

A Survey on Unified Phishing and Email Spam Detection Systems Using Machine Learning and NLP

Ritish Sharma¹, Sumanth Mohan², Tejaswi B N³, Jeevan Y R⁴, Mr. Keerthi K.S.⁵

^{1,2,3,4}Student, Department of Computer Science and Engineering, Malnad College of Engineering, Hassan, Karnataka, India

Guide: Mr. Keerthi K.S.

Assistant Professor, Department of Computer Science and Engineering, Malnad College of Engineering, Hassan, Karnataka, India

¹sharmaritish144@gmail.com, ²sumanthmohan877@gmail.com, ³tejaswibn.992005@gmail.com
⁴jeevanyryr@gmail.com

Abstract—Phishing websites and email spam have become two of the most persistent and damaging cyber threats facing users and organisations today. What makes modern attacks particularly difficult to handle is that they rarely rely on a single trick — attackers routinely combine deceptive URLs, social engineering, fake branding, and crafted email content all at once, making it easy to slip past conventional security tools. Systems built around rigid rule-based filters simply cannot keep up with how quickly these tactics evolve. Over the past several years, machine learning, natural language processing (NLP), deep learning, and explainable artificial intelligence (XAI) have opened up much more effective paths forward. This survey reviews existing approaches to phishing URL detection, spam classification, feature engineering, ensemble methods, semantic analysis, visual similarity detection, and real-time threat intelligence. We examine algorithms ranging from Naive Bayes and SVM through to gradient boosting and deep learning models. We also identify a persistent gap: most systems treat phishing detection and spam filtering as entirely separate problems. This survey argues that a unified pipeline enriched with LLM-based explainability and live threat intelligence feeds is the most promising direction for next-generation cybersecurity systems.

Keywords—Phishing Detection, Spam Email Classification, Machine Learning, NLP, Explainable AI, Random Forest, XGBoost, SVM, Naive Bayes, Cybersecurity, Threat Intelligence.

I. INTRODUCTION

Think about how much of daily life now runs through the internet — banking, shopping, remote work, cloud collaboration, and personal communication. Each of these activities leaves a trail that cybercriminals are eager to exploit. Among all the threats that security teams deal with, phishing and email spam stand out as the most widespread and, in many ways, the most difficult to fully eliminate [16]. The Anti-Phishing Working Group (APWG) reported over 4.7 million phishing attacks in 2022 alone — a figure that underscores how dramatically the threat landscape has scaled alongside digital adoption. The core challenge is that these attacks prey on human trust as much as they exploit technical weaknesses. A convincing fake login page, a spoofed sender address, or an email that looks like it came from a manager can fool even careful users. Attackers do not rely on a single method — they mix deceptive URLs, forged branding, malware-laced attachments, and socially tailored

messages to maximise their chances of success [2]. The convergence of these techniques into coordinated multi-vector campaigns means that single-purpose detection tools are structurally incapable of providing comprehensive coverage.

Machine learning changed the game considerably. Rather than relying on manually written rules that attackers can study and evade, ML-based systems learn directly from data and adapt as threat patterns shift [18]. NLP takes this further by giving systems the ability to read and understand text content — not just count keywords, but grasp context, detect urgency language, and spot impersonation cues. The introduction of transformer-based models such as BERT [6] represented a qualitative leap: where earlier systems processed words in isolation, transformers model bidirectional context, enabling them to understand that 'account suspended' carries a very different weight in a banking notification versus a marketing email.

Despite this progress, a frustrating gap remains: the vast majority of research treats phishing detection and spam classification as entirely separate problems [16]. Phishing URL detectors operate independently of the email clients through which those URLs arrive. Spam filters assess message text without inspecting embedded hyperlinks. Visual similarity systems check rendered page screenshots without any awareness of the originating email chain. This siloed architecture gives attackers a structural advantage — a message that passes each individual check in isolation may still be deeply malicious when its components are considered together.

