IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

JSPM's, SBERCT's BHAGWANT INSTITUTE OF TECHNOLOGY, BARSHI COMPUTER SCIENCE & ENGINEERING DEPARTMENT "Phishing URL Detection Web Tool"

Guide Prof. Mali A.K

Shashank Kate BIT, Barshi

Anushka Jadhav BIT, Barshi

Janhavi Pandit BIT, Barshi

Sarika Gadade BIT, Barshi

Abstract:-

Phishing attacks persist as a critical and financially destructive threat in the digital domain, exploiting user weaknesses to acquire sensitive data such as bank credentials and login details. 1 The annual worldwide financial impact of these attacks has been estimated to be as high as US$5 billion, demonstrating the urgent need for robust detection mechanisms. 1 Traditional detection methodologies, particularly blacklisting, are inherently incapable of addressing zero-hour threats due to the attackers' frequent use of fast-flux networks and algorithmic URL generation. 1 This manuscript introduces the TrustLayer system, a novel hybrid architecture that seamlessly integrates a comprehensive, multi-tiered heuristic feature model with optimized ensemble Machine Learning (ML) classification. The synthesized heuristic model incorporates 16 distinct features, focusing on sophisticated structural evasion tactics (e.g., Punycode homograph attacks, excessive dot counts, and URL shortening services). 1 By leveraging empirically proven ensemble methods, notably the Random Forest algorithm, which achieved a detection accuracy of 97.14% and a low False Negative Rate (FNR) of 3.14% in prior studies 1 , TrustLayer is engineered to provide superior detection fidelity and resilience. The proposed system employs a staged classification pipeline optimized for low latency, ensuring real-time capability essential for mitigating modern cybersecurity threats.

Keywords-

Phishing detection, URL scanner, URL safety check, Malicious link detector, Suspicious URL detection, Phishing URL checker, Online link scanner, Website safety checker, Fake website detector, Cybersecurity tool

Introduction:-

The Pervasive and Economic Impact of Phishing Attacks

Phishing remains a prominent concern for security researchers due to the relative ease with which fake websites, meticulously crafted to resemble legitimate platforms, can be deployed. 1 While experts can often identify these deceptive sites, the general user population frequently becomes a victim, leading to significant corporate and personal losses. 1 The economic consequences are staggering, with annual losses exceeding US$2 billion for businesses in the United States alone. 1 Beyond financial losses, protecting users from phishing is vital for maintaining public trust and confidence in online services and platforms, directly addressing regulatory compliance requirements concerning data protection. 1 The increasing sophistication of these attacks, moving toward highly targeted social engineering, necessitates real-time, zero-hour detection capabilities that can identify threats immediately upon deployment.

Limitations of Traditional and Static Detection Systems

The general method for detecting malicious websites involves maintaining a database of known blacklisted URLs and Internet Protocols (IPs). 1 However, this reactive blacklist approach cannot detect "zero-hour" phishing attacks—those that have not yet been reported—because attackers actively use creative obfuscation techniques, such as fast-flux hosting and algorithmic URL modification, to evade these static databases. 1

The TrustLayer Hybrid Approach and Manuscript Contributions

To address the shortcomings of purely static or purely heuristic systems, the TrustLayer project proposes a hybrid architecture founded on advanced feature engineering and machine learning technology. 1 This approach analyzes various blacklisted and legitimate URLs, extracting predictive features to accurately detect phishing websites, including zero-hour instances. 1

The core contributions of this manuscript are three-fold:

1.Synthesis of a Multi-Tiered Heuristic Model: Formalizing a comprehensive set of 16 structural, deceptive, and dynamic features derived from contemporary research to maximize detection coverage. 1

2. Architectural Optimization: Defining a sequential two-stage classification pipeline that manages the trade-off between real-time processing latency and detection depth.

3. Ensemble Classification Strategy: Justifying and deploying a supervised ensemble ML model (including Random Forest, ExtraTree Classifier, and Support Vector Machine) specifically tuned to minimize False Negative Rate (FNR) while maintaining a minimal operational False Positive Rate (FPR).

