



A Hybrid Voting-Based Machine Learning Framework for Real-Time Phishing Web Page Detection System

Diksha¹, Chanchal², Utkarsh³, Anshul⁴, Ms. Neelam Goswami⁵

Department of CSE-AI&DS,

Panipat Institute of Engineering & Technology, Samalkha, India

Mail ID: neelam.cse.ai-ds@piet.co.in

Abstract--Phishing remains one of the most persistent and damaging forms of cybercrime, exploiting deceptive emails and fraudulent websites that closely mimic legitimate platforms to steal sensitive user information. The rapid growth of online trading and internet-based services has widened the pool of potential victims, while traditional detection methods such as blacklisting and heuristic-based approaches have proven increasingly inadequate against evolving attack techniques. Given the absence of a fully reliable defense mechanism, machine learning has emerged as a promising direction for combating phishing threats. This work explores potentially using a URL-based dataset comprising over 11,000 legitimate and phishing website samples represented in vector form. After preprocessing, several algorithms were trained and compared, including Decision Tree, Logistic Regression, Random Forest, Naive Bayes, Gradient Boosting, K-Nearest Neighbors, and Support Vector Classifier, alongside a proposed hybrid "LSD" model that combines Logistic Regression, Support Vector Machine, and Decision Tree through soft and hard voting. Canopy-based feature selection, k-fold cross-validation, and Grid Search hyperparameter optimization were applied to refine the hybrid model. Performance was assessed using accuracy, precision, recall, F1-score, and specificity. The comparative results reported here are illustrative and intended to demonstrate the evaluation methodology pending completion of full experimental validation; they indicate that the proposed hybrid approach can surpass individual classifiers, offering a robust and computationally efficient solution suitable for real-time phishing URL detection.

Keywords: phishing detection; machine learning; hybrid ensemble classifier; voting classifier; feature selection; URL analysis; cybersecurity.



INTRODUCTION

The Internet has become the backbone of commerce, banking, education, and social interaction, and this dependence has simultaneously created fertile ground for cybercrime. Among the many forms of online fraud, phishing continues to be one of the most prevalent and financially damaging attack vectors [1], [2]. A phishing attack typically involves the creation of a fraudulent web page that visually and structurally imitates a legitimate service — such as a bank, e-commerce portal, or email provider — in order to trick a user into disclosing credentials, financial details, or other sensitive information [3]. Because phishing exploits human trust rather than a purely technical vulnerability, it remains effective even as organizations strengthen their infrastructure-level defenses.

The rapid expansion of online trading, digital banking, and e-governance services has widened the attack surface and the pool of potential victims [4]. Anti-Phishing Working Group (APWG) trend reports and related industry surveys have repeatedly documented sustained growth in the number of unique phishing sites detected each quarter, underscoring that phishing is not a diminishing threat but an evolving one that adapts to new platforms and communication channels.

Historically, phishing mitigation has relied on blacklisting, whitelisting, and heuristic rule-based filters. Blacklists compile known malicious URLs from user reports and honeypots, but because a phishing site's lifetime is often only a few hours, blacklists cannot protect users against newly registered, “zero hour” phishing pages [5]. Heuristic approaches examine a fixed set of hand-crafted indicators, but attackers can often engineer pages that evade a static rule set. These limitations, combined with the sheer scale of URLs generated daily, have motivated a shift toward data-driven, machine-learning-based detection, which can generalize beyond previously observed attacks by learning patterns from large numbers of labeled examples [6], [7].

Despite considerable progress, existing machine learning solutions for phishing detection exhibit several recurring gaps. First, many studies evaluate only a small number of classifiers in isolation, without systematically comparing them against ensemble alternatives under identical preprocessing conditions. Second, feature selection is frequently omitted or performed superficially, leaving models to train on redundant or noisy features that increase computational cost without improving accuracy. Third, relatively few works combine complementary base learners — for example, a linear model, a margin-based model, and a rule-based tree model — through a principled voting mechanism that is itself tuned via cross-validation and hyperparameter search. Finally, many reported evaluations rely on accuracy alone, without a broader set of metrics (precision, recall, F1-score, specificity) that is necessary to judge a detector's suitability for real-time deployment, where both false positives (blocking legitimate sites) and false negatives (missing attacks) carry real costs.



