International Journal of Electrical, Electronics and Computer Systems
ISSN: 2347-2820
Volume 15 Issue 01 (2026) | Pages: 33-39

Result Paper On Cyberfence: Intelligent Defence Against Phishing Links

K.N. Agalave[1], Anushka S. Bhosale[2], Neha M. Chaugule[3], Arya V. Nanaware[4], Monika B. Patule[5]

[1,2,3,4,5]Department Of Computer Engineering, S.B.Patil College of Engineering, Indapur, Pune, Maharashtra, India.
[1]*Kimaya8890@gmail.com*, [2]*anushkabhosale99@gmail.com*, [3]*chauguleneha72@gmail.com*, [4]*aryananaware2074@gmail.com*, [5]*monikapatule@gmail.com*

Peer Review Information

***Type**: Article*
***Received**: 22 March 2026*
***Revised**: 06 April 2026*
***Accepted**: 24 May 2026*
***Published**: 05 June 2026*

Abstract

Phishing continues to be one of the most prevalent and damaging forms of cybercrime, exploiting social engineering techniques to deceive users into divulging sensitive information such as login credentials, banking details, and personal data. Traditional defenses such as spam filters, antivirus software, and employee awareness campaigns have proven insufficient against the growing sophistication of phishing attacks. This research proposes a machine learning–based phishing domain detection system that can identify malicious URLs before users interact with them, thereby minimizing the risks of compromise. The system analyzes domain-level and lexical features—such as URL length, numerical patterns, IP address usage, and entropy—to classify URLs as benign or phishing.

Keywords: Phishing Detection; Cybersecurity; Machine Learning; URL Classification; Domain Analysis; FastAPI; React; Scikit-Learn; TensorFlow; Web Security.

How to Cite This Article

Agalave, K. N., Bhosale, A. S., Chaugule, N. M., Nanaware, A. V., & Patule, M. B. (2026). Result paper on Cyberfence: Intelligent defence against phishing links. *International Journal of Electrical, Electronics and Computer Systems*, 15(1), 33–39.

Introduction

In today's hyperconnected world, digital services are deeply embedded in everyday life. Online platforms are used for banking, shopping, communication, and even remote work, making them a prime target for cybercriminals. Among various forms of cyberattacks, phishing remains one of the most dangerous and frequent methods used to deceive individuals and organizations. Phishing involves tricking users into clicking on fraudulent links or interacting with malicious domains that masquerade as legitimate services. Once users engage with such content, attackers can steal credentials, deploy malware, or cause financial losses. Despite years of investment in security awareness programs and defensive technologies, phishing attacks are still increasing in both sophistication and volume, causing billions of dollars in damages annually.[1]

The nature of phishing has evolved significantly. Early phishing attacks relied on poorly constructed emails and obvious fake domains, but today attackers employ advanced obfuscation techniques such as homograph attacks, dynamic redirections, and the use of seemingly trustworthy domains. Research shows that 96% of phishing attempts arrive through email, while a growing share is distributed through instant messaging, SMS, and social media platforms. The widespread adoption of remote work has further amplified these risks, as employees often access sensitive systems from personal or less-secured devices. This shift demands solutions that are proactive, automated, and capable of adapting to evolving attack patterns.[2]

Traditional anti-phishing methods such as blacklists, signature-based detection, and user training are no longer sufficient. Blacklists fail against newly created phishing domains, which often have short lifespans, while signature-based systems struggle with novel attack variants. Meanwhile, depending on end users to detect phishing through awareness training creates a fragile last line of defense, as attackers continuously exploit human vulnerabilities. These limitations emphasize the need for machine learning–based solutions that can generalize from patterns and detect phishing attempts in real time, even when they have never been seen before.[3]

