

Hybrid Ensemble Learning For Malicious URL Detection: A Literature Review of Machine Learning, Deep Learning, and Feature-Fusion Approaches

Ashwini Sable¹, Vijay More²

Independent Researcher

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150900024>

Received: 11 September 2026; Accepted: 16 September 2026; Published: 01 October 2026

ABSTRACT

The rapid evolution of phishing, malware distribution, and other web-based attacks has made malicious Uniform Resource Locator (URL) detection an important cybersecurity research problem. Traditional blacklist and rule-based mechanisms provide efficient protection against known malicious addresses but may fail when attackers generate previously unseen URLs, employ URL shortening, manipulate lexical structures, or rapidly change domains. Consequently, machine learning and deep learning approaches have increasingly been investigated for detecting malicious URLs using lexical, structural, host-based, contextual, and semantic information. More recently, hybrid and ensemble learning architectures have emerged as promising alternatives to individual classifiers because they combine complementary representations and learning mechanisms. This structured review synthesizes twenty studies published primarily between 2020 and 2026 across IEEE, Elsevier, Springer, and related peer-reviewed venues. The analysis is organized around representation diversity rather than reported accuracy alone and covers handcrafted lexical and structural features, sparse character n-grams, convolutional and recurrent neural representations, transformer-based contextual models, graph neural networks, and heterogeneous ensemble or fusion architectures. The evidence indicates a clear transition from single-representation classifiers toward multi-branch models that combine complementary statistical, sequential, contextual, and relational signals. However, reported benchmark performance is not directly comparable across studies because datasets, class distributions, preprocessing, temporal splits, and operating thresholds vary substantially. Persistent research challenges include cross-dataset generalization, temporal drift, domain-level leakage, class imbalance, false-positive control, adversarial robustness, explainability, and deployment latency. Based on these gaps, the review proposes a four-branch hybrid ensemble architecture and a formal evaluation plan emphasizing ablation, temporal and cross-dataset validation, adversarial testing, low-false-positive operating points, and computational efficiency.

Keywords— malicious URL detection, phishing detection, ensemble learning, hybrid learning, machine learning, deep learning, BERT, Random Forest, boosting, cybersecurity.

INTRODUCTION

The increasing dependence on Internet-based services has expanded exposure to phishing, malware delivery, credential theft, and other attacks that use malicious URLs as the initial interaction point. A URL is therefore not merely an address; it is also a compact artifact containing lexical, structural, and sometimes contextual evidence that can support automated security decisions [1]– [5].

Conventional defenses such as blacklists, reputation systems, and static rules remain important because they are fast and operationally simple. Their principal limitation is coverage: newly registered domains, rapidly changing infrastructure, URL shortening, homograph-like token manipulation, and previously unseen paths can evade exact-match or fixed-rule mechanisms [4] [5] [11]. Machine-learning (ML) and deep-learning (DL) methods address this problem by learning discriminative patterns from benign and malicious examples rather than depending exclusively on prior enumeration.

Over time, the research focus has broadened from selecting a single classifier to combining multiple forms of evidence. Tree-based models are effective for engineered numerical features; linear models efficiently exploit sparse character n-grams; CNNs and recurrent models learn local and sequential patterns; transformers model longer-range contextual dependencies; and graph neural networks can encode relations among URL components or infrastructure entities [2] [7]-[10] [12]-[17]. This diversity motivates hybrid ensemble learning in which different branches contribute genuinely complementary information.

Early machine-learning approaches to phishing and malicious URL detection largely relied on handcrafted characteristics and conventional classifiers. Common algorithms include Logistic Regression (LR), Decision Trees (DT), Support Vector Machines (SVM), Random Forests (RF), Naive Bayes (NB), and k-Nearest Neighbors (KNN) [3] [4] [6] [7] [11]. These approaches can provide relatively efficient classification using lexical, structural, and other engineered characteristics. However, individual classifiers differ in how they represent decision boundaries and exploit the available feature space, motivating research into approaches that combine multiple models or feature representations [4] [6].

Ensemble and hybrid learning approaches address this limitation by integrating complementary classifiers, feature sets, or representations. Methods such as bagging, boosting, voting, and stacking can exploit diversity among individual learners so that the final prediction is not determined by the limitations of a single classifier. This principle is particularly relevant to malicious URL detection because different models can capture different characteristics of an attack: tree-based models can effectively process engineered statistical features, linear models can exploit sparse lexical representations, and neural models can learn more complex representations directly from URL sequences [4] [8] [9].

Deep-learning approaches have further expanded malicious URL detection by reducing dependence on manually designed features and learning representations directly from URL data. Neural architectures and representation-learning approaches have demonstrated the ability to capture patterns that may not be readily expressed through conventional handcrafted features [2] [7]. More recent research also incorporates transformer-based representations and graph-based learning into hybrid architectures, reflecting a broader transition from isolated classifiers toward systems that integrate multiple forms of evidence [8]-[11].

Graph-based approaches are particularly relevant when maliciousness is associated not only with an individual URL but also with relationships among URL components or other entities. [8], for example, proposed a hybrid malicious URL detection approach combining machine-learning classifiers with Graph Neural Networks (GNNs). Such architecture illustrates how structural relationships can complement the statistical and lexical characteristics processed by conventional machine-learning models.

Transformer-based models provide another important direction. Rather than representing URLs exclusively through manually engineered attributes, transformer architectures can learn contextual representations from tokenized URL sequences. Recent work has consequently investigated hybrid systems in which transformer-derived representations are combined with conventional machine-learning components [9] [10]. Lee and Kwon [9] investigate hybrid ensemble learning for malicious URL detection using BERT and boosting models, while Özmen and Yildirim [10] present a DistilBERT-based hybrid architecture for phishing URL detection. These studies demonstrate the growing interest in combining learned contextual representations with complementary classification mechanisms.

The progression of literature therefore suggests a transition from selecting a single optimal classifier toward integrating complementary feature spaces, representations, and learning algorithms. Handcrafted lexical and structural characteristics, sparse character-level representations, transformer-derived contextual representations, graph-based relationships, and ensemble classifiers each capture different aspects of malicious URLs [4] [8]-[10]. A hybrid ensemble architecture can potentially exploit these complementary strengths while reducing dependence on the assumptions and limitations of any individual model.

This review therefore emphasizes representation diversity, evaluation quality, and deployability. Rather than ranking papers solely by reported accuracy, it examines how each study represents URLs, how models are

combined, which datasets and metrics are used, and whether the evaluation design supports claims about generalization. Accordingly, following are the research questions that the paper addresses:

RQ1: Which machine-learning and deep-learning techniques have been used for malicious URL and phishing URL detection?

RQ2: How do ensemble and hybrid approaches improve upon individual classifiers?

RQ3: Which feature representations are most useful for malicious URL classification?

RQ4: What limitations prevent current high-performing models from being reliably deployed in real-world cybersecurity environments?

RQ5: Which research directions are most promising for future hybrid ensemble malicious URL detection systems?

LITERATURE REVIEW

Table I summarizes the 20 studies included in the final synthesis according to publication year, URL representation, model or fusion strategy, principal contribution, and major methodological limitation. Reported performance values are presented only when they are explicitly available in the original studies. Because datasets, class distributions, preprocessing pipelines, and evaluation strategies differ considerably across publications, the values should be interpreted within the context of each study rather than as direct cross-study comparisons.

