Research Article

Integrating Artificial Intelligence and Cyber Law For Phishing and Fake Content Detection: A Legal Intelligence Framework

Nisarg Mehta[1]*, Dinesh Swami[2], and Pratik Patel[3]

[1*]Department of Computer Science and Engineering, Parul University, India Email: 2503032010009@paruluniversity.ac.in , nisargmehta3010@gmail.com
[2]Department of Computer Science and Engineering, Parul University, India. Email: dinesh.swami38762@paruluniversity.ac.in
[3]Department of Computer Science and Engineering, Parul University, India. Email: pratik.patel2988@paruluniversity.ac.in

Abstract

Phishing and deceptive digital content combine URL obfuscation, impersonation, social engineering, and generated content. This paper reviews AI-based detection and presents a modular framework integrating URL, content, behavioral, explainability, and cyber-law intelligence. The framework connects technical threat evidence to candidate legal provisions, jurisdictional information, evidence requirements, and reporting guidance for human review. A reproducible validation protocol is specified for temporal drift, adversarial robustness, multilingual detection, computational efficiency, legal mapping, and human review. No experimental performance is claimed for the proposed integrated framework; the contribution is a structured architecture and an evaluation pathway that can be implemented and tested.

Index Terms—Phishing detection, fake content detection, machine learning, deep learning, natural language processing, cyber law, legal intelligence, explainable AI, adversarial robustness, cybersecurity.

1. INTRODUCTION

The expansion of Internet services, online banking, social networking, and cloud communication has increased opportunities for phishing, impersonation, malicious advertisements, and deceptive digital content. Contemporary campaigns may combine URL manipulation, domain spoofing, visual impersonation, social engineering, and dynamically generated text. Previously observed indicators may therefore be insufficient for some unseen attacks. Research has progressed from blacklists and heuristics toward machine learning (ML), deep learning (DL), natural language processing (NLP), and hybrid detection. Prior studies consider URL, webpage, host, visual, behavioral, and content features, but random train–test splits can overestimate practical performance when attacks evolve over time.

Evaluation in realistic settings requires temporal separation, adversarial testing, multiple performance measures, and computational measurements. A second limitation is the gap between technical detection and cyber law. A classifier may label content as benign, suspicious, or phishing, but that label does not establish legal liability. Investigators may also need supporting evidence, applicable provisions, jurisdiction, and reporting pathways. This paper links technical detection to a legal-intelligence layer without treating an AI prediction as an autonomous legal conclusion. The legal module returns candidate provisions and reporting guidance for review by a human investigator or qualified legal professional.

The paper makes five contributions: (1) it synthesizes URL-, host-, content-, behavioral-, ML-, DL-, NLP-, and hybrid detection approaches; (2) it identifies gaps in temporal generalization, real-time deployment, multilingual and multimodal detection, adversarial robustness, efficiency, and explainability; (3) it proposes a modular architecture combining URL, content, and behavioral evidence; (4) it adds candidate legal provisions, jurisdiction, evidence requirements, explanations, and reporting support; and (5) it specifies comparative legal analysis and a reproducible validation protocol that separates literature-based findings from future experimental results.

The review is guided by three research questions: which features and AI models are used most often for phishing and fake-content detection; what limits robust real-time, multilingual, temporally stable, and adversarially resilient protection; and how can cyber-law information be connected to technical detection without treating AI output as a legal determination?

2. LITERATURE REVIEW AND RELATED WORK

Phishing detection has evolved from manually curated indicators toward data-driven and representation-learning methods. Reviews identify URL, webpage, host, visual, behavioral, and hybrid detection as major directions [1], [4], [5], [8]. The techniques are complementary: reputation-based methods screen known threats, structured ML learns patterns from engineered features, DL learns richer representations, NLP models analyze deceptive language, and hybrid systems combine different evidence sources.

A. Blacklist and Heuristic Detection

Early defenses relied on blacklists, reputation services, and manually designed rules. They are computationally simple but typically require a malicious URL to be known before it can be blocked. This makes them less effective against newly registered domains and short-lived campaigns. Heuristics examine URL length, unusual characters, subdomain depth, IP-address use, suspicious tokens, encoding, entropy, and redirects. Such indicators remain useful for first-stage screening, but syntax can be altered while malicious intent remains unchanged.