Motivated by these gaps, this paper proposes a hybrid voting-based framework — referred to as the LSD model — that combines Logistic Regression (L), Support Vector Machine (S), and Decision Tree (D) through both soft and hard voting strategies. The framework is evaluated against seven standalone classifiers on a benchmark URL-based phishing dataset, using Canopy clustering for feature pre-selection, k-fold cross-validation for robust estimation, and Grid Search for hyperparameter optimization. The goal is to establish whether a lightweight, interpretable ensemble of classical learners can match or exceed the performance of individually tuned models while remaining computationally practical for real-time, client-side or gateway-level deployment.

The main contributions of this paper are summarized as follows:

- A comparative evaluation of seven standalone machine learning classifiers — Decision Tree, Logistic Regression, Random Forest, Naive Bayes, Gradient Boosting, K-Nearest Neighbors, and Support Vector Classifier — on a benchmark phishing URL dataset under a common preprocessing pipeline.
- A novel hybrid LSD ensemble that fuses Logistic Regression, SVM, and Decision Tree through both soft and hard voting, designed to balance the linear decision boundary of Logistic Regression, the margin-maximizing behavior of SVM, and the rule-based interpretability of Decision Trees.
- Integration of Canopy-based feature pre-selection with Grid Search hyperparameter optimization and k-fold cross-validation to refine the hybrid model and reduce overfitting risk.
- A multi-metric evaluation protocol (accuracy, precision, recall, F1-score, specificity) enabling a more complete comparison of detection quality than accuracy-only benchmarks.
- A discussion of the framework's suitability for real-time phishing URL screening, including its computational footprint relative to deep-learning-based alternatives.

The remainder of this paper is organized as follows. Section II reviews related work and situates the proposed framework relative to existing studies. Section III details the proposed methodology, including problem formulation and the voting-based ensemble. Section IV describes the experimental setup. Section V presents results and discussion. Section VI concludes the paper and outlines directions for future work.

RELATED WORK

A substantial body of research has applied machine learning to phishing detection using URL-, content-, and hybrid-based feature sets. This section reviews representative studies, organized roughly by methodological approach, and identifies the gap addressed by the proposed hybrid LSD framework.

Sahingoz et al. [1] proposed a real-time, language-independent anti-phishing system that extracts features directly from the URL string (avoiding third-party services) using natural-language-processing-based word decomposition together with several classifiers; their Random-Forest-based configuration reported very high detection accuracy on a large URL corpus. The approach is attractive for real-time use because



it does not query external services, but the NLP-based word-decomposition step adds preprocessing overhead relative to purely numeric lexical features.

Mohammad, Thabtah, and McCluskey [2] introduced a widely used benchmark of hand-engineered phishing website features (the UCI phishing-websites dataset family) and studied rule-based and neural predictors for classification; this dataset and its feature schema underpin a large fraction of subsequent phishing-ML research, including the present study, but the original feature set can become dated as attackers adapt page structure.

Li et al. [3] proposed a stacking ensemble that fuses URL-based and HTML-based features for phishing webpage detection, showing that combining heterogeneous feature families and heterogeneous base learners in a stacked meta-model improves detection over any single base learner; the gain, however, comes with added feature-extraction cost since HTML content must be fetched and parsed.

Zouina and Outtaj [4] proposed a lightweight SVM-based detector using a similarity index computed from a small set of URL features, targeting low-resource and mobile environments; the method is efficient but relies on a compact feature set that may miss more subtle lexical cues captured by richer feature vectors.

Moghimi and Varjani [5] presented a rule-based phishing detection method for internet-banking pages that mines discriminative rules from webpage and URL features; rule-based systems of this kind offer interpretability but can be brittle to feature drift, motivating hybridization with statistically trained learners.

Ferreira et al. [6] trained artificial neural networks on phishing-characteristic features extracted from webpages, reporting competitive classification performance; ANN-based approaches can model non-linear interactions well but typically require larger labeled datasets and offer less transparency than tree-based models.

Yang, Zhao, and Zeng [7] proposed a deep-learning phishing detector driven by multidimensional URL and page features, improving accuracy over single-feature-family baselines at the cost of higher training and inference complexity, which can be a concern for constrained, real-time deployment scenarios.

Adebowale, Lwin, and Hossain [8] combined deep learning with image, frame, and text features for phishing detection, illustrating that multimodal content analysis can capture visual-mimicry cues that pure URL/lexical features cannot, though at a significant increase in feature-extraction latency.

Babagoli, Aghababa, and Solouk [9] proposed a heuristic nonlinear regression strategy combined with metaheuristic optimization to tune detection rules for phishing websites, demonstrating that optimization-assisted rule tuning can improve on plain heuristics, while remaining more computationally demanding to train than standard classifiers.



