Research Trends International Journal of Technology and Innovation
An International Peer-Reviewed, Refereed & Open-Access Journal
ISSN : [ISSN]
Available Online at :
https://rtijti.com/
doi : https://doi.org/[DOI-PREFIX]/RTIJTI

Explainable AI Driven Phishing Detection in Email, SMS and URL with a Unified System

Om Jagtap, Nimish Dhawale, Vinayak Arote, Pankaj Chandre

Department of Computer Engineering, Pune, Maharashtra, India

ARTICLEINFO

Article History:

Accepted : 06 August 2026
Published : 08 August 2026

Publication Issue
Volume 01, Issue 03
July-September 2026

Page Number
32-38

ABSTRACT

Phishing posed a continuing risk where bad guys use false emails, messages or links to trick people. The authors unveiled a new tool named PhishGuard and claimed it is a hybrid multi-channel phishing detection system that combines machine learning, rule-based analysis, and explainable AI to improve both the phishing detection accuracy and user trust. The system uses handcrafted thirty lexical, domain, and content-based attributes for URL analysis, which are fed to an ensemble of Random Forest, Gaussian Naive Bayes, and residual multilayer perceptron model. The model outputs are then ensembled using a weighted decision fusion technique while being supported by real-time blacklist confirmation and structural risk rules. On the other hand, SMS and email phishing detection methods use text preprocessing steps such as tokenization, stop-word removal, stemming, and TF-IDF vectorization, before the texts are classified with Support Vector Machines. Performing phishing detection purely from URLs, the system has reached the accuracy of 96.28% while having a good balance with other evaluation metrics as well. As well, PhishGuard further has a two-level explanation component that returns both feature-level and natural-language explanations to assist interpretation. The discussed framework is a multi-channel phishing detection system that is not just scalable but also personal to the user.

Index Terms - Phishing Detection, Explainable AI, Weighted Decision Fusion, URL Feature Engineering, Support Vector Machine, Multi-Channel Security

I. INTRODUCTION

1.1 Background and Motivation

Phishing remains one of the most persistent entry points for compromise because it targets the user rather than the system. An attacker does not need a software vulnerability if a convincing message can persuade a recipient to enter credentials on a page that merely looks familiar. The delivery channel has also broadened. Where phishing was once dominated by email, campaigns now arrive as SMS messages containing shortened links, as chat messages, and as

bare URLs shared through social platforms, and each of these channels carries different signals and different amounts of text.

Detection tools have tended to specialise. URL scanners reason about the structure of a link, email filters reason about message text and headers, and SMS filters reason about very short text with heavy abbreviation. A user, however, does not experience these as separate problems; a single campaign may reach them by two channels at once. A system that covers only one channel leaves the others open, and running three unrelated tools produces three inconsistent verdicts with no shared notion of confidence.

1.2 Problem Statement

We set out to build a phishing detection system with three properties. First, it must cover URLs, email, and SMS within one system rather than requiring a separate tool for each, so that a verdict has the same meaning regardless of where the content arrived. Second, it must not depend on a single classifier or a single kind of evidence, because learned models and blacklists fail in different and largely uncorrelated ways: a model generalises to unseen domains but can be fooled by a well-constructed imitation, while a blacklist is exact but always lags a fresh campaign. Third, it must explain its verdict in terms a non-specialist can act on, because a phishing warning that a user does not understand is a warning the user will eventually dismiss.

1.3 Contributions

This paper makes four contributions. We present PhishGuard, a hybrid multi-channel phishing detection system that handles URL, email, and SMS input through one interface and one confidence scale. We describe a URL analysis path built on thirty handcrafted lexical, domain, and content-based attributes, classified by an ensemble of Random Forest, Gaussian Naive Bayes, and a residual multilayer perceptron, whose outputs are combined by weighted decision fusion. We describe how learned classification is supported by real-time blacklist verification and a set of structural risk rules, so that known-bad links are caught immediately and structurally suspicious links are escalated even when the ensemble is uncertain. Finally, we describe a two-level explanation component that returns both feature-level attributions and a natural-language summary, so that the same verdict can serve an analyst reviewing evidence and an end user deciding whether to click.

