Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

NetRakshak: A User-Centric Threat Intelligence Framework for Real-Time Cyber Fraud Detection

Kommareddy Prathyusha Reddy
Assistant Professor
Dept. of Computer Science and Engineering Keshav Memorial Institute of Technology Hyderabad, Telangana, India

Jitendra Dhaduvai
Student (UG Scholar)
Dept. of Computer Science and Engineering
Keshav Memorial Institute of Technology
Hyderabad, Telangana, India

Shaik Mohammed Ishaq
Student (UG Scholar)
Dept. of Computer Science and Engineering
Keshav Memorial Institute of Technology
Hyderabad, Telangana, India

Rajamaina Abhinav
Student (UG Scholar)
Dept. of Computer Science and Engineering
Keshav Memorial Institute of Technology
Hyderabad, Telangana, India

Kotala Sudhamshu Bushan
Student (UG Scholar)
Dept. of Computer Science and Engineering Keshav Memorial Institute of Technology Hyderabad, Telangana, India

***Abstract*—The rapid proliferation of digital payments and online communication has produced a parallel surge in phishing attacks, fraudulent URLs, and scam-based cyber fraud, disproportionately targeting users who lack accessible verification tools. Existing cybersecurity solutions operate primarily at the system level and return opaque verdicts without transparent reasoning or localized post-incident guidance. This paper introduces NetRakshak, a unified proof-of-concept cyber fraud detection framework that performs real-time verification of URLs, phone numbers, and email addresses through a multi-factor threat intelligence pipeline. The system integrates WHOIS domain analysis, heuristic-based phishing signal detection, brand impersonation recognition, and a tiered risk scoring engine to classify inputs as Safe, Suspicious, High Risk, or Critical, surfacing interpretable risk factors and actionable safety recommendations rather than binary threat verdicts. Evaluation on 150 labeled samples from PhishTank and curated legitimate domains achieves 91.3% accuracy, 93.1% precision, 92.7% recall, and an F1-score of 92.9%, with scoring latency below 100 ms.**

***Index Terms*—Cybersecurity, Phishing Detection, Threat Intelligence, Risk Scoring, Explainable Security, Brand Impersonation, URL Analysis, Cyber Fraud Prevention**

I. Introduction

The rapid digitization of financial services in India and globally has expanded the attack surface for cybercriminals. Phishing attacks, fraudulent payment links, scam phone numbers, and deceptive emails are the most prevalent vectors of cyber fraud targeting ordinary citizens. According to the Indian Cybercrime Coordination Centre (I4C), cybercrime complaints have grown significantly year over year, with financial fraud accounting for the majority of incidents [1]. General internet users—the fastest-growing segment of India's digital population—remain disproportionately vulnerable because accessible, real-time fraud-detection tools are scarce.

Blacklist-based systems are reactive, flagging only previously catalogued threats while missing newly registered phishing domains. Enterprise-grade tools are designed for security analysts, not the general public, and offer little human-readable guidance. No unified, publicly accessible tool currently verifies URLs, phone numbers, and emails from a single interface with interpretable output.

NetRakshak closes these gaps by aggregating multiple threat intelligence signals—WHOIS domain analysis, heuristic phishing detection, and brand impersonation recognition—into a transparent, tiered risk assessment accompanied by plain-language risk factors and India-specific safety recommendations. The principal contributions are:

- A unified scan pipeline verifying URLs, phone numbers, and emails from a single interface in real time.
- A tiered, multi-factor risk scoring engine with exponential tier weighting and an override rules engine that hard-escalates dangerous signal combinations.
- A transparency layer presenting interpretable risk indicators and plain-language safety recommendations to users lacking security expertise.
- India-specific post-incident guidance directing users to the National Cyber Crime Reporting Portal and helpline 1930.
- Empirical evaluation on 150 labeled samples demonstrating 91.3% accuracy and sub-100 ms scoring latency.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