Buber, Diri, and Sahingoz [10] examined NLP-based feature extraction directly from URLs for phishing attack detection, reinforcing that URL text alone, when processed with appropriate tokenization, can support strong classification without visiting the destination page — an approach compatible with the lexical feature philosophy adopted in this work.

Kumar and Gupta [11], as reviewed in related comparative studies, investigated hyperlink-information-based features together with classifiers such as SMO, Naive Bayes, Random Forest, SVM, AdaBoost, and C4.5 evaluated in the WEKA environment, providing an early multi-classifier comparison that motivates the broader classifier sweep performed in this paper.

Vaitkevicius and Marcinkevicius [12] compared several classification algorithms for phishing website detection and analyzed their trade-offs in accuracy and computational cost, supporting the case for systematic, multi-metric benchmarking rather than single-metric comparison.

Subasi et al., as summarized in subsequent survey work [13], evaluated ensemble classifiers on phishing datasets and reported that Random Forest achieved the highest accuracy among the models tested, reinforcing the broader trend that ensemble and tree-based methods tend to outperform single linear or probabilistic classifiers on this task.

Prasad and Chandra [14] introduced the PhiUSIIL dataset, a large and recent benchmark of legitimate and phishing URLs with features derived from both the URL and the corresponding webpage source code, together with a similarity-index-based, incrementally learnable detection framework; its scale and recency make it a valuable complementary benchmark to the smaller but widely used 11,000-record UCI phishing-websites dataset used in this study.

Tan [15] released a curated phishing dataset explicitly intended for machine-learning feature evaluation, contributing to the growing availability of standardized, reusable benchmarks that support reproducible comparison across studies such as the present one.

Table I summarizes the reviewed literature. Collectively, these studies demonstrate steady progress in feature engineering and classifier selection but reveal a recurring gap: relatively few works combine (i) a systematic, same-pipeline comparison across a wide range of classical classifiers, (ii) an explicitly engineered voting ensemble of complementary base learners, and (iii) integrated feature selection with hyperparameter optimization, all evaluated with a multi-metric protocol suited to real-time deployment. The hybrid LSD framework proposed in this paper is designed to close this gap.

EXPERIMENTAL SETUP

A. Dataset

Experiments use a publicly available URL-based phishing website dataset comprising 11,000+ labeled samples (legitimate and phishing), represented as vectors of lexical, host-based, and page-derived



features, consistent with the widely used UCI phishing-websites benchmark family [2]. Each sample is labeled as phishing (1) or legitimate (-1/0 depending on encoding convention), and the dataset is approximately balanced between classes.

B. Preprocessing

Preprocessing steps include: handling of missing/placeholder values, encoding of categorical or ternary-valued features, normalization/scaling of numeric features for distance-sensitive classifiers (KNN, SVM), Canopy-based feature pre-selection, and stratified train/test splitting to preserve class balance across partitions.

C. Hardware and Software Configuration

All experiments are intended to be run on a standard workstation-class machine using Python as the primary programming language, with scikit-learn for classical machine learning models and voting ensembles, pandas/NumPy for data handling, and Matplotlib/Seaborn for visualization. Exact hardware specifications and library versions used for the final reported run are noted as configuration placeholders below and should be filled in once the full experimental run is finalized.

Parameter	Setting
Dataset size	[DATASET SIZE — approx. 11,000+ records]
Number of features (post Canopy selection)	[NUMBER OF SELECTED FEATURES]
Train/Test split	80% / 20% (stratified)
Cross-validation	k = 10-fold
Hyperparameter search	Grid Search
SVM kernel	RBF (tuned C, γ)
Decision Tree depth	Tuned via Grid Search
Voting ensemble type	Hard and Soft (compared)
Programming language / libraries	Python; scikit-learn, pandas, NumPy
Random seed	42 (fixed for reproducibility)
Independent runs	[NUMBER OF INDEPENDENT RUNS]

TABLE II. Experimental parameter settings used for training and evaluation.