1.4 Organisation of the Paper

Section II reviews URL-based and text-based phishing detection, ensemble methods, and explainability in security classifiers. Section III sets out the system architecture. Section IV details the methodology for each channel, the fusion step, the rule and blacklist layers, and the explanation component. Section V covers the implementation. Section VI presents the evaluation. Section VII discusses the design trade-offs, Section VIII states the limitations, and Section IX concludes with future work.

II. RELATED WORK

2.1 URL-Based Phishing Detection

Feature-based URL classification is well established. Handcrafted attributes drawn from the URL string, such as length, the count of dots and hyphens, the presence of an IP address in place of a hostname, and the use of suspicious tokens in the path, have been shown to separate phishing from legitimate links with high accuracy across several classifier families [1]. Related work adds attributes derived from the registered domain, including registration age and record consistency, on the grounds that phishing infrastructure is typically short-lived [2]. Fast feature extraction directly from the URL, without retrieving the page, has been studied specifically because retrieval is often too slow for interactive use and exposes the checking system to the target [10].

2.2 Text-Based Phishing Detection in Email and SMS

Email phishing detection has historically combined header analysis with lexical features of the message body, and early work established that a modest set of such features supports accurate classification [9]. SMS filtering is a harder text problem in one specific respect: messages are short, heavily abbreviated, and often contain shortened links that carry no lexical signal at all. Public collections assembled for SMS spam research

remain the standard basis for evaluating this channel [3]. Across both channels, term frequency-inverse document frequency representations followed by a margin-based classifier remain a strong and computationally light baseline.

2.3 Ensemble and Hybrid Approaches

Ensembles are attractive in this domain because the constituent models make different mistakes. Random Forests handle heterogeneous tabular attributes and interactions between them without extensive tuning [4]. Gaussian Naive Bayes is fast and calibrates reasonably on well-separated continuous attributes, though its independence assumption is clearly violated by correlated URL features. Neural classifiers capture interactions the other two miss, and residual connections make deeper networks trainable without degradation [8]. Hybrid systems that pair a learned model with hyperlink or structural analysis have reported gains over either component alone [11].

2.4 Explainability in Security Classifiers

Model-agnostic explanation methods produce local attributions that indicate which input features drove an individual prediction [6], and unified attribution frameworks give these attributions a consistent theoretical basis [7]. In a security setting the audience is split. An analyst benefits from a ranked list of contributing features, whereas an end user receiving a warning on a phone benefits from a short sentence naming the concrete reason. Most reported phishing systems provide the first form of output, if any, and leave the translation to the reader.

2.5 Gap Addressed

The gap we address is the combination of three things that are usually found separately: multi-channel coverage under one verdict scale, hybrid evidence that pairs a learned ensemble with blacklist verification and structural rules, and an explanation layer that serves both the analyst and the end user. Prior systems generally cover one channel well, rely on one form of evidence, and stop at a probability score.

III. SYSTEM ARCHITECTURE

3.1 Overall Structure

PhishGuard is organised as a single entry point in front of three analysis paths. Submitted content is routed by type: a bare link enters the URL path, message text with headers enters the email path, and short message text enters the SMS path. All three paths terminate in a common decision layer that produces a label, a confidence value on a shared scale, and an explanation object, so that downstream consumers do not need to know which path produced the verdict.

3.2 Component Overview

The URL path contains a feature extractor producing thirty attributes, three classifiers operating on those attributes, and a fusion module that combines their outputs. Alongside it sit a blacklist verification client and a structural rule engine, both of which can raise the final risk assessment independently of the ensemble. The text paths contain a preprocessing pipeline and a Support Vector Machine classifier per channel, trained separately for email and SMS because the two differ in length and register. The explanation component sits after the decision layer and consumes whichever evidence the active path produced.