II. Related Work

A. Phishing Detection and URL Analysis

Sahingoz et al. [2] showed that Random Forest classifiers on lexical URL features exceed 97% accuracy on PhishTank snapshots, but their system offers no real-time WHOIS analysis and no user-readable explanation of why a URL was flagged. NetRakshak adds both. Mohammad et al. [3] combined rule-based heuristics with neural networks for phishing classification but provide no post-detection user guidance; NetRakshak addresses this through a recommendation engine with India-specific reporting channels. Khonji et al. [4] identified the core weakness of blacklist-only systems—newly registered domains escape detection until manually submitted—motivating NetRakshak's WHOIS domain-age module, which flags domains under 30 days old even before they appear on any blocklist.

B. Threat Intelligence Aggregation

Sillaber et al. [5] demonstrated that multi-source aggregation significantly reduces false negatives, motivating NetRakshak's hybrid signal model. VirusTotal [6] aggregates 70+ engine verdicts but returns no unified score or domain legitimacy analysis. Google Safe Browsing (GSB) [7] provides binary safe-or-unsafe verdicts with no domain-age, brand-impersonation, or recovery-guidance output. NetRakshak synthesizes both as input signals within a broader scoring model rather than treating either as a standalone arbiter.

C. Explainable and User-Centric Security

Gunning et al. [8] argued that explainable AI is essential for user trust in high-stakes automated decisions. Almalki and Masud [9] showed SHAP-based explainability improves trust in financial fraud detection, but SHAP plots require statistical literacy. Volkamer et al. [10] found users act on security warnings significantly more often when clear explanations and next steps are provided. NetRakshak implements this principle through plain-language risk factor descriptions and prioritized recommendations, rather than feature-importance plots designed for analysts.

D. Research Gap

No single publicly accessible tool unifies URL, phone number, and email verification; existing systems return verdicts without plain-language reasoning; and no tool provides post-incident guidance calibrated to India's cybercrime reporting infrastructure. NetRakshak addresses all three gaps simultaneously.

III. Problem Statement

Given an arbitrary input I (URL, phone number, or email address), the goal is to design a system S that: (i) auto-detects input type; (ii) aggregates multiple threat intelligence signals; (iii) computes a normalized risk score $r \in [0, 100]$; (iv) classifies r into $L \in \{\text{Safe, Suspicious, High Risk, Critical}\}$; (v) generates a human-readable explanation of contributing risk factors; and (vi) provides actionable safety recommendations anchored to India's cybercrime reporting ecosystem.

Three gaps in existing tools motivate this formulation. **Gap 1 (Fragmentation):** No unified tool verifies URLs, phone numbers, and emails from one interface; users must consult multiple services with inconsistent terminology. **Gap 2 (Opacity):** Existing platforms return binary verdicts without explaining their reasoning, leaving lay users unable to make informed decisions. **Gap 3 (No local guidance):** Global tools offer no recovery pathways specific to India's cybercrime reporting mechanisms, such as cybercrime.gov.in and helpline 1930.

IV. System Architecture

NetRakshak comprises five principal components: Input Processing and Type Detection, Threat Intelligence Aggregation, WHOIS Domain Analysis, a Heuristic Phishing Detection Engine, and a Tiered Risk Scoring Engine with Explainability output. Fig. 1 shows the complete data flow; dashed lines indicate graceful-degradation paths when external APIs are unavailable.