B. Machine-Learning Detection

ML methods learn relationships among lexical, host, and webpage features. Random Forest, Support Vector Machine (SVM), gradient boosting, and related ensembles have been studied on structured feature sets [3], [8]. Random Forest is attractive as a first-stage classifier because it can process heterogeneous inputs and provide feature-importance estimates. However, performance may decline when the distribution of attacks changes or when evaluation contains leakage between training and test samples.

C. Deep-Learning Detection

DL reduces dependence on manually designed features. Character-level convolutional neural networks can learn local URL patterns, whereas recurrent models capture sequential dependencies. Hybrid architectures combine representations to improve coverage [6], [11]. Feature-free approaches and large-language-model-based detectors indicate a shift toward richer representations rather than reliance on handcrafted URL features alone [9], [10].

D. NLP and Fake-Content Detection

NLP is useful when phishing campaigns contain deceptive emails, social-media messages, advertisements, or webpage text. Bag-of-Words and TF–IDF provide efficient baselines, while contextual language models offer richer semantic representations. Fake-content detection is more difficult than URL classification because factual correctness, intent, context, satire, quotation, and source credibility may need to be considered. A classifier should not automatically equate “false” content with “illegal” content.

E. Behavioral Analysis

Behavioral analysis considers events after a user opens a webpage or receives a message. Indicators may include repeated authentication failures, abnormal navigation, unexpected redirection, unusual session duration, geographic anomalies, and deviations from expected behavior. Behavioral evidence complements URL and content analysis, but monitoring also creates privacy concerns. Deployment should apply data minimization, purpose limitation, access control, retention limits, and privacy-preserving processing.

F. Adversarial and Temporal Robustness

A conventional evaluation often assumes that future attacks resemble training data. In practice, attackers change domains, URL tokens, webpage templates, and text to evade detection. Adversarial research therefore motivates controlled feature perturbation and temporally separated test data [3], [7]. Static random splits alone are not sufficient to establish generalization to future campaigns.

3. ESTABLISHED DETECTION METHODOLOGIES

Detection approaches can be organized into complementary layers. URL analysis provides rapid screening; content analysis provides semantic evidence; behavioral analysis observes post-interaction events; and a hybrid detector combines these signals. Fig. 1 summarizes the main feature groups. Fig. 2 illustrates the evolution from rule-based screening to AI, NLP, robustness analysis, and legal intelligence.

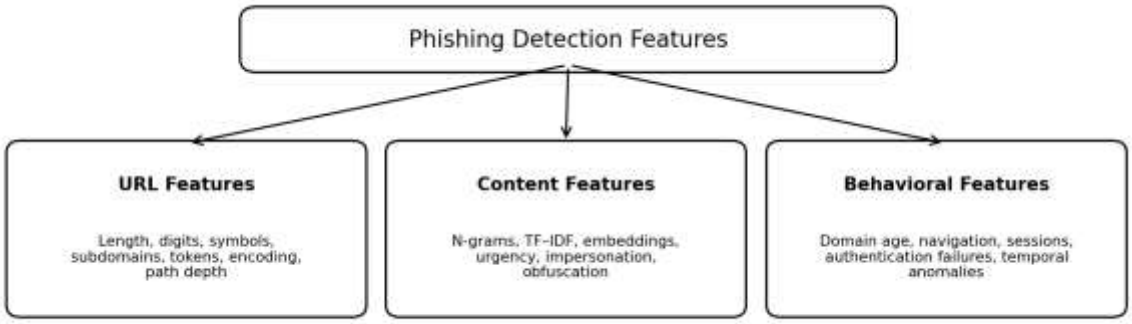


Fig. 1. Taxonomy of phishing-detection features based on URL, content, and behavioral characteristics.

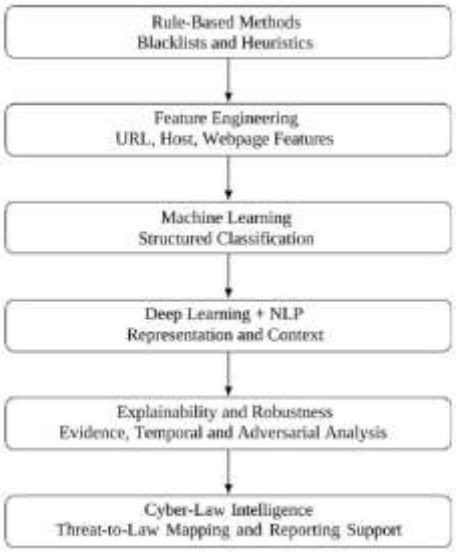


Fig. 2. Evolution from rule-based methods toward ML, DL, NLP, explainability, robustness, and cyber-law intelligence.