3.3 Data Flow

For a URL, the extractor computes the attribute vector without retrieving the page, the three classifiers score it in parallel, and the fusion module produces a weighted probability. The blacklist client is queried concurrently; a positive match short-circuits to a phishing verdict at maximum confidence. If there is no match, the rule engine checks structural conditions and can raise the risk band assigned by fusion. For email and SMS, the text is normalised, vectorised, and classified, and the resulting margin is mapped to the same confidence scale used by the URL path. In all cases the decision layer emits the verdict together with the evidence that produced it, which the explanation component then renders at two levels of detail.

IV. METHODOLOGY

4.1 URL Feature Engineering

Thirty attributes are extracted from each URL and fall into three groups. Lexical attributes describe the string itself: overall length, the length of the hostname and path, counts of dots, hyphens, digits, and special characters, the presence of an embedded IP address, the depth of the path, the number of subdomains, and the presence of tokens frequently used in credential-harvesting paths such as login, verify, secure, and account. Domain attributes describe the registered domain and its records, including the age of registration, the validity and issuer class of the certificate, and whether the hostname belongs to a URL shortening service. Content-based attributes describe structural properties observable without full page rendering, such as the presence of an at-symbol redirect, protocol mismatch, and unusual port specification. All attributes are computed from the URL and its associated records rather than from retrieved page content, which keeps extraction fast enough for interactive use.

4.2 URL Classification Ensemble

The attribute vector is scored by three classifiers chosen for their complementary failure modes. A Random Forest handles the mixed and interacting tabular attributes and is robust to attributes that are uninformative on their own. A Gaussian Naive Bayes classifier provides a fast, independently calibrated second opinion on the continuous attributes. A residual multilayer perceptron captures higher-order interactions that neither of the others represents, using residual connections between hidden blocks so that additional depth does not degrade training. Each classifier outputs a class probability rather than a hard label, which is what makes the fusion step possible.

4.3 Weighted Decision Fusion

The three probabilities are combined by weighted decision fusion rather than by majority vote. Each classifier carries a fixed weight set from its validation performance, and the fused probability is the weighted mean of the individual probabilities. This preserves the distinction between unanimous confidence and a narrow split, which a majority vote discards. A URL on which all three classifiers agree strongly therefore receives a decisively higher score than one where a single classifier carries the decision, and the difference is visible to the risk banding step that follows.

4.4 Blacklist Verification and Structural Risk Rules

Two non-learned components run alongside the ensemble. Real-time blacklist verification checks the hostname and full URL against reputation sources, and a positive match is treated as conclusive, since a confirmed report is stronger evidence than any model estimate. Structural risk rules encode conditions that are individually suspicious regardless of the learned decision boundary, such as an IP address used in place of a hostname, a brand name appearing in a subdomain rather than the registered domain, or an at-symbol placed so that the visible portion of the URL differs from its actual destination. A satisfied rule raises the assigned risk band. The two layers exist because they fail in opposite directions: the blacklist is exact but always lags a new campaign, while the rules generalise to unseen domains but are coarse.

4.5 Text Preprocessing for Email and SMS

Message text is normalised before classification. Text is lowercased and tokenised, punctuation and non-informative symbols are stripped, stop-words are removed, and the remaining tokens are reduced to their stems so that surface variants of the same term collapse together. URLs, currency amounts, and phone numbers are replaced with placeholder tokens rather than deleted, because their presence is informative even when their specific value is not. The processed tokens are converted to a term frequency-inverse document frequency representation, which weights terms that are distinctive to a message against those common across the corpus.

4.6 Classification for the Email and SMS Channels

Each text channel is classified by a Support Vector Machine, which suits sparse high-dimensional representations and remains effective when the number of features exceeds the number of training examples [5]. Separate models are trained for email and SMS rather than one shared model, because the two channels differ in length, register, and the degree of abbreviation; a decision boundary fitted to full email

bodies transfers poorly to messages of a dozen tokens. The distance from the decision boundary is mapped to the shared confidence scale so that a text verdict and a URL verdict can be compared directly.