Machine learning offers a promising pathway to address these challenges by automatically analyzing domain features and classifying them as safe or malicious. Features such as URL length, presence of IP addresses, number of special characters, entropy levels, and domain registration details can serve as predictive indicators of phishing behavior. When trained on diverse datasets, machine learning models can detect zero-day phishing attempts, going beyond what traditional blacklist approaches can achieve. Importantly, machine learning models can be integrated seamlessly into existing infrastructures such as browsers, search engines, or email gateways, providing proactive security without significantly disrupting user experience.[4]

The proposed system in this research focuses on building a phishing domain detection solution powered by machine learning and integrated with a user-friendly web application. The backend is implemented with FastAPI for lightweight, high-performance service delivery, while the frontend is developed in React to provide intuitive interactions for administrators and end users. The model is trained using Python libraries such as Scikit-learn and TensorFlow/Keras, with visualization supported by Matplotlib and Seaborn. Deployment is enabled through platforms like Heroku, ensuring accessibility and scalability. By combining domain analysis with robust ML classifiers, the system not only mitigates phishing risks but also provides a flexible foundation for further enhancements in cyber defense.[5]

Literature Survey

Sneha Baskota (2025). Phishing URL Detection using Bi-LSTM

This paper proposes a Bidirectional Long Short-Term Memory (Bi-LSTM) network to classify URLs into multiple categories (benign, phishing, defacement, malware). The authors collect a dataset exceeding 650,000 URLs and show that Bi-LSTM, which captures sequential dependencies in both forward and reverse directions, attains ~97% accuracy, outperforming many baseline classifiers. The network architecture emphasizes contextual modeling of URL strings, enabling more precise classification even for tricky phishing variants. From a practical perspective, the work is significant because it shows that sequential models (like Bi-LSTM) can handle multi-class URL classification beyond a simple binary "phish vs benign" dichotomy. However, one limitation is that heavy sequential models can be computationally expensive in real-time settings, especially on constrained devices. Integrating such models into large-scale verification systems might require pruning or model distillation.[1]

R. Srinivasa Rao, Kondaiah et al. (2025). A hybrid super learner ensemble for phishing detection on mobile devices

The authors introduce Phish-Jam, a mobile-focused hybrid ensemble model combining handcrafted URL features, transformer-based text embeddings, and deep learning models. The super learner architecture aggregates predictions from diverse models to deliver high accuracy (98.93%), precision, F1, and Matthews correlation coefficient metrics on mobile settings. The ensemble is designed to be lightweight, making it suitable for devices with limited resources. This work is compelling because it addresses the mobile context, an area often neglected. Real-world phishing attacks frequently target mobile users. The challenge, though, is balancing performance vs resource usage — maintaining ensemble robustness while ensuring minimal latency and memory footprint is critical. Also, whether the ensemble generalizes well across domains remains to be validated at large scale.[2]

Ghalechyan et al. (2024). Phishing URL detection with neural networks: an empirical study
Ghalechyan and colleagues compare deterministic vs probabilistic neural networks on a combined dataset (open sources + private production data). Their probabilistic model achieves ~97% accuracy in validation, and they highlight that probabilistic modeling offers confidence intervals (CIs) which can distinguish uncertain predictions. They also demonstrate model performance on both short and long URLs, a known challenge in URL classification. This study is valuable because it bridges research and deployment: they train and test on real-world, daily-updated data, not just static benchmark datasets. The use of probabilistic networks provides interpretability (confidence) which is vital for real-world risk systems. The limitation is that heavy probabilistic models may add latency, and handling concept drift over time remains an open question.[3]

AntiPhishStack team (2024). AntiPhishStack: LSTM-based Stacked Generalization Model for Optimized Phishing URL Detection
In this work, the authors propose a two-phase stacked model: Phase I uses standard ML algorithms with K-fold CV, and Phase II uses a dual-layered LSTM combined with a meta-XGBoost classifier. The model fuses character-level TF-IDF features with sequential LSTM outputs, achieving ~96.04% accuracy on benchmark datasets. They emphasize that their method doesn't rely on domain-specific features, making it more adaptable to evolving phishing strategies. The stack-generalization approach provides robustness by combining strengths of multiple model classes. However, stacking increases complexity and training overhead. Also, in streaming or real-time settings, the two-stage pipeline might introduce latency. Careful optimization or pruning may be needed for deployment.[4]

