Hybrid Machine Learning and Deep Learning Framework for Malicious URL Classification

Gangala Naga Sai Kiran[1], Ms. Ch. Dhana Lakshmi[2],Mr.M.Chiranjeevi[3]
[1]M.Tech Scholar (Reg. No. 24811D5803), Department of Computer Science and Engineering
[2]Assistant Professor, Department of Computer Science and Engineering
Avanthi Institute of Engineering and Technology (Autonomous)
[3]Associate professor, Department of Computer Science and Engineering
Avanthi Institute of Engineering and Technology (Autonomous)
Tamaram, Makavarapalem, Narsipatnam, Anakapalli District – 531116, Andhra Pradesh, India

***Abstract*—** The rapid expansion of the internet has dramatically increased the risk of malicious URLs, which serve as primary vectors for phishing attacks, malware distribution, ransomware, and data breaches. This paper presents a comprehensive comparative study of machine learning (ML) and deep learning (DL) techniques for the classification of URLs as benign or malicious. Traditional ML algorithms—Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, and Support Vector Classifier (SVC)—alongside a Multi-Layer Perceptron (MLP) deep learning model are evaluated. Key URL features including protocol, domain, path, subdomains, HTTPS status, and domain age are extracted and engineered. Class imbalance is addressed using SMOTE (Synthetic Minority Over-sampling Technique). Models are assessed using accuracy, precision, recall, and F1-score. Gradient Boosting achieved the highest classification accuracy of 99.21%, closely followed by Random Forest at 98.88%. The findings demonstrate that ensemble learning methods consistently outperform individual classifiers, offering a robust and scalable approach to real-time malicious URL detection in cybersecurity applications.

Keywords: Malicious URL detection, phishing, machine learning, deep learning, Gradient Boosting, Random Forest, Convolutional Neural Networks, SMOTE, feature engineering, cybersecurity.

1. INTRODUCTION

The rapid advancement of digital platforms has significantly broadened the attack surface for cybercriminals. Malicious URLs are among the most pervasive tools employed by attackers, functioning as gateways to phishing scams, malware downloads, ransomware campaigns, and command-and-control servers. Phishing URLs impersonate legitimate websites to deceive users into disclosing sensitive credentials or financial information, while others redirect users to sites hosting drive-by downloads that compromise system integrity.

Traditional URL detection methods—blacklists, whitelists, and pattern-based heuristics—are inherently reactive, requiring continuous manual updates and failing to detect zero-day threats or URLs employing typosquatting and subtle obfuscation. For example, replacing 'google.com' with 'g00gle.com' evades keyword-matching rules while deceiving human users. DNS and IP-based filtering, though useful, are easily circumvented by domain rotation tactics.

Machine learning offers a paradigmatic shift: instead of relying on static rules, ML models learn discriminative patterns from labeled datasets of benign and malicious URLs. Feature extraction pipelines capture lexical attributes (URL length, special character frequency, suspicious keywords), host-based attributes (IP address, WHOIS registration, domain age), and network-based attributes (DNS activity, HTTP traffic patterns). These features enable classifiers to generalise across previously unseen URL structures.

Deep learning further extends detection capability by eliminating manual feature engineering. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs/LSTMs) analyse raw URL character sequences directly, capturing structural patterns and contextual semantics—such as the presence of deceptive phrases like 'secure-login' or 'account-verification'—that simpler models may overlook. The integration of ML and DL with real-time threat intelligence feeds further reduces detection latency, enabling proactive defence against fast-emerging threats.

1.1 Motivation

The motivation for this research arises from the inadequacy of existing URL detection systems in handling the volume, velocity, and variety of modern cyber threats. With billions of URLs generated daily, manual or rule-based classification is computationally intractable. Moreover, adversarial techniques such as URL obfuscation, domain generation algorithms (DGAs), and semantic mimicry continue to evolve faster than blacklist-based systems can respond. There is an urgent need for adaptive, data-driven classification models that can detect novel threats with high accuracy, low false-positive rates, and real-time processing capability. This study is motivated by the goal of identifying the optimal ML and DL combination for practical, deployable malicious URL detection.

1.2 Problem Statement

Despite significant research on malicious URL detection, current systems face several unresolved challenges: (1) Traditional methods cannot detect zero-day or obfuscated threats; (2) Existing ML models struggle with non-linear, high-dimensional URL feature spaces; (3) Deep learning models require extensive preprocessing and computational resources; (4) Class imbalance in URL datasets leads to biased classifier performance; (5) Lack of contextual awareness—domain reputation history and temporal traffic patterns—limits detection accuracy. This study addresses these gaps through comprehensive feature engineering, SMOTE-based imbalance correction, and comparative evaluation of eight ML and DL classifiers.

