Hybrid Explainable Phishing URL Detection Using Transformer-Based Embeddings

Pragati Priyadarshinee[1]*, Aravind Chandra S.[2], Varun Reddy P.[3], and Jyothika Thunam[4]

[1,2,3,4]Department of Information Technology, Chaitanya Bharathi Institute of Technology (A), Hyderabad, India

*priyadarshineepragati10@gmail.com, sangammagariaravind123@gmail.com, varunreddy1127@gmail.com, thunamjyothika@gmail.com

Abstract. Phishing has always been a prevalent cybersecurity threat, using human trust and vulnerabilities on the internet to acquire sensitive information. Standard machine learning and deep learning models have improved the accuracy of phishing URL detection. However, they continue to strive to adjust to the growing severe attack patterns and integration with real world security systems and lack explainability. This paper introduces a hybrid framework for detecting phishing URLs that blends transformer based semantic comprehension with rule-based cybersecurity intelligence to improve robustness and Legibility. Our methodology improves the BERT Phish Finder model by applying MiniLM embeddings for optimized semantic representation, along with lexical, structural, and heuristic URL characteristics. A Random Forest classifier, combined with a bespoke Trust Index, rule-engine and Deep Learning Model delivers multi-dimensional scoring to categorize URLs as Safe, Suspicious, or Phishing. Additionally, by visualizing the model's decision factors, Explainable AI (XAI) with Sharley Additive exPlanations (SHAP) improves transparency. Real-time detection capabilities and interpretable outputs are demonstrated by the initial implementation using streamlet. In order to lay the groundwork for cross-domain integration across network monitoring, database systems, and big data security analytics, this research attempts to reduce the gap between pure AI models and useful cybersecurity applications.

Keywords: Phishing URL Detection, Threat Detection Systems, Semantic-Structural Fusion, Hybrid Machine Learning Model, Semantic Feature Representation, Cybersecurity Rule Engine, Trust Index Evaluation, XAI, SHAP.

1 Introduction

Phishing attacks, that utilize manipulated websites, emails, and URLs to exploit users, have become one of the most deceptive and quickly expanding cybercrimes [1], [2].

© The Author(s) 2026

S. Bhalerao et al. (eds.), *Proceedings of the 2nd International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2026)*, Advances in Intelligent Systems Research 207,
https://doi.org/10.2991/978-94-6239-678-4_26

Attackers impersonate trustworthy organizations to obtain private information, including login and banking credentials, and personal data. The dynamic nature of phishing techniques continues to surpass traditional detection techniques, even with cybersecurity advancements.

The cornerstones of early detection models were heuristic based techniques and blacklists, which are neither very scalable nor flexible [1]. Later, methods for machine learning (ML) and deep learning (DL) emerged, utilizing attributes such as URL length, domain age, and lexical patterns [2], [6]. However, these models usually do not understand URL contexts semantically and do not explain their decisions, which damages their credibility and prevents them from being widely used in enterprise security systems.

There are now more options for extracting contextual features from URLs thanks to recent developments in transformer-based architectures like BERT and MiniLM and Natural Language Processing (NLP) [3], [4], [14], [15]. But these models by themselves are difficult to interpret and computationally demanding.

A hybrid phishing URL detection system is recommended by this study that addresses these problems by combining the following: semantic embeddings derived from transformer models (MiniLM), feature-based heuristics extracted from URL structure, a cybersecurity rule engine to enhance interpretability, and mechanisms of Explainable AI (XAI) to display the model's reasoning process. This integration's goals are to reduce false positives, enhance the precision of phishing detection, and provide a framework that is interpretable and applicable to different fields such as database threat logging, big data security analytics, and network traffic filtering.

2 Related Work

2.1 URL Feature Based and Heuristic Systems

A significant portion of early phishing detection studies still use manually determined features - these are lexical, host and content-based features [1], [2], [6]. A lot of studies pay attention to the length of the URL, token amounts, WHOIS presence, whether there's an IP in play, special characters, domain age. All of these are compounded into a comprehensive model through logistic regression, SVMs and random forests. These work well because they're easily understandable for humans, too. In addition, these are operable by limited systems meaning they can be implemented quickly. Yet they also assume too much from what can be expected from predictable patterns of certain features. Thus, they are susceptible to character homoglyphs, URL shortening, dynamic domains. In addition, they're ineffective at combatting new strategies unless features are constantly tweaked.