Authors and Paper Title	Publisher / Year	Representation and Model / Fusion	Research Gaps Identified
1. Maria Sameen, Kyunghyun Han & Seong Oun Hwang — PhishHaven—An Efficient Real-Time AI Phishing URLs Detection System	IEEE Access, Vol. 8 / 2020. DOI: 10.1109/ACCESS.2020.2991403	Representation: Lexical URL features and URL HTML encoding. Model/Fusion: Ensemble machine-learning architecture, PhishHaven, designed for real-time classification of both AI-generated and human-crafted phishing URLs.	Strong reliance on lexical information. More extensive temporal, cross-source and cross-dataset validation is required to determine robustness to evolving phishing patterns. Relational and contextual representations are not extensively exploited.
2. Jianting Yuan, Guanxin Chen, Shengwei Tian & Xinjun Pei — Malicious URL Detection Based on a Parallel Neural Joint Model	IEEE Access, Vol. 9 / 2021, pp. 9464–9472. DOI: 10.1109/ACCESS.2021.3049625	Representation: Semantic and visual URL information. Model/Fusion: Parallel neural joint architecture combining complementary learned representations.	Multi-representation neural processing increases model complexity. Broader cross-dataset, temporal and computational-efficiency evaluation would strengthen evidence of practical generalization.
3. Nadar et al. — Detection of Phishing Websites Using Machine Learning	IEEE, 2nd Global Conference for Advancement in Technology (GCAT) / 2021. DOI: 10.1109/GCAT52182.2	Representation: Handcrafted phishing/website characteristics. Model/Fusion: Logistic Regression, AdaBoost, Gradient Boosting and	Dependence on manually engineered features limits automatic representation learning. Richer contextual and sequential representations, stronger

Approach	021.9587682	stacking ensemble; stacking reported approximately 85.6% accuracy.	external datasets and temporal testing are needed.
4. Malak Aljabri et al. — Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions	IEEE Access / 2022	Representation: Survey of lexical, host/network, content and related malicious-URL features. Model/Fusion: Comparative review of machine-learning approaches and datasets.	Reveals substantial dataset and experimental heterogeneity. There remains a need for standardized datasets, unified evaluation protocols and directly comparable benchmarking of different representation/model families.
5. Survey on Malicious URL Detection Techniques — Survey on Malicious URL Detection Techniques	IEEE, 6th International Conference on Trends in Electronics and Informatics (ICOEI) / 2022. DOI: 10.1109/ICOEI53556.2022.9777221	Representation: Survey taxonomy of webpage/URL features. Model/Fusion: Blacklist-based, rule-based, ML and DL detection taxonomy.	Provides broad taxonomy but predates much of the recent work involving Transformers, GNNs and heterogeneous fusion. No controlled unified benchmark is provided.
6. Joney Kumar — Hybrid Feature-Based Machine Learning Method for Phishing URL Detection	IEEE, Third International Conference on Secure Cyber Computing and Communication (ICSCCC) / 2023. DOI: 10.1109/ICSCCC58608.2023.10176901	Representation: NLP features, word-vector representations, PCA and engineered attributes. Model/Fusion: Five classifiers evaluated; Random Forest with NLP/word-vector hybrid features reported 99.75% accuracy.	Very high within-study performance requires further cross-dataset and temporally separated evaluation. Fusion is predominantly at the feature level rather than heterogeneous model-level fusion.
7. Rawshon Ferdaws & Nahid Ebrahimi Majd — Phishing URL Detection Using Machine Learning and Deep Learning	IEEE World AI IoT Congress (AIIoT) / 2024. DOI: 10.1109/AIIoT61789.2024.10579005	Representation: Engineered features for conventional ML and raw/sequential representations for DL. Model/Fusion: ML compared with CNN and LSTM. Best DL achieved approximately 98.1% accuracy and 98.7% F1-score.	ML and DL components are primarily compared rather than fused. A heterogeneous ensemble could exploit complementary engineered and automatically learned representations.
8. Bimanna Alaladinni, Gangu Dharmanaraju, Yamuna Vadada et al. — A Hybrid Approach for Malicious URL	IEEE, 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI) / 2025. DOI: 10.1109/ICDICI66477.2025.11135390	Representation: URLs tokenized into subdomain, domain and path elements to form token graphs; handcrafted features such as URL length, HTTPS and special-character counts. Model/Fusion: GNN +	Graph construction adds computational overhead. More extensive cross-dataset, temporal and real-time scalability evaluation is needed. The study also motivates stronger mechanisms for genuinely

Detection Using ML Classifiers and Graph Neural Networks		Random Forest classifier framework.	integrating graph and statistical predictions.
9. Lee & Kwon — BERT + structural-feature malicious/phishing URL study (exact title/full author metadata from your source list still requires verification)	Reported as 2026 in your source set; exact venue requires confirmation.	Representation: BERT contextual representations + structural URL features. Model/Fusion: XGBoost/CatBoost together with BERT using weighted ensemble fusion.	Transformer inference introduces latency and resource cost. Further work is needed on calibration, temporal drift, lightweight deployment and independent external validation.
10. Ülkü Özmen & Esra Odabaş Yildirim — DistilBERT-Based Hybrid Architecture for Phishing URL Detection	IEEE Access, Vol. 14 / 2026, pp. 71720–71737. DOI: 10.1109/ACCESS.2026.3684855	Representation: DistilBERT-based semantic representation + handcrafted statistical URL attributes. Model/Fusion: Hybrid late-fusion architecture. Reported accuracy 98.79%, ROC-AUC 0.9983; your source notes detection of 149/150 later phishing URLs.	External temporal evaluation is encouraging but the temporal sample is small and phishing focused. Larger balanced benign/malicious temporal and cross-source datasets, calibration analysis and computational evaluation are needed.
11. Nuria Reyes-Dorta, Pino Caballero-Gil & Carlos Rosa-Remedios — Detection of Malicious URLs Using Machine Learning	Springer, Wireless Networks / 2024	Representation: Engineered malicious-URL features. Model/Fusion: Comparison of conventional ML/DL approaches with proof-of-concept quantum ML experiments.	Broad comparative investigation rather than a dedicated heterogeneous fusion framework. Further cross-source validation and practical real-world deployment evaluation are required.
12. Ruitong Liu, Yanbin Wang, Zhenhao Guo, Haitao Xu, Zhan Qin, Wenrui Ma & Fan Zhang — TransURL: Improving Malicious URL Detection with Multi-Layer Transformer Encoding and Multi-Scale Pyramid Features	Elsevier, Computer Networks / 2024. DOI: 10.1016/j.comnet.2024.110707	Representation: Character-aware Transformer embeddings containing character- and subword-level information. Model/Fusion: Multi-layer Transformer encoding + multi-scale feature learning + spatial pyramid attention.	Importantly, TransURL already evaluates cross-dataset, class-imbalance and adversarial scenarios. Remaining gaps relate more strongly to Transformer computational cost, lightweight deployment, latency and continued robustness under future temporal drift.