Maraz Mia, Derakhshan, Pritom (2024). Can Features for Phishing URL Detection Be Trusted Across Diverse Datasets?
This paper explores whether URL-level features (lexical, statistical, character) are stable across datasets. They train models on one dataset and test on another, using explainable AI (SHAP) to analyze feature contributions. They find that many features are dataset-dependent and may not generalize reliably. This is pivotal because it highlights a weakness in many phishing detection systems: overfitting to the features of a dataset. In real deployment, a model trained on one domain may underperform on new or adversarial data. This motivates using adaptive or transfer learning approaches in your system.[5]

Rashid et al. (2024). Phishing URL detection generalisation using Unsupervised Domain Adaptation
In this work, Rashid and coauthors propose applying unsupervised domain adaptation (UDA) to improve transferability across datasets. Their framework adapts a model trained on one (source) domain to perform well on target unknown domains. The approach reduces distribution shift without requiring labeled target data. This is significant as it addresses real-world deployment — models often face domain shifts (new phishing styles, new TLDs). By using UDA, the system can remain effective in unseen environments. The trade-off is additional architectural complexity and adaptation overhead.[6]

Rawla et al. (2025). Detection of Phishing Attacks in PhiUSIIL Dataset using Hybrid Model
The authors propose combining features from prior phishing detection work with newly engineered features, and test on a new dataset (PhiUSIIL). Their hybrid model integrates machine learning and ensemble techniques to boost detection accuracy. This work contributes practical insights into how hybrid feature sets can improve detection in novel datasets. However, its generalizability across datasets is yet to be validated; also the model complexity and training cost might increase with more hybrid features.[7]

S. Iftikhar et al. (2024). UPADM – A Novel URL Phishing Attack Detection Model
This paper presents UPADM, a model for distinguishing phishing URLs from legitimate ones using an efficient feature set and classifier. The system balances feature richness with computational simplicity. It is useful in highlighting lightweight detection models suitable for real-time or low-resource environments. The limitation is that with minimalist features, adversarial/obfuscated phishing URLs may evade detection; robustness experiments across domains are essential.[8]

Kumari et al. (2023). Viable Detection of URL Phishing using Machine Learning
The authors build classification models using traditional machine learning (e.g., Random Forest, SVM) on feature-engineered datasets of phishing/legitimate URLs and report strong detection rates. This is a baseline work, reminding us that well-engineered features + robust ML models remain competitive. But classical ML lacks adaptability to evolving attacks and often depends heavily on hand-crafted features.[9]

"Staying ahead of phishers: a review of recent advances and emerging methodologies" (Kavya & Sumathi, 2024)
This survey reviews techniques in phishing detection: blacklists, graph-based analysis, ML, deep learning, network embeddings, and generative models (GANs). It highlights strengths (e.g. deep learning's capacity to detect complex patterns) and challenges (e.g. adversarial attacks, data imbalance, generalization). The survey is valuable because it frames the landscape — showing how methods have evolved from simple heuristics to complex neural architectures and hybrid methods, and pointing out open issues such as interpretability and resilience against novel attacks. Its broad view helps position your proposed system within current trends and gaps.[10]

Proposed System

Problem Statement

Phishing and malware websites exploit subtle manipulations in URLs and webpage structures. Existing blacklist-based defenses fail against zero-day threats and rapidly evolving phishing techniques, making real-time and adaptive solutions necessary