2.2 Deep Representation and Transformer Based Methods

Where URL/page content generation is concerned, transformer methods are a game changer as they accommodate meaning and subtleties on a sub token level. Fine-tuned encoders understand URLs and web text components in relative context and within other URL strings/text better than other methods, outperforming on standardized, test collections [3], [4], [5]. Even distilled or tiny versions maintain a good precision recall trade off which is important for real-time validation. Transformers will achieve yields that were never before possible, however, they are still often used as black boxes. Their returns do not afford any investigators any explanation nor do they integrate any implied heuristics that come with real-life, human-validation endeavors.

2.3 Ensemble, Stacking, and Hybrid Architectures

Some solutions try to engineer some features and learn others to get the best of both worlds [5]–[7]. Therefore, hybrid or ensemble arrangements are used [5]– [7]. These are usually linked semantic embeddings of transformers and lexical and host features feeding tree ensembles or stacked arrangements. These have the highest accuracy. They also perform better through cross dataset tests. This is because they possess signals that validate one another. However, most arrangements do not include explicit cyber security semantics. They have low interpretable support for identifying malicious URLs. Moreover, the ensembles complicate deployment and maintenance even more. Rules must be adjusted after the fact and retraining is even worse.

2.4 Explainability, Operationalization, and Research Gaps

Papers on surveys and methodology show these continuous concerns which prevent transition from prototypes to operational detectors [9]–[10]. First, explainability is insufficient. Few detected models extend feature-based explanations which are SOC process compliant [11], [12]. Second, Analyst accommodation is insufficient. Models seldom include or show heuristics processes (they might note extensions or dubious TLDs and keyword analyses). But these are everyday employed by cybersecurity personnel. Third, operational accommodation is insufficient. For any operational setup to work, real time inference, situation-based rule adjustment and interpretation and decision-making assessment pathways must be crucial and competent [9], [10]. These concerns delay operational practicality. With statutory governance, compliance regulations, and ethical considerations, there must be systems implemented which not only boast transformer level accuracies but also, in particular, rule-based assessments with trust scores for predicted results. Our proposed approach fulfils all these needs, as it combines semantic embeddings, transformer advantages and an already trained, host-

based set of features. It supports a cybersecurity, rule-based engine and an intelligible Trust Index with explainable components. Thus, it retains the detection strengths of such trained, advanced systems while also incorporating transparency, transparency of user intention, and easily documentable audit trails to ensure it can operate within daily, operational needs. These types of mistakes mean nothing to a regulator. It only makes it worse.

3 Proposed Methodology

The overall explainable and transparent mixed method framework for phishing URL detection is based on such embedded transformative integrations of semantics [3], [14], lexical feature generation, a cybersecurity-informing rule set and human interpretable conclusions.

3.1 System Architecture Overview

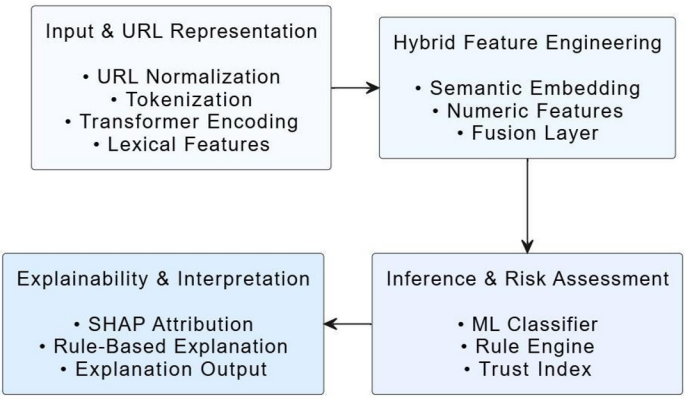

Fig.1. System Architecture

The proposed phishing URL detection system follows a layered hybrid architecture that integrates semantic understanding, structural feature analysis, rule-based reasoning, and explainable decision-making. The architecture (Fig. 1) is designed to balance detection accuracy, interpretability, and deployment feasibility in real-world cybersecurity environments.

The pipeline begins with URL input acquisition and preprocessing, where raw URLs are normalized and parsed. In parallel, two complementary representations are generated. First, the textual URL string is processed using a lightweight transformer encoder (MiniLM) to extract semantic embeddings that capture contextual and sub-

token patterns indicative of phishing behavior. Second, a set of lexical and structural features—such as URL length, special character frequency, digit-to-character ratio, presence of suspicious tokens, and structural anomalies—are extracted to provide explicit, human-interpretable signals.