13. Ruitong Liu, Yanbin Wang, Haitao Xu, Zhan Qin, Fan Zhang, Yiwei Liu & Zheng Cao — PMANet: Malicious URL Detection via Post-Trained Language Model Guided Multi-Level Feature Attention Network	Elsevier, Information Fusion, Vol. 113 / 2025, Article 102638. DOI: 10.1016/j.inffus.2024.102638	Representation: Character-level + subword-level URL representations, hierarchical Transformer layers, local/global information. Model/Fusion: Post-trained language model + layer-wise attention + spatial pyramid pooling.	PMANet already includes extensive robustness testing. Remaining challenges include computational complexity, inference efficiency, resource-constrained deployment and continuous adaptation to changing URL distributions.
14. Nguyet Quang Do et al. — An Integrated Model Based on Deep Learning Classifiers and Pre-Trained Transformer for Phishing URL Detection	Elsevier, Future Generation Computer Systems, Vol. 161 / 2024, pp. 269–285. DOI: 10.1016/j.future.2024.06.031	Representation: Character-, word- and contextual-level URL representations. Model/Fusion: RasNet + TCMA + MPNet. RasNet integrates embeddings/ResNet; TCMA combines TCN and multi-head self-attention; MPNet provides pretrained contextual representation.	Highly integrated architecture introduces substantial model complexity. Further work should address architecture simplification, explainability, latency, memory usage and resource-efficient deployment.
15. Nguyet Quang Do, Ali Selamat, Ondrej Krejcar & Hamido Fujita — Detection of Malicious URLs Using Temporal Convolutional Network and Multi-Head Self-Attention Mechanism	Elsevier, Applied Soft Computing, Vol. 169 / 2025, Article 112540. DOI: 10.1016/j.asoc.2024.112540	Representation: Character-level and word-level embeddings. Model/Fusion: Temporal Convolutional Network (TCN) + Multi-Head Self-Attention (MHSA); reported 98.78% accuracy.	Needs stronger temporal and cross-source evaluation, adversarial robustness testing, and direct comparison/fusion with contextual Transformer and conventional feature-based models.
16. Kousik Barik, Sanjay Misra & Raghini Mohan — Web-Based Phishing URL Detection Model Using Deep Learning Optimization Techniques	Springer, International Journal of Data Science and Analytics, Vol. 20 / 2025, pp. 4449–4471. DOI: 10.1007/s41060-025-00728-9	Representation: URL/content/external features; StandardScaler and Variational Autoencoder (VAE) for preprocessing/representation. Model/Fusion: Enhanced Grid Search Optimization + CNN (EGSO-CNN). Reported 99.44% accuracy, 99.21% recall and 99.32%	Complex preprocessing and optimization pipelines may increase training and deployment costs. Further independent temporal, adversarial and resource-efficiency evaluation is desirable.

		F1-score.	
17. Tirth Doshi, Vishva Patel, Nemil Shah, Debabrata Swain, Debabala Swain & Biswaranjan Acharya — PhishHunter-XLD: An Ensemble Approach Integrating Machine Learning and Deep Learning for Phishing URL Classification	Elsevier/Franklin Open, Vol. 12 / 2025, Article 100349. DOI: 10.1016/j.fraope.2025.100349	Representation: Explicit lexical features + sequential URL information + DistilBERT contextual semantics. Model/Fusion: XGBoost + LSTM + DistilBERT weighted voting. Reported 99.83% accuracy with a low-latency claim.	This is close to the heterogeneous ensemble direction of your review. Major open questions include independent temporal/adversarial replication, ensemble calibration, ablation/component-contribution analysis, pruning/quantization and zero-day robustness.
18. Authors require confirmation from IEEE metadata — Phishing URL Detection Using Machine Learning: A Comparative Study	IEEE, 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI) / 2025. DOI: 10.1109/ICMCSI64620.2025.10883082	Representation: Engineered URL features. Model/Fusion: Logistic Regression, KNN, SVC, Naive Bayes, Decision Tree and Random Forest comparison. Random Forest reported 94.5% accuracy.	Conventional feature/model comparison does not exploit representation diversity. Future work should evaluate learned representations, ensemble fusion, temporal adaptation and cross-dataset generalization.
19. Jaqueline D. Duarte, Pedro Chagas Junior, João Paulo Javidi da Costa, Elena J. da Costa, Laerte Peotta de Melo, Rafael Rabelo Nunes, Carlos V. N. Gabriel Soares & Thiago Erivan da Cunha Silva — Machine Learning for Early Detection of Phishing URLs in Parked Domains: An Approach Applied to a Financial	IEEE Access, Vol. 13 / 2025, pp. 145736–145753. DOI: 10.1109/ACCESS.2025.3599454	Representation: Newly registered/parked-domain information, SSL certificate-derived and engineered domain characteristics. Model/Fusion: LightGBM, feature engineering and feature selection. Dataset contained 211,659 URLs; reported 97.28% accuracy and 96.02% recall on balanced data with 10-fold CV.	Highly valuable for early detection but specialized toward newly registered and parked domains. Generalization to broader phishing/malicious-URL scenarios and integration with lexical, contextual and content features require further study.

Institution			
20. Ye Tian, Yanqiu Yu, Jianguo Sun & Yanbin Wang — From Past to Present: A Survey of Malicious URL Detection Techniques, Datasets and Code Repositories	Elsevier, Computer Science Review / 2025	Representation: Modality-oriented taxonomy spanning blacklists, handcrafted ML features, DL representations, Transformers, GNNs and LLM-oriented approaches. Model/Fusion: Comprehensive survey rather than a detector; also emphasizes datasets and code repositories.	Identifies the continuing lack of standardized reproducible benchmarking, adequate dataset coverage, modality-aware comparisons and consistent evaluation protocols. As a survey, it does not provide controlled experimental comparison under a single dataset/protocol.

Table I. Comparative summary of the twenty studies included in the revised synthesis.

Summary of Research Gap

Despite reported classification accuracies approaching or exceeding 98–99% in several studies, malicious URL detection cannot yet be considered a solved problem. Recent studies and surveys indicate that high benchmark accuracy does not necessarily imply robustness under changes in data distribution, attack strategies, temporal conditions, or deployment environments [20]–[26].

Cross-Dataset Generalization

Many malicious URL detection studies evaluate models using random partitions of a single benchmark dataset. Such evaluation may not adequately demonstrate whether a model can generalize URLs collected from different sources or under different distributions. Cross-dataset and independent-source evaluation is therefore important for determining whether learned representations capture transferable malicious patterns rather than dataset-specific characteristics [21] [22].

Temporal Generalization and Concept Drift

Phishing campaigns, domain-registration practices, URL structures, hosting infrastructure, and obfuscation strategies continuously evolve. Consequently, models trained on historical datasets may experience performance degradation when exposed to newly emerging attacks. Temporal train/test separation and evaluation under distribution shift should therefore complement conventional random-split experiments [22] [23].

Dataset Leakage

Dataset construction can substantially influence reported performance. Exact duplicates, near-duplicate URLs, augmented variants, or multiple URLs associated with the same domain may create information leakage when closely related samples occur in both training and test partitions. Recent work has consequently emphasized de-duplication and leakage-aware splitting before model training and evaluation [24]. Domain-aware or grouped splitting can provide an even stronger assessment of generalization.

Class Imbalance and Low-FPR Operation

Real-world malicious URL traffic is typically imbalanced, with benign URLs substantially outnumbering malicious examples. Under such conditions, classification accuracy alone may obscure poor malicious class recall or an operationally unacceptable number of false positives. Evaluation should therefore include precision, recall, F1-score, ROC-AUC, PR-AUC, false-positive rate (FPR), and true-positive rate (TPR), particularly at low FPR thresholds relevant to large-scale security systems [22], [25].

Adversarial Robustness

Malicious URL detection operates in an adversarial environment in which attackers can deliberately modify URL tokens, character encodings, path segments, separators, subdomains, homoglyphs, and other lexical or structural characteristics to evade classifiers. Although recent models have begun evaluating adversarial and obfuscated URLs, explicit robustness testing remains an important requirement beyond conventional random-split evaluation [22] [25] [26].