Related Work and Foundational Algorithms

Review of Machine Learning in Phishing URL Detection

The reliance on machine learning has become a linchpin in modern phishing URL detection, allowing algorithms to automatically discern patterns and traits associated with malicious Uniform Resource Locators (URLs). 1 This automation overcomes the limitations of manual or static database methods. Feature extraction is fundamental to this process, involving the analytical breakdown of URL components such as domain names, sub-domains, path segments, and query parameters. 1

Research efforts typically utilize large datasets composed of benign URLs, often sourced from reputable indexes like www.alexa.com , and malicious URLs collected from repositories such as www.phishtank.com and PhishStorm. 1 For example, one foundational dataset contained 36,711 URLs, split between benign and phishing examples. 1 Feature engineering applied to these datasets heightens detection accuracy by highlighting anomalies and suspicious markers in the URL structure. 1

Empirical Comparison of Classification Algorithms -

Multiple studies have investigated the efficacy of various machine learning algorithms, with ensemble methods consistently demonstrating superior performance in classifying phishing URLs. Classification models such as Decision Tree, Random Forest (RF), and Support Vector Machine (SVM) were benchmarked against a 36,711 URL dataset across various training-testing data split ratios (50:50, 70:30, and 90:10). 1

Random Forest and Accuracy Optimization

Empirical evidence strongly favors the Random Forest algorithm. At the optimal 90:10 training-testing split ratio, RF achieved the highest detection accuracy of 97.14%, marginally outperforming the Decision Tree algorithm (97.11%). 1 Furthermore, RF demonstrated the lowest FNR (3.14%) in this configuration. This observed behavior—where detection accuracy increases as more data is used for training—supports the strategy of maximizing the training data volume for the core classifier. 1

The Importance of Scale: ExtraTree Classifier.

While Random Forest excels on well-defined feature sets and moderately sized datasets, the complexity introduced by massive, real-world data streams presents a unique challenge. In experiments involving a "huge" dataset of approximately 620,000 unique URLs, the ExtraTree Classifier (ET) proved to be the most effective algorithm, achieving an accuracy of 80.67%, which was superior to both AdaBoost Classifier (78.51%) and Logistic Regression (68.61%).

The discrepancy between the high accuracy reported by RF (97.14%) on smaller, potentially cleaner datasets and the relatively lower accuracy of ET (80.67%) on significantly larger datasets underscores a critical architectural consideration: single models often become fragile when exposed to high-variance or noisy real-time data, necessitating models capable of handling high volume and complexity. The optimal system design must leverage RF's precision for highly correlated features and ET's inherent robustness in high-volume environments.

False Positive Rate Management with SVM

The operational risk of a security system is defined not just by missed attacks (FNR) but also by unnecessary blockages (FPR). The cost of a False Negative is typically catastrophic in cybersecurity; however, high FPR renders a system unusable. Although Random Forest achieved the highest overall accuracy, the Support Vector Machine (SVM) consistently registered the lowest False Positive Rate across all test splits, reaching as low as 2.08% at the 50:50 split and 2.34% at the 90:10 split. 1 This characteristic suggests that SVM is highly effective in establishing a robust boundary defining the legitimate (benign) class, making it valuable as a specialized filter within an overall ensemble system to manage operational overhead.

The following table summarizes the key comparative performance metrics of the established classification algorithms evaluated:

Table 1: Comparative Performance Summary of Established URL Detection Classifiers

Classifier	Max Accuracy (%)	Min FNR (%)	Dataset Context	Citation Source
Random Forest (RF)	97.14	3.14	Dataset Context Citation Source Small/Medium Dataset (~36k URLs, 90:10 split)	1
Decision Tree	97.11	3.18	Small/Medium Dataset (~36k URLs, 90:10 split)	1
Support Vector Machine (SVM)	96.51	4.73	Small/Medium Dataset (~36k URLs, 90:10 split)	1
ExtraTree Classifier (ET)	80.67	N/A	Large Dataset (~620k URLs, 80:20 split)	1