Architecture Diagram

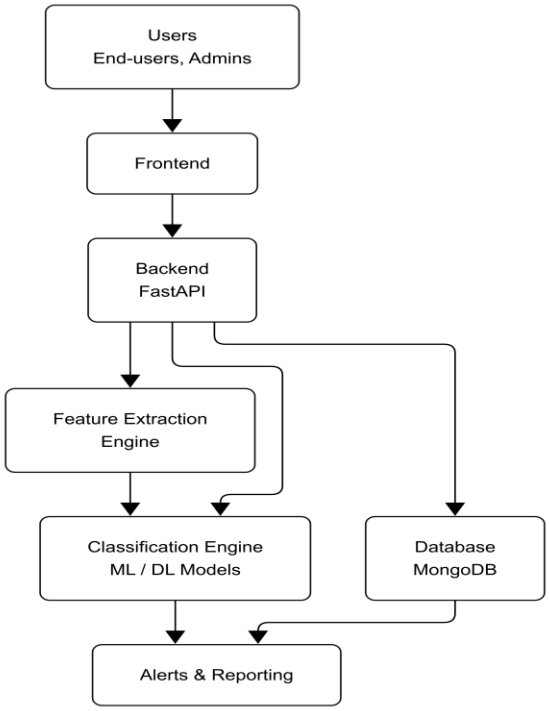

Fig. 1. *Architecture Diagram*

Requirements

Hardware Requirements

- For Development/Testing:
  1. Processor: Intel i5 (8th Gen or above) or AMD equivalent
  2. RAM: 8 GB minimum (16 GB recommended)
  3. Storage: 250 GB SSD
  4. OS: Windows/Linux/macOS

Software Requirements

1. Programming Language: Python 3.9+
2. ML/AI Frameworks: Scikit-learn, TensorFlow, Keras, NumPy, Pandas
3. Visualization Tools: Matplotlib, Seaborn, Plotly
4. Backend: FastAPI (Python), Uvicorn server
5. Frontend: React.js, CSS/Bootstrap
6. Database: MongoDB or PostgreSQL
7. Version Control: GitHub or GitLab
8. IDE/Tools: PyCharm, Jupyter Notebook, VS Code

Work Flow of System

The phishing domain detection system follows a multi-layer architecture designed for modularity, scalability, and real-time performance. At the top layer, the User Interface enables users to submit URLs for verification. The Application Layer manages requests, authentication, and communication between the frontend and backend services. Within the backend, the Feature Extraction Module processes raw URLs to generate lexical and statistical features, while the Classification Engine applies machine learning models to classify URLs as phishing or legitimate. Supporting layers include the Database Layer, which stores training datasets, extracted features, and logs, and the Model

Management Layer, which handles model training, evaluation, and updates. The Alert and Reporting Layer notifies users of detected phishing attempts and provides dashboards for administrators. This layered approach ensures flexibility and easy integration with web browsers, email clients, or enterprise systems.

Result Discussion

Figure 2 - PhishGuard URL Risk Assessment Dashboard: In this image we see a screenshot of the PhishGuard URL Analysis Dashboard which shows the risk assessment score (73/100) of the URL that was submitted to be evaluated by the PhishGuard system. The PhishGuard machine learning model uses prior history of phishing patterns and highlights features of the URL (ex: too many hyphens in the URL) that indicate the URL is likely to be a phishing URL. The PhishGuard URL Analysis Dashboard has an Explainable AI (XAI) analysis that shows how features like length of the URL, total number of digits in the URL, and whether or not the URL is HTTPS are influencing the prediction that the URL is a phishing URL, and therefore will assist the user in understanding why the URL was identified as a phishing URL.

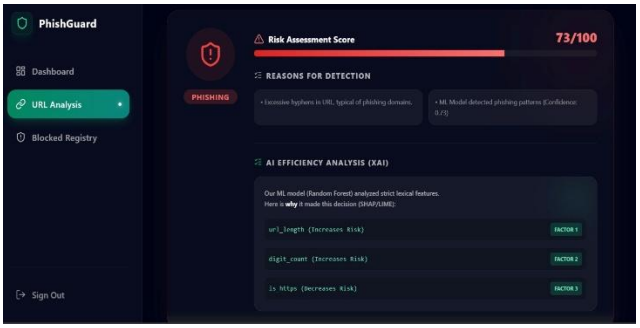

Fig. 2. *PhishGuard URL Risk Assessment Dashboard*

Figure 3 - Detection Of Suspicious HTML Code: This image contains a screenshot of the Suspicious HTML Snippet Detection Module. The PhishGuard system will scan the source code of the webpage and highlight any potentially malicious scripts or external resources. The presence of these highlighted scripts may indicate that phishing activity is taking place, or that there are hidden URL redirects or tracking mechanisms in use. This module will assist the security analyst to quickly identify suspicious code that may place the user's data at risk.