1.3 Objectives

The primary objectives of this research are as follows:

- To collect and preprocess a comprehensive labeled dataset of benign and malicious URLs from publicly available sources.
- To engineer and select high-predictive-power features from URL structures using lexical, host-based, and network-based feature extraction.
- To apply dimensionality reduction (PCA) and handle class imbalance using SMOTE.
- To implement and comparatively evaluate eight ML and DL classifiers: Logistic Regression, KNN, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting, MLP, and SVC.
- To identify the optimal model based on accuracy, precision, recall, and F1-score metrics.

- To provide actionable recommendations for real-world cybersecurity deployment.

2. LITERATURE REVIEW

[1] Shafin (2024) proposed an Explainable AI (XAI)-based feature selection framework integrating SHAP and LIME for web phishing detection. Random Forest and XGBoost models combined with the SLA-FS framework achieved 97.41% and 97.21% accuracy respectively, emphasising the role of interpretability in ML-based phishing detection [1].

[2] Saravanan and Subramanian (2020) presented a Genetic Algorithm (GA)-based feature selection method coupled with an ARTMAP neural network classifier. The Gen Fea module optimised feature selection while Phi Dec performed classification, outperforming traditional Naïve Bayes, Random Forest, and SVM baselines with lower error rates [2].

[3] Subasi and Kremic (2020) compared AdaBoost and Multi-Boosting ensemble techniques for phishing detection. AdaBoost with SVM achieved the highest accuracy of 97.61%, confirming ensemble methods' superior classification performance, particularly in handling noisy and diverse phishing datasets [3].

[4] Prabakaran et al. (2023) proposed an enhanced phishing detection mechanism integrating Variational Autoencoders (VAE) with deep neural networks. The VAE model automatically extracted latent URL features through input reconstruction, achieving 97.45% accuracy with a response time of 1.9 seconds [5].

[5] Wei and Sekiya (2022) demonstrated that ensemble ML methods achieve superior detection rates on phishing website classification tasks, confirming Random Forest and Gradient Boosting as leading approaches when evaluated across multiple accuracy, AUC, and F1-score dimensions [6].

[6] Alnemari and Alshammari (2023) proposed a domain feature-based ML approach for phishing domain detection, highlighting the critical importance of domain registration patterns and WHOIS data in distinguishing malicious from legitimate domains [7].

[7] Tang and Mahmoud (2021) conducted a comprehensive survey of ML-based phishing website detection, cataloguing feature types, datasets, and algorithm performance across 40+ studies. Their taxonomy informs the feature engineering strategy adopted in this study [8].

[8] Kapan and Gunal (2023) performed a comprehensive evaluation of classifiers and URL features for improved phishing detection, identifying optimal feature-classifier combinations that maximise detection performance while maintaining computational efficiency [11].

[9] Divakaran and Oest (2022) reviewed ML and DL approaches for phishing detection, noting that hybrid CNN-RNN architectures achieve state-of-the-art performance on sequential URL character modelling tasks, motivating the inclusion of the MLP model in this comparative study [17].

[10] Yerima and Alzaylaee (2020) demonstrated that CNN-based models achieve high accuracy (>97%) in phishing detection through character-level URL analysis, validating deep learning's advantage in capturing structural anomalies without manually engineered features [18].

3. EXISTING SYSTEM

Existing systems for malicious URL detection rely predominantly on three approaches, each with distinct functional characteristics and inherent limitations.

Blacklist and Heuristic-Based Systems: These systems maintain curated repositories of known malicious URLs and apply rule-based pattern matching using regular expressions and keyword lookups. While simple to implement and effective for known threats, they are inherently reactive. Their inability to detect zero-day threats, newly registered malicious domains, and obfuscated URLs—such as those employing typosquatting, Unicode homoglyphs, or excessive subdomain nesting—renders them insufficient as standalone defences. Constant manual curation is required to maintain relevance.

Traditional Machine Learning Methods: Logistic Regression, SVM, and early Random Forest implementations trained on manually extracted URL feature sets improved upon blacklisting by enabling generalisation to unseen URLs. However, these models are limited in capturing complex non-linear relationships within high-dimensional URL feature spaces. Feature engineering is labour-intensive, and model accuracy degrades when attackers deliberately engineer URLs to evade known feature patterns. Class imbalance in real-world URL datasets further compounds accuracy limitations.