The TrustLayer Hybrid System Architecture

System Modality: Optimized Two-Stage Detection

The TrustLayer system is designed specifically to overcome the inherent latency associated with dynamic feature extraction, such as those requiring external API lookups (e.g., WHOIS or SSL certificate checks). A purely dynamic system would be too slow for real-time traffic inspection. Therefore, TrustLayer implements a rigorous, sequential two-stage screening process based on the feature's acquisition time:

1. **Stage 1 (Static Analysis):** This stage relies on instantaneous parsing of Level 1 and Level 2 heuristic features that can be extracted directly from the URL string without external communication. This process provides rapid filtration. URLs identified as definitively legitimate or definitively phishing proceed directly to their final classification. URLs displaying moderate suspicious markers or ambiguity are flagged for escalation to Stage 2.

2. **Stage 2 (Dynamic & ML Analysis):** Only URLs flagged as Suspicious enter this stage. This justifies the higher latency required for dynamic lookups (Level 3 features: SSL age, WHOIS data) and subsequent, more computationally intensive complex ensemble ML processing. This staged approach ensures that the vast majority of web traffic, particularly legitimate traffic, is processed at low latency, while maintaining the necessary depth for detecting zero-hour threats.

Feature Acquisition Pipeline

The detection methodology commences with a dedicated feature extraction pipeline, typically implemented using programming languages such as Python. 1 This pipeline is responsible for parsing the raw URL string and translating its characteristics into structured input vectors for the machine learning model. Key components extracted include the overall URL length, detailed domain and subdomain analysis, the presence of special characters or suspicious keywords, URL structure (e.g., number of subdirectories), and indicators of redirection or shortening. 1 This structured data forms the input to the heuristic model and the final ML classifier.

Classification Outcomes and Decision Tiers

The TrustLayer system utilizes a three-tier output classification to manage uncertainty and inform the two-stage process:

- **Legitimate (0):** Confirmed safe or falls below all suspicious thresholds.

- **Suspicious (0.5):** Displays features that warrant further dynamic inspection but are not definitively malicious. This outcome triggers Stage 2 processing. Suspicion is raised if the URL length falls into the intermediate range (54 to $\le$ 75 characters) 1 or if HTTPS is present but the certificate issuer is not trusted. 1

- **Phishing (1):** Definitively flagged by high-confidence Level 1/2 heuristics or by the final ensemble ML model in Stage 2.

TrustLayer Heuristic Model Enhancement and Feature Synthesis

The TrustLayer detection system's resilience is built upon a synthesized heuristic layer that incorporates 16 distinct features, categorized by their processing latency and complexity, ensuring robust identification of contemporary phishing evasion techniques.

Level 1 Heuristics: Structural Anomalies (Instantaneous Parsing)

These features represent the most basic and rapid checks, often providing high-confidence f lags for rudimentary or heavily obfuscated attacks:

● **IP Address Usage:** Phishers often use the raw IP address (e.g., http://125.98.3.123/fake.html) or its hexadecimal representation (http://0x58.0xCC.0xCA.0x62/...) instead of a domain name to host the phishing site. 1 Since most benign sites use descriptive domain names, the presence of an IP address in the domain part of the URL immediately raises a high-confidence Phishing flag. 1

● **Special Character (@) Evasion:** The inclusion of the '@' symbol in a URL manipulates the browser, causing it to ignore all information preceding the symbol, and the actual address often follows the '@'. 1 This mechanism is a known obfuscation tactic, resulting in an Immediate Phishing flag .

● **Hostname Dot Count Threshold:** Phishing URLs frequently employ complex subdomain structures to impersonate legitimate entities (e.g., http://shop.fun.amazon.phishing.com). The average number of dots in benign URLs is empirically found to be three. 1 If the hostname contains more than three dots, the feature is flagged as malicious.