4.7 Two-Level Explanation Component

Every verdict is accompanied by an explanation at two levels. The feature level reports the attributes or terms that contributed most to the decision, with the direction and relative magnitude of each contribution, which suits an analyst reviewing a queue of alerts. The natural-language level renders the same evidence as a short statement naming the concrete reason, for example that the visible brand name appears in a subdomain rather than in the registered domain, or that the registered domain is only days old. The second level is generated from the first rather than produced independently, so the two cannot disagree.

4.8 Evaluation Method

We evaluated the URL path on a labelled dataset of phishing and legitimate links, reporting accuracy together with precision, recall, and F1-score, since accuracy alone is misleading when the operational cost of a false positive differs from that of a false negative. We evaluated the email and SMS classifiers separately on labelled corpora for each channel. We also compared fused ensemble performance against each constituent classifier in isolation to establish whether fusion contributed beyond the best single model, and inspected the explanation output for agreement with the features that drove each decision.

V. IMPLEMENTATION

5.1 Technology Stack

The classifiers are implemented in Python. The Random Forest and Gaussian Naive Bayes models and the TF-IDF and Support Vector Machine pipelines use scikit-learn; the residual multilayer perceptron is implemented in a deep learning framework and exported for inference. Feature extraction, blacklist querying, and rule evaluation are implemented as separate modules behind a service interface, so that a channel can be updated without redeploying the others. Trained models and vectoriser vocabularies are serialised and loaded once at start-up rather than per request.

5.2 Inference Pipeline

A submission is dispatched to the appropriate path, and within the URL path feature extraction, the three classifiers, and the blacklist query run concurrently, so that overall latency is bounded by the slowest component rather than their sum. The blacklist query is given a timeout, and on expiry the system proceeds on ensemble and rule evidence alone rather than blocking, with the degraded evidence state recorded in the explanation. Results are cached by URL and by message hash for a short window so that repeated submissions during a single campaign do not repeat the full pipeline.

5.3 Deployment Considerations

Because no attribute requires retrieving the target page, the URL path can run without ever contacting the suspected host, which avoids both the latency of retrieval and the disclosure of the checking request to the attacker. Model updates are handled by versioning the serialised artefacts and the feature extractor together, since an attribute vector produced by one extractor version is not interchangeable with a model trained on another.

VI. RESULTS AND EVALUATION

6.1 URL Channel Performance

On the URL evaluation set the fused ensemble reached an accuracy of 96.28 percent, with precision, recall, and F1-score in close agreement, indicating that the classifier is not achieving its accuracy by trading one error type against the other. Errors concentrated in two groups: legitimate links on newly registered domains, which share the domain-age signal that phishing infrastructure exhibits, and phishing links hosted on compromised legitimate domains, where the domain attributes are genuinely benign and only the path carries signal.

6.2 Email and SMS Channel Performance

The email classifier performed better than the SMS classifier, which is consistent with the amount of text

available to each. Email messages provide enough tokens for the TF-IDF representation to be discriminative, whereas SMS messages are short and frequently reduce to a greeting, an urgency phrase, and a shortened link. For the shortest SMS messages the text signal is weak enough that the embedded link, routed through the URL path, carries most of the decision.

6.3 Effect of Decision Fusion

The fused ensemble outperformed each of its three constituent classifiers evaluated in isolation. The Random Forest was the strongest single model, the residual multilayer perceptron was close behind and disagreed with the forest on a distinct subset of cases, and Gaussian Naive Bayes was the weakest alone but contributed on examples where the continuous attributes were well separated. The gain from fusion was largest precisely on the cases where the individual classifiers disagreed, which is the behaviour weighted fusion is intended to produce.

6.4 Explanation Quality

The features surfaced by the explanation component corresponded to those that drove each decision, and the natural-language summaries named concrete and checkable conditions rather than restating the confidence value. The most frequently cited reasons across flagged URLs were brand names appearing in a subdomain, recently registered domains, and excessive path depth combined with credential-related tokens, which matches the attributes the ensemble weighted most heavily.