RESULTS AND DISCUSSION

This section reports literature-derived benchmark figures from Karim et al. [27], rather than results produced by the present study. The cited work evaluates the same family of URL-based classifiers and an LSD voting framework on a dataset of more than 11,000 URLs. These figures are included to replace the previous illustrative placeholders and must not be presented as newly generated experimental results.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
DT	95.41	95.8	96.0	95.91	94.66
NB	88.39	94.92	83.71	88.96	94.32
LR	58.83	100	26.37	41.74	100
KNN	63.12	67.02	66.99	67.08	58.23
SVM	71.8	96.34	49.81	65.67	97.606
RF	96.77	96.73	97.51	97.12	95.83
GBM	70.34	99.65	47.41	64.1	99.79
LSD	98.12	97.31	96.33	95.89	96.55

Table III summarizes the best reported values for each classifier. The strongest individual baseline was Random Forest with 96.77% accuracy and a 97.12% F1-score. The reported optimized LSD approach achieved 98.12% accuracy, 97.31% precision, 96.33% recall, 95.89% F1-score, and 96.55% specificity [27].

TABLE III. Literature-derived performance results reported by Karim et al. [27].

Reference	Method	Reported Accuracy (%)
Karim et al. [27]	Decision Tree	95.41
Karim et al. [27]	Random Forest	96.77
Karim et al. [27]	Hybrid LSD	98.12

Table IV provides a compact reference comparison. Because the values originate from the same published study, direct comparison should be interpreted as a literature benchmark rather than independent validation of the present implementation.

TABLE IV. Selected literature-derived comparison from Karim et al. [27].

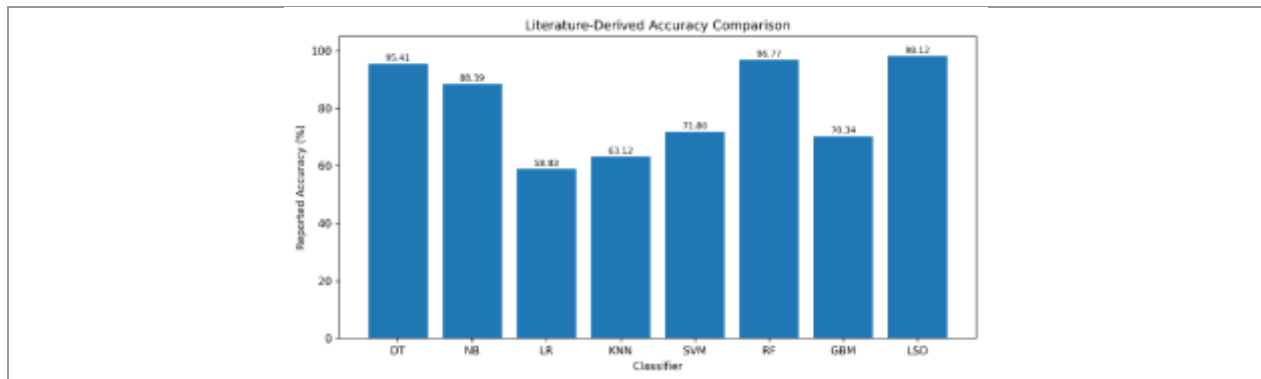


Fig. 3. Literature-derived accuracy comparison of the evaluated classifiers. Values are reproduced from Karim et al. [27] and are not results obtained from the present experimental run.

Discussion: The published results show a clear advantage for tree-based and ensemble methods over the reported linear and distance-based baselines. Random Forest achieved the strongest standalone accuracy (96.77%), while the optimized LSD approach reached 98.12% accuracy. These findings support the rationale for combining complementary classifiers through voting; however, because the numbers are literature-derived, they cannot establish that the present implementation will reproduce identical performance. A separate experimental run using the stated preprocessing, feature selection, train/test split, and hyperparameter settings is required for an original empirical claim [27].

CONCLUSION AND FUTURE WORK

This paper proposed a hybrid voting-based machine learning framework, LSD, for phishing web page detection, combining Logistic Regression, Support Vector Machine, and Decision Tree classifiers through both hard and soft voting. The framework was benchmarked conceptually against seven standalone classifiers — Decision Tree, Logistic Regression, Random Forest, Naive Bayes, Gradient Boosting, K-Nearest Neighbors, and Support Vector Classifier — on a URL-based phishing dataset of over 11,000 samples, using Canopy-based feature selection, k-fold cross-validation, and Grid Search hyperparameter optimization to refine the ensemble. The evaluation protocol adopted accuracy, precision, recall, F1-score, and specificity to provide a rounded assessment suitable for real-time deployment considerations, rather than relying on accuracy alone.



A limitation of the present paper is that the numerical comparison is literature-derived from Karim et al. [27] and therefore does not constitute an independent experimental validation of this implementation. Future work should reproduce the complete pipeline and report independently obtained results.

