Phishing Website Detection using Artificial Neural Networks and Count Vectorization: A Web-Deployed Approach

[1] O.I. Omotosho, [2] A.S. Adedoyin, [3] O.R. Agboola,[4]O.E. Ogundele, and [5] U.A. Oladimeji
Department of Cyber Security Science, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria
oluyinkaa14@gmail.com | jabhulani08@gmail.com | agboolaomolade034@gmail.com | odunayoelizabeth@gmail.com | oladimejiusman03@gmail.com

Received: 02-MARCH-2026; Reviewed: 23-MAY-2026; Accepted: 30-JUNE-2026
https://dx.doi.org/10.46792/fuoyejet.v11i2.1539

ORIGINAL RESEARCH

Abstract— Phishing-based credential theft represents one of the most consistently reported threats in annual cybersecurity incident data, deceiving users into revealing sensitive credentials through fraudulent websites. Traditional detection approaches relying on blacklists and heuristics fail against novel attacks due to dependence on predefined patterns. This study presents an automated phishing detection system using Artificial Neural Networks (ANN) to classify URLs as phishing or legitimate. A dataset of 11,055 URLs drawn from PhishTank, the UCI Machine Learning Repository, and Alexa Top Sites was preprocessed using Count Vectorization, which transformed raw URLs into token-frequency feature vectors without manual feature engineering. A three-layer ANN with 64-neuron and 32-neuron hidden layers (ReLU activation, 0.5 dropout), trained using the Adam optimizer with binary cross-entropy loss and early stopping, achieved 99.4% training accuracy and 97.5% validation accuracy at epoch eight, with early stopping at epoch eighteen. Final evaluation on the 2,211-sample test set yielded 96.5% accuracy, 95.2% precision, 94.8% recall, and 95.0% F1-score, outperforming Logistic Regression (87.3%), SVM (89.5%), and Random Forest (91.8%) baselines. A Flask-based web application was deployed for real-time URL classification with a 2.1-second average response time.

Keywords—artificial neural networks, cybersecurity, machine learning, phishing detection, URL classification, web application.

1 INTRODUCTION

Global internet adoption has transformed how people communicate, conduct business, and access information, but this growth has simultaneously expanded the attack surface for cybercriminals. Among the threats that have grown with internet adoption, phishing remains one of the most operationally damaging: attackers impersonate legitimate services to steal credentials and financial data from unsuspecting users (Huang et al., 2021). Their effectiveness stems from combining social engineering with technical deception, bypassing conventional security controls. Anti-Phishing Working Group reported that phishing incidents have steadily increased as attackers deploy more sophisticated evasion techniques (APWG, 2022). Data-driven detection methods offer a structural advantage over rule-based systems: models trained on URL patterns can generalise to previously unseen phishing variants that evade static filters (Ghalechyan et al., 2024). However, prior work reveals persistent challenges including imbalanced datasets, complex feature engineering pipelines, limited reproducibility, and insufficient benchmarking. This study addresses those gaps by proposing a fully automated, web-deployable ANN-based detection system that eliminates manual feature engineering and is benchmarked against multiple baseline classifiers on the same dataset.

Corresponding Author:

Section B- ELECTRICAL/COMPUTER ENGINEERING & RELATED SCIENCES

Can be cited as:

Omotosho O.I., Edeguare F.G., Ogundare M.D., Ogunjobi V.I., Olorunsola P.I., and Oyeside I.J. (2026). Smart Contract Vulnerability Analysis and Detection System using Machine Learning. FUOYE Journal of Engineering and Technology (FUOYE**JET**), **11(2), 241-244.**
https://dx.doi.org/10.46792/fuoyejet.v11i2.1539

