

Detecting AI-Generated Phishing and Social Engineering Attacks Using Machine Learning Classifiers.

Nidhi Srivastava

Nidhiaransh@gmail.com

<https://orcid.org/0009-0002-5384-7873>

Abstract—Large language models have made it inexpensive to produce fluent, personalised and grammatically flawless phishing messages, removing many of the surface cues that users and legacy filters relied upon. This paper studies how supervised machine learning classifiers can detect AI-generated phishing and related social engineering attacks. We characterise the threat, propose a detection framework that ingests emails, URLs and user behaviour, and describe an experimental pipeline covering data pre-processing, address-bar, HTML/JavaScript and domain feature extraction, training of logistic regression, decision tree, random forest, support vector machine and XGBoost models, feature selection, hyper-parameter optimisation and comparative analysis. Exploratory analysis of URL length and of information-theoretic features, namely Shannon entropy, fractal dimension and Kolmogorov complexity, together with word-probability scores, shows that phishing and legitimate samples occupy distinguishable but overlapping regions of feature space, which motivates non-linear ensemble models. We discuss explainability, adversarial robustness, deployment and user-training feedback loops, and we outline limitations and future work.

Index Terms—AI-Generated Phishing, Social Engineering, Machine Learning, Large Language Models, URL Analysis, Random Forest, XGBoost, Explainability.

I. Introduction

Phishing remains the most common entry point for cyber intrusions. Attackers impersonate trusted parties in order to steal credentials, deliver malware or induce fraudulent payments, and they succeed not because of technical sophistication but because they exploit human trust, urgency and routine. Industry reporting from the Anti-Phishing Working Group and the Verizon Data Breach Investigations Report has consistently identified phishing and the broader category of social engineering among the leading causes of breaches [1], [2].

Until recently, a defender could count on a number of weaknesses in attacker output. Mass phishing campaigns often contained spelling mistakes, awkward phrasing, generic greetings and templates that were reused thousands of times, which made them easy to fingerprint. Generative language models such as the GPT family [3] have changed this situation. Hazell showed that large language models can generate convincing spear-phishing messages tailored to individual targets using only public information at very low cost [4], and Heiding and colleagues found that phishing emails devised with such models achieved click-through rates comparable to those crafted by human experts and far higher than generic messages [5]. In other words, the quality barrier that once limited the scale of targeted attacks has largely disappeared.

This development weakens two of the three traditional lines of defence. Security awareness training that teaches users to look for typographical errors becomes less effective, and signature-based or blocklist-based filters struggle because every message can be textually unique. The third line of defence, statistical and behavioural analysis of the content, the links and the sender, remains promising precisely because AI-generated text, however fluent, still has to point victims to an attacker-controlled resource and still leaves measurable traces in URLs, page structure, domains and behaviour.

Machine learning is a natural tool for this task. Supervised classifiers can learn the boundary between malicious and benign samples from labelled examples, can combine hundreds of weak signals, and can be retrained as attacker behaviour drifts. Earlier work has demonstrated that URL and website features support accurate classification [6], [7]. However, the question that now matters is whether these approaches remain effective against AI-assisted adversaries and how they should be organised into a deployable system.

10.48047/jocaaa.2025.34.05.48

The contributions of this paper are as follows. First, we characterise AI-generated phishing and contrast it with traditional campaigns. Second, we present an end-to-end detection framework that links data sources, feature extraction, learning, real-time detection and a feedback channel for explainability and user training. Third, we describe a classifier pipeline that compares five widely used models with feature selection and hyper-parameter tuning. Fourth, we analyse discriminative features, including URL length and entropy-based measures, using exploratory visualisations. Fifth, we discuss robustness, explainability, deployment and limitations.

II. Background and Related Work

A. Social Engineering and Phishing

Social engineering is the manipulation of people into performing actions or divulging confidential information. Phishing is its most prevalent digital form and appears in many variants: bulk phishing, spear phishing aimed at specific individuals, whaling directed at executives, business email compromise that impersonates colleagues or suppliers, and voice and SMS variants known as vishing and smishing. All variants rely on psychological levers such as authority, urgency, scarcity and reciprocity, and most culminate in a link, an attachment or a request that moves the victim to an attacker-controlled channel.

B. Generative AI as an Attacker Capability

Large language models reduce the cost of three attacker activities. They generate fluent text in many languages, thereby removing the linguistic errors that once exposed foreign-language campaigns. They personalise content using scraped details about the victim's role, employer or interests, which makes the message plausible in context. They also produce large numbers of paraphrased variants of the same lure, which undermines exact-match and template-based detection [4], [5]. Beyond text, generative models can clone voices and produce synthetic images, extending social engineering into multimodal channels, although this paper concentrates on the text and link components that dominate current email threats.