A. URL-Based Feature Analysis

URL analysis is suitable for first-stage screening because it can often be performed without loading a complete webpage. The URL length is

$$L_URL = |u| \quad (1)$$

where u is the URL string. The special-character ratio can be expressed as

$$R_sp = N_sp / L_URL \quad (2)$$

where Nsp is the number of selected special characters. Other lexical indicators include digit count, subdomain count, suspicious tokens, encoded characters, path depth, entropy, and use of an IP address in place of a domain. Host-based indicators include domain age, registration information, DNS consistency, nameserver and certificate characteristics, redirects, URL-shortening

services, and hosting reputation. These are indicators rather than deterministic proof: legitimate organizations may also use new domains, content-delivery networks, shorteners, or frequently changing infrastructure.

B. Content-Based Detection Using NLP

Content detection examines text and structure from webpages, emails, messages, and social media. Potential linguistic indicators include urgency or pressure, requests for credentials or financial information, emotional manipulation, suspicious grammar, impersonation language, inconsistent claims, and abnormal calls to action. TF–IDF is a common baseline:

$$TF\text{–}IDF(t,d) = TF(t,d)\cdot\log[(1+N)/(1+DF(t))] \quad (3)$$

where N is the number of documents and DF(t) is the number containing term t. The Shannon entropy of a URL character distribution is

$$H(U) = -\Sigma_{i=1}^{N} p_i \log_2(p_i) \quad (4)$$

Transformer models represent contextual relationships using self-attention:

$$Attention(Q,K,V) = softmax(QK^T/\sqrt{d_k})V \quad (5)$$

where Q, K, and V are query, key, and value matrices. Contextual models help interpret text but should not be treated as factual-verification or legal-decision systems by themselves.

C. Behavioral and Hybrid Analysis

Behavioral signals may include abnormal navigation, repeated login failures, unusual session duration, unexpected credential submission, abnormal redirection, and geographic or temporal anomalies. A simple event-rate measure is

$$r = N_events / \Delta t \quad (6)$$

where N_events is the number of events during interval Δt. A practical layered architecture can apply URL screening first, then content analysis, behavioral analysis, and legal intelligence. The layers should expose evidence and uncertainty so that downstream decisions do not depend only on a single probability score.

4. MACHINE-LEARNING MODELS FOR DETECTION

A. Support Vector Machines and Recurrent Models

SVMs can perform effectively when the feature space is well defined and the dataset is comparatively structured. A linear separating hyperplane is

$$w^Tx + b = 0 \quad (7)$$

with decision function

$$f(x) = sign(w^Tx + b) \quad (8)$$

SVMs provide a useful baseline, though performance may decline under substantial feature-distribution shifts. Long Short-Term Memory (LSTM) models capture sequential relationships. Their gate equations are

$$i_t = \sigma(W_i\, x_t + U_i\, h_{(t-1)} + b_i) \quad (9)$$

$$f_t = \sigma(W_f\, x_t + U_f\, h_{(t-1)} + b_f) \quad (10)$$

$$o_t = \sigma(W_o\, x_t + U_o\, h_{(t-1)} + b_o) \quad (11)$$

$$c_t = f_t\cdot c_{(t-1)} + i_t\cdot\tanh(W_c\, x_t + U_c\, h_{(t-1)} + b_c) \quad (12)$$

$$h_t = o_t \odot \tanh(c_t) \quad (13)$$

Transformer-based models can provide contextual representations for deceptive text and email content. A principal limitation for browser-level deployment is computational cost and inference latency.