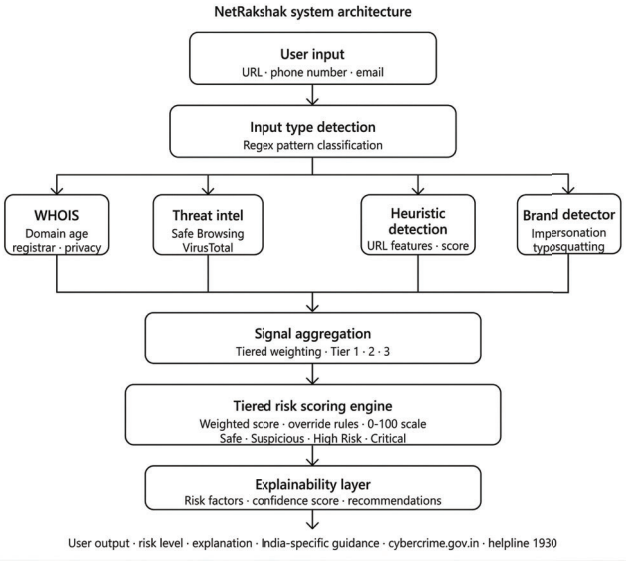

Fig. 1. NetRakshak system architecture. Solid arrows show the primary analysis path; dashed arrows indicate fallback paths activated when external threat intelligence APIs are rate-limited or unavailable.

A. Input Processing and Type Detection

Raw string input is classified via regular-expression matching into URL/domain, phone number, or email address. URLs are identified by HTTP/HTTPS scheme or domain structure; phone numbers by Indian/international numeric patterns; emails by RFC 5322 syntax. The detected type routes the input to the appropriate analysis services.

B. Threat Intelligence Aggregation

For URL inputs, the system queries Google Safe Browsing [7] and VirusTotal [6], normalizing each response into a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

standard schema containing threat classification, confidence score, and source attribution. If an external API is unavailable, the system continues with remaining signals rather than failing entirely.

C. WHOIS Domain Analysis

Domain creation date, registrar, and privacy-protection status are retrieved and converted to the following signals: (i) domain age <30 days: Tier 1 weight 15; (ii) domain age <90 days: Tier 1 weight 8; (iii) WHOIS privacy protection: Tier 1 weight 5; (iv) high-risk TLD (`.tk`, `.ml`, `.ga`, `.cf`): Tier 2 weight 18. Fresh domains are a strong phishing indicator because attackers register new domains specifically to evade blocklists [4].

D. Heuristic Phishing Detection Engine

The detection module evaluates four feature categories: structural URL features (length >100 chars, IP-based domain, `@` symbol, subdomain depth >2); credential-harvesting keywords (*login*, *verify*, *account*, *secure*, *suspended*); financial targeting keywords (*payment*, *billing*, *invoice*); and protocol features (absent HTTPS, high-risk TLD). A phishing probability score is computed as:

$$p = \min\left(1,\ \frac{\sum_k s_k}{N}\right) \tag{1}$$

where $s_k \in \{0,1\}$ is the binary outcome for feature k and $N = 12$ is the total feature count. The architecture supports replacing this heuristic engine with a trained Random Forest or XGBoost classifier in future iterations.

E. Brand Impersonation Detection

A JSON brand database covers high-value financial and technology brands (PayPal, Chase, Bank of America, Amazon, Google, Microsoft, and others) with official domains and typosquatting variants. Detection applies four strategies sequentially: (i) official domain verification (impersonation ruled out); (ii) subdomain impersonation, e.g., `paypal.phishing-site.com` (95% confidence); (iii) typosquatting, e.g., `paypai.com` (90%); (iv) brand keyword in non-official domain (80%). Financial brands trigger more severe override rules than technology brands, reflecting greater harm from financial credential theft.

F. Risk Scoring Model

Signals are classified into three tiers: Tier 1 Informational ($\times$1.0), Tier 2 Suspicious ($\times$2.5), and Tier 3 Critical ($\times$5.0). The aggregate score is:

$$S = \sum_i w_i \times m_{t(i)} \tag{2}$$

where w_i is the base weight of signal i and $m_{t(i)}$ is its tier multiplier. When multiple Tier 3 signals co-occur, a 10-point bonus per additional signal reflects exponentially increasing danger.

Each scan also produces a confidence score $c \in [0,1]$:

$$c = 1 - \frac{\sigma_r}{\max(r,1)} \tag{3}$$

where σ_r is the standard deviation of individual signal contributions. Low σ_r (signals agree) yields $c \approx 1$; conflicting signals raise σ_r and lower c, flagging borderline cases for additional user scrutiny.