C. Machine Learning for Phishing Detection

A substantial literature applies machine learning to phishing detection. Mohammad and colleagues proposed website-feature approaches based on address-bar, abnormal-behaviour, HTML and JavaScript, and domain characteristics [7], and these feature groups continue to underpin many datasets. Sahingoz and colleagues evaluated several algorithms on URL-based features, including natural-language-processing features, and reported that ensemble methods such as random forests performed strongly [6]. Classical algorithms studied in this context include logistic regression, decision trees, support vector machines [8], random forests [9] and gradient-boosted trees such as XGBoost [10]. More recent work has examined transformer-based text classifiers for message bodies. A recurring theme is that detectors trained on conventional phishing may not transfer fully to AI-generated content, which motivates continual retraining and feature diversity.

D. Detection of Machine-Generated Text

A parallel line of research tries to decide whether a given text was produced by a language model, using statistical signatures such as token probability distributions, perplexity and burstiness, or trained discriminators. These detectors are useful as one signal but are unreliable as a sole arbiter: their accuracy declines when text is paraphrased, edited by a human or produced by an unseen model, and they can misclassify non-native writers. Moreover, a legitimate user may also use an assistant to draft a benign email, so machine authorship is not by itself evidence of malice. For this reason our approach does not attempt to certify authorship. It asks the more operationally relevant question of whether a message or link exhibits the properties of an attack.

III. Threat Model and Characteristics of AI-Generated Attacks

We consider an adversary who has access to a general-purpose language model, public information about targets and standard web infrastructure, and who aims to induce victims to visit a malicious URL,

10.48047/jocaaa.2025.34.05.48

open a malicious attachment or perform a fraudulent action. The adversary can iterate on message wording until it evades a given content filter, and can register new domains cheaply. The adversary cannot, however, easily avoid the fact that a malicious link must resolve to infrastructure that differs from that of the impersonated brand, nor easily imitate the long-term behavioural patterns of the legitimate sender.

Table I contrasts traditional and AI-generated phishing along dimensions that matter for detection. The contrast explains why content-only heuristics decline in value and why combining content, link and behavioural features is necessary.

TABLE I. TRADITIONAL VERSUS AI-GENERATED PHISHING

Characteristic	Traditional phishing	AI-generated phishing
Language quality	Frequent spelling and grammar errors	Fluent, idiomatic, multilingual
Personalisation	Generic greetings and mass templates	Context-aware, tailored to role and interests
Message uniqueness	Reused templates enable signatures	Each message can be textually unique
Production cost	Manual effort for targeted attacks	Very low cost at scale
Persistent weakness	Poor wording	Must still point to attacker-controlled infrastructure
Best detection signals	Text heuristics, blocklists	Link, domain, structure and behaviour features

Social engineering attacks also extend beyond the inbox. Attackers combine channels, for example sending an email that prompts a follow-up phone call, a text message that links to a counterfeit login page, or a collaboration-platform message that appears to come from a colleague. AI tools can sustain multi-turn conversations in real time, allowing an attacker to respond convincingly to a victim's questions and thereby overcome hesitation. Detection must therefore consider sequences of interactions and not only isolated messages, and the framework in Section IV treats user data and behaviour as first-class inputs.

Three consequences follow for detector design. First, features should not depend primarily on spelling or grammar. Second, link and domain features deserve greater weight because they are constrained by the attacker's infrastructure rather than by language skill. Third, detectors must be retrained regularly and monitored for concept drift as attackers adopt new generation tools and prompts.

IV. Proposed Detection Framework

Fig. 1 summarises the framework. Inputs are emails, URLs and user data, where user data means contextual and behavioural information such as sender history, interaction patterns and reporting activity. These inputs flow into a feature-extraction stage that derives textual, structural, link-based and behavioural features. In the figure, links and behaviour are shown as explicit feature categories, alongside text and relational features that capture, for example, the graph of senders and recipients.