B. Convolutional Models and Performance Evaluation

A one-dimensional convolution can be written as

$$y_i = \sigma(\Sigma_{k=o}^{K-1} w_k\, x_{(i+k)} + b) \quad (14)$$

For binary classification, accuracy, precision, recall, and F1-score are defined as follows:

$$Accuracy=(TP+TN)/(TP+TN+FP+FN) \quad (15)$$

$$Precision = TP / (TP + FP) \quad (16)$$

$$Recall = TP / (TP + FN) \quad (17)$$

$$F1=2(Precision \cdot Recall)/(Precision+Recall) \quad (18)$$

Accuracy should not be used alone where classes are imbalanced. Recall is especially relevant in security because false negatives can allow malicious content to reach users. Random Forest provides a useful structured-feature baseline. For a forest of B trees, the estimated probability for class c is

$$P(y=c|x) = (1/B)\Sigma_{\beta=1}^{B} I[h_b(x)=c] \quad (19)$$

where hb(x) is the prediction of tree b. Its advantages include relatively low inference complexity and feature-importance information; it can still generalize poorly to unseen attacks or temporal drift.

5. CYBER-LAW INTEGRATION IN DETECTION FRAMEWORKS

Technical threat detection and legal interpretation are distinct functions. An ML model estimates a class from observed evidence, whereas legal analysis depends on legislation, jurisdiction, facts, intent, procedural context, and the applicable version of the law. The proposed legal-intelligence module does not declare that a person or organization committed a crime. It produces candidate provisions, factual prerequisites, jurisdictional assumptions, evidence requirements, known uncertainty, and reporting guidance for human review.

TABLE I ILLUSTRATIVE INDIAN LEGAL MAPPING

Threat category	Candidate provision	Interpretive limitation
Identity misuse	IT Act, Sec. 66C	Requires factual analysis of identity credentials or identifying features and unauthorized use.
Online impersonation	IT Act, Sec. 66D	Depends on the elements of cheating by personation using a computer resource.
Privacy-related conduct	IT Act, Sec. 66E	Applies only where the specific conduct and statutory requirements are met.
Computer-related offence	IT Act, Sec. 66	Requires analysis of the underlying computer-related conduct and statutory elements.

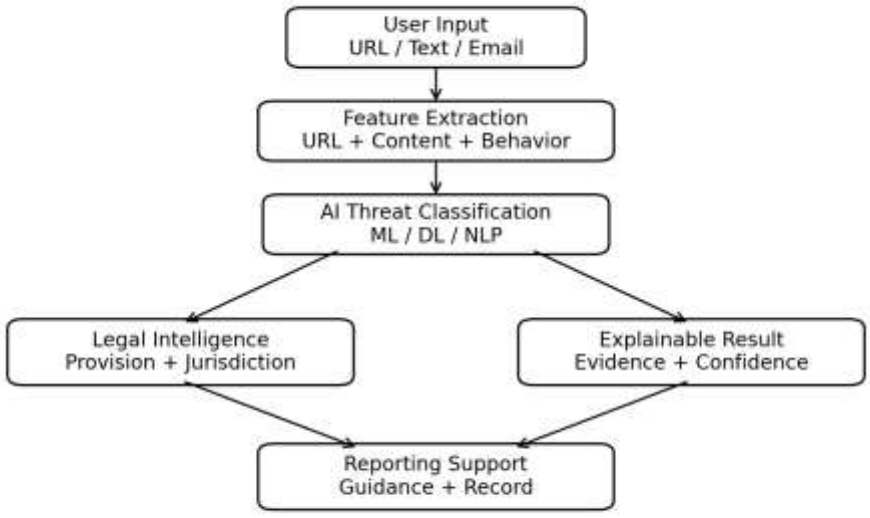


Fig. 3. Legal-intelligence flow connecting threat detection to candidate provisions, jurisdiction, explanation, and reporting support.

A. Illustrative Indian Legal Mapping

The Information Technology Act, 2000 contains provisions potentially relevant to identity theft, cheating by personation, privacy-related conduct, and computer-related offences. The following mapping is illustrative and is not a legal determination. Applicability depends on the statutory elements and facts of a particular incident.

B. Comparative Legal Analysis

The European Union has separate instruments for privacy, cybersecurity, and platform governance. GDPR may be relevant where personal-data processing is involved, while the Digital Services Act addresses aspects of digital-services governance. The United States has a fragmented framework spanning federal, state, sector-specific, and regulatory requirements; 18 U.S.C. §1030 addresses specified unauthorized access, damage, fraud, password trafficking, and related computer activity. The system should retrieve candidate rules based on facts and jurisdiction rather than mapping every phishing event to one statute.