This survey maps out where the field stands, identifies what is missing, and points toward what a truly unified approach might look like. We review algorithms, feature engineering strategies, NLP techniques, and system architectures, then articulate a framework that treats phishing and spam as facets of a single detection problem rather than independent tasks.

II. BACKGROUND AND CYBERSECURITY THREATS

Before diving into technical approaches, it helps to understand the nature of the threats themselves and how they have developed over time.

A. Phishing Attacks

At its core, phishing is an act of impersonation. Attackers build fake websites or craft emails that closely mimic trusted brands — banks, government agencies, popular services — to trick people into handing over credentials or personal information [20]. Modern phishing pages can be nearly pixel-perfect copies of the real thing, hosted on domains that look plausible at a glance. The adoption of HTTPS by phishing sites — now exceeding 80% according to industry reports — has further eroded the reliability of the padlock icon as a trust signal for end users.

B. Spam Emails

Spam started as a nuisance — bulk advertising that clogged inboxes. It has evolved into a serious security threat. Today's spam campaigns often carry phishing links, malware triggers, or ransomware payloads [16]. The boundary between "spam" and "attack" has become increasingly blurry, and the techniques used to evade spam filters — image-based text, character substitution, multipart MIME structures — are now regularly borrowed by phishing campaigns as well.

C. Spear Phishing and Zero-Day Attacks

Unlike bulk phishing, spear phishing targets specific individuals or organisations. Attackers research their

targets and craft messages that feel personally relevant, dramatically increasing success rates [2]. Zero-day threats exploit phishing infrastructure that has never been seen before [19]. Any model trained on historical data will have blind spots here — the strongest argument for supplementing ML models with live threat intelligence feeds such as VirusTotal [28] and Google Safe Browsing [27].

III. LITERATURE REVIEW

A substantial body of research has grown around phishing and spam detection over the past two decades. The following review traces key contributions across five research directions.

A. NLP-Based Phishing Email Recognition

Anilkumar et al. [1] conducted a comprehensive survey of NLP-based approaches to phishing email recognition, covering classifiers such as SVM, Random Forest, KNN, and Naive Bayes alongside TF-IDF and Count Vectorization. The work provides a solid overview of individual techniques but stops short of proposing any integrated architecture. Verma et al. [7] applied CNN and LSTM networks directly to phishing email content, achieving strong performance on benchmark datasets, though their approach was limited to the email body without URL or visual analysis. Peng et al. [10] extended the NLP approach by extracting urgency language and brand impersonation cues, but still lacked real-time threat intelligence integration.

B. URL-Based Detection Approaches

Moheshkumar et al. [2] built a Logistic Regression classifier assessing URLs based on length, subdomain structure, and HTTPS status. It worked reasonably well within its scope, but URL-only analysis meant it was blind to anything hidden in page content or visual design. Ma et al. [9] demonstrated that lexical and host-based URL features — including domain age, token entropy, and hyphen density — are powerful signals for detecting malicious websites, though they deliberately excluded email semantics and visual brand similarity. Blum et al. [11] introduced an online learning approach to lexical URL detection that updates incrementally from new examples, providing adaptability to evolving attack patterns, though it had no mechanism for analysing email body content or rendered page appearance.

C. Ensemble and Browser-Based Systems

Prasad et al. [3] embedded a Random Forest classifier directly into a browser extension backed by a Flask API. Real-time classification was a genuine contribution, but the system could not explain its decisions and could not

analyse emails at all. Dharmaraju et al. [4] investigated combining Random Forest, Gradient Boosting, Extra Trees, and Logistic Regression, finding that ensemble approaches consistently outperformed single models by reducing variance through diversity. Accuracy improved, but the work remained a research exercise with no deployment and no email integration. Basnet et al. [14] similarly demonstrated the power of rich feature engineering paired with SVM and Random Forest, achieving strong benchmark results while acknowledging the absence of visual and real-time components.