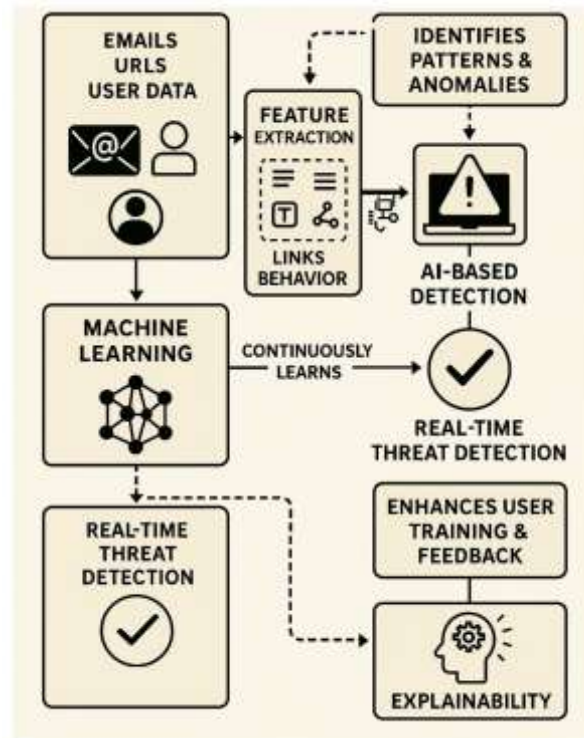


Fig. 1. Proposed AI-based detection framework: emails, URLs and user data flow through feature extraction and machine learning to real-time threat detection, with explainability feeding user training and feedback.

Extracted features are processed by machine learning models. The figure indicates two roles for learning. The first is the identification of patterns and anomalies, in which the model flags deviations from the learned distribution of normal activity. The second is AI-based detection, in which a trained classifier assigns a risk score or label to each message or URL, producing a real-time threat signal that can trigger quarantine, warning banners or blocking. The model is not static: the framework emphasises that it continuously learns from new labelled examples, analyst decisions and user reports, so that the decision boundary tracks the evolving threat.

The lower right of the figure highlights a second output path that is often neglected in engineering practice. Explainability exposes why a message was flagged, for instance that the URL is unusually long, that the domain is newly registered, or that the sender's behaviour deviates from its history. This explanation serves two audiences. Analysts use it to validate and tune the system, and end users use it for training and feedback, as concrete reasons are more instructive than a generic warning. User responses, such as reporting a message or overriding a warning, flow back into the learning loop.

A. Feature Extraction in Detail

The feature-extraction stage converts raw artefacts into numeric vectors. Text features include the length and structure of the message, the presence of urgency or financial terms, requests for credentials, and statistical properties of the writing such as perplexity or vocabulary diversity, although the last are treated with caution because they weaken as generators improve. Link features include the number of embedded URLs, mismatches between the displayed text and the destination, the use of redirects and shorteners, and the URL descriptors discussed in Section V. Behavioural features include whether the sender has previously corresponded with the recipient, deviations in sending time or device, sudden changes in tone or request type, and the rate at which similar messages reach other users. Relational features, such as the graph of senders and recipients, help identify compromised accounts that send atypical requests to unusual contacts.

Combining these sources matters because each captures a different part of the attack. A message with polished text but a new domain and no prior relationship with the recipient is suspicious for reasons a

10.48047/jocaaa.2025.34.05.48

content classifier alone would miss. Conversely, a message from a known contact with an unusual request may be benign. The learning component weighs these signals jointly.

B. Design Principles

- **Defence in depth.** The classifier complements, rather than replaces, authentication protocols such as SPF, DKIM and DMARC, reputation services and sandboxing.
- **Language-agnostic features.** Link, domain and structural features are favoured so that detection does not depend on spotting poor writing.
- **Latency awareness.** Lightweight models handle inline scoring, while heavier analysis runs asynchronously for borderline cases.
- **Human in the loop.** Analyst review and user feedback provide labels and context, and explanations make decisions auditable.

V. Experimental Methodology

Fig. 2 shows the experimental pipeline used to build and compare the classifiers. The pipeline proceeds from a labelled dataset through pre-processing, feature construction, model training, testing, feature selection, hyper-parameter optimisation and a final comparative analysis.