VII. DISCUSSION

The central design choice in PhishGuard is to treat learned classification, blacklist verification, and structural rules as three independent forms of evidence rather than folding everything into one model. Each fails in a different way. A blacklist is authoritative but only after a campaign has been observed and reported, so it is precise and permanently late. A learned ensemble generalises to links it has never seen but can be defeated by an attacker who constructs a URL that looks statistically ordinary. Structural rules catch specific deceptions such as brand names in subdomains regardless of what the model has learned, but they are coarse and would produce many false positives if used alone. Combining them means a failure in one layer does not become a failure of the system.

A second point concerns the explanation component. It is tempting to treat explainability as a reporting feature added after the classifier works, but in a phishing product it is closer to a functional requirement. Users routinely dismiss warnings they do not understand, and a warning that says only that a link is suspicious with high confidence gives no basis for a decision. A statement that the visible brand name appears in a subdomain rather than the registered domain is something a user can verify by looking at the link, which makes the warning actionable rather than merely authoritative.

Serving both channels through one confidence scale also has a practical consequence: a message and the link it contains can be assessed together, so a short SMS whose text is uninformative can still be flagged on the strength of its embedded URL. This is the case a single-channel filter handles worst.

VIII. LIMITATIONS

This work has four limitations we want to state plainly. The reported figures come from evaluation on labelled datasets rather than from a live deployment, and phishing distributions shift as campaigns change, so accuracy under sustained real-world use may differ from these figures and will require periodic retraining. Second, the system has not been evaluated against an adversary who has knowledge of the feature set; an attacker aware of which thirty attributes are extracted can construct URLs that score as benign on most of them, and adversarial robustness has not been tested. Third, the SMS channel is limited by the amount of text available, and for the shortest messages the text classifier contributes little beyond what the embedded URL already provides. Fourth, the explanation component reports the features that drove each decision, but this is not the same as establishing that

those features are the causally correct reason to distrust a link; feature attribution describes the model, not the world.

IX. CONCLUSION AND FUTURE WORK

9.1 Conclusion

We presented PhishGuard, a hybrid multi-channel phishing detection system covering URL, email, and SMS content under a single verdict and confidence scale. The URL path extracts thirty lexical, domain, and content-based attributes and classifies them with an ensemble of Random Forest, Gaussian Naive Bayes, and a residual multilayer perceptron combined by weighted decision fusion, supported by real-time blacklist verification and structural risk rules. The email and SMS paths apply tokenisation, stop-word removal, stemming, and TF-IDF vectorisation before Support Vector Machine classification. The URL path reached 96.28 percent accuracy with balanced precision, recall, and F1-score, and fusion outperformed every constituent classifier taken alone. A two-level explanation component returns both feature-level attributions and natural-language reasons, so the same verdict serves an analyst and an end user.

9.2 Future Work

Future work includes adversarial evaluation against attackers with knowledge of the feature set, incremental retraining so that the ensemble tracks campaign drift without full retraining, extension of the text paths to messaging and social platforms beyond email and SMS, a user study measuring whether the natural-language explanations actually change click behaviour rather than merely being available, and multilingual support for the text channels, which are at present trained on English corpora.

REFERENCES

[1] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," Expert Systems with Applications, vol. 117, pp. 345-357, 2019.

[2] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Computing and Applications, vol. 25, no. 2, pp. 443-458, 2014.

[3] T. A. Almeida, J. M. Gomez Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in Proc. 11th ACM Symp. Document Engineering, 2011, pp. 259-262.

[4] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[5] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 1135-1144.

[7] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 4765-4774.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[9] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in Proc. 16th Int. Conf. World Wide Web (WWW), 2007, pp. 649-656.

[10] R. Verma and A. Das, "What's in a URL: Fast Feature Extraction and Malicious URL Detection," in Proc. 3rd ACM Int. Workshop on Security and Privacy Analytics (IWSPA), 2017, pp. 55-63.

[11] A. K. Jain and B. 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.