D. Multichannel and Semantic Systems

Mayadevi et al. [5] designed PHISHNET to handle phishing threats in URLs, SMS, and email — an ambitious scope, but each channel ran its own pipeline with no shared intelligence between modules. Kumar et al. [6] combined BERT with TF-IDF for richer semantic features, pairing them with a Random Forest classifier. The semantic understanding was a step forward, but the dataset was small and there was no visual analysis. Fette et al. [8] pioneered email header analysis for phishing detection, leveraging IP-URL mismatches and HTML content ratios alongside Naive Bayes and SVM, establishing foundational features that remain relevant today though the approach predates modern NLP-based semantic analysis.

E. Survey and Comparative Studies

Abu-Nimeh et al. [13] compared five ML techniques — Logistic Regression, Bayesian Additive Regression Trees, CART, Random Forest, and SVM — on a shared phishing email dataset, providing useful accuracy baselines but offering no NLP semantics or explainability. William et al. [15] offered a broader landscape review of AI in cybersecurity covering intrusion detection and phishing together. Mujtaba et al. [16] surveyed email classification research trends over a decade and identified siloed architectures as a persistent open problem. Salloum et al. [12] reviewed ML approaches to text mining and noted the growing applicability to phishing, while Sethi et al. [17] demonstrated fast spam detection using Count Vectorization and TF-IDF without a phishing URL or visual module.

The comparative summary of the reviewed references is provided in Table 1.

Table 1: Comparison of Existing Detection Techniques

N o.	Author(s)	Methodology	Target Area	Key Outcome
1	Anilkumar et al.	NLP-based Email Classif. (SVM, RF, KNN, NB)	Phishing Email Detection	Survey of NLP classifiers; no unified pipeline
2	Moheshkumar et al.	URL Feature Analysis (Logistic Regression)	Phishing URL Detection	Good URL accuracy; no content or visual analysis
3	Prasad et al.	Browser Extension (Random Forest + Flask)	Real-time Phishing Detection	Live classification; no explainability or email analysis
4	Dharmaraju et al.	Ensemble Learning (XGBoost, RF, Extra Trees)	Phishing Classification	Strong accuracy; no deployment or email integration
5	Mayadevi et al.	Multichannel PHISHNET (Gradient Boost)	URL + SMS + Email	Ambitious scope; isolated modules, no shared intelligence
6	Kumar et al.	BERT + TF-IDF + Random Forest	NLP + Semantic Email Analysis	Richer semantics; small dataset, no visual analysis
7	Verma et al.	CNN + LSTM (Deep Learning)	Phishing Email Detection	Strong accuracy; no URL features or visual analysis
8	Fette et al.	Email Header Analysis	Phishing Email Detection	Foundational header features;

N o.	Author(s)	Methodology	Target Area	Key Outcome
		(Naive Bayes, SVM)		predates modern NLP
9	Ma et al.	Lexical + Host-based URL Features (SVM, LR)	Malicious URL Detection	Effective URL signals; no email body or visual analysis
10	Peng et al.	NLP Linguistic Features (SVM, RF)	Phishing Detection via NLP	Urgency & impersonation cues; no live threat intelligence
11	Blum et al.	Online Learning (Lexical URL Tokens)	Phishing URL Detection	Incremental learning; no email body or visual similarity
12	Salloum et al.	ML Survey (SVM, RF, DNN)	Phishing Detection Survey	Broad ML coverage; no unified or real-time system
13	Abu-Nimeh et al.	Comparative ML Study (LR, BN, CART, RF, SVM)	Phishing Email Classification	Benchmark comparison; no NLP semantics or XAI
14	Basnet et al.	ML-based Phishing Detection (SVM, RF)	Phishing Attack Detection	Feature-rich approach; no visual or real-time analysis
15	William et al.	AI in Cybersecurity (Intrusion + Phishing)	Broad Cybersecurity Review	Landscape overview; no concrete unified system
16	Mujtaba et al.	Email Classification Survey	Spam & Email Classification	Comprehensive survey; no URL-email joint pipeline