Early Deep Learning Approaches: CNN and LSTM-based URL classifiers introduced end-to-end character-level URL analysis, reducing dependence on manual feature engineering. These models demonstrated improved performance on obfuscated URLs. However, they require extensive training data, are computationally intensive, and are susceptible to adversarial perturbations. The lack of explainability—the 'black box' problem—hinders analyst trust and operational deployment in regulated cybersecurity environments.

Collectively, existing systems suffer from: limited non-linear pattern detection capability; over-reliance on static known-threat repositories; high-dimensional data challenges leading to overfitting or noisy predictions; and absence of contextual awareness incorporating domain reputation history and temporal traffic behaviour. This study addresses these limitations through a multi-algorithm comparative framework augmented with comprehensive feature engineering and class-balancing techniques.

4. PROPOSED SYSTEM

The proposed system is a multi-stage, ML and DL-integrated pipeline designed to classify URLs as phishing (malicious) or legitimate (benign) with high accuracy, interpretability, and real-time applicability. The system is structured across five functional layers: data collection, preprocessing and feature engineering, model training and optimisation, evaluation and comparison, and deployment.

Data Collection: The Kaggle phishing dataset (phishing.csv) is used as the primary data source, providing a labelled collection of URL instances with features spanning protocol, domain, path, query parameters, SSL certificate status, domain age, subdomains count, anchor URL patterns, and website traffic ranking. This dataset is periodically supplemented with entries from PhishTank and Open Phish for real-time validation.

Preprocessing and Feature Engineering: Null values are handled and duplicates removed. Outlier detection employs Z-score and IQR methods. Features are standardised using StandardScaler. Key URL attributes extracted include: HTTPS status (SSL certificate presence), domain age (newly registered domains correlate with phishing activity), subdomains count (phishing sites frequently use multiple subdomains), website traffic (legitimate sites typically exhibit higher traffic volumes),

and anchor URL patterns (whether hyperlinks point to external or internal domains). Correlation analysis is performed to identify and retain high-predictive-power features. Dimensionality reduction via PCA eliminates redundant features.

Class Imbalance Handling: SMOTE (Synthetic Minority Over-sampling Technique) generates synthetic samples for the underrepresented phishing class, ensuring balanced class distribution in the training set and preventing majority-class bias in classifier predictions.

The system implements eight classification models:

- Logistic Regression — linear baseline classifier.
- K-Nearest Neighbors (KNN) — distance-based non-parametric classifier.
- Naïve Bayes (Gaussian) — probabilistic classifier based on Bayes' theorem.
- Decision Tree — interpretable tree-based model.
- Random Forest — ensemble of decision trees reducing overfitting.
- Gradient Boosting — iterative boosting ensemble for complex feature interactions.
- Multi-Layer Perceptron (MLP) — deep learning neural network.
- Support Vector Classifier (SVC) — kernel-based classifier optimised via Grid Search CV.

Performance is evaluated using accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC. The best-performing model is serialised using Python's pickle library for deployment. The system architecture supports integration with browser extension interfaces and REST API endpoints for real-time URL scanning, with future-scope integration of Google Safe Browsing and VirusTotal threat intelligence APIs for cross-validation.

Advantages of the Proposed System: High classification accuracy through ensemble learning; automated feature-based analysis eliminating manual URL review; scalable architecture extensible to email, social media, and mobile phishing detection; real-time processing with minimal false positives; and user-facing deployability as a browser extension or web application.

5. RESULTS AND DISCUSSIONS

5.1 Performance Metrics

All eight models were trained on the SMOTE-balanced training set (80%) and evaluated on a held-out test set (20%) using standard classification metrics. The evaluation formulae are:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives.

5.2 Model Comparison Results

Table I presents the comprehensive evaluation results for all eight classifiers across accuracy, precision, recall, and F1-score dimensions.

Model	Accuracy	Precision	Recall	F1-Score

Logistic Regression	96.31%	96.00%	96.00%	96.00%
K-Nearest Neighbors	96.09%	96.00%	96.00%	96.00%
Naïve Bayes	92.40%	92.00%	92.00%	92.00%
Decision Tree	98.42%	98.00%	98.00%	98.00%
Random Forest	98.88%	99.00%	99.00%	99.00%
Gradient Boosting	99.21%	99.00%	99.00%	99.00%
Multi-Layer Perceptron	97.63%	98.00%	98.00%	98.00%
Support Vector Classifier	97.08%	97.00%	97.00%	97.00%

TABLE I. CLASSIFICATION PERFORMANCE METRICS OF ALL MODELS (%)

5.3 Feature Importance Analysis

Table II summarises the relative importance of the top URL features identified through correlation analysis and feature ranking from the Random Forest and Gradient Boosting models.