Explainability

Increasing model complexity creates an additional challenge concerning interpretability. Feature importance can be obtained relatively naturally from several conventional machine-learning models, whereas explanations become more difficult for transformers, graph neural networks, and heterogeneous multi-branch architectures. Future detection systems should therefore provide analyst-facing explanations that identify the lexical, structural, contextual, semantic, or relational evidence contributing most strongly to a prediction [21] [23].

Computational Efficiency

Transformers, graph-, and hybrid deep-learning architectures can substantially increase representation capacity, but their computational requirements may limit deployment in browsers, gateways, endpoints, mobile devices, and other resource-constrained environments. Recent research on lightweight transformers and knowledge distillation demonstrates the importance of considering parameter count, training cost, memory requirements, inference latency, and computational overhead in addition to predictive performance [25] [26].

Reproducibility and Benchmark Standardization

Recent survey evidence continues to identify fragmented datasets, heterogeneous preprocessing procedures, inconsistent evaluation protocols, and limited availability of reproducible implementations as obstacles to direct comparison among malicious URL detection methods [20] [23]. Public dataset definitions, explicit train/validation/test splits, de-duplication procedures, source and collection-date information, source code, model configurations, and standardized evaluation metrics would enable more credible and reproducible comparisons. Collectively, these gaps indicate that the central challenge in malicious URL detection is no longer limited to achieving high classification accuracy on a static benchmark. A more important research objective is to develop detection systems that maintain reliable performance when the data source, collection period, attack strategy, class distribution, and operational constraints change [21]–[26]. This observation motivates the emphasis of the present review on representation diversity, heterogeneous model fusion, cross-dataset and temporal robustness, leakage control, adversarial resilience, low false-positive operation, explainability, reproducibility, and computational efficiency.

REVIEW METHODOLOGY

Databases and Search Date

The review used targeted searches of IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect/Elsevier. The search and verification pass used for this revision was completed on 18 September 2026. Cross-publisher bibliographic pages and DOI metadata were used to verify titles, publication years, and article identifiers when necessary.

Search Strategy

Search strings combined the threat object (malicious URL, phishing URL, fraudulent URL, malicious website) with learning and representation terms. Representative Boolean forms were: ("malicious URL" OR "phishing URL") AND (machine learning OR deep learning); ("malicious URL" OR "phishing URL") AND

(ensemble OR stacking OR voting OR hybrid); ("phishing URL" OR "malicious URL") AND (BERT OR DistilBERT OR transformer); and ("malicious URL" OR "phishing URL") AND (graph neural network OR GNN). Additional targeted searches were performed for feature fusion, character n-grams, temporal convolution, attention, adversarial robustness, and dataset generalization.

Eligibility Criteria

Studies were included when they: (1) directly addressed malicious, phishing, or fraudulent URL/web-resource detection; (2) used ML, DL, transformer, graph, ensemble, or hybrid methods; (3) provided sufficient methodological or evaluation detail for synthesis; (4) appeared in peer-reviewed journals or conference proceedings; and (5) were relevant to the 2020–2026 technological progression, with earlier foundational work used only when needed to explain a major method family. Studies were excluded when URL detection was incidental, the work concerned an unrelated recognition task, methodological detail was insufficient, or the paper duplicated an already represented contribution without adding a distinct perspective.

Screening and Study Selection

A PRISMA-style screening procedure was followed to improve transparency in study selection. Records were identified through targeted searches of IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect/Elsevier. Duplicate records were removed before title and abstract screening. After de-duplication, 32 candidate records remained for assessment. Title and abstract screening excluded eight studies because they were off topic, duplicated an already represented contribution, or were not primarily focused on malicious or phishing URL detection. The remaining 24 studies were assessed for eligibility through full-text examination. Four studies were excluded because of insufficient methodological detail or limited direct relevance to the objectives of this review. Consequently, 20 studies were retained for the final synthesis. The screening process was designed to capture representative studies across machine learning, deep learning, transformer-based learning, graph-based learning, ensemble methods, and hybrid feature-fusion approaches rather than claim exhaustive coverage of every malicious-URL detection publication. The resulting literature set therefore represents a structured and reproducible synthesis of recent methodological developments, primarily between 2020 and 2026.

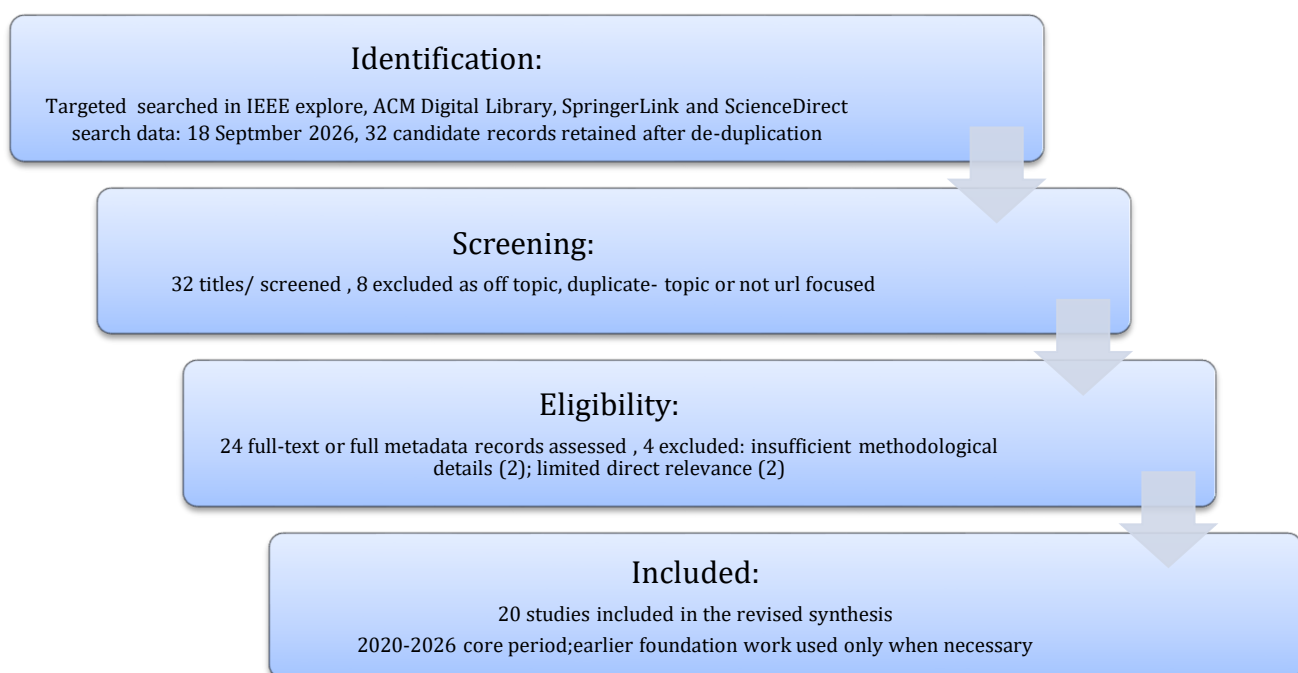


Fig. 1. PRISMA-style identification, screening, eligibility, and study-selection process used in the structured literature review.

Data Extraction and Synthesis

For each included study, the review recorded publication year, representation type, model family, fusion strategy where applicable, dataset or validation design, reported metrics, deployment considerations, and limitations. Reported performance values were checked against the original publication or publisher metadata when available. Because studies use different datasets, sampling policies, class balances, preprocessing procedures, and train/test partitions, performance values are treated as within-study results rather than a common leaderboard.