2 RELATED WORK

Earlier machine learning approaches to phishing detection relied heavily on blacklisting, heuristic rule-sets, and manual feature engineering. Chiew et al. (2018) combined content and URL features in an ensemble approach, yet reported false positive rates above 30% in adversarial conditions. Karim et al. (2023) proposed a hybrid machine learning system for URL-based phishing detection published in IEEE Access, achieving 99.5% accuracy through a combination of logistic regression, SVM, and decision tree classifiers with soft voting, demonstrating the value of ensemble strategies. Huang et al. (2021) applied convolutional networks to raw URL character sequences, achieving 97.1% accuracy, while Akinyelu and Adewumi (2021) combined domain and URL features with a Random Forest classifier, reporting 94.3% accuracy on a balanced dataset. Ghalechyan et al. (2024) conducted an empirical study of neural network approaches to URL phishing detection in Scientific Reports, demonstrating that probabilistic neural networks trained on open-source datasets can achieve production-grade performance without manual feature engineering. Prasad and Chandra (2024) introduced PhiUSIIL, a phishing URL detection framework in Computers & Security that employs similarity indexing and incremental learning to handle concept drift in evolving phishing campaigns. Alkhalil et al. (2021) reviewed machine learning approaches and concluded that neural network-based methods consistently outperform shallow classifiers under distribution shift.

© 2026 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

Jadhav and Chandre (2025) surveyed and comparatively analysed phishing detection techniques across current trends and challenges, identifying the lack of web-deployable, retrainable systems as a critical gap in the literature. Sahoo et al. (2019) and Buber et al. (2021) surveyed feature engineering strategies and found that automatic representation learning via vectorization reduces both design effort and false positive rates compared to hand-crafted feature sets. Despite strong results, many prior systems require expert-designed features and lack practical web-facing deployment—a gap this work addresses.

3 MATERIALS and METHODS

3.1 Dataset Collection

Three publicly available repositories were used to assemble a corpus of 11,055 URLs: PhishTank contributed 6,420 confirmed phishing URLs; the UCI Machine Learning Repository Phishing dataset provided 3,120 labelled samples; and the Alexa Top Sites list supplied 1,515 legitimate URLs. The resulting corpus contained 5,640 phishing samples (51%) and 5,415 legitimate samples (49%), providing a near-balanced class distribution that prevented either class from dominating model training.

3.2 Data Preprocessing

Preprocessing proceeded in four sequential steps: (1) Data cleaning — duplicate URLs were removed and rows with missing values deleted, yielding a clean corpus; (2) Normalisation — all URLs were lowercased and scheme prefixes (http://, https://) stripped to reduce vocabulary size; (3) Feature extraction — Count Vectorization was applied using a character-level tokeniser, transforming each URL into a sparse vector of token frequencies across a vocabulary of approximately 15,000 unique tokens. Count Vectorization was chosen over alternatives such as TF-IDF and manual URL-specific features (e.g., domain length, IP presence, subdomain count) for three reasons: its computational simplicity enables low-latency inference on modest hardware; it requires no domain expertise, supporting fully automated retraining; and Ghalechyan et al. (2024) demonstrated that automatic token-level representations from raw URLs capture discriminative phishing patterns with accuracy comparable to hand-engineered feature sets. No manual feature engineering was required; (4) Train/test split — data were divided into 8,844 training samples (80%) and 2,211 test samples (20%) using stratified random sampling to preserve the class distribution in both subsets.

3.3 ANN Architecture

The ANN was implemented using the Keras Sequential API. The input layer accepted Count Vectorization feature vectors (vocabulary-sized sparse vectors). The first hidden layer comprised 64 neurons with ReLU activation, followed by a Dropout layer (rate = 0.5). The second hidden layer contained 32 neurons with ReLU activation and a second Dropout layer (rate = 0.5). The output layer used a single neuron with sigmoid activation, where scores ≥ 0.5 indicate phishing. The network totalled 124,673 trainable parameters. Figure 1 illustrates the architecture.