N o.	Author(s)	Methodology	Target Area	Key Outcome
17	Sethi et al.	Count Vectorization + TF-IDF (Spam Detection)	Email Spam Classification	Fast spam features; no phishing URL or visual module

As Table 1 illustrates, no existing work brings together phishing URL detection, email spam classification, visual brand similarity, real-time threat intelligence, and explainable AI in a single coherent framework. This persistent gap motivates the unified pipeline architecture described in Section VI.

IV. MACHINE LEARNING TECHNIQUES

Understanding what each algorithm brings — and where it falls short — helps explain why ensemble approaches have become the dominant strategy in cybersecurity detection. The following subsections examine the most commonly employed algorithms and their respective trade-offs in phishing and spam detection contexts.

A. Naive Bayes

Naive Bayes is one of the oldest tools in the spam-detection toolbox and it still earns its place [26]. Built on Bayes' theorem, it estimates the probability that a message is spam based on word frequencies. It trains quickly, needs relatively little data, and handles high-dimensional text features gracefully. Its core weakness — the independence assumption among features — is less damaging in practice than theory suggests, because even a partially correct probability estimate is often sufficient for classification. In spam filtering specifically, where the feature space is the entire vocabulary of the corpus, Naive Bayes consistently outperforms more complex models in low-data regimes.

B. Support Vector Machine (SVM)

SVM [22] approaches classification geometrically, finding the hyperplane with the widest possible margin between phishing and legitimate samples in high-dimensional feature space. This margin-maximisation objective makes SVM particularly resistant to overfitting — a critical property when training on datasets where phishing examples are scarce. The kernel trick allows SVM to capture non-linear decision boundaries without explicitly computing high-dimensional feature transformations. In phishing email detection, SVM paired with TF-IDF feature vectors has consistently produced state-of-the-art results among classical classifiers [13].

C. Random Forest

Random Forest [21] builds a large collection of decision trees, each trained on a random subset of data and features, then combines their votes through majority voting. The diversity among trees — each trained on a different bootstrap sample and feature subset — makes the ensemble robust: errors that individual trees make tend to cancel out. It handles mixed feature types extremely well, which matters when combining URL statistics, text features, visual similarity scores, and header metadata in a unified model. Feature importance scores from Random Forest are also readily interpretable, providing a natural basis for explainability.

D. XGBoost

XGBoost [23] has become the go-to choice for structured data classification across many domains, including cybersecurity. It builds trees sequentially, with each new tree focused on correcting the residual errors of the ensemble so far. This boosting strategy captures subtle non-linear feature interactions that single classifiers miss — for example, the interaction between URL entropy and domain age that independently carry limited signal but together strongly indicate a freshly registered phishing domain. Paired with Random Forest in a stacked ensemble, XGBoost consistently produces state-of-the-art results on phishing detection benchmarks.

E. Deep Learning Models

CNNs, RNNs, and LSTMs [26] learn useful representations directly from raw input rather than relying on hand-crafted features. Character-level CNNs applied to raw URL strings can learn that certain character n-gram patterns are associated with phishing without being given explicit rules. LSTMs capture long-range sequential dependencies in email text, tracking how the framing of a message shifts from neutral to urgent across paragraphs. The trade-off is substantially more data, compute, and opacity — a challenge addressed in modern systems through SHAP-based post-hoc explanation.

F. Logistic Regression

Logistic Regression [2] remains a sensible and interpretable baseline for binary phishing classification. It outputs calibrated probabilities rather than hard labels, which is useful for setting custom alert thresholds and for downstream risk scoring. Its linearity makes it fast to train and to serve at inference time, and its coefficient weights offer direct interpretability — a feature that ensemble methods sacrifice. In practice, Logistic Regression is

most useful as a component in stacked ensembles or as a calibration layer on top of a more powerful classifier.