Quality Assessment

To improve the reliability of the synthesis, each primary experimental study was qualitatively assessed according to several methodological criteria. These included: clarity of dataset provenance, reporting of class distribution, description of preprocessing and feature extraction, handling of duplicate or highly related URLs, appropriateness of the train-test validation strategy, reporting of multiple evaluation metrics, comparison with relevant baseline models, use of external or temporal validation, computational or deployment-related evaluation, and availability of sufficient methodological detail for reproducibility.

The quality assessment was not used to generate a numerical ranking of the included studies because the papers differ substantially in objectives, datasets, representations, and evaluation designs. Instead, the assessment was used to interpret the strength of reported results and to distinguish benchmark performance from stronger evidence of generalization. Studies using independent-source, temporal, domain-separated, or multi-dataset validation were considered to provide stronger evidence of robustness than studies relying exclusively on random train-test partitions of a single dataset.

Machine-Learning and Engineered-Feature Approaches

Conventional ML remains highly relevant because URL strings expose inexpensive lexical and structural attributes: total length, number of subdomains, digit and special-character ratios, suspicious tokens, path depth, entropy, protocol indicators, and host-related characteristics. Random Forest, gradient boosting, SVM, logistic regression, and decision trees can process such features with low inference cost [3] [6] [18] [19].

Kumar [6] illustrates feature-level hybridization: NLP-related features, word vectors, PCA, and Random Forest are combined rather than treating each feature family independently. The reported 99.75% accuracy is notable but must be interpreted within the study's dataset and validation protocol. More recent comparative ML studies continue to find tree-based ensembles competitive when the feature space is structured [18] [19].

The principal advantage of engineered-feature models is operational efficiency and partial interpretability. Their weakness is that attackers can deliberately manipulate visible surface characteristics. Consequently, engineered features are most valuable as one branch of heterogeneous architecture rather than as the sole evidence source.

Deep-Learning and Sequence Representations

Deep Learning-Based Malicious URL Detection

Deep learning reduces dependence on manually designed variables by learning discriminative representations directly from raw or minimally processed URL sequences. Unlike conventional machine-learning approaches that often rely on manually extracted lexical, structural, host-based, or statistical features, neural architectures can automatically identify complex patterns within URL strings.

Convolutional Neural Networks (CNNs) are particularly effective at detecting local character- or token-level motifs, such as suspicious substrings, unusual symbol combinations, domain patterns, and path structures, while LSTM-like recurrent models capture ordered and sequential dependencies across URL components

[2], [7]. Yuan et al. [2] further demonstrated the benefit of processing complementary semantic and visual views in parallel, suggesting that multiple representations can capture information that may be missed by a single feature space.

Character-level modeling is particularly relevant to malicious URL detection because attackers frequently manipulate individual characters, subdomains, directory structures, query parameters, and delimiters to imitate legitimate websites. Neural models operating directly on character sequences can learn such patterns without requiring every possible manipulation to be explicitly represented through handcrafted rules. Word- or token-level representations, in contrast, can capture higher-level relationships among meaningful URL components. Combining character- and token-level representations therefore provides an opportunity to capture both fine-grained lexical irregularities and broader semantic patterns [7], [15].

Hybrid deep-learning architecture has consequently received increasing attention. CNN-LSTM and related architectures exploit the complementary strengths of convolutional and recurrent processing: convolutional layers extract informative local patterns, while recurrent layers model sequential dependencies among those patterns. Such architecture can provide richer representations than either component independently, although their additional complexity may increase computational requirements and complicate model interpretation.

The evolution beyond basic CNN/LSTM models is visible in Temporal Convolutional Network (TCN) and attention-based designs. Do et al. [15] combine character- and word-level embeddings with a Temporal Convolutional Network and Multi-Head Self-Attention, reporting 98.78% accuracy. TCN provides a mechanism for modeling long-range sequential relationships while supporting parallel computation, whereas the attention mechanism allows the model to assign different importance to different portions of the URL. This design addresses long-range dependency and feature-selection limitations that can affect conventional sequence models [15]. Deep models may also be vulnerable to adversarial URL manipulation. Attackers can alter characters, insert benign-looking tokens, modify subdomains or paths, change separators, use encoding techniques, or introduce visually similar characters while preserving the malicious destination or intent. Consequently, evaluation against intentionally obfuscated and adversarial modified URLs can provide a more realistic assessment of robustness than conventional benchmark testing alone.

Class imbalance represents another practical challenge. Because operational traffic can contain substantially more benign than malicious URLs, high overall accuracy may coexist with poor malicious-class recall or an excessive false-positive rate. Neural malicious-URL studies should therefore report precision, recall, F1-score, ROC-AUC and PR-AUC together with false-positive and true-positive rates rather than relying primarily on accuracy.

Finally, neural approaches generally increase training and inference cost and can be sensitive to the collection period and composition of their training data. A model achieving marginally higher benchmark accuracy may not necessarily be preferable for real-time deployment if it requires substantially greater computational resources. For deployment-oriented research, predictive performance should therefore be evaluated together with inference latency, model size, memory footprint, computational complexity, throughput, false-positive behavior, and robustness under distribution shift. These considerations are particularly important for browser extensions, network gateways, endpoint-security applications, and mobile or resource-constrained environments.

Overall, the progression from CNN and recurrent architectures toward TCNs, attention mechanisms, transformers, and multi-view neural systems demonstrates a broader transition from single-representation learning toward heterogeneous representation learning. This trend provides a strong foundation for hybrid ensemble malicious URL detection, where complementary models can be combined to exploit lexical, sequential, semantic, structural, contextual, and relational information while reducing dependence on any single representation.

Transformer-Based and Multi-Level Representations

Transformer-based URL models are increasingly important because they can learn contextual relations

among URL tokens while supporting transfer learning. Recent work also recognizes that generic language modeling is not sufficient by itself: URLs contain character-level irregularities, separators, subdomains, paths, and non-natural-language structures.

TransURL [12] combines a character-aware Transformer with multi-layer encoding, multi-scale feature learning, and spatial pyramid attention. PMANet [13] similarly fuses character- and subword-level signals and uses multi-level attention. These approaches treat URL representation as a multi-granularity problem rather than relying on a single pooled transformer embedding.

Do et al. [14] integrate character-, word-, and sentence-level processing using RasNet, a TCN-attention module, and MPNet, reporting the highest detection rate of 99.71% on the 1M-PD dataset and evaluation across four datasets. Lee and Kwon [9] and Özmen and Yildirim [10] take a different route: transformer outputs are combined with structured or boosting-based branches, directly motivating heterogeneous ensemble design.

Graph and Relational Representations

Graph learning introduces relational evidence that cannot be represented fully by an independent flat feature vector. In the hybrid framework of Alaladinni et al. [8], tokenized URL components form graph nodes and edges, while Random Forest processes conventional URL attributes. The design illustrates an important principle: hybridization is strongest when branches receive meaningfully different evidence. The trade-off is additional graph construction and message-passing cost, which must be justified for real-time use.