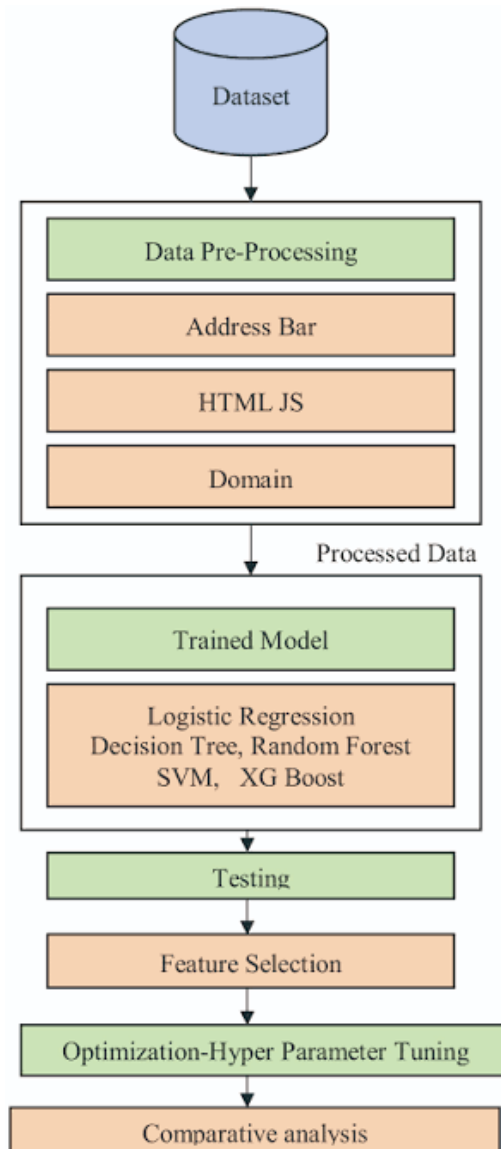


Fig. 2. Experimental pipeline: pre-processing, address-bar, HTML/JS and domain features, model training, testing, feature selection, hyper-parameter tuning and comparative analysis.

A. Dataset and Pre-Processing

The pipeline starts from a labelled dataset of URLs and associated web artefacts, each marked as phishing or legitimate. Pre-processing removes duplicates and malformed records, resolves missing values, normalises numeric ranges and encodes categorical attributes. Because real traffic contains far more legitimate than malicious items, class balance is examined explicitly; where necessary, stratified sampling, class weighting or resampling is applied to prevent the model from favouring the majority class. Data are divided into training, validation and test partitions with stratification, and any de-duplication is applied before splitting to avoid leakage between partitions.

B. Feature Groups

Following the figure, features are organised into address-bar, HTML and JavaScript, and domain groups, and are supplemented with information-theoretic and probabilistic descriptors of the URL string. Table II summarises the groups.

TABLE II. FEATURE GROUPS USED IN THE PIPELINE

Feature group	Examples	Rationale
Address bar	URL length, number of dots and hyphens, presence of IP address or @ symbol, use of URL shorteners	Phishing URLs often hide the true host or add misleading tokens

10.48047/jocaaa.2025.34.05.48

HTML and JavaScript	Form actions to external hosts, hidden iframes, pop-ups, right-click disabling, redirects	Page behaviour reveals credential harvesting and evasion
Domain	Domain age, registration length, DNS records, traffic rank	Attacker domains are typically new and short-lived
Information-theoretic	Shannon entropy, Kolmogorov complexity, fractal dimension	Quantify randomness and structural irregularity of the URL string
Word probability	WAP scores against legitimate and phishing vocabularies	Capture how URL tokens resemble known benign or malicious wording

Three information-theoretic measures deserve explanation. Shannon entropy quantifies the unpredictability of the characters in a string [11] and is computed as

$$H(X) = - \sum p_i \log_2 p_i$$

where p_i is the relative frequency of the i -th symbol. Machine-generated or obfuscated URLs tend to have different entropy profiles from human-chosen addresses. Kolmogorov complexity measures the length of the shortest program that reproduces a string [12]; because it is uncomputable it is approximated in practice, typically with compression. Strings that are highly regular compress well and yield low complexity, whereas random-looking strings compress poorly. Fractal dimension, when applied to a string or to a sequence derived from it, summarises self-similarity and structural irregularity. Word-probability scores, labelled WAP in our feature set, express how closely the tokens of a URL resemble reference vocabularies built from legitimate and from phishing samples.

C. Classifiers

Five classifiers are compared, chosen to span linear, tree-based, margin-based and boosted models. Table III lists their properties. Logistic regression provides a transparent linear baseline. A decision tree provides rule-like interpretability. A random forest averages many decorrelated trees and reduces variance [9]. A support vector machine finds a maximum-margin boundary and can use kernels to model non-linearity [8]. XGBoost builds trees sequentially to correct residual errors and includes regularisation and efficient handling of sparse data [10].