V. NATURAL LANGUAGE PROCESSING IN CYBERSECURITY

NLP gives modern detection systems the ability to read and understand text rather than merely counting words. Several preprocessing steps and feature extraction techniques work together to convert raw email and webpage text into representations that a classifier can act on effectively.

A. Tokenization and Text Preprocessing

Everything starts with tokenization — breaking raw text into individual tokens. Subword tokenization, used by transformer models like BERT [6], handles unusual or misspelled words much better than simple word splitting, making it particularly valuable in the phishing context where attackers intentionally introduce character substitutions to evade keyword filters. Stop word removal eliminates high-frequency function words that carry little discriminative signal. Stemming and lemmatization collapse morphological variants of the same word — "verify", "verifying", "verified" — into a common root, improving feature consistency across documents.

B. TF-IDF Vectorization

TF-IDF [17] is the workhorse of text-based phishing and spam detection. The term frequency component rewards words that appear often within a given document, while the inverse document frequency component down-weights words that appear across the entire corpus — distinguishing document-specific vocabulary from generic language. A phrase such as "verify your account immediately" scores high in a phishing corpus because it appears frequently in malicious messages but rarely in legitimate email. This property makes TF-IDF particularly effective at isolating the urgency and action-demanding language that characterises phishing content.

C. Word Embeddings and Transformer Models

Word2Vec [24] and GloVe represent words as dense vectors in a continuous semantic space where words with similar meanings sit close together. This allows classifiers to generalise across paraphrases — recognising that "confirm your credentials" and "validate your login" carry the same phishing signal even though they share no tokens. BERT [6] takes this further by modelling bidirectional context: the meaning assigned to each word depends on all surrounding words, enabling the model to understand that "suspended" in "your account has been suspended" carries a different connotation than in "the meeting was suspended". Fine-

tuned BERT models have achieved state-of-the-art results on phishing email classification benchmarks.

D. Semantic Feature Engineering

Beyond automated embeddings, domain-specific semantic features capture phishing-relevant signals that general-purpose models may miss. These include urgency score (density of time-pressuring phrases), impersonation score (similarity of sender display name to known brand names), action-demand density (frequency of imperative verb constructions), and link-to-text ratio (the proportion of hyperlink anchor text that mismatches the actual destination URL). Combined with TF-IDF vectors and word embeddings, these engineered features produce a rich multi-level representation of email semantics [1].

VI. FEATURE ENGINEERING FOR PHISHING AND SPAM

Good features remain the foundation of good detection, even in an era of deep learning. Phishing and spam systems draw on three complementary categories of features that together provide a multi-dimensional view of each potential threat.

A. URL Structural Features

A URL carries a surprising amount of information about its trustworthiness [9]. Key features include URL length (longer URLs with many path segments are more likely to be obfuscation attempts), subdomain count (legitimate domains rarely have more than two levels of subdomains), presence of IP addresses in the hostname (a strong phishing signal), use of URL shorteners (which mask the true destination), hyphen density in the domain name, entropy of the URL string (high entropy indicates randomly generated character sequences), and domain registration age (phishing domains are typically registered days before an attack). While HTTPS status has weakened as a standalone signal — attackers routinely obtain valid certificates — it still contributes when combined with other structural features.

B. Email Header and Metadata Features

Beyond the message body, email headers tell their own story [8]. Mismatches between the displayed sender name and the actual sending domain (From header spoofing), routing through unusual or unfamiliar relay servers, reply-to addresses that differ from the from address, missing or malformed DKIM and SPF authentication records, and unusual MIME boundary patterns are all established indicators of phishing or spam origin. These header features are highly complementary to body text features: an attacker who carefully crafts believable body

text may still leave telltale traces in the routing infrastructure.