These semantic and numerical features are then combined through a late fusion mechanism to form a unified hybrid feature vector. This fused representation is passed to a machine learning classifier (Random Forest), which estimates the probability of the URL being malicious based on learned patterns. In parallel, a cybersecurity rule engine evaluates the URL using domain-specific heuristics, including suspicious top-level domains, excessive subdomains, URL shorteners, and known phishing indicators.

The outputs of the classifier and rule engine are aggregated using a Trust Index, which produces a graded risk score and categorizes URLs as Safe, Suspicious, or Phishing. To ensure transparency, an explainability module based on SHarley Additive exPlanations (SHAP) highlights the most influential semantic, structural, and rule-based factors contributing to each decision. This layered architecture enables accurate phishing detection while providing interpretable, audit-friendly outputs suitable for operational security systems.

3.2 Dataset Representation and Preprocessing

The standard phishing URL dataset contains word, host and content attributes. To allow hybrid modelling to work without stress, the hybrid framework I intend to create will pre-process universally into a semantic and numeric hybrid for exploration. This means URL normalization (string size), removal of N/A, then, a value mapping to simple binaries of phishing versus legit occurrence. Thereafter, we can extract features from word and host-based features. This includes URL length, number of dots and hyphens, number of digits vs. characters, presence of an '@' symbol and presence of phishing specific keywords. This will create a numeric vector. Continuous features will be standardized to prevent scale differences between attributes. The goal is a non-discerning integration with downstream transformer-based embeddings.

Next, we extract indicators from words and structure. These cover length of the URL, how often dots and hyphens show up, ratio of digits to characters, if there is an "@" symbol, and these are keywords linked to phishing. These make a vector of numbers. Continuous parts get standardized, so scales do not mess things up across features. What comes out gives a solid base to mix with transformer embeddings later on.

3.3 Semantic Embeddings and Hybrid Feature Fusion

To get the context and small token patterns in phishing URLs, the framework will use an encoder based on transformers like MiniLM [14], to make semantic embeddings. For any URL input u, the encoder will give a dense representation:

$$e_{url} \in R^{d_e}, \tag{1}$$

where d_e denotes the embedding dimensionality.

In parallel, the lexical and structural indicators form a numerical feature vector:

$$x_{num} \in R^{d_n} \tag{2}$$

A late fusion technique is used to make a unified hybrid feature representation:

$$Ⅎb\{X\}\_\{hybrid\} = [e_{url} \| x_{num}] \in R^{d_e+d_n} \tag{3}$$

This hybridization will preserve the contextual depth of transformer embeddings while inserting explicit structural cues that are traceable and well aligned with cybersecurity reasoning.

3.4 Hybrid Inference: Classifier, Rule Engine, and Trust Index

A classifier based on learning works on the hybrid feature space X_{hybrid} to estimate the chance a URL is a phishing URL. We suppose a model like Random Forest [13] from tree ensembles. It is challenging, yet simple to run, and easy to interpret on its own. The classifier gives a probability estimate:

$$P_{ml} = Pr\ Pr\left(X_{hybrid}\right). \tag{4}$$

To add insight from cybersecurity to the stats learning, a rule engine will check heuristic indicators that stand out. These include odd top-level domains, shorteners for URLs, too many subdomains, and traces of tokens for phishing. The engine figures a score from rules that is normalized:

$$RuleScore = \frac{Number\ of\ triggered\ rules}{Total\ rules} \tag{5}$$

The last decision will come from a Trust Index. It combines the probability from stats with certainty from heuristics:

$$TrustIndex = w_{ml}(1 - P_{ml}) + w_{rule}(1 - RuleScore), \tag{6}$$

where w_{ml} and w_{rule} indicate weighting factors such that $w_{ml} + w_{rule} = 1$. The Trust Index evaluates safety of the URL on a continuous scale. We map it to groups for triage, like Safe, Suspicious, Phishing, using set thresholds.

3.5 Explainable Reasoning and Interpretable Output

The explainability component is explainable. There exists a reason for each decision made. Embeddings from a transformer are not explainable. Therefore, with the

proposed model, the explainability component of SHAP [11] for model feature attribution or a similar fallback - depending on the model type - is applied to the classifier to indicate the most relevant components within the hybrid and rules that were applied. This is an explainable output justified by what an analyst would assume. The explainability component output is a formal report. It reduces a decision to a subset of drivers based upon semantics and formal structure. It derives factors from rules determining increased risk. It factors in the Matched Trust Index. Thus, it provides a diagnosis that's easy to understand, making it easy to audit each URL. This helps support automated filtering to take place autonomously and operations requiring cybersecurity to take place where human intervention is needed.