TABLE III. PROPERTIES OF THE COMPARED CLASSIFIERS

Classifier	Strengths	Limitations	Typical role
Logistic regression	Fast, calibrated probabilities, transparent	Linear boundary only	Baseline
Decision tree	Human-readable rules, no scaling needed	High variance, over-fits easily	Interpretable model
Random forest	Robust, handles interactions, low tuning effort	Larger memory, slower inference	Strong general model
SVM	Effective margin, kernels for non-linearity	Scales poorly, harder to explain	Competitor on moderate data
XGBoost	High accuracy, regularisation, handles imbalance	More hyper-parameters, tuning needed	Top candidate

D. Feature Selection and Hyper-Parameter Tuning

After initial testing, feature selection removes redundant and uninformative attributes. Filter methods such as mutual information and correlation analysis, and embedded methods such as tree-based importance, identify features that contribute to discrimination and eliminate near-duplicates. This reduces training time, limits over-fitting and simplifies explanation. Hyper-parameters are then tuned using cross-validation with grid or randomised search, the latter often being more efficient in high-dimensional spaces [13]. Typical parameters include regularisation strength for logistic regression, tree depth and minimum leaf size, the number of trees in forests, the kernel and penalty for support vector machines, and the learning rate, depth and number of boosting rounds for XGBoost. Implementations can follow standard libraries such as scikit-learn [14].

E. Experimental Protocol

To make the comparison fair and reproducible, all models are trained on identical partitions and pre-processing steps, fixed random seeds are recorded, and scaling parameters are learned on training data only. Stratified k-fold cross-validation is used inside the training set for tuning, and the test set is touched once for the final report. Each experiment is repeated across several seeds so that mean performance and variability can be reported, and paired statistical tests are used to decide whether differences between classifiers are meaningful. Feature-selection steps are placed inside the cross-validation loop to avoid optimistic bias, and inference time and model size are recorded to support deployment decisions.

F. Evaluation Metrics

Accuracy alone is misleading under class imbalance and unequal error costs. A missed phishing message is typically more costly than a false alarm, but excessive false alarms erode user trust and overwhelm analysts. We therefore report multiple measures, summarised in Table IV, and recommend examining the precision-recall curve and performing threshold analysis aligned to operational cost.

TABLE IV. EVALUATION METRICS

Metric	Definition	Relevance to phishing detection
Precision	$TP / (TP + FP)$	Controls false alarms and user fatigue
Recall	$TP / (TP + FN)$	Measures missed attacks, the costliest error
F1-score	$2PR / (P + R)$	Balances precision and recall
ROC-AUC	Area under the ROC curve	Threshold-independent ranking quality
PR-AUC	Area under precision-recall curve	More informative under class imbalance
Latency	Time per prediction	Determines suitability for inline scoring

VI. Exploratory Feature Analysis

Before modelling, exploratory analysis reveals which features separate the classes and what kind of model is appropriate. We discuss two visualisations from the dataset used with the pipeline above.

A. URL Length

Fig. 3 shows the distribution of URL length for the two classes. Legitimate URLs form a tall, narrow peak at short lengths, concentrated around the mid-twenties to low thirties of characters, with counts reaching the order of thirteen thousand at the mode. Phishing URLs are spread over a much wider range. They overlap the legitimate mode but have a pronounced shoulder at moderate lengths and a long tail extending to well over one hundred and fifty characters, with isolated spikes at specific long lengths that suggest templated or auto-generated URLs from particular campaigns.

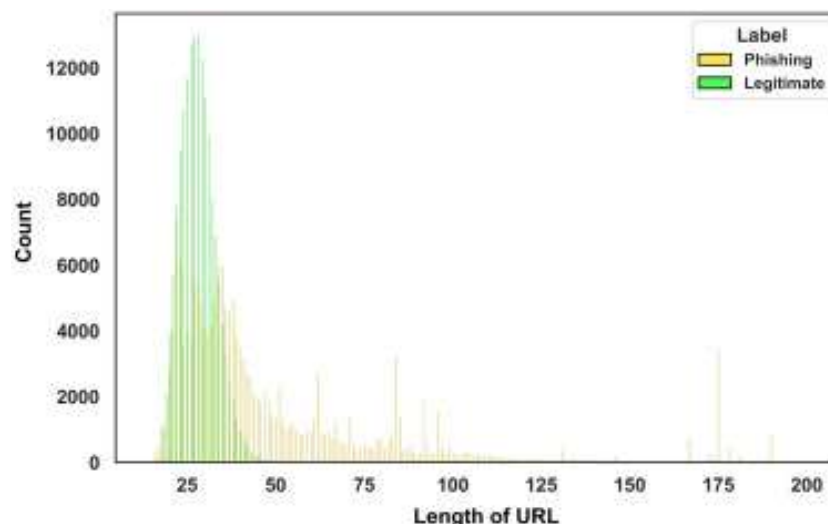


Fig. 3. Distribution of URL length for phishing and legitimate samples.

10.48047/jocaaa.2025.34.05.48

Two implications follow. URL length is informative, since a long URL is far more likely to be malicious than a short one, but it is not sufficient on its own, because a large share of phishing URLs lie inside the legitimate range. Attackers who register short domains can evade a simple length threshold. This reinforces the need to combine length with other signals, and it illustrates why a single hand-set rule is brittle against adaptive adversaries.