Feature	Description	Importance Rank
HTTPS Status	Presence of valid SSL certificate	1
Domain Age	Days since domain registration	2
Subdomains Count	Number of nested subdomains	3
URL Length	Total character count of URL	4
Website Traffic	Alexa/traffic rank of domain	5
Anchor URL Ratio	Ratio of external to internal links	6
Special Characters	Count of @, -, //, %20, etc.	7
IP Address Usage	IP used instead of domain name	8

TABLE II. TOP URL FEATURE IMPORTANCE RANKING

5.4 Discussion

Gradient Boosting Classifier achieved the highest overall accuracy of 99.21%, followed by Random Forest at 98.88% and Decision Tree at 98.42%. The superior performance of ensemble methods is attributable to their ability to combine multiple weak learners, reducing variance and overfitting while capturing complex non-linear interactions among URL features that individual models cannot effectively model.

The Multi-Layer Perceptron achieved 97.63% accuracy, demonstrating deep learning's strong representation capability even with tabular URL features. However, MLP required significantly longer training time and more careful hyperparameter tuning—including dropout regularisation, learning rate scheduling, and batch normalisation—compared to tree-based ensemble methods.

Naïve Bayes registered the lowest accuracy at 92.40%, attributable to its conditional independence assumption, which is violated in URL feature datasets where lexical, host-based, and network-based features exhibit significant inter-correlation. Logistic Regression (96.31%) and KNN (96.09%) provided competitive baselines but are limited by linear decision boundaries and computational inefficiency at scale, respectively.

SMOTE-based class balancing demonstrably improved minority-class (phishing) recall across all models, reducing false-negative rates that would otherwise allow malicious URLs to bypass detection. PCA-based dimensionality reduction further improved training efficiency without measurable accuracy degradation, confirming feature redundancy in the raw URL feature space.

The SVC, optimised through Grid Search hyperparameter tuning, achieved 97.08% accuracy with strong kernel-based non-linear classification capability. Its performance is competitive but computationally more expensive than Random Forest for equivalent accuracy levels, making Random Forest the preferred choice for deployment in resource-constrained real-time detection environments.

6. CONCLUSION

This paper presents a comprehensive comparative evaluation of eight ML and DL classifiers for malicious URL detection. The Kaggle phishing dataset was subjected to rigorous preprocessing—null value handling, outlier treatment via Z-score and IQR methods, feature standardisation, SMOTE class balancing, and PCA-based dimensionality reduction—to ensure optimal data quality for model training.

Among the classifiers evaluated, Gradient Boosting achieved the highest accuracy of 99.21%, closely followed by Random Forest (98.88%), Decision Tree (98.42%), and MLP (97.63%). Ensemble learning methods consistently outperformed individual classifiers due to their inherent ability to reduce variance and model complex feature interactions within high-dimensional URL datasets.

The study demonstrates that effective malicious URL detection requires a holistic approach combining robust feature engineering, class-imbalance correction, and ensemble-based classification. The Random Forest model represents the optimal balance between accuracy (98.88%), interpretability, and computational efficiency for real-time deployment. The Gradient Boosting model is recommended for batch processing or offline threat analysis pipelines where maximal classification accuracy is paramount.

Future work will explore the integration of LSTM and CNN models for character-level sequential URL analysis, enabling detection of sophisticated obfuscation techniques. Explainable AI frameworks (SHAP, LIME) will be incorporated to enhance model transparency and analyst trust. Real-time browser extension deployment with VirusTotal and Google Safe Browsing API integration is planned to validate performance against live phishing campaigns in production cybersecurity environments.

REFERENCES

[1] S. S. Shafin, "An Explainable Feature Selection Framework for Web Phishing Detection with Machine Learning," Data Science and Management, Aug. 2024. doi: 10.1016/j.dsm.2024.08.004.

[2] P. Saravanan and S. Subramanian, "A Framework for Detecting Phishing Websites using GA-based Feature Selection and ARTMAP-based Website Classification," Procedia Computer Science, vol. 171, pp. 1083–1092, 2020. doi: 10.1016/j.procs.2020.04.116.

[3] A. Subasi and E. Kremic, "Comparison of AdaBoost with Multi Boosting for Phishing Website Detection," Procedia Computer Science, vol. 168, pp. 272–278, 2020. doi: 10.1016/j.procs.2020.02.251.