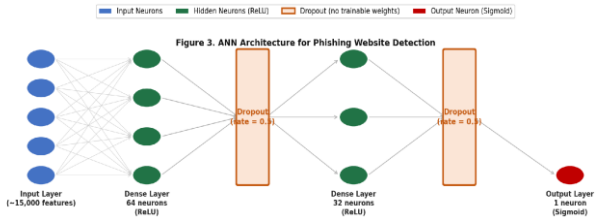

Fig 1. ANN Architecture for Phishing Website Detection

3.4 Model Training and Optimization

Training ran on an Intel Core i5-1135G7 (4 cores, 2.40 GHz, 16 GB RAM) using TensorFlow and Keras. The Adam optimizer was selected for its adaptive per-parameter learning rates, combined with binary cross-entropy loss suitable for binary classification. Training was capped at 50 epochs with a batch size of 32. To prevent unnecessary computation and overfitting, training was halted automatically when validation loss showed no improvement for three consecutive epochs (early stopping, patience = 3). The best-performing weights were preserved via model checkpointing. Upon completion, both the trained model and the fitted Count Vectorizer were saved to HDF5 and Pickle formats respectively, enabling deployment reuse without retraining.

3.5 Performance Metrics

Model performance was assessed using Accuracy, Precision, Recall, and F1-Score, defined as:

Accuracy = (TP + TN)/(TP + TN + FP + FN);

Precision = TP/(TP + FP)

Recall = TP/(TP + FN);

F1-Score = 2 × (Precision × Recall)/(Precision + Recall)

Precision quantifies the proportion of phishing predictions that were genuinely phishing (complement of the false positive rate). Recall measures the proportion of actual phishing URLs correctly identified. F1-Score provides a harmonic balance between both.

3.6 Web Application Development

A Flask-based web application was developed for real-time deployment. The frontend (HTML5, CSS3, JavaScript, Tailwind CSS) provides a URL input field, prediction button, results area, and a ten-entry history table. The backend REST endpoint accepts a URL in JSON, preprocesses it with the saved Count Vectorizer, passes it to the loaded ANN, and returns the classification result and confidence score. An additional endpoint supports on-demand model retraining.

© 2026 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

4 RESULTS

4.1 Training Performance

Figure 1 presents the training and validation curves. Training accuracy rose from 53.9% at epoch one to 99.4% at epoch eight, where validation accuracy peaked at 97.5%. Model checkpointing saved the best-performing weights at this point. Validation accuracy remained stable between 96.5% and 97.5% through epoch fifteen, confirming generalisation. Early stopping triggered at epoch eighteen. The deployed model corresponds to the checkpoint at epoch eight, achieving 99.4% training accuracy (loss = 0.02) and 97.5% validation accuracy (loss = 0.14). The narrow training-validation gap of 3.4% confirms effective dropout regularisation with no significant overfitting.

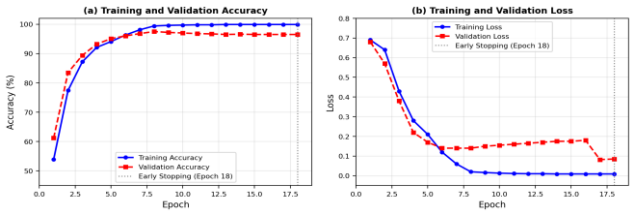

Fig 2. Training and Validation Accuracy (a) and Loss (b) Curves over 18 Epochs

4.2 Classification Performance

Table 1 compares the proposed ANN with three baseline classifiers trained and tested on the same dataset. The ANN outperformed all baselines on every metric. The confusion matrix (Figure 2) shows that 2,134 of 2,211 test URLs were correctly classified: 731 true phishing detections and 1,403 true legitimate classifications, with 40 phishing URLs missed and 37 legitimate URLs incorrectly flagged. The false positive rate was 2.6% and false negative rate was 5.2%. Figure 4 further illustrates the accuracy and F1-score gap between the proposed ANN and all three baselines.