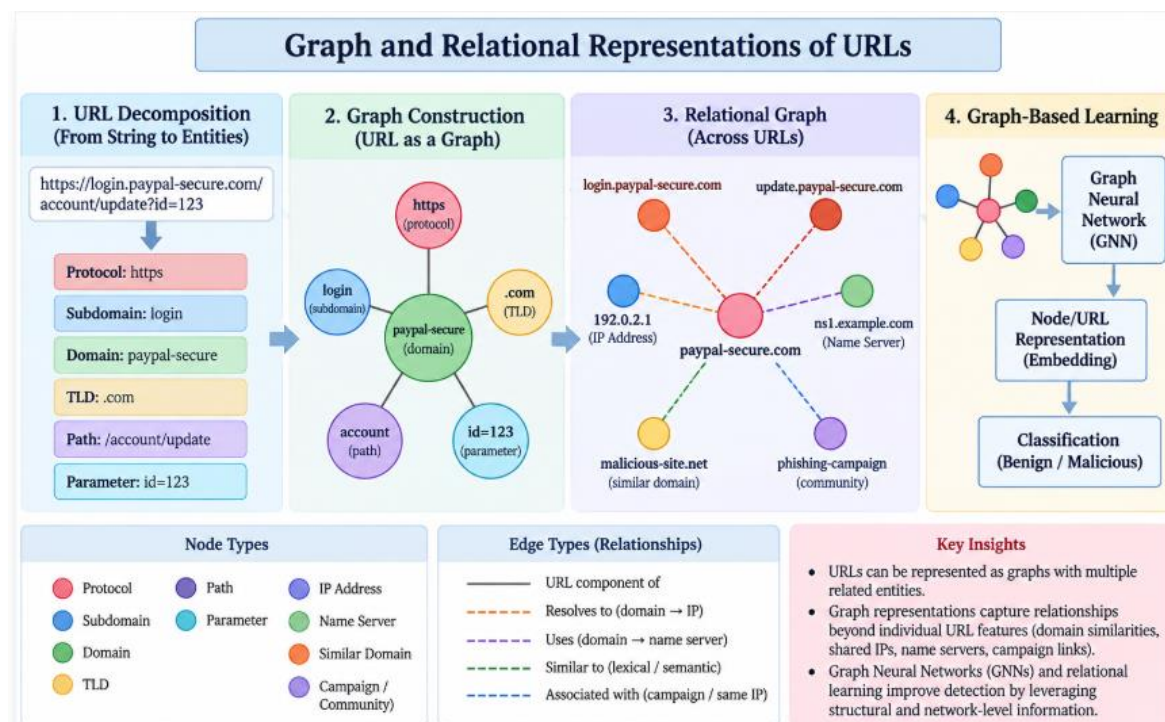


Fig 2. Graph and relation features of a URL for malicious URL detection

Ensemble Learning and Feature Fusion

Ensemble methods combine classifiers or representations through bagging, boosting, voting, stacking, or late fusion. Early systems such as PhishHaven [1] and the stacking experiments of Nadar et al. [3] demonstrate classifier diversity. More recent systems increasingly combine model families that process different input modalities. Lee and Kwon [9] integrate BERT with XGBoost and CatBoost. Özmen and Yildirim [10] combine DistilBERT-derived semantic information with handcrafted statistical features through late fusion. PhishHunter-XLD [17] extends this principle by combining XGBoost, LSTM, and DistilBERT using weighted voting, thereby assigning explicit roles to lexical, sequential, and contextual learners.

The reviewed evidence suggests that representation diversity may be more informative than simply increasing classifier count. Combining several classifiers that operate on highly similar feature spaces can produce correlated predictions and therefore provide limited additional benefits. In contrast, heterogeneous ensembles that integrate structurally different inputs—such as engineered statistical features, sparse lexical n-grams, sequential representations, contextual transformer embeddings, or relational graph features—have greater potential to capture complementary properties of malicious URLs. The value of a hybrid ensemble should therefore be assessed according to the distinct information contributed by each branch rather than by the total number of constituent models.

Malicious Url Representation and Feature Engineering

Lexical and Structural Features

Lexical and structural features are directly derived from the URL string and commonly include URL length, domain length, number of subdomains, path depth, digit ratio, special-character frequency, IP-address usage, suspicious tokens, protocol indicators, and entropy-related characteristics [1] [4] [6] [8] [20]. Their main advantage is efficiency because most of these features can be computed without retrieving webpage content. However, attackers can deliberately manipulate visible URL structure, making lexical features more suitable as one source of evidence within a broader hybrid architecture.

Character n-grams and TF-IDF

Character-level n-grams preserve fine-grained lexical patterns across domains, paths, parameters, and other portions of a URL. When transformed using TF-IDF, they produce high-dimensional sparse vectors that can be classified efficiently by linear models such as SVM. Such representations reduce dependence on manually defined suspicious keywords but provide only limited contextual understanding compared with deep neural or transformer-based representations [4] [12] [20].

Learned Sequence Features

CNN, LSTM, and TCN architectures learn local or sequential representations directly from encoded URLs and therefore reduce reliance on manually selected characteristics [2] [7] [15]. CNN-based methods can identify local patterns, whereas recurrent and temporal convolutional architectures can capture ordered or longer-range dependencies. These benefits generally come at the cost of increased data and computational requirements.

Transformer Contextual Features

Transformer-based models such as BERT-family architectures can learn contextual relationships among tokenized URL components. Recent URL-specific approaches demonstrate that character-aware, subword-aware, and multi-level representations can be particularly important because URLs differ structurally from ordinary natural-language text [9], [10], [12], [13]. Such approaches provide expressive contextual representations but may introduce greater inference cost and calibration requirements.

Graph and Infrastructure Features

Graph representations can model relationships among URL components and, where available, external entities such as domains, IP addresses, name servers, certificates, and hosting infrastructure [8] [20]. Such relational information may reveal structural dependencies that are difficult to encode using flat feature vectors. However, graph construction and external data acquisition introduce additional processing and deployment overhead.

Fusion Implication

A future hybrid system should avoid redundant branches. Each branch should be justified by the information

it contributes and evaluated through ablation to establish whether the combined system adds measurable value.

Comparative Analysis of the Reviewed Studies

Because direct comparison of reported accuracy values is methodologically unreliable across heterogeneous datasets and experimental designs, Table II compares the principal roles, strengths, and limitations of major representation and model families. The objective is not to identify a universally superior algorithm, but to clarify how different approaches can contribute complementary evidence within a heterogeneous ensemble.

Approach	Principal Strength	Principal Limitation	Role in Hybrid Architecture
Handcrafted lexical/statistical [1], [6]	Fast, computationally efficient, and relatively interpretable	Feature-dependent and potentially vulnerable to URL manipulation/obfuscation	Structured evidence branch
TF-IDF + Linear SVM [5], [6]	Efficient representation of sparse lexical patterns and URL tokens	Limited ability to capture contextual and semantic relationships	Sparse lexical branch
CNN / LSTM / TCN [7], [14], [15]	Automatically learns local and sequential patterns from character/word-level URL representations	Higher computational/data requirements; individual architectures may have difficulty capturing all long-range dependencies	Sequence-learning branch
BERT / DistilBERT / URL Transformers [10], [12], [13]	Learns contextual, character/subword, and multi-level representations with reduced dependence on manual feature engineering	Greater inference/computational cost; confidence calibration and efficient deployment remain important	Contextual representation branch
Boosting — XGBoost / CatBoost / LightGBM [9], [17], [19]	Strong nonlinear modeling of structured and engineered features; effective component of heterogeneous ensembles	Performance depends strongly on the quality and relevance of engineered input features	Structured boosting branch
Graph Neural Networks (GNN) [8], [20]	Captures relational and structural information that independent URL representations may miss	Requires graph construction and relational information, increasing preprocessing and computational overhead	Relational evidence branch
Stacking / Voting / Late Fusion [3], [9], [10], [17]	Combines complementary representations and model errors; allows heterogeneous ML, DL, and Transformer models to contribute to the final prediction	Requires leakage-safe training, careful weight/meta-classifier selection, probability calibration, and additional computational resources	Final decision/fusion layer

Table II. Representation-oriented comparison of model families.