4 Comparative Analysis

This section presents a comparison of the proposed hybrid explainable Phishing URL Detection Framework with established, major categories of phishing detection literature. These major, foundational categories exist relative to lexical machine learning models [1], [2], [6], transformer-based semantics [3], [4], hybrid approaches [5], [7] and ensemble methods [6]– [8] NOT relative to individual works. Therefore, Table 1 a relative assessment of these categories and a standard set of review criteria. Thereafter, each major category is assessed in less than one paragraph relative to pros and cons of such approaches that necessitate the development of a hybrid explainable model. Survey works acknowledge shortcomings of phishing detection systems [9], [10].

4.1 Comparative Criteria

Approach type is the type of modelling that will dominate the approaches implemented in these systems. That is, lexical ML, semantic transformers, hybrid approaches and ensemble approaches. Advantages assess what MAJOR advantages arise from each type of approach. This will include accuracy, processing speeds and heightened semantic understanding. Gaps assess what MAJOR weaknesses prevent feasible application in the real world. This impacts practical application, general robustness and transparent understanding.

The comparative analysis notes clear distinctions between conflicting modelling paradigms currently and the various means by which these methods tackle the phishing detection problem. For example, lexical based machine learning paradigms are fast and interpretable yet fall victim to adversarial transformations that twist a URL's appearance without transforming its actual meaning [1], [2], [6]. Pure transformer-based systems avoid such a drawback since they rely upon meaning and contextual emphasis [3], [4], however, they themselves are black-box systems with higher

computational requirements that make them difficult to deploy in security minded and resource-limited environments.

Table 1. Comparative Analysis of Phishing URL Detection Approaches

Method	Advantages	Gaps
Lexical–ML Models	Simple, fast, interpretable; effective for handcrafted URL features (length, tokens, symbols). [1], [2], [6]	Feature-dependent and brittle against obfuscation; fails on semantic variations.
Transformer Models	Capture deep semantic/sub-token patterns; achieve state-of-the-art accuracy. [3], [4]	Opaque (black-box); high computational cost; no heuristic or rule-based reasoning.
Hybrid (Transformer + ML)	Combines semantic richness with structured URL features, improved robustness. [5], [7]	Lacks integrated rule engine; explainability remains limited; higher complexity.
Ensemble / Stacking Models	Better generalization; leverage strengths of multiple learners (CatBoost, XGBoost, etc.). [6]–[8]	Complex to tune and deploy; no domain specific heuristics; limited interpretability.
Proposed Hybrid Explainable Model	Unifies transformer embeddings, engineered features, rule engine, and Trust Index; provides interpretable and graded outputs.	Requires empirical validation on multisource datasets; tuning Trust Index weights is non-trivial.

Thus, hybrid systems seek to create a middle ground, yet many merely amalgamate features without domain-relevant thresholds or explainable aspects for human investigators [5], [7]. Meanwhile, system ensembles and stacking systems complicate design yet fail to actually implement cybersecurity domain knowledge for explainable recommendations [6]–[8]. Broader assessments of previous works also show operational gaps, including limited alignment with real SOC workflows and challenges in achieving deployment-ready behavior [9], [10].

5 Future Work

The way phishing campaigns keep changing, with clever ways to hide information and automate attacks on a large scale, highlight key areas for further research. This could strengthen the hybrid explainable detection framework. The following sections outline the main directions for future work.

5.1 Advances in Transformer Architectures and Robust Representation Learning

Transformer based solutions for small segments of noisy text have begun populating the URL detection space [3], [4]. However, a greater breadth of character level or even byte level models exist that could implement smaller theft detection (e.g. look alike letters, fraudulent Unicode or inversed characteristics of URL components) than this study was able to accommodate. URL generation/synthesis and adversarial training could be extended out to implement robustness against malicious alterations. Furthermore, contrastive pretraining on web domain datasets of immense size could produce appropriate embeddings to support stronger differentiation between legitimate URL features versus malicious [14], [15].

5.2 Explainability Enhancement and Dynamic Rule Adaptation

SHAP and rule-based explanations are enough for this model but transparency of semantic vs. structural features and interaction could be more thoroughly explored in future work. Probabilistic rules or rules based on subject matter experts interacting or embedded threat intelligence would allow for interaction over time. Calibrated explanations could be more focus-driven to make the Trust Index more interpretable for analysts to see if and how every input contributed to or desired final decision [11], [12].