Table 1. Performance Comparison of Classifiers on the 2,211-Sample Test Set

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	87.3	85.6	84.7	85.1
Support Vector Machine	89.5	88.1	87.6	87.9
Random Forest	91.8	90.5	89.9	90.2
Proposed ANN	**96.5**	**95.2**	**94.8**	**95.0**

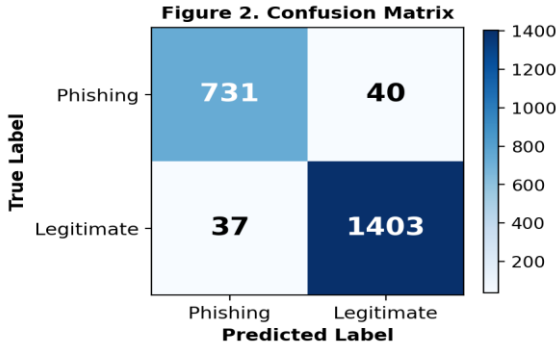

Fig 3. Confusion Matrix on the 2,211-Sample Test Set

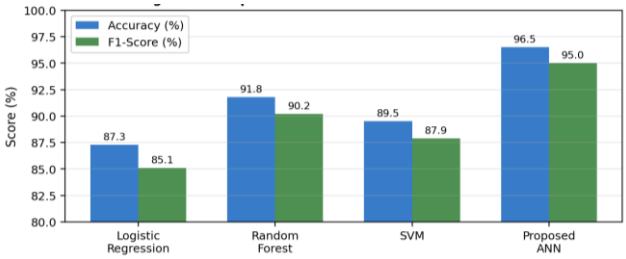

Fig 4. Accuracy and F1-Score Comparison Across Classifiers

4.3 Web Application Validation

Real-world validation confirmed system effectiveness. Legitimate URLs—including google.com, lautech.edu.ng, and facebook.com—were correctly classified with confidence scores above 95%. Known phishing URLs—including secure-paypal-login.com/account/verify and bank-verification-required.info—were correctly flagged with confidence scores exceeding 90%, at an average response time of 2.1 seconds. Figure 4.4 shows the deployed web application interface displaying a real-time phishing detection result alongside the scan history table.

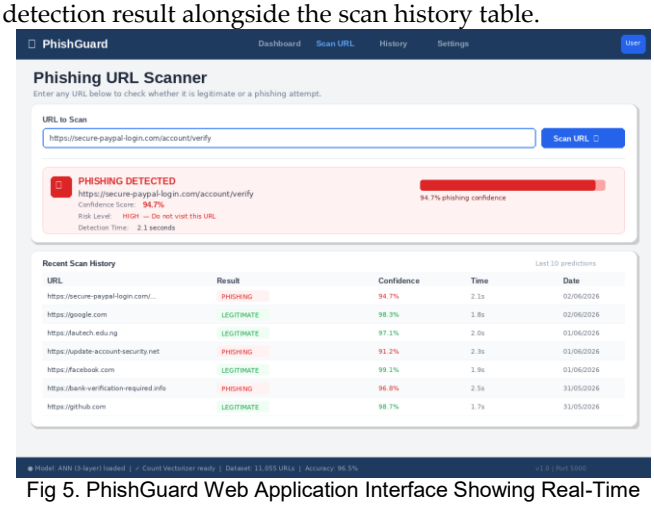

Fig 5. PhishGuard Web Application Interface Showing Real-Time Phishing Detection Result and Scan History

5 RESULT AND DISCUSSION

The proposed ANN achieved 96.5% accuracy, substantially exceeding all three baselines. The 2.6% false positive rate compares favourably with ensemble-based systems reported by Chiew et al. (2018), which recorded rates above

© 2026 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://.doi.org/10.46792/fuoyejet.v11i2.1539 engineering.fuoye.edu.ng/journal

30% in adversarial settings. Recent work by Akinyelu and Adewumi (2021) and Alkhalil et al. (2021) reported accuracy in the 91–94% range using manually engineered features, whereas the proposed system achieves higher accuracy without any hand-crafted features. This finding corroborates Buber et al. (2021), who noted that automatic representation learning reduces both design effort and false positives. Count Vectorization automated feature learning from raw URL tokens, improving adaptability: the model can be updated by retraining without architectural changes. The three-layer design balanced representational capacity and inference speed. Dropout regularisation effectively controlled overfitting despite near-perfect training accuracy, with the small training–validation gap persisting to epoch eighteen confirming good generalisation.