C. Visual and Semantic Page Features

When an attacker clones a bank's website, the HTML content can be indistinguishable from the real thing [20]. Perceptual hashing compares rendered screenshots against a reference library of legitimate brand identities and flags pages that look suspiciously similar to known brands without being hosted on their actual domains. Semantic page features — extracted from HTML structure, anchor text distributions, form action URLs, and external resource loading patterns — complement visual similarity by capturing functional characteristics of phishing pages that pixel-level comparison would miss. Together, visual and semantic page features close the gap that URL-only and email-only systems leave open.

VII. CHALLENGES IN EXISTING SYSTEMS

Even with all this progress, there are genuine obstacles the field has not fully resolved. Understanding these challenges is essential for designing systems that can operate effectively in real-world production environments.

A. Lack of Unified Architectures

Phishing detectors and spam filters are almost always built as separate tools with no shared knowledge [16]. An email carrying a phishing URL might pass the spam filter because the text looks clean, and the URL analyser may never see it. This siloed architecture is not merely an engineering convenience — it reflects a conceptual separation in the research community that has persisted for two decades. The result is that real attacks, which deliberately span multiple channels and techniques, are systematically underdetected by single-purpose tools.

B. Zero-Day Threat Detection

Any model trained on yesterday's data will struggle with today's brand-new phishing domains registered hours before an attack campaign launches [19]. The only real solution is combining trained models with live threat intelligence that updates continuously. Services such as VirusTotal [28] and Google Safe Browsing [27] aggregate threat signals from millions of endpoints and provide API access that can be integrated directly into detection pipelines, ensuring that even attacks too new for any trained model to recognise are still flagged.

C. Explainability and Black-Box Decisions

A detection system that labels an email as phishing without explaining why is not much better than one that offers no detection at all from a user trust perspective [25]. Administrators who cannot audit model decisions

cannot improve them. Users who receive unexplained warnings frequently dismiss them. Integrating SHAP-based feature attribution or LLM-generated plain-language explanations transforms the detection result from an opaque label into actionable intelligence — identifying specifically which URL pattern, which header anomaly, or which urgency phrase triggered the alert.

D. Dataset Imbalance and Real-Time Constraints

In the real world, the vast majority of emails are legitimate. Training on imbalanced datasets can produce classifiers that achieve high overall accuracy by simply predicting "legitimate" for every message. Techniques such as SMOTE oversampling, cost-sensitive learning, and threshold calibration are essential for ensuring that the minority malicious class is correctly identified [16]. Separately, a phishing detector that takes several seconds to return a verdict is unusable in a browser extension context where the user is already waiting on a page load. Balancing detection thoroughness against inference latency is a real engineering challenge that many published research systems sidestep by simply not deploying at all.

VIII. UNIFIED PHISHING AND SPAM DETECTION FRAMEWORK

The most meaningful gap in the field is the absence of systems that bring phishing and spam detection together into a coherent whole. We propose a four-layer unified pipeline that addresses each of the challenges identified in Section VII.

A. Data Collection and Ingestion Layer

The foundation is a rich, multi-source data collection layer gathering: structural URL features (length, subdomain count, HTTPS status, domain age, entropy); email headers, subject lines, and full body content; rendered webpage screenshots for visual analysis; raw HTML, JavaScript, and embedded metadata; and live threat intelligence feeds from VirusTotal [28] and Google Safe Browsing [27]. All data sources are normalised into a shared feature schema before being passed to downstream modules.

B. Phishing Detection Module

Rather than relying on a single signal, this module analyses three parallel streams and fuses them through an ensemble: (1) URL structural feature analysis using XGBoost [23] over lexical and host-based features; (2) visual brand similarity detection using perceptual hashing against a reference library of 500+ legitimate brand identities; and (3) NLP-based page content analysis using a fine-tuned BERT [6] model trained on phishing page

text. A Random Forest [21] meta-classifier combines the three stream outputs into a final phishing probability score, with uncertainty quantification provided by conformal prediction intervals.