A legal-intelligence component should consider substantive applicability, jurisdiction, evidence, and uncertainty. The same technical event can have different legal implications depending on location, victim, actor, data type, intent, authorization, and applicable law. The legal module may assign a decision-support confidence score

$$C_{legal} = \alpha C_{threat} + (1-\alpha)C_{rule}, 0 \le \alpha \le 1 \quad (20)$$

Crule represents confidence in a retrieved legal rule. This score is not the probability that an offence occurred; it represents support for a candidate mapping.

6. PROPOSED METHODOLOGY

The proposed methodology combines technical detection with explainable legal intelligence and reporting support. The normalized technical threat score is

$$S_{threat} = w_u S_{URL} + w_c S_{content} + w_b S_{behavior}; w_u + w_c + w_b = 1 \quad (21)$$

The weights are design parameters to be selected or learned using validation data; they are not experimentally established in this manuscript. The framework includes input acquisition, normalization and sanitization, URL/content/behavioral feature extraction, AI classification, score fusion, explanation, legal retrieval, jurisdiction filtering, reporting support, and analyst feedback. Fig. 4 summarizes the architecture.

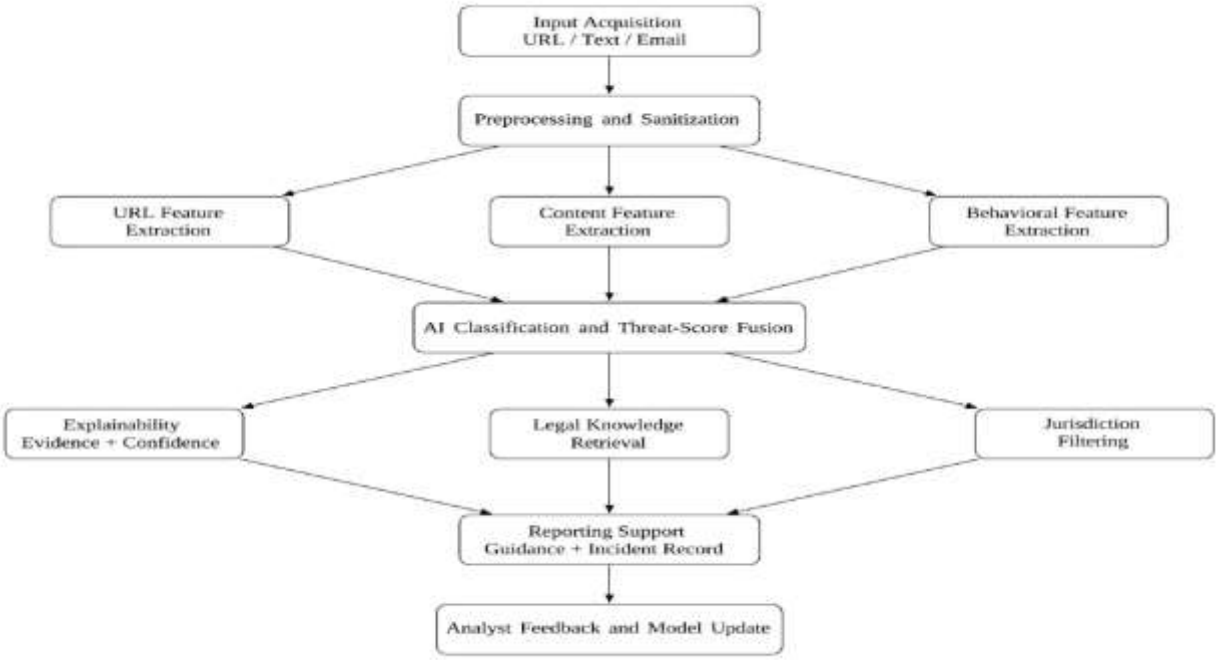


Fig. 4. Detailed architecture of the proposed AI and cyber-law integrated phishing and fake-content detection framework.

A. Detection and Legal-Mapping Algorithm

The workflow returns a technical threat label and evidence, then retrieves candidate rules only when the classification is suspicious or malicious. Missing legal facts and jurisdictional assumptions must remain visible to the reviewer.