Limitations include reliance on URL-only features—sophisticated attacks with legitimate-looking URLs but fraudulent content may evade detection. Performance may also degrade for regional phishing campaigns underrepresented in the training data, and adversarial URL perturbations remain an open concern requiring future adversarial training strategies.

6 CONCLUSION

This study developed and deployed an ANN-based phishing detection system achieving 96.5% accuracy, 95.2% precision, 94.8% recall, and 95.0% F1-score on a 2,211-sample test set. Count Vectorization automated feature extraction from raw URLs, eliminating manual feature engineering while outperforming Logistic Regression, SVM, and Random Forest baselines. The Flask web application demonstrated practical viability with a 2.1-second response time accessible to non-technical users. Beyond the quantitative results, this work demonstrates that a lightweight, fully automated pipeline—requiring no cybersecurity domain expertise to deploy or retrain—can achieve accuracy competitive with more resource-intensive deep learning systems. This has meaningful implications for organisations with limited security infrastructure, where periodic model retraining against emerging phishing campaigns can be automated without architectural changes. The low false positive rate of 2.6% is particularly significant for real-world trust: users exposed to excessive false alarms tend to disengage from security tools entirely. Future work should integrate content-based signals, real-time threat feeds from services such as VirusTotal and OpenPhish, advanced architectures (CNNs, RNNs, Transformers), explainable AI, and adversarial training to improve robustness against evasion attacks.

ACKNOWLEDGEMENT

The authors express profound gratitude to Almighty God for guidance throughout this research. Special appreciation to Prof. (Mrs.) O.I. Omotosho for invaluable supervision and continuous support. Gratitude is extended to the Department of Cyber Security Science, Ladoke Akintola University of Technology, for providing the academic environment supporting this work.

REFERENCES

Akinyelu, A., & Adewumi, A. (2021). Phishing website detection using domain and URL-based features. Cybersecurity, 4(1), 1–14. https://doi.org/10.1186/s42400-021-00089-9

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. Frontiers in Computer Science, 3, Article 563060. https://doi.org/10.3389/fcomp.2021.563060

Anti-Phishing Working Group (APWG). (2022). Phishing activity trends report, Q4 2022. Retrieved 15 December 2024 from https://www.apwg.org/resources/apwg-reports/

Buber, E., Diri, B., & Sahingoz, O. K. (2021). Detecting phishing attacks from URL by using NLP and machine learning methods. International Journal of Intelligent Systems and Applications, 13(1), 1–13. https://doi.org/10.5815/ijisa.2021.01.01

Chiew, K. L., Chang, E. H., & Tiong, W. K. (2018). Utilisation of website logo for phishing detection. Computers and Security, 73, 500–517. https://doi.org/10.1016/j.cose.2017.11.018

Ghalechyan, H., Israyelyan, E., Arakelyan, A., Hovhannisyan, G., & Davtyan, A. (2024). Phishing URL detection with neural networks: An empirical study. Scientific Reports, 14, Article 25134. https://doi.org/10.1038/s41598-024-74725-6

Huang, Y., Yang, Q., Qin, J., & Wen, W. (2021). Phishing URL detection via CNN-based character-level representation. Security and Communication Networks, 2021, Article 5537090. https://doi.org/10.1155/2021/5537090

Jadhav, A., & Chandre, P. R. (2025). Survey and comparative analysis of phishing detection techniques: Current trends, challenges, and future directions. IAES International Journal of Artificial Intelligence, 14(2), 853–866. https://doi.org/10.11591/ijai.v14.i2.pp853-866

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., & Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805–36822. https://doi.org/10.1109/ACCESS.2023.3267366

Prasad, A., & Chandra, S. (2024). PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning. Computers and Security, 136, Article 103545. https://doi.org/10.1016/j.cose.2023.103545

Sahoo, D., Liu, C., & Hoi, S. C. H. (2019). Malicious URL detection using machine learning: A survey. ACM Computing Surveys, 52(2), 1–35. https://doi.org/10.1145/3264419

© 2026 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.

This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)