5.3 Operational Scalability and Learning in Real-World Environments

Phishing detection as a protective measure should be implemented in actual operational security systems; it's not enough to have accurate prediction but also near-real-time inference capabilities and continued learning to adopt and proliferate across systems. Future work could entail incremental learning, consistent updates, federated options that learn without centralizing the knowledge across operators/departments. Beyond operational, the potential to expansively include other phishing environments - email phishing, QR phishing, chat phishing would give an overall risk score instead of compartmentalized efforts for weaker social engineering efforts [16], [17].

6 Conclusion

Phishing represents one of the most omnipresent, problematic, and advancing attack techniques against cybersecurity. Such advancements are fueled by increasingly advanced URL obfuscation and malicious jokes. Yet, surprisingly, while traditional, older, off-the-shelf machine learning applications yield impressive findings concerning word-based observation, they're ill-prepared to handle such new developments.

Conversely, transformer-based applications with meaning-based approaches achieve high levels of accuracy but relatively lower levels of transparency and greater day-to-day application challenges. Ultimately, an ideal approach in the future would be an ideal approach with an advanced understanding of meaning and provision of transparent articulation to practitioners and standardized reasoning approaches [11], [12].

Thus, this paper presents a hybrid approach to phishing URL detection based on meaning aware embeddings from transformers, manual word and structure-based features, flexible rules and a tangible Trust Index that evaluates proportions of doubt. The approach is also hierarchical, showing over the course of the manuscript how meaning making embeddings can be combined with straightforward security rules for a more extensive and transparent evaluation of URL threats. Moreover, since both likelihood probabilities and rule-based support are given, the more extensive approach can provide better trusted justifications for such detections, especially for tools that require explainability and rigorously accountable assessments. In addition to enhanced explainability, the hybrid approach fosters the ease of swapping in new parts with new attack patterns and the ease of expanding with additional data whether DNS data or nodes of threat in graphs. This especially benefits future work on withstand deception, cross media phishing detection and rule adjustments to accommodate live threats. As phishing continues to transform in relation to applications and means of communication, approaches like ours that combine yet maintain explainable, separated operations seem a valid approach to more secure but practical security solutions.

References

1. A. Basit, M. Zafar, A. Javed, and M. A. Khan: A comprehensive survey on phishing attacks and countermeasures. *IEEE Access*, 9:104655–104682 (2021)
2. A. Jain and B. B. Gupta: Towards detection of phishing websites on client-side using machine learning based approach. *Telematics and Informatics*, 64:101676 (2021)
3. S. Sivaguru, R. Raj, and M. Saiful: BERT-PhishFinder: A robust model for accurate phishing URL detection with optimized DistilBERT. In: *Proc. IEEE*, (2024)
4. J. Lee and J. Kim: Phishing URL detection via transformer encoder with subword tokenization. *IEEE Access*, 11:122300–122312, (2023)
5. R. Thomas and S. Mehta: Advanced learning for phishing URLs detection to secure consumer-centric applications. In: *Proc. IEEE*, (2024)
6. H. Song, M. Lee, and S. Park: A hybrid machine learning approach for phishing URL detection. *Applied Sciences*, 12(5):2551, (2022)
7. L. Liu, J. He, and F. Chen: A method for detecting phishing websites based on Tiny-BERT stacking. In: *Proc. IEEE*, (2023)
8. T. Wang, B. Liu, and X. Zhang: Lightweight transformer models for URL classification. In: *Proc. IEEE ICMLA* (2023)

9. R. Gupta and A. Kuppusamy: Phishing webpage detection: Unveiling the threat landscape and investigating detection techniques. In: *Proc. IEEE*(2024)
10. A. Alqarni, M. Meulen, and P. Watters: A systematic literature review of phishing attacks and defenses. *Computers & Security*, 125:103046 (2023)
11. S. M. Lundberg and S.-I. Lee: A unified approach to interpreting model predictions. In: *Proc. NIPS*, pp. 4765–4774 (2017)
12. A. Guidotti, R. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi: A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93, 2019.
13. L. Breiman: Random forests. *Machine Learning*, 45(1):5–32 (2001)
14. W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou: MiniLM: Deep self-attention distillation for transformer models. In: *Proc. NeurIPS* (2020)
15. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin: Attention is all you need. In: *Proc. NIPS* (2017)
16. S. Bahnsen, D. Torres, C. G. L. Rojas, and E. C. Caicedo: Adversarial attacks on machine learning based phishing URL detectors. In: *Proc. IEEE ICDM* (2020)
17. J. Tian and K. Chang: Robust phishing detection under evasion attacks. *IEEE Trans. Inf. Forensics Security*, 18:5112–5127 (2023)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.