C. Email Spam Classification Module

The email module preprocesses messages through the NLP pipeline (tokenization, stop word removal, lemmatization), extracts TF-IDF features [17] alongside BERT embeddings [6] and engineered semantic features, and classifies messages with a Naive Bayes and XGBoost ensemble into one of four categories: legitimate, spam, phishing, or malware. Crucially, any URLs found in the email body are automatically extracted and passed to the phishing detection module — a spam email carrying a phishing link is caught at both levels simultaneously, closing the gap that siloed architectures leave open.

D. LLM-Based Explainability and Threat Intelligence Layer

By feeding detection results, SHAP feature importance scores, and confidence levels into a large language model [25], the system generates plain-language explanations: which URL pattern triggered the alert, which header anomaly was detected, which urgency phrases were identified, and what the user should do next. Simultaneously, all analysed URLs are checked against VirusTotal [28] and Google Safe Browsing [27] APIs. Any URL flagged by these services is escalated to confirmed malicious status regardless of what the ML model scores, providing a continuously updated safety net that complements the trained classifiers.

IX. FUTURE RESEARCH DIRECTIONS

Several directions look particularly promising for closing the gaps described above and advancing the field toward production-grade unified detection systems.

Truly unified pipelines that share learned representations across phishing, spam, malware, and intrusion detection — rather than treating each as a separate silo — would allow threat intelligence gathered from one channel to immediately inform detectors on others. Transformer-based models such as BERT [6] applied directly to raw URLs as character sequences have shown early promise at capturing morphological phishing patterns without explicit feature engineering.

SHAP-integrated LLM explainability [25] that translates numeric feature importance scores into plain-language guidance a non-expert can act on represents one of the most impactful near-term opportunities. Adaptive learning systems that update model weights continuously from confirmed new threat examples — rather than

requiring periodic full retraining cycles — would substantially reduce the zero-day detection gap [19]. Federated learning protocols would allow multiple organisations to collaboratively improve a shared detection model without sharing sensitive email or browsing data, addressing the privacy and legal barriers that currently prevent large-scale data pooling.

Graph neural networks modelling the relationships between domains, senders, URLs, and recipients capture structural attack patterns — such as coordinated domain registration clusters — that pure content-based approaches miss entirely. Multimodal detection architectures that treat email text, embedded URLs, attached documents, images, and sender behavioural history as a unified input represent the long-term direction the field is moving toward.

X. CONCLUSION

Phishing and spam are not going away. If anything, the tools available to attackers are becoming more sophisticated — generative AI has already made it easier to craft convincing, personalised phishing messages at scale, and LLM-assisted attacks can now dynamically tailor content to individual targets based on publicly available social data. The research community's response needs to match that sophistication.

This survey traced the evolution of detection techniques from simple Bayesian filters through ensemble machine learning and into NLP-enriched and deep learning approaches. Algorithms including SVM [22], Naive Bayes, Random Forest [21], Logistic Regression, and XGBoost [23] each contribute something distinct, and the strongest systems today combine several of them. NLP tools — TF-IDF [17], word embeddings [24], and contextual models like BERT [6] — have transformed what it means to analyse email and webpage content. We reviewed 17 representative works spanning URL detection, email classification, semantic analysis, ensemble methods, and survey studies, summarised in Table 1.

What this survey keeps returning to is the gap between individual technical advances and systems that actually work together. Isolated architectures, unexplainable decisions, and blind spots for zero-day threats [19] are the gaps that real attackers slip through. Unified frameworks that connect phishing detection to email classification, explain their reasoning in plain language, and stay current through live threat intelligence APIs [27, 28] represent the direction the field needs to move in. The building blocks are fully available — what is needed now is the

architectural vision to assemble them into a coherent whole.