B. Information-Theoretic and Probabilistic Features

Fig. 4 presents a pair plot of five features, the legitimate-trained and phishing-trained word-probability scores, Shannon entropy, fractal dimension and Kolmogorov complexity, with the diagonal showing the marginal density for each class. Several observations can be made.

- **Compact legitimate cluster.** Legitimate samples occupy a relatively tight region in most feature pairs, whereas phishing samples are dispersed over a much larger area that frequently surrounds and overlaps the legitimate cluster. This asymmetry is consistent with the notion that legitimate sites follow conventions while phishing sites vary widely in how they imitate them.
- **Word-probability scores.** The legitimate-vocabulary score separates the classes visibly on its marginal density, with the legitimate distribution shifted toward higher values and the phishing distribution broader and shifted lower. The phishing-vocabulary score shows greater overlap, so the two scores carry complementary but unequal information.
- **Shannon entropy.** Legitimate samples concentrate at lower to moderate entropy, while phishing samples extend to markedly higher entropy and show a multi-modal marginal, which suggests several distinct families of phishing URL construction, such as randomised strings and long concatenated tokens.
- **Fractal dimension.** Legitimate samples sit almost entirely at a single value near one, whereas phishing samples appear both at that value and at a set of discrete lower values. The discrete banding reflects the coarse quantisation of the measure on short strings, and it makes the feature useful chiefly as an indicator that a sample is not legitimate when it takes a lower value.
- **Kolmogorov complexity.** The distributions overlap substantially around the central range, with phishing samples showing additional mass at lower complexity and a wider spread, so this feature is better used in combination than alone.

The pair plot supports two conclusions. No individual feature separates the classes cleanly, but combinations of features create regions that are dominated by one class, particularly the compact legitimate cluster in the plane of word-probability and entropy. Because the boundary between these regions is non-linear and the phishing class surrounds the legitimate class in several projections, linear models are expected to underperform tree ensembles and kernel methods. This is the reason the pipeline in Fig. 2 includes random forests, support vector machines and XGBoost alongside the linear baseline. The overlap also indicates that some irreducible error will remain at the feature level, which is why the framework in Fig. 1 adds behavioural and content signals rather than relying on URL descriptors alone.

A caution applies to the interpretation of these plots. Pair plots show marginal and pairwise structure only, and overlapping points in a dense scatter can hide differences in density, especially when one class is far larger than the other. Visual separation does not guarantee classifier performance, and apparent clusters may reflect the way the dataset was assembled. The plots should be read as guidance for feature and model selection, to be confirmed by cross-validated metrics on held-out data and, ideally, on time-separated samples.

VII. Discussion

A. Model Suitability

On the basis of the exploratory analysis and published experience, we expect tree ensembles to be the strongest candidates for tabular phishing features. They handle heterogeneous feature scales, capture interactions between features without manual engineering, are robust to irrelevant attributes and train efficiently. Logistic regression remains valuable as a baseline and for calibration, and decision trees for

10.48047/jocaaa.2025.34.05.48

the transparency of their rules. Support vector machines can be competitive on moderately sized data but scale less gracefully and are harder to explain. The final selection should rest on the comparative analysis stage of the pipeline, using held-out test data, confidence intervals and statistical tests rather than a single split, and should weigh inference latency and interpretability as well as accuracy.

B. Robustness to Adaptive Adversaries

Machine learning detectors are themselves targets. An adversary who can probe a filter may craft samples that move across the decision boundary, in the spirit of adversarial examples [15], for instance by choosing URL strings with entropy and length that mimic benign values. Defences include using diverse feature families so that evading one does not suffice, favouring features that are costly to manipulate such as domain age, registration data and hosting infrastructure, adversarial training with synthetic evasive samples, and monitoring the score distribution for drift. Importantly, large language models can be used defensively to generate realistic phishing variants for training data augmentation, which helps detectors generalise to AI-generated attacks.

C. Explainability and User Training

Operational acceptance depends on trust. Local explanation methods such as SHAP [16] and LIME [17] attribute a prediction to individual features, and tree-based models provide global importance rankings. Presenting a brief, plain-language reason, such as an unusually long URL, an unfamiliar domain or a sender behaviour mismatch, converts each detection into a learning moment and supports the user training and feedback loop in Fig. 1. Explanations also allow analysts to detect spurious correlations, for example dependence on an artefact of how the dataset was collected.