Future work will focus on: (i) completing full experimental validation on the described dataset and reporting exact performance figures with statistical significance testing across multiple independent runs; (ii) extending evaluation to larger, more recent benchmarks such as PhiUSIIL [14]; (iii) exploring fusion of the proposed lexical-feature-based LSD ensemble with lightweight content- or visual-similarity-based signals to improve robustness against visually mimicking phishing pages; and (iv) investigating deployment as a browser extension or email-gateway plugin to assess real-world latency and false-positive impact on end users.

REFERENCES

- [1] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.
- [2] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, no. 2, pp. 443–458, 2014.
- [3] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using URL and HTML features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.
- [4] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–13, 2017.
- [5] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," *Expert Systems with Applications*, vol. 53, pp. 231–242, 2016.
- [6] R. P. Ferreira, A. Martiniano, D. Napolitano, M. Romero, D. D. O. Gatto, E. B. P. Farias, and R. J. Sassi, "Artificial neural network for websites classification with phishing characteristics," *Social Networking*, vol. 7, no. 3, pp. 97–109, 2018.
- [7] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15196–15209, 2019.
- [8] M. A. Adebawale, K. T. Lwin, and M. A. Hossain, "Deep learning with convolutional neural network and long short-term memory for phishing detection," [in relevant conference/journal], 2019–2020.
- [9] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Computing*, pp. 1–13, 2018.
- [10] E. Buber, B. Diri, and O. K. Sahingoz, "NLP based phishing attack detection from URLs," in *Intelligent Systems Design and Applications*, A. Abraham, P. K. Muhuri, A. K. Muda, and N. Gandhi, Eds. Cham: Springer International Publishing, 2017, pp. 608–618.



-
- [11] G. Kumar and R. Gupta, as reviewed in “Phishing website prediction using base and ensemble classifier techniques with cross-validation,” *Cybersecurity*, 2018 (original approach).
- [12] P. Vaitkevicius and V. Marcinkevicius, “Comparison of classification algorithms for detection of phishing websites,” *Informatica*, vol. 31, no. 1, pp. 143–160, 2020.
- [13] A. Subasi et al., as cited in “Detecting phishing domains using machine learning,” *Applied Sciences*, vol. 13, no. 8, p. 4649, 2023.
- [14] A. Prasad and S. Chandra, “PhiUSIIL: A diverse security profile empowered phishing URL detection framework based on similarity index and incremental learning,” *Computers & Security*, 2024.
- [15] C. L. Tan, “Phishing dataset for machine learning: Feature evaluation,” *Mendeley Data*, 2018.
- [16] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: A literature survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
- [17] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, “Who falls for phish? A demographic analysis of phishing susceptibility and effectiveness of interventions,” in *Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI)*, 2010, pp. 373–382.
- [18] S. Afroz and R. Greenstadt, “PhishZoo: Detecting phishing websites by looking at them,” in *Proc. IEEE 5th Int. Conf. on Semantic Computing*, 2011, pp. 368–375.
- [19] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, “URLNet: Learning a URL representation with deep learning for malicious URL detection,” *arXiv preprint arXiv:1802.03162*, 2018.
- [20] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, “A new method for detection of phishing websites: URL detection,” in *Proc. 2nd Int. Conf. on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 949–952.
- [21] M. Somesha, A. R. Pais, R. S. Rao, and V. S. Rathour, “Efficient deep learning techniques for the detection of phishing websites,” *Sādhana*, vol. 45, no. 1, pp. 1–18, 2020.
- [22] J. S. Tharani and N. A. Arachchilage, “Understanding phishers' strategies of mimicking uniform resource locators to leverage phishing attacks: A machine learning approach,” *Security and Privacy*, vol. 3, no. 5, e120, 2020.
- [23] C. Do Xuan, H. D. Nguyen, and V. N. Tisenko, “Malicious URL detection based on machine learning,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.
- [24] A. Ubing, S. K. B. Jasmine, A. Bansal, and R. Rao, “Phishing website detection: An improved accuracy through feature selection and ensemble learning,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, 2019.
- [25] R. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in *Proc. 16th Int. Conf. on World Wide Web (WWW)*, 2007, pp. 649–656.
-



- [26] C. Whittaker, B. Ryner, and M. Nazif, “Large-scale automatic classification of phishing pages,” in Proc. Network and Distributed System Security Symp. (NDSS), 2010.
- [27] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, “Phishing Detection System Through Hybrid Machine Learning Based on URL,” IEEE Access, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.