● **Prefix or Suffix Dash Separation:** The dash symbol (-) is rarely used in legitimate domain names but is a common tactic by phishers to separate brand keywords, such as creating http://www.online-amazon.com to confuse users accustomed to http://www.onlineamazon.com. 1 This structural anomaly warrants a strong risk classification.

● **URL Redirection:** The existence of the string // within the URL path, distinct from the initial protocol definition, is indicative of forced redirection to another website. 1

Level 2 Heuristics: Deceptive Encoding and Path Analysis

These features detect sophisticated methods used to mislead users or hide malicious destinations:

● **URL Shortening Service Detection:** Services like TinyURL or bit.ly are utilized by phishers to hide long, suspicious URLs behind a shorter, innocuous-looking address. 1 The detection of these known shortening service domains triggers a high-confidence Phishing flag. 1

● **Deceptive HTTPS Token:** Attackers sometimes embed the "HTTPS" token directly into the domain part of an HTTP URL (e.g., http://https-www-paypal...) to deceptively create an appearance of security. 1 This feature is classified as an attempt to spoof security protocols.

● **Punycode Homograph Attacks:** Phishers exploit Unicode characters, which are rendered using the xn-- prefix (Punycode), to visually mimic legitimate domains (e.g., making a user see "apple.com" while redirecting to "xn--80ak6aa92e.com," a phishing site). 1 This is a definitive indicator of intentional visual deception.

● **URL Length Tiered Analysis:** Excessive URL length is a recognized characteristic of phishing sites designed to push the deceptive components out of the visible address bar. 1 URL lengths exceeding 75 characters are flagged as Phishing , while lengths between 54 and 75 characters are classified as Suspicious , triggering dynamic verification. 1

● **Sensitive Keywords in Path:** Phishing sites frequently use sensitive terms like 'confirm', 'account', 'banking', 'secure', 'paypal', or 'password' within their URL path to instill a false sense of legitimacy and urgency in the user. 1

● **Number of Slashes in URL:** The empirical average number of slashes in benign URLs is f ive. 1 If the number of slashes exceeds this threshold, it is flagged as potentially malicious, indicating a deep, unusual directory structure often associated with temporary or throwaway phishing hosts.

Level 3 Heuristics: Dynamic & Content Verification (Stage 2 Dependencies)

These features are crucial for detecting sophisticated zero-hour attacks but require external communication or page crawling, justifying their classification into the high-latency Stage 2 analysis:

● **SSL Certificate Age and Trust:** While the mere presence of HTTPS is insufficient proof of legitimacy, checking the certificate's age and the trustworthiness of its issuer is vital. 1 Benign certificates typically have a minimum age between one and two years. 1 Certificates younger than one year, or those issued by a non-trusted authority, are f lagged as Suspicious . 1

● **Website Rank (Popularity):** Legitimate, high-value target websites usually possess high internet traffic rankings. The system utilizes databases like Alexa to compare the website's rank; if the rank is greater than 100,000, the site is designated as high-risk. 1

● **Abnormal URL (WHOIS Check):** This involves querying the WHOIS database to ascertain the registered identity of the domain. If the hostname presented in the URL is inconsistent with the primary identity derived from WHOIS data, the URL is deemed "Abnormal". 1

● **IFRAME Usage:** Phishers often embed invisible IFRAMEs (web pages without frame borders) into a legitimate-looking webpage to capture sensitive information without the user realizing the input field is hosted elsewhere. 1 Source code crawling is required to detect this structural deception.