Input: URL/text/email X, jurisdiction J
1. Xn ← NormalizeAndSanitize(X)
2. U ← ExtractURLFeatures(Xn)
3. C ← ExtractContentFeatures(Xn)
4. B ← ExtractBehaviorFeatures(Xn)
5. sURL ← URLModel(U); sCONT ← ContentModel(C); sBEH ← BehaviorModel(B)
6. s ← w_u·sURL + w_c·sCONT + w_b·sBEH
7. label ← Classify(s)
8. evidence ← GenerateExplanation(U, C, B, label)
9. if label is MALICIOUS or SUSPICIOUS:
10. rules ← RetrieveLegalRules(label, J)
11. mapping ← RankLegalRules(rules, evidence)
12. missing ← IdentifyMissingLegalFacts(label, J)
13. guidance ← GenerateReportingGuidance(mapping, missing)
14. else:
15. mapping ← NONE; guidance ← NONE
16. return label, s, mapping, evidence, guidance

B. Explainability and Legal-Use Limitation

The explanation should separate model evidence from legal evidence. Model evidence may include important URL features, suspicious lexical tokens, host indicators, influential content terms, behavioral anomalies, and model confidence. Legal evidence should state why a provision was retrieved as a candidate. Missing facts should be explicit. The system is not intended to replace lawyers, investigators, courts, regulators, or law-enforcement authorities, and it should not automatically submit complaints or make a legal finding.

7. RESEARCH GAP

Six practical gaps remain. First, high-capacity models may add latency and computational cost. Second, concept drift reduces the validity of static benchmarks. Third, datasets often remain concentrated on English or a single modality, limiting multilingual and multimodal generalization. Fourth, attackers can manipulate visible features. Fifth, technical classifiers rarely provide structured legal intelligence. Sixth, legal mapping, jurisdiction selection, explanation completeness, and reporting utility are not consistently evaluated. These gaps motivate joint evaluation of technical detection and downstream legal decision support.

TABLE II COMPARATIVE LEGAL-INTELLIGENCE ANALYSIS

Dimension	India	European Union / United States
Primary framework	Information Technology Act, 2000	GDPR and Digital Services Act in relevant EU contexts; federal, state, sector-specific, and regulatory laws in the United States.
Identity misuse	Section 66C may be relevant depending on the facts.	GDPR may be relevant to personal-data processing; identity-related and computer-access laws may also apply.
Impersonation	Section 66D may be relevant where statutory conditions are met.	Potentially governed by national or platform rules; U.S. fraud or computer-related provisions may apply.

Dimension	India	European Union / United States
Privacy	Section 66E may apply to specified privacy-related conduct.	GDPR establishes personal-data requirements; U.S. privacy frameworks vary by context.
Platform/content issues	Fact- and statute-dependent.	The DSA addresses platform governance; U.S. rules vary by statute and context.

TABLE III PROPOSED ABLATION STUDY

ID	Configuration	Purpose
A1	URL only	Measure lexical and host-feature contribution.
A2	URL + content	Measure additional semantic evidence.
A3	URL + content + behavior	Measure contribution of behavioral information.
A4	Full technical model	Evaluate the complete technical architecture.
A5	Full model + legal intelligence	Evaluate legal mapping and reporting support.

8. EXPERIMENTAL VALIDATION METHODOLOGY

The framework should be tested using controlled experiments rather than a single random train–test split. The protocol distinguishes literature-derived findings from future results of an implementation.

A. Dataset Construction and Splitting

The dataset should include benign and phishing URLs, web-page text, email/message content, host features, and ethically and legally collected behavioral events where available. It should document data sources, collection dates, class distribution, language, duplicate handling, missing values, features, and preprocessing. External data should be checked for duplicates and temporal leakage. Three partitions are recommended: historical training data, historical validation data for model selection, and a temporal test set containing later observations not available during training.

B. Baselines and Ablation

Baselines should include rule/blacklist methods, Logistic Regression, SVM, Random Forest, CNN, LSTM, transformer-based content classification, and URL-only, URL-plus-content, and URL-plus-content-plus-behavior hybrids. Ablations should measure the contribution of each evidence source without implying that legal intelligence directly improves classification accuracy.