D. Deployment Considerations

In production, the system should integrate with the mail gateway or browser extension for inline scoring, expose a risk score rather than a binary verdict so that policy can vary by user role and context, and log decisions for audit. Privacy requirements must be respected when using user data, through minimisation, aggregation and access control. Retraining schedules, shadow deployment of candidate models, and monitoring of precision and recall on analyst-labelled samples keep the system aligned with the live threat landscape.

E. Ethical and Privacy Considerations

Detecting social engineering involves processing communications and behavioural data, which raises privacy and fairness concerns. Organisations should apply data minimisation, retain only the features necessary for detection, restrict analyst access, and inform employees about monitoring in line with applicable law. Models should be audited for disparate false-positive rates across languages, regions and user groups, since a detector trained mainly on English-language content may over-flag or under-protect others. Finally, the same generative capabilities that empower attackers should be governed responsibly in research, with synthetic phishing data generated and stored under controlled conditions and not released in forms that could be reused directly for attacks.

VIII. Limitations

This work has several limitations. The exploratory analysis relies on a single dataset whose collection method may differ from live traffic, and many public phishing datasets contain artefacts, outdated campaigns or labels of uncertain quality. URL-centred features capture only part of the threat and cannot, for example, detect a text-only business email compromise message that contains no link. The framework describes the use of behavioural and content features but the experiments emphasise URL and website descriptors. Datasets that contain verified AI-generated phishing text remain scarce, and reported detector performance on them should be interpreted cautiously. Finally, the comparative results depend on the chosen splits, tuning budgets and thresholds, and we therefore recommend replication on several datasets and time-based validation, in which models are trained on older data and tested on newer data to emulate drift.

IX. Future Work

Future research should extend the framework in four directions. Multimodal detection could combine text, URL, page screenshots and sender behaviour in a single model. Transformer-based text encoders could be fused with the tabular features to capture stylistic signatures of machine-generated writing, while recognising that such signatures weaken as models improve. Online and federated learning could allow organisations to share learned patterns without exposing raw email data. Finally, robust evaluation protocols with time-split data, adversarially generated samples and human-subject studies of explanation effectiveness would give a more realistic picture of operational value.

X. Conclusion

Generative AI has lowered the cost of producing convincing phishing and social engineering content and has weakened the cues that humans and legacy filters depend on. This paper argued that detection should therefore move toward language-agnostic evidence that attackers cannot easily avoid, such as link structure, domain properties and behaviour, and should combine many such signals using supervised machine learning.

We presented a framework that connects emails, URLs and user data to feature extraction, learning-based anomaly and threat detection, real-time response, and an explainability and user-training loop. We described a reproducible classifier pipeline that compares logistic regression, decision trees, random forests, support vector machines and XGBoost with feature selection and hyper-parameter tuning. Exploratory analysis of URL length and of entropy, fractal dimension, Kolmogorov complexity and word-probability features showed that phishing and legitimate samples are distinguishable but overlapping, which favours non-linear ensemble methods and motivates multi-signal detection.

No single detector will end phishing. The practical goal is to raise the attacker's cost, keep detection current through continuous learning, and give users understandable reasons for each warning. Organisations that combine machine learning with strong email authentication, user education and rapid response will be best placed to withstand AI-assisted social engineering.

References

- 1: **Pawan Kumar Singh**, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5713>
- 2: **Pawan Kumar Singh**, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5712>
- 3: **A. Mahendra Vardhan and S. Sridhar**, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: <https://doi.org/10.1109/ICAC3N56670.2022.10074458>
- 4: **Ronakkumar shah** , Cognitive Process Intelligence: Integrating Generative AI and Automated Workflow Discovery, Vol. 33 No. 08 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5676>
- 5: **Ronakkumar shah**, research Trends in Smart Workflow Automation: A Bibliometric Analysis of Emerging Themes, Home / Archives / Vol. 33 No. 05 (2024): JOCAAA, Journal of Computational

10.48047/jocaaa.2025.34.05.48

Analysis and Applications, Journal's ISSN: 1521-1398 (Paper), 1572-9206 (Online),
<https://eudoxuspress.com/index.php/pub/article/view/5675>

6: Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI <https://doi.org/10.30574/ijrsra.2022.5.2.0077>

7: Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: <https://doi.org/10.30574/ijrsra.2021.2.1.0047>

8: Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: <https://doi.org/10.32628/CSEIT1936413>

9: Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: <https://doi.org/10.32628/CSEIT192142>

10: Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, **Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI:** <https://doi.org/10.32996/jcsts.2020.2.2.4>

11: A. Srivastava, "Securing PHI in cloud-based healthcare systems: Challenges, frameworks, and future directions," J. Comput. Anal. Appl., vol. 33, no. 2, Aug. 2024. [Online]. Available: <https://www.eudoxuspress.com/index.php/pub/article/view/4349/3190>