● **URL of Anchor Analysis:** This involves crawling the source code and analyzing the anchor tags (). If the majority of hyperlinks originate from a domain different from the main URL's host, it suggests a redirection or link farm setup typical of malicious sites. 1

● **Information Submission to Email:** Detecting the use of functions such as mail() or mailto: within the URL path or source code indicates that the attacker is configured to redirect submitted user information directly to a personal email address. 1

The synthesized feature set, forming the core of TrustLayer's zero-hour detection capability, is summarized below:

Table 2: TrustLayer Multi-Tiered Heuristic Feature Synthesis

Feature Category	**Heuristic Feature**	**TrustLayer Rule Threshold/ Indicator**	Latency Tier	Citation Source
Structural (L1)	IP Address Usage (Domain Part	Presence of IPv4/IPv6 or Hexadecimal Equivalent.	Low	1
Structural (L1)	Special Character ('@')	Presence of the '@' symbol in the URL.	Low	1
Structural (L1)	Hostname Dot Count	Number of dots in the primary hostname segment > 3.	Low	1
Structural (L1)	Prefix or Suffix Dash (-)	Domain name separated by a dash symbol.	Low	1
Structural (L1)	URL Redirection (//)	Presence of // within the URL path.	Low	1
Deceptive (L2)	URL Shortening Service	Use of known shortener domains (e.g., bit.ly).	Low	1
Deceptive (L2)	Deceptive HTTPS Token	"HTTPS" string embedded within the domain name.	Low	1
Deceptive (L2)	Punycode/Unicode Use	Detection of the Punycode prefix (xn--).	Low	1
Deceptive (L2)	URL Length Tiering	Length > 75 chars (Phishing) or 54 $\le$ length $\le$ 75 (Suspicious).	Low	1
Deceptive (L2)	Sensitive Keywords in Path	Presence of high-risk terms ('paypal', 'signin', 'password').	Low	1
Deceptive (L2)	Number of Slashes	Number of slashes in the URL > 5.	Low	1
Dynamic/ Content (L3)	SSL Certificate Age/Trust	Age < 1 Year or untrusted issuer.	High (External)	1
Dynamic/ Content (L3)	Website Rank	Alexa Rank > 100,000.	High (External)	1
Dynamic/ Content (L3)	Abnormal URL (WHOIS)	Host Name is not consistent with WHOIS registered identity.	High (External)	1
Dynamic/ Content (L3)	IFRAME Usage	Detection of invisible IFRAME borders in source code.	Medium (Crawl)	1
Dynamic/ Content (L3)	URL of Anchor Analysis	Maximum number of hyperlinks from other domains.	Medium (Crawl)	1

Dynamic/ Content (L3)	Info Submission to Email	Presence of mail() or mailto: functions.	Medium (Crawl)	1

Machine Learning Modeling and Validation

Classifier Selection Justification and Training Objective

The primary objective of the machine learning stage is to minimize the False Negative Rate (FNR), as the cost of a missed phishing attack is operationally unacceptable. Based on empirical evidence, the Random Forest (RF) algorithm, having demonstrated a minimal FNR of 3.14% and a peak accuracy of 97.14% 1 , is selected as the core predictive engine for the structured features. The model training strategy requires utilizing a maximized data split, such as the 90:10 training-testing ratio, which research confirms leads to enhanced classifier performance. 1 Furthermore, given the known imbalance in real-world data, where malicious URLs are scarce compared to legitimate ones, specialized techniques like undersampling or oversampling must be applied during the preparation phase to ensure the model trains effectively. 1

Ensemble Stacking and Fusion Strategy

TrustLayer deploys a supervised ensemble approach, often referred to as stacking, to combine the complementary predictive capabilities of multiple, high-performing algorithms. This heterogeneity is essential for system resilience:

1. **Random Forest (RF):** Serves as the high-precision base classifier, optimized for FNR reduction.

2. **ExtraTree Classifier (ET):** Included specifically to address data scalability challenges. As observed, the performance of single models degrades significantly when exposed to huge datasets (e.g., 620,000 URLs), making the ET's robust handling of high-variance input crucial. 1

3.Support Vector Machine (SVM): Utilized as a high-precision filter, particularly on ambiguous or suspicious cases. Its empirically low FPR (ranging from 2.08% to 2.34%) allows the ensemble to maintain strong security (low FNR) while tightly controlling the risk of erroneously blocking benign traffic. 1