C. Adversarial and Temporal Robustness

Controlled manipulations should include redundant subdomains, character substitutions, URL encoding, lexical perturbation, HTML modification, textual paraphrasing, multilingual translation, and changes to nonessential webpage elements. Adversarial degradation can be measured as

$$D_{adv} = F1_{clean} - F1_{adv} \quad (22)$$

where the two F1-scores are measured on clean and adversarial test sets. Temporal degradation is

$$D_{temporal} = F1_{validation} - F1_{temporal} \quad (23)$$

Smaller degradation indicates greater stability under the selected test conditions. Baselines should use identical or clearly documented splits.

D. Efficiency, Multilingual and Legal Evaluation

For real-time deployment, measure inference latency, throughput, peak memory, CPU/GPU utilization where applicable, model size, and preprocessing time. Average inference latency is

$$\bar{T}=(1/N)\Sigma_{i=1}^{N}T_i \quad (24)$$

where Ti is the inference time for sample i. Multilingual evaluation should compare English-only, multilingual, translated, code-mixed, and language-specific settings. Legal-intelligence evaluation should use manually reviewed scenarios annotated for threat type, facts, jurisdiction, relevant and alternative provisions, and unavailable facts. Legal mapping accuracy and coverage are

$$Accuracy_legal = N_correct / N_total \quad (25)$$

$$Coverage_legal = N_mapped / N_applicable \quad (26)$$

Additional measures should include jurisdiction accuracy, legal-rule precision and recall, explanation completeness and fidelity, reviewer agreement, and reporting-guidance correctness.

E. Literature-Based Synthesis

Studies report strong benchmark performance for several URL, webpage, DL, and hybrid methods [1]–[3], [6], [8], [11].

Direct numerical ranking is inappropriate because datasets, class distributions, features, collection periods, and protocols differ. URL features remain useful for rapid screening; structured ML provides baselines for URL and host data; DL learns representations with less reliance on handcrafted features; NLP and transformer models add semantic analysis. Temporal and adversarial robustness remain significant challenges. No experimental accuracy is claimed for the integrated framework until implementation and evaluation are completed.

TABLE IV RECOMMENDED EXPERIMENTAL VALIDATION PROTOCOL

Evaluation	Measures	Purpose
Baseline comparison	Accuracy, precision, recall, F1, ROC–AUC	Compare individual models with the proposed technical architecture.
Ablation	F1, recall, false-positive rate	Quantify contribution of URL, content, and behavioral evidence.
Temporal evaluation	F1 degradation, recall, false-positive rate	Measure concept-drift robustness.
Adversarial evaluation	Recall and F1 degradation	Measure robustness to controlled feature manipulation.
Efficiency	Latency, throughput, memory, model size	Determine real-time deployment feasibility.
Multilingual	Per-language F1 and recall	Measure language generalization.
Legal mapping	Accuracy, precision, recall, coverage	Measure threat-to-law mapping quality.
Jurisdiction	Jurisdiction-selection accuracy	Measure legal-framework selection.
Explainability	Completeness, fidelity, consistency, human agreement	Measure interpretability and reviewer usefulness.
Reporting	Completeness and correctness	Evaluate incident-report usefulness without unsupported legal conclusions.

9. DISCUSSION AND RESEARCH CHALLENGES

Technical challenges include concept drift, class imbalance, adversarial manipulation, computational constraints, multilingual text, and multimodal attacks that combine text, images, URLs, and webpage behavior. Legal challenges include jurisdictional complexity and the need to version the knowledge base as statutes, regulations, penalties, and reporting procedures change. Integration challenges include real-time operation, evidence preservation, explainable outputs, and meaningful human review. Behavioral analysis also requires privacy safeguards and purpose limitation.

A deployed system should produce a structured incident record containing a timestamp, suspicious URL or content, threat label, supporting evidence, model confidence, candidate legal provisions, jurisdictional assumptions, missing information, and recommended reporting channel. Evaluation should check required-field completeness, correctness, and reviewer usefulness. Explanations should distinguish why the technical classifier marked an input as suspicious from why a legal provision was retrieved as a candidate.

10. FUTURE RESEARCH DIRECTIONS

Future work should implement the architecture and conduct the validation protocol. Priorities include lightweight real-time models, multilingual and code-mixed detection, multimodal analysis, continual learning, adversarial robustness, privacy-preserving behavioral processing, explainable AI, supervised legal retrieval, knowledge-base versioning, and human-in-the-loop review. Each legal rule should record its source and effective date, and updates should be validated before production use.