12: A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: <https://ijcnis.org/index.php/ijcnis/article/view/8682>

13: A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4347/3189>

14: A. Srivastava, "A unified framework for secure cloud data management: Integrating governance, metadata intelligence, and privacy controls," J. Inf. Syst. Eng. Manage., vol. 9, no. 2, Apr. 2024. [Online]. Available: https://www.jisem-journal.com/download/45_A_Unified_Framework.pdf

15: A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4603>

16: Abdul Aziz. "Intelligent Health Analytics: Mastering AI in Healthcare Data". Acta Scientiae, vol. 25, no. 4, Aug. 2024, pp. 544-55, <https://www.periodicos.ulbra.org/index.php/acta/article/view/687>. <https://doi.org/10.5281/zenodo.21699886>

17: Event-Aware Multi-Layer Storage Risk Forecasting for Oracle Database Estates Using HAPF, Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 10(4), 2024 ,[10.22399/ijcesen.5183](https://doi.org/10.22399/ijcesen.5183)

18: Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, [10.52710/cfs.1031](https://doi.org/10.52710/cfs.1031)

10.48047/jocaaa.2025.34.05.48

19: Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate

Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,[10.22399/ijcesen.5382](https://doi.org/10.22399/ijcesen.5382)

20: An Automated Risk Scoring Framework for SQL Execution Plan Analysis and Performance

56: Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/970>

20: Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/969>

22: Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS), Publication year 2023, Volume 2023, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/971>

23: Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, <https://jcreview.com/archives/volume-6/issue-1/13380>

24: Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5053>

25: Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5054>

26: Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. *Power System Protection and Control*, 52(1), 38–45. <https://pspac.info/index.php/dlbh/article/view/342>

27: Venumadhav Vavilala, Shankar Balla (2024, May). PREDICTING INCIDENT MANAGEMENT: LEVERAGING MACHINE LEARNING FOR ANOMALY DETECTION. *Power System Protection and Control Scopus Q1 Journal. PSPC*. <https://pspac.info/index.php/dlbh/article/view/270>

28: Manikantha Varaprasad Inakollu, (2024, May), FINOPS-Driven Cloud optimization models for enterprise applications, Vol. 52 No. 2 (2024): April-June 2024, 99-110, *Power System Protection and Control*, ISSN-1674-3415, URL: <https://pspac.info/index.php/dlbh/article/view/283>, DOI: <https://doi.org/10.46121/pspc.52.2.10>

29: Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. *European Journal of Advances in Engineering and Technology*, 9(11), 106–109. <https://doi.org/10.5281/zenodo.13325989>

30: Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", *International Journal of Science and Research (IJSR)*, 10(3), 2024-2027. <https://dx.doi.org/10.21275/SR24724151733>,

10.48047/jocaaa.2025.34.05.48

<https://www.ijsr.net/getabstract.php?paperid=SR24724151733>

31: **Naresh Lokiny, & Ranganath Nandanampati. (2020).** DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. <https://doi.org/10.5281/zenodo.13348695>

32: **Naresh Lokiny, & Pradip Reddy. (2021).** Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. <https://doi.org/10.5281/zenodo.13325845>

33: **Jayanth Para, 6G Internet Technology** Cyber Threat Notification & Alert System, Vol. 52 No. 4 (2024): October-December 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/170> , DOI: <https://doi.org/10.46121/pspc.52.4.9>

34: **Jayanth Para, AI Based Cloud Computation Observational Method & Process**, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/171> , DOI: <https://doi.org/10.46121/pspc.51.4.3>

35: **Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES**, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/294>

36: **Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS**, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/364>

37: **Kaleshwar Aryasomayajula, Micro services: “A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments.**, Vol. 51 No. 2 (2023): April-June 2023 , Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/374>

38: **Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS**, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/359>

39: **Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS**, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/362>

40: **Vikas Suresh Bagora, SECURE EMBEDDED NETWORKING ARCHITECTURES USING TRUSTED EXECUTION ENVIRONMENTS IN BROADBAND GATEWAYS**, Vol. 52 No. 2 (2024): April-June 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/363>

41: **Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices**, DOI:

10.48047/jocaaa.2025.34.05.48

<https://doi.org/10.63282/3050-9246.IJETCSIT-V1I4P104>, Vol. 1 No. 4 (2020),<https://ijetcsit.org/index.php/ijetcsit/article/view/108>

42: Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI:

<https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105>,<https://ijaibdcms.org/index.php/ijaibdcms/article/view/78>

43: Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA,

<https://eudoxuspress.com/index.php/pub/article/view/4118>