[4] A. E. Bolock and S. Madany, "Phishing Susceptibility through Personality, Age, Education Level, and Gender," Medicon Engineering Themes, vol. 8, no. 2, pp. 37–48, Feb. 2025. doi: 10.55162/MCET.08.263.

[5] M. K. Prabakaran, P. M. Sundaram, and A. D. Chandrasekar, "An Enhanced Deep Learning-Based Phishing Detection Mechanism using Variational Autoencoders," IET Information Security, vol. 17, no. 3, pp. 423–440, 2023.

[6] Y. Wei and Y. Sekiya, "Sufficiency of Ensemble Machine Learning Methods for Phishing Websites Detection," IEEE Access, vol. 10, pp. 124103–124113, 2022.

[7] S. Alnemari and M. Alshammari, "Detecting Phishing Domains Using Machine Learning," Applied Sciences, vol. 13, no. 8, 2023.

[8] L. Tang and Q. H. Mahmoud, "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," Machine Learning and Knowledge Extraction, vol. 3, no. 3, pp. 672–694, 2021.

[9] V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing Detection Using Machine Learning Techniques," arXiv preprint arXiv:2009.11116, 2020.

[10] H. Ali et al., "A Review on Data Preprocessing Methods for Class Imbalance Problem," International Journal of Engineering and Technology, vol. 8, no. 3, pp. 390–397, 2019.

[11] S. Kapan and E. S. Gunal, "Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features," Applied Sciences, vol. 13, no. 24, 2023.

[12] S. Raschka, J. Patterson, and C. Nolet, "Machine Learning in Python: Main Developments and Technology Trends," Information, vol. 11, no. 4, 2020.

[13] V. Chang et al., "An Artificial Intelligence Model for Heart Disease Detection Using Machine Learning Algorithms," Healthcare Analytics, vol. 2, p. 100016, 2022.

[14] A. Ullah et al., "Enhancing Phishing Detection Leveraging Deep Learning Techniques," Journal of Computing & Biomedical Informatics, 2024.

[15] T. Ige et al., "An Investigation into the Performances of Current State-of-the-Art Naive Bayes, Non-Bayesian and Deep Learning Based Classifiers for Phishing Detection," arXiv preprint arXiv:2411.16751, 2024.

[16] S. Gopali, A. S. Namin, F. Abri, and K. S. Jones, "The Performance of Sequential Deep Learning Models in Detecting Phishing Websites Using Contextual Features of URLs," arXiv preprint arXiv:2404.09802, 2024.

[17] D. M. Divakaran and A. Oest, "Phishing Detection Leveraging Machine Learning and Deep Learning: A Review," arXiv preprint arXiv:2205.07411, 2022.

[18] S. Y. Yerima and M. K. Alzaylaee, "High Accuracy Phishing Detection Based on Convolutional Neural Networks," arXiv preprint arXiv:2004.03960, 2020.

[19] T. Philippon, "On FinTech and Financial Inclusion," NBER Working Paper No. 26330, National Bureau of Economic Research, 2019.

[20] S. Singh and R. Malik, "Digital Banking Satisfaction Among Indian Millennials," Journal of Retailing and Consumer Services, vol. 58, p. 102340, 2021.

Authors:

Gangala Naga Sai Kiran,
pursuing M.tech ,Department of computer science and Engineering.
Avanthi Institute of Engineering and Technology (Autonomous)
Tamaram,Makavarapalem,Narsipatnam
Anakapalli 531116
E.Mail:gangala.nagasaikiran@gmail.com

CH.DHANA LAKSHMI M.Tech(CSE)
received B.Tech degree in Computer Science and Engineering from JNTU,Kakinada and received M.Tech degree in Computer Science and Engineering from JNTU,Kakinada.Presently working as an Assistant Professor in Department of Computer Science and Engineering in Avanthi Institute of Engineering and Technology(Autonomous),Makavarapalem,Anakapalli,A.P. Her research interests include Data Warehousing and Data Mining,Cloud Computing ,Cyber Security and Cryptography.
E.Mail:dhanalakshmichitikala@gmail.com

M.CHIRANJEEVI M.Tech(CSE) received B.Tech degree in Computer Science and Engineering from JNTU, Hyderabad and received the M.Tech degree in Computer Science and Engineering from Berhampur University.Presently working as an Associate Professor in Department of Computer Science and Engineering in Avanthi Institute of Engineering and Technology(Autonomous),Makavarapalem,Anakapalli,A.P. His research interests include Data Warehousing and Data Mining ,Cloud Computing ,Network Security and RDBMS.He has published more than 15 papers in various national and internationals journals.
E.Mail:chiru.mf@gmail.com