The final prediction is achieved through **confidence score fusion** , where the outputs of the base classifiers are weighted and combined by a meta-classifier, rather than relying on a simple majority vote. 2 This fusion strategy is proven to maximize reliability and push detection capabilities toward the high-accuracy limits demonstrated in advanced studies, potentially achieving accuracy rates up to 98.77%. 3

Feature Importance and Model Explainability (XAI)

A significant advantage of ensemble decision-tree-based methods is their inherent capability to provide measures of feature importance. This model explainability (XAI) is critical for system transparency, allowing security analysts to understand precisely which heuristic flags or combination of features were most significant in classifying a URL as malicious. 1 This process not only validates the model but also highlights which features are most indicative of current threat landscapes, aiding user trust and providing actionable feedback for continuous threat modeling.

Table 3: TrustLayer ML Model Structure and Rationale

Component	**Classifier**	**Primary Role/Rationale**	Optimization Target
Core Engine 1	Random Forest	High generalized accuracy and precision on structured feature vectors.	Minimize FNR (Baseline: 3.14%)
Core Engine 2	ExtraTree Classifier	High scalability and robustness for handling large, high-variance datasets.	Resilience under data scale stress
Supplementary Filter	Support Vector Machine (SVM)	High-precision filter for ambiguous/suspicious cases.	Minimize FPR (Baseline: 2.08% - 2.34%)
Meta-Classifier	Weighted Fusion/Stacking	Combines output confidence scores for maximized reliability.	Achieve theoretical $\ge$ 98% accuracy 3

Discussion of Performance and Future Work

Expected Performance Gains and Resilience

The hybrid TrustLayer architecture successfully bridges the gap between the speed of static URL analysis and the detection depth afforded by dynamic, content-aware checks. The implementation of the two-stage classification strategy ensures that the system can handle traffic in real-time by rapidly filtering low-risk content based on instantaneous Level 1/2 heuristics. By reserving the computationally intensive Level 3 features and the

advanced ensemble ML for Suspicious cases, the overall architectural latency is optimized without sacrificing detection fidelity. This setup provides superior zero-hour threat detection compared to static blacklist methods, while the inclusion of the SVM layer minimizes the high FPR traditionally associated with pure heuristic models.

TrustLayer Phase II: Advancing Beyond URL Features

The ongoing battle against phishing necessitates continuous investigation into how attackers actively attempt to bypass detection systems. 1 As machine learning models become standard defenses, attackers will target the model boundaries. Consequently, the feature thresholds and training parameters within TrustLayer must be subject to constant refinement through monitoring shifting threat patterns, such as new URL shortening services, changes in TLD usage, and novel Punycode applications. To maintain operational readiness, adaptive systems using online learning and transfer learning techniques should be implemented to swiftly modify the model's structure and weights in response to emerging, undocumented evasion techniques. 1

Conclusion

The TrustLayer system establishes a technically rigorous and operationally optimized framework for phishing URL detection. By formally synthesizing 16 crucial heuristic features—spanning immediate structural indicators like IP address usage and Punycode detection, to high-latency dynamic checks like SSL certificate age and WHOIS validation—TrustLayer provides comprehensive coverage against zero-hour threats. The subsequent deployment of a heterogeneous ensemble machine learning stack, prioritizing the Random Forest and ExtraTree classifiers alongside the low-FPR Support Vector Machine filter, maximizes detection accuracy (projected near 98%) while ensuring real-time operational speed via the two-stage pipeline. This hybrid feature engineering approach, grounded in empirical evidence regarding classification performance under varying data scales, provides a robust, evidence-based foundation critical for defending against the evolving tactics of cybercriminals.

References

1. Efficient Phishing URL Detection Using Graph-based Machine Learning and Loopy Belief Propagation - arXiv, accessed on November 15, 2025, https://arxiv.org/html/2501.06912v1

2. Phishing website detection using novel machine learning fusion approach - ResearchGate, accessed on November 15, 2025, https://www.researchgate.net/publication/350927392_Phishing_website_detection_using_novel_machine_learning_fusion_approach