The comparison indicates that the principal value of hybrid ensemble learning lies in complementary

representation spaces. Engineered features provide efficient and relatively interpretable statistical evidence; TF-IDF and linear SVM provide sparse lexical modeling; CNN, LSTM, and TCN architectures capture local and sequential patterns; transformers provide contextual and multi-level representations; boosting methods model nonlinear interactions among structured variables; and GNNs introduce relational evidence. A well-designed hybrid architecture should therefore combine only those branches whose information content and prediction errors are sufficiently different to justify the added computational complexity.

Proposed Future Research Direction

Based on the reviewed evidence, future architecture should combine four complementary branches rather than duplicating similar classifiers. Branch 1 processes handcrafted lexical and structural features using Random Forest. Branch 2 converts character n-grams into TF-IDF vectors and classifies them using a Linear SVM. Branch 3 uses a lightweight transformer such as DistilBERT to generate contextual URL representations. Branch 4 applies XGBoost or CatBoost to structured and, where available, host-related features. Calibrated branch probabilities are then supplied to a leakage-safe stacking meta-classifier, such as logistic regression.

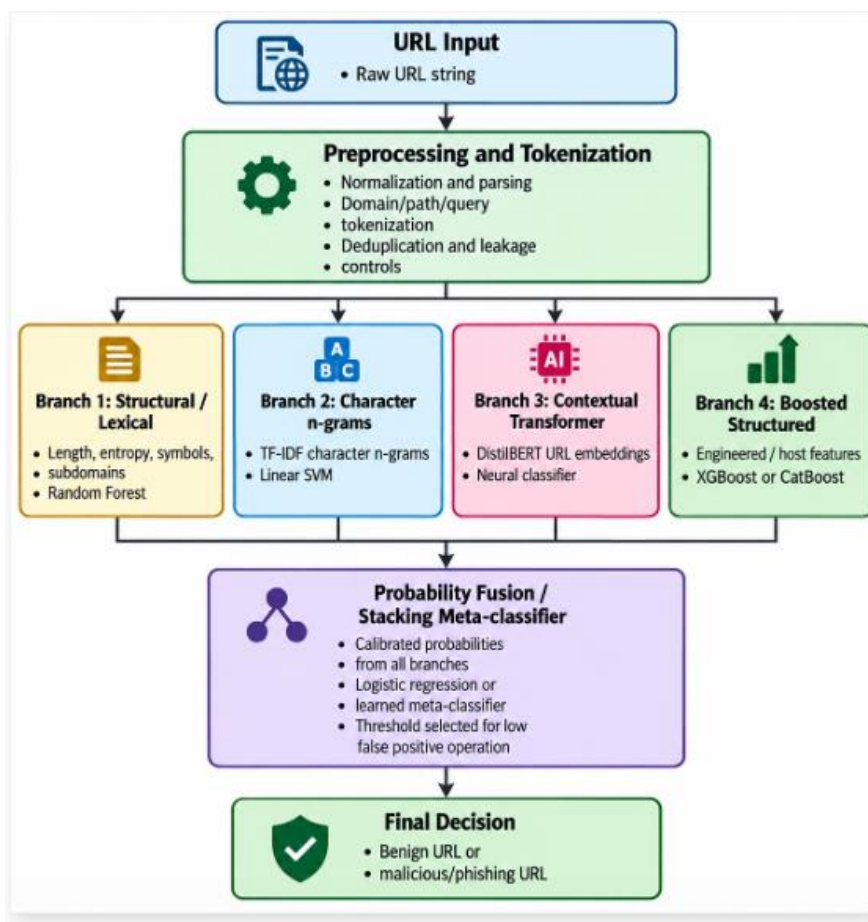


Fig 3. Proposed heterogeneous hybrid ensemble architecture for malicious URL detection

The proposed architecture is deliberately heterogeneous because each branch is designed to process a different representation of the same URL. The Random Forest branch models interpretable lexical and structural characteristics, while the TF-IDF–Linear SVM branch captures high-dimensional character-level lexical patterns. The transformer branch provides contextual representations that are difficult to express through manually engineered variables, and the boosting branch models nonlinear interactions among structured and host-related features. Rather than applying fixed majority voting, calibrated prediction probabilities from the four branches are supplied to a stacking meta-classifier. To prevent information leakage, the meta-classifier should be trained only on out-of-fold predictions produced by the base learners

during training. The final decision threshold can subsequently be selected on validation data according to the operational trade-off between phishing detection rate and false-positive rate.

Formal Experimental Evaluation Plan

Baseline and Ablation Experiments

Each branch should first be evaluated independently to establish its standalone contribution. At minimum, the evaluation should include Random Forest, Linear SVM with character-level TF-IDF, DistilBERT, and XGBoost or CatBoost. Progressive ablation experiments should then compare combinations such as RF+SVM, RF+Boosting, SVM+Transformer, RF+SVM+Transformer, and the complete four-branch ensemble. Removing one branch at a time from the complete system should additionally reveal whether each component contributes unique predictive information. These experiments should determine whether performance improvements result from genuine complementarity rather than simply increasing model complexity.

Random, Domain-Separated, Temporal, and Cross-Dataset Evaluation

Four evaluation settings should be considered. First, a stratified random split may be used as a conventional benchmark reference. Second, domain-grouped splitting should ensure that URLs belonging to the same registered domain do not occur across training and test partitions. Third, temporal evaluation should use older observations for training and newer URLs for testing to examine concept drift. Fourth, cross-dataset evaluation should train the model using one data source and test it using an independently collected source. The latter three settings provide substantially stronger evidence of generalization than random splitting alone.

Adversarial and Obfuscation Evaluation

Controlled adversarial transformations should be applied to determine whether the proposed system remains stable under superficial URL manipulation. Representative transformations may include subdomain insertion, benign-token padding, path or query modification, percent encoding, separator variation, character substitution, homoglyph-like substitutions where technically valid, and redirection through URL-shortening services. The resulting degradation in precision, recall, F1-score, ROC-AUC, PR-AUC, and malicious-class detection rate should be measured relative to the clean test set.

Metrics and Operating Points

Evaluation should report accuracy, precision, recall or true-positive rate, specificity, F1-score, ROC-AUC, PR-AUC, false-positive rate, and confusion-matrix statistics. Because operational cybersecurity systems may process very large volumes of legitimate URLs, low-FPR performance is particularly important. Where the number of test samples permits reliable estimation, the true-positive rate should therefore be reported at fixed false-positive rates such as 1%, 0.5%, and 0.1%. Confidence intervals should accompany the principal evaluation metrics.

Efficiency and Resource Evaluation

Deployment-oriented evaluation should report average and percentile-based per-URL inference latency, throughput, peak memory usage, model size, and training time on explicitly specified hardware. The complete ensemble should be compared with reduced variants in order to quantify the marginal computational cost associated with the transformer branch and any external host-related feature acquisition.

Statistical and Reproducibility Controls

Random seeds, preprocessing rules, deduplication procedures, feature definitions, hyperparameters, model configurations, and dataset split identifiers should be documented. Duplicate and domain-related samples

should be removed or grouped before splitting. Hyperparameter optimization and threshold calibration should rely exclusively on training and validation data, leaving the final test set untouched until model selection is complete. When multiple training runs are computationally feasible, results should be reported using mean and standard deviation together with bootstrap confidence intervals or appropriate paired statistical comparisons.