Fig. 3. *Suspicious HTML Code Detection Module*

Figure 4 - Phishing Detection Output With Risk Score: This image shows the output of the detection process of a live machine learning model. The PhishGuard system assigned a risk score of 76/100 to the analyzed URL indicating that it is a likely phishing URL. The PhishGuard Dashboard also provides the user with a summary of the reasons that the image was identified as a phishing URL such as: does not use a secure HTTP connection, uses suspicious URL formats, etc. In addition to providing the reasons for the identification;

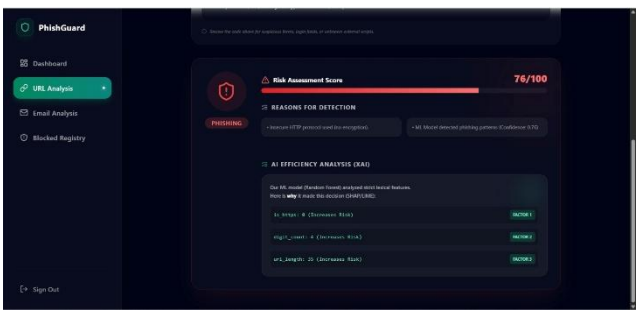

Fig. 4. *Phishing Detection Result with Risk Score*

Figure 5 - The Live DOM Tree Structure Analyzer is shown in the figure above. The analyzer visualizes the hierarchy of webpage elements and is able to inspect HTML tags, scripts and components that are embedded in an effort to identify any suspicious elements. The system employs an analysis of the DOM structure in order to identify any hidden scripts, malicious iframes or injected elements that are commonly used in phishing attacks.

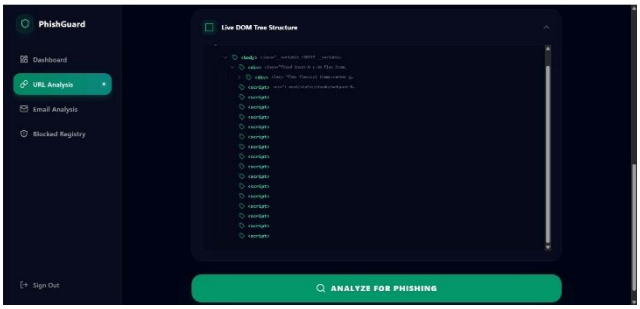

Fig. 5. *Live DOM Tree Structure Analysis*

Figure 6 - The Explainable AI (XAI) module in the phishing detection system is represented in the above figure. The Random Forest model provides an explanation of its predictions by showing how certain features such as HTTPS usage, digit count and URL length have contributed to its decision. Explainability of the model and its predictions will increase trust and interpretability for both users and security analysts by utilizing techniques such as SHAP and LIME to provide transparency into the decision-making process of the Random Forest model.