11. LIMITATIONS AND ETHICAL CONSIDERATIONS

The framework is not a completed production implementation and no new numerical performance value is claimed. Literature-derived results cannot be directly compared across differing datasets and protocols. The legal analysis is methodological rather than legal advice; applicability depends on facts and jurisdiction. Behavioral analysis may create privacy risks and requires data minimization and retention controls. An AI label of "suspicious" should not alone cause content to be treated as illegal. Technical classification, factual evidence, legal applicability, and legal responsibility must remain distinct.

12. CONCLUSION

Phishing and deceptive digital content require detection systems that combine technical, contextual, and legal awareness. This paper reviewed URL screening, ML, DL, NLP, behavioral analysis, hybrid architectures, and robustness issues, and proposed a modular framework that extends technical threat detection with a cyber-law intelligence layer. The layer provides candidate provisions, jurisdictional information, explanations, evidence requirements, and reporting guidance for human review rather than autonomous legal determinations. A reproducible validation pathway covering baselines, ablation, temporal and adversarial testing, multilingual evaluation, efficiency, legal mapping, jurisdiction, explainability, and reporting support is specified. Future work should implement the system, construct temporally separated multilingual datasets, evaluate robustness, and conduct expert-reviewed legal mapping.

REFERENCES

[1] S. El Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," IEEE Access, vol. 8, pp. 22170–22192, 2020, doi: 10.1109/AC-CESS.2020.2969780.
[2] M. Sanchez-Paniagua, E. Fidalgo, E. Alegre, W. Al-Nabki, and V. Gonzalez-Castro, "Phishing URL detection: A real-case scenario through login URLs," IEEE Access, vol. 10, pp. 42949–42960, 2022, doi: 10.1109/ACCESS.2022.3168681.
[3] 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.
[4] S. Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, "A survey of intelligent detection designs of HTML URL phishing attacks," IEEE Access, vol. 11, pp. 6421–6443, 2023, doi: 10.1109/ACCESS.2023.3237798.
[5] R. Zieni, L. Massari, and M. C. Calzarossa, "Phishing or not phishing? A survey on the detection of phishing websites," IEEE Access, vol. 11, pp. 18499–18519, 2023, doi: 10.1109/ACCESS.2023.3247135.
[6] O. K. Sahingoz, E. Buber, and E. Kugu, "DEPHIDES: Deep learning based phishing detection system," IEEE Access, vol. 12, pp. 8052–8070, 2024, doi: 10.1109/ACCESS.2024.3352629.
[7] S. R. Kavya and D. Sumathi, "A study on adversarial sample resistance and defense mechanism for multimodal learning-based phishing website detection," IEEE Access, vol. 12, pp. 137805–137824, 2024, doi: 10.1109/ACCESS.2024.3436812.
[8] W. Li, S. Manickam, Y.-W. Chong, W. Leng, and P. Nanda, "A state-of-the-art review on phishing website detection techniques," IEEE Access, vol. 12, pp. 187976–188012, 2024, doi: 10.1109/ACCESS.2024.3514972.
[9] R. W. Purwanto, A. Pal, A. Blair, and S. Jha, "PhishSim: Aiding phishing website detection with a feature-free tool," IEEE Trans. Inf. Forensics Security, vol. 17, pp. 1497–1512, 2022, doi: 10.1109/TIFS.2022.3164212.
[10] T. Koide, H. Nakano, and D. Chiba, "ChatPhishDetector: Detecting phishing sites using large language models," IEEE Access, vol. 12, pp. 154381–154400, 2024, doi: 10.1109/ACCESS.2024.3483905.

[11] U. Zara, K. Ayyub, H. U. Khan, A. Daud, T. Alsahfi, and S. G. Ahmad, “Phishing website detection using deep learning models,” IEEE Access, vol. 12, pp. 167072–167087, 2024, doi: 10.1109/ACCESS.2024.3486462.
[12] Government of India, Information Technology Act, 2000, India Code, Sections 66, 66B, 66C, 66D, 66E and related provisions.
[13] European Parliament and Council, “Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data,” Official Journal of the European Union, 2016.
[14] European Parliament and Council, “Regulation (EU) 2022/2065 on a Single Market for Digital Services (Digital Services Act),” Official Journal of the European Union, L 277, 2022.
[15] United States Congress, “18 U.S.C. Section 1030: Fraud and related activity in connection with computers,” United States Code.