CONCLUSION

Malicious URL detection has evolved from blacklist-based mechanisms and manually engineered lexical features toward representation-learning systems that integrate statistical, lexical, sequential, contextual, and relational evidence. This structured review of 20 studies demonstrates that conventional machine-learning methods remain valuable because of their efficiency and effectiveness on engineered URL characteristics, while deep neural networks, transformers, and graph-based approaches extend the ability of detection systems to learn complex representations that cannot easily be specified manually.

The reviewed literature also indicates that high benchmark accuracy alone does not provide sufficient evidence of practical effectiveness. Differences in dataset composition, collection period, class balance, preprocessing, duplicate handling, train-test partitioning, and operational thresholds make direct cross-study ranking unreliable. Moreover, temporal drift, cross-dataset generalization, domain-level leakage, adversarial URL manipulation, false-positive control, computational cost, explainability, and reproducibility remain significant challenges.

A central conclusion of this review is that the value of hybrid ensemble learning lies in both model diversity and representation diversity. Combining several classifiers is unlikely to provide substantial benefit when all models operate on highly correlated information. In contrast, architectures that integrate engineered structural features, sparse character-level representations, contextual transformer embeddings, and nonlinear ensemble models can exploit complementary evidence.

Based on these observations, the review proposes a leakage-safe heterogeneous stacking architecture combining Random Forest, TF-IDF with Linear SVM, a lightweight transformer such as DistilBERT, and gradient-boosting models. Future evaluation of such architectures should include ablation analysis, domain-separated testing, temporal validation, independent cross-dataset testing, adversarial and obfuscation stress tests, low false-positive operating points, computational measurements, and reproducibility controls. Such evaluation would provide stronger evidence of real-world deployment readiness than further optimization of classification accuracy on a single static benchmark.

REFERENCES

1. M. Sameen, K. Han, and S. O. Hwang, "PhishHaven—An efficient real-time AI phishing URLs detection system," *IEEE Access*, vol. 8, pp. 83425–83443, 2020, doi: 10.1109/ACCESS.2020.2991403.
2. J. Yuan, G. Chen, S. Tian, and X. Pei, "Malicious URL detection based on a parallel neural joint model," *IEEE Access*, vol. 9, pp. 9464–9472, 2021, doi: 10.1109/ACCESS.2021.3049625.
3. V. K. Nadar, B. Patel, V. Devmane, and U. Bhawe, "Detection of phishing websites using machine learning approach," *Proc. 2nd Global Conf. Advancement in Technology (GCAT)*, 2021, pp. 1–8, doi: 10.1109/GCAT52182.2021.9587682.
4. M. Aljabri et al., "Detecting malicious URLs using machine learning techniques: Review and research directions," *IEEE Access*, vol. 10, pp. 121395–121417, 2022, doi: 10.1109/ACCESS.2022.3222307.
5. A. Saleem Raja, R. Madhubala, N. Rajesh, L. Shaheetha, and N. Arulkumar, "Survey on malicious URL detection techniques," in *Proc. 6th Int. Conf. Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, Apr. 28–30, 2022, pp. 778–781, doi: 10.1109/ICOEI53556.2022.9777221.
6. J. A. Kumar, "Hybrid feature-based machine learning method for phishing URL detection," *Proc. 3rd Int. Conf. Secure Cyber Computing and Communication (ICSCCC)*, 2023, pp. 222–227, doi: 10.1109/ICSCCC58608.2023.10176901.

7. R. Ferdaws and N. E. Majd, "Phishing URL detection using machine learning and deep learning," *Proc. IEEE World AI IoT Congress (AIIoT)*, 2024, pp. 485–490, doi: 10.1109/AIIoT61789.2024.10579005.
8. B. Alaladinni et al., "A hybrid approach for malicious URL detection using ML classifiers and graph neural networks," *Proc. 6th Int. Conf. Data Intelligence and Cognitive Informatics (ICDICI)*, 2025, doi: 10.1109/ICDICI66477.2025.11135390.
9. J. Lee and H. Kwon, "Hybrid ensemble learning for malicious URL detection with BERT and boosting models," *IEEE Access*, vol. 14, pp. 62045–62058, 2026, doi: 10.1109/ACCESS.2025.3605302.
10. Ü. Özmen and E. O. Yildirim, "DistilBERT-based hybrid architecture for phishing URL detection," *IEEE Access*, vol. 14, pp. 71720–71737, 2026, doi: 10.1109/ACCESS.2026.3684855.
11. N. Reyes-Dorta, P. Caballero-Gil, and C. Rosa-Remedios, "Detection of malicious URLs using machine learning," *Wireless Networks*, vol. 30, pp. 7543–7560, 2024, doi: 10.1007/s11276-024-03700-w.
12. R. Liu, Y. Wang, Z. Guo, H. Xu, Z. Qin, W. Ma, and F. Zhang, "TransURL: Improving malicious URL detection with multi-layer Transformer encoding and multi-scale pyramid features," *Computer Networks*, 2024, doi: 10.1016/j.comnet.2024.110707.
13. R. Liu, Y. Wang, H. Xu, Z. Qin, F. Zhang, Y. Liu, and Z. Cao, "PMANet: Malicious URL detection via post-trained language model guided multi-level feature attention network," *Information Fusion*, vol. 113, 102638, 2025, doi: 10.1016/j.inffus.2024.102638.
14. N. Q. Do, A. Selamat, H. Fujita, and O. Krejcar, "An integrated model based on deep learning classifiers and pre-trained transformer for phishing URL detection," *Future Generation Computer Systems*, vol. 161, pp. 269–285, 2024, doi: 10.1016/j.future.2024.06.031.
15. N. Q. Do, A. Selamat, O. Krejcar, and H. Fujita, "Detection of malicious URLs using Temporal Convolutional Network and Multi-Head Self-Attention mechanism," *Applied Soft Computing*, vol. 169, 112540, 2025, doi: 10.1016/j.asoc.2024.112540.
16. K. Barik, S. Misra, and R. Mohan, "Web-based phishing URL detection model using deep learning optimization techniques," *International Journal of Data Science and Analytics*, vol. 20, pp. 4449–4471, 2025, doi: 10.1007/s41060-025-00728-9.
17. T. Doshi et al., "PhishHunter-XLD: An ensemble approach integrating machine learning and deep learning for phishing URL classification," *Franklin Open*, vol. 12, 100349, 2025, doi: 10.1016/j.fraope.2025.100349.
18. S. Lokesh R, "Phishing URL detection using machine learning: A comparative study," in *Proc. 6th Int. Conf. Mobile Computing and Sustainable Informatics (ICMCSI)*, Goathgaun, Nepal, Jan. 7–8, 2025, pp. 1524–1532, doi: 10.1109/ICMCSI64620.2025.10883082.
19. J. D. Duarte, P. Chagas Junior, J. P. J. da Costa, E. J. da Costa, L. P. de Melo, R. R. Nunes, C. V. N. G. Soares, and T. E. da Cunha Silva, "Machine learning for early detection of phishing URLs in parked domains: An approach applied to a financial institution," *IEEE Access*, vol. 13, pp. 145736–145753, 2025, doi: 10.1109/ACCESS.2025.3599454.
20. Y. Tian, Y. Yu, J. Sun, and Y. Wang, "From past to present: A survey of malicious URL detection techniques, datasets and code repositories," *Computer Science Review*, vol. 58, Art. no. 100810, 2025, doi: 10.1016/j.cosrev.2025.100810.