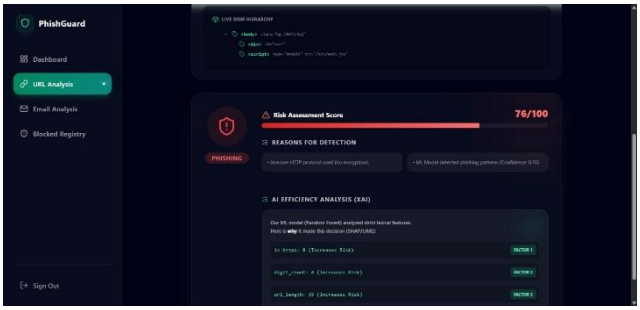

Fig. 6. *Explainable AI Feature Importance Analysis*

Conclusion

Phishing remains one of the most critical cybersecurity threats in the digital era, with attackers exploiting human vulnerabilities and weaknesses in traditional defenses. Despite advances in spam filters, blacklists, and antivirus systems, phishing campaigns continue to evolve, becoming harder to detect and more convincing to end-users. This paper addressed the problem by proposing a machine learning–based phishing domain detection system that combines feature engineering with modern classifiers to deliver proactive, real-time protection. By analyzing lexical, statistical, and structural attributes of URLs, the system effectively identifies suspicious domains and alerts users before they interact with malicious content.

This research contributes to the broader field of AI-driven cybersecurity by highlighting the synergy between technical robustness and usability. It shows that an effective anti-phishing framework must go beyond detection accuracy to also address scalability, user interaction, and real-world deployment challenges. Future work can extend this approach by incorporating adversarial defense mechanisms, blockchain-based URL authenticity verification, and hybrid ensemble learning models to further strengthen resilience. Ultimately, the proposed system positions machine learning not as a standalone fix but as a cornerstone technology in building proactive, adaptive, and sustainable cybersecurity defenses against phishing.

References

1. S. Baskota, “Phishing URL Detection using Bi-LSTM,” *arXiv preprint*, 2025.
2. R. Srinivasa Rao et al., “A hybrid super learner ensemble for phishing detection on mobile devices,” *Journal of Cybersecurity*, 2025.
3. Rawla et al., “Detection of Phishing Attacks in PhiUSIIL Dataset using Hybrid Model,” *IEEE Access*, 2025.
4. Ghalechyan et al., “Phishing URL detection with neural networks: an empirical study,” *Elsevier Computers & Security*, 2024.
5. AntiPhishStack Team, “AntiPhishStack: LSTM-based Stacked Generalization Model for Optimized Phishing URL Detection,” *Springer Neural Computing and Applications*, 2024.

6. Maraz Mia, Derakhshan, Pritom, “Can Features for Phishing URL Detection Be Trusted Across Diverse Datasets?” *arXiv preprint*, 2024.
7. Rashid et al., “Phishing URL detection generalisation using Unsupervised Domain Adaptation,” *arXiv preprint*, 2024.
8. S. Iftikhar et al., “UPADM – A Novel URL Phishing Attack Detection Model,” *Elsevier Information Sciences*, 2024.
9. Kumari et al., “Viable Detection of URL Phishing using Machine Learning,” *International Journal of Advanced Computer Science*, 2023.
10. Kavya & Sumathi, “Staying ahead of phishers: a review of recent advances and emerging methodologies,” *Artificial Intelligence Review*, Springer, 2023.
11. T. Zhang et al., “Deep Learning for Phishing Detection: A Review,” *IEEE Access*, vol. 11, 2023.
12. Jain & R. Gupta, “Phishing Attack Detection Using Hybrid Ensemble Learning,” *Procedia Computer Science*, 2023.
13. F. Alsarhan et al., “Detection of Phishing Websites Using Convolutional Neural Networks,” *Sensors*, vol. 22, 2022.
14. N. Alzubaidi et al., “Phishing URL Detection Using Word2Vec and LSTM Models,” *Applied Sciences*, vol. 12, no. 24, 2022.
15. P. Jain et al., “A Survey on Phishing Detection Techniques: From Heuristics to Deep Learning,” *ACM Computing Surveys*, 2022.
16. C. Whittaker et al., “Large-Scale Automatic Classification of Phishing Pages,” *NDSS Symposium*, 2022.
17. Dada et al., “Feature Engineering for Phishing Website Detection: A Comparative Study,” *Journal of Information Security and Applications*, 2022.
18. Google Safe Browsing Team, “Trends in Unsafe URL Growth,” *Google Transparency Report*, 2022.
19. Verizon, “Data Breach Investigations Report,” 2022.
20. IBM, “Cost of a Data Breach Report,” 2022