REFERENCES

- [1] C. Anilkumar, A. Karrothu, N. Sri Mouli, and C. Bhanu Tej, "Recognition and Processing of Phishing Emails Using NLP: A Survey," in Proc. Int. Conf. Computer Communication and Informatics (ICCCI), 2023.
- [2] A. R. More, O. B. Shendkar, K. G. Pathak, P. D. Sanap, S. N. Pachore, and P. William, "Survey on Framework for Governing Modern Cyber Attacks in Computer Network," in Proc. ICCAKM, 2023.
- [3] A. Jain and B. Gupta, "A Machine Learning Based Approach for Phishing Detection Using Hyperlinks Information," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 5, pp. 2015–2028, 2019.
- [4] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical Feature Based Phishing URL Detection Using Online Learning," in Proc. ACM Workshop on AI and Security, pp. 54–60, 2010.
- [5] S. Garera, N. Provos, M. Chew, and A. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," in Proc. ACM Workshop on Recurring Malcode, pp. 1–8, 2007.
- [6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019.
- [7] P. Verma, V. Goyal, and Y. Verdhan, "Phishing Website Detection Using Neural Network and Deep Learning," International Journal of Computer Applications, vol. 176, no. 39, pp. 1–6, 2020.
- [8] I. Fette, N. Sadeh, and A. Tomasic, "Learning to Detect Phishing Emails," in Proc. Int. Conf. World Wide Web, pp. 649–656, 2007.
- [9] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in Proc. ACM SIGKDD, pp. 1245–1254, 2009.
- [10] T. Peng, I. Harris, and Y. Sawa, "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning," in Proc. IEEE 12th Int. Conf. Semantic Computing (ICSC), 2018.
- [11] A. Blum, B. Wardman, T. Solorio, and G. Warner, "Lexical Feature Based Phishing URL Detection Using Online Learning," in Proc. ACM Workshop on AI and Security, pp. 54–60, 2010.

- [12] S. Salloum, M. Al-Emran, A. Khalaf, and K. Shaalan, "A Survey of Text Mining in Social Media: Facebook and Twitter Perspectives," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 1, pp. 127–133, 2017.
- [13] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," in *Proc. eCrime Researchers Summit*, pp. 60–69, 2007.
- [14] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of Phishing Attacks: A Machine Learning Approach," *Studies in Computational Intelligence*, vol. 226, pp. 373–383, 2009.
- [15] P. William, S. Aggarwal, A. Gupta, S. Alshehri, and T. R. Gadekallu, "Implementation of Machine Learning Classification Techniques for Intrusion Detection System," in *Proc. ICIEM*, 2023.
- [16] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email Classification Research Trends: Review and Open Issues," *Artificial Intelligence Review*, vol. 50, pp. 221–250, 2017.
- [17] M. Sethi, A. Bhandari, and B. Kohli, "Spam Detection Using Count Vectorization and TF-IDF," *Procedia Computer Science*, vol. 184, pp. 456–463, 2021.
- [18] P. Verma et al., "Phishing Email Detection Using NLP and Machine Learning," *IEEE Access*, vol. 8, pp. 11276–11285, 2020.
- [19] N. Kumar et al., "Machine Learning Techniques for Spam Detection," *Int. Journal of Advanced Computer Science*, vol. 11, no. 2, pp. 100–110, 2020.
- [20] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
- [21] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [22] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [23] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. ACM KDD*, 2016.
- [24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv:1301.3781*, 2013.
- [25] A. Ng, "Machine Learning and Artificial Intelligence in Cybersecurity," *Communications of the ACM*, vol. 64, no. 7, pp. 34–41, 2021.
- [26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [27] Google, "Google Safe Browsing API Documentation," [Online]. Available: <https://safebrowsing.google.com/>
- [28] VirusTotal, "VirusTotal API Documentation," [Online]. Available: <https://developers.virustotal.com/reference/overview>