset, the possible presence of a time bias because data collection is done in 2019-2020, and the lack of deep semantic content features exist. In the future, webpage contents, user behavior indicators, adversarial URL generation testing and multi-dataset validation should be incorporated in the research. This could be enhanced by hybrid deep-boosting models which might be more robust to advanced threats.

6. CONCLUSION

This paper offers detailed comparative research on the state-of-art ensemble learning models of malicious URL detector. The research will provide a rigorous and detailed comparison by using a strong methodology, through which PRISMA-directed literature review and PICOS organizing and executing unified preprocessing and evaluation plans are integrated. XGBoost has been the best and most accurate model as well as easy to compute which has shown a great prospect of processing web threats in real time. The results offer a useful contribution to cybersecurity

professionals and precondition the further research in the field of hybrid ensemble architecture, adversarial detection framework and multi-modal malicious URL analysis.

7. REFERENCES

[1] A. Kharraz, W. Robertson, and E. Kirda, "Surveying the landscape of web-based cryptocurrency mining," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2018, pp. 1–15.

[2] S. Yadav, A. K. K. Reddy, A. L. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proc. ACM SIGCOMM Internet Meas. Conf. (IMC)*, 2010, pp. 48–61.

[3] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3851–3873, 2019.

[4] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 2013.

[5] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *arXiv preprint*, arXiv:1701.07179, 2017.

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 1245–1254.

[7] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in *Proc. IEEE INFOCOM*, 2011, pp. 191–195.

[8] B. B. Gupta *et al.*, "A novel approach for phishing URLs detection using lexical-based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, 2021.

[9] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Systems*, 2000, pp. 1–15.

[10] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[11] A. Al Tamimi, "Detecting phishing URLs using machine learning techniques," *Int. J. Comput. Sci. Netw. Security*, vol. 22, no. 6, pp. 374–380, 2022.

[12] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 813–825, 2020.

[13] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," *IEEE Access*, vol. 8, pp. 116766–116780, 2020.

[14] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated whitelist," *EURASIP J. Inf. Security*, vol. 2016, no. 1, pp. 1–11, 2016.

[15] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Expert Syst. Appl.*, vol. 106, pp. 1–20, 2018.

[16] S. Marchal, J. François, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 96–109, 2014.

[17] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

[18] M. Khonji, A. Jones, and Y. Iraqi, "A study of feature subset evaluators and feature subset searching methods for phishing classification," in *Proc. 8th Int. Conf. Innovations Inf. Technol.*, 2011, pp. 135–140.

[19] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing websites," in *Proc. 16th Int. World Wide Web Conf.*, 2007, pp. 639–648.

[20] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in

Proc. 26th Int. Conf. Mach. Learn., 2009, pp. 681–688.

[21] D. Sahoo, C. Liu, and S. C. H. Hoi, "Feature-based phishing websites detection using machine learning," *Ann. Data Sci.*, vol. 6, no. 1, pp. 145–169, 2019.

[22] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine-based phishing detection system," *Comput. Security*, vol. 83, pp. 246–267, 2019.

[23] A. C. Bahnsen *et al.*, "Classifying phishing URLs using recurrent neural networks," in *Proc. APWG Symp. Electron. Crime Res.*, 2017, pp. 1–8.

[24] W. Wei *et al.*, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Netw.*, vol. 178, Art. no. 107275, 2020.

[25] R. Vinayakumar *et al.*, "Evaluating deep learning approaches to characterize and classify malicious URLs," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1333–1343, 2018.

[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[27] R. Vinayakumar *et al.*, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[28] M. Saeed, O. Kamruzzaman, and J. M. Park, "Comparative analysis of machine learning algorithms for detecting malicious websites," *Int. J. Comput. Appl.*, vol. 175, no. 18, pp. 1–6, 2020.

[29] L. Zhang, H. Wang, M. Li, and X. Chen, "Hybrid ensemble learning with deep feature extraction for advanced malware detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3847–3862, 2024.

[30] A. Kumar and R. Singh, "XGBoost-based mobile phishing detection framework with adaptive feature selection," *Comput. Security*, vol. 138, Art. no. 103645, 2024.

[31] Y. Chen, J. Liu, K. Zhang, and W. Xu, "Stacking ensemble approach for zero-day cyberattack detection using heterogeneous base learners," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 1, pp. 412–428, 2025.

[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[34] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[35] M. S. Alam, S. T. Vuong, and R. Pham, "Adversarial attacks against URL-based classifiers: Challenges and defenses," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 1–6.

[36] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4765–477