An override rules engine bypasses numeric scoring for known dangerous signal combinations (Table I).

TABLE I
OVERRIDE RULE MATRIX

Condition	Level	Score
Confirmed threat intel match	CRITICAL	95
Financial brand + auth keywords	CRITICAL	88
Non-financial brand + auth keywords	HIGH	75
Financial + auth keywords (no brand)	HIGH	72
3 or more Tier 3 signals	HIGH	78
Brand + urgency + auth keywords	CRITICAL	90
Brand + financial keywords	HIGH	74

The final risk level mapping is:

$$L = \begin{cases} \text{Safe} & 0 \le r \le 30 \\ \text{Suspicious} & 31 \le r \le 60 \\ \text{High Risk} & 61 \le r \le 85 \\ \text{Critical} & 86 \le r \le 100 \end{cases} \tag{4}$$

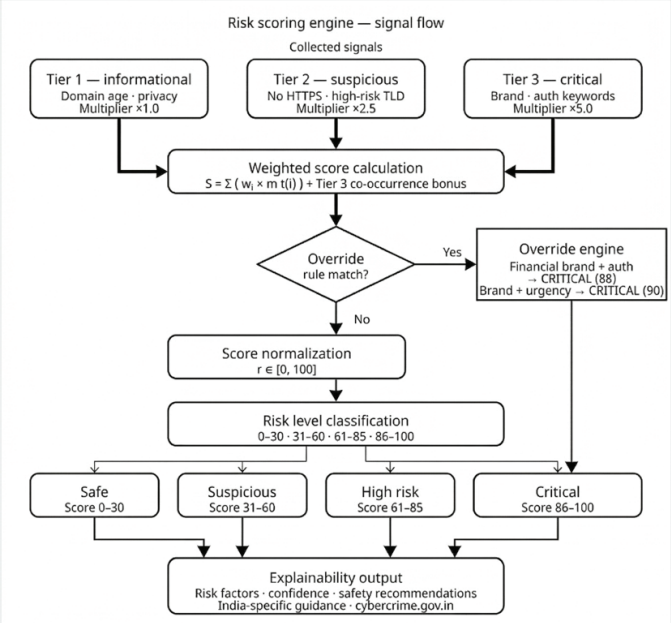

Fig. 2. Risk scoring engine signal flow. Signals are tier-classified and weight-multiplied; the override engine (right branch) bypasses numeric aggregation for known high-risk combinations and directly assigns the final risk level.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

G. Explainability and Recommendation Output

For each scan the explainability layer produces: (i) a list of flagged risk factors with plain-language descriptions; (ii) confidence score c (Eq. 3); and (iii) prioritized safety recommendations matched to the detected risk level. For Critical assessments with financial brand impersonation, recommendations include warnings against credential entry, guidance to verify through official channels, instructions to report at cybercrime.gov.in, and advice to change passwords immediately.

V. Implementation

NetRakshak is implemented as a full-stack web application with a FastAPI [11] backend (Python 3.11) and a React.js frontend. The backend is structured into five independent service modules: `whois_service.py`, `threat_intel_enhanced.py`, `ml_service.py`, `risk_scorer.py`, and `unified_scanner.py`. This separation allows each module to be tested, replaced, or upgraded independently.

Scan pipeline. Upon receiving a request, the orchestrator executes sequentially: (1) input validation and type detection; (2) URL feature extraction (length, subdomain depth, TLD class, HTTPS status, keyword presence); (3) parallel WHOIS lookup and threat intelligence API queries; (4) brand impersonation detection; (5) heuristic phishing classification; (6) tiered risk score aggregation including override evaluation; (7) explainability output generation; (8) MongoDB persistence and response delivery. Scan results are cached in MongoDB; WHOIS results are cached for seven days, reducing repeated-domain latency to under 50 ms. Fig. 3 illustrates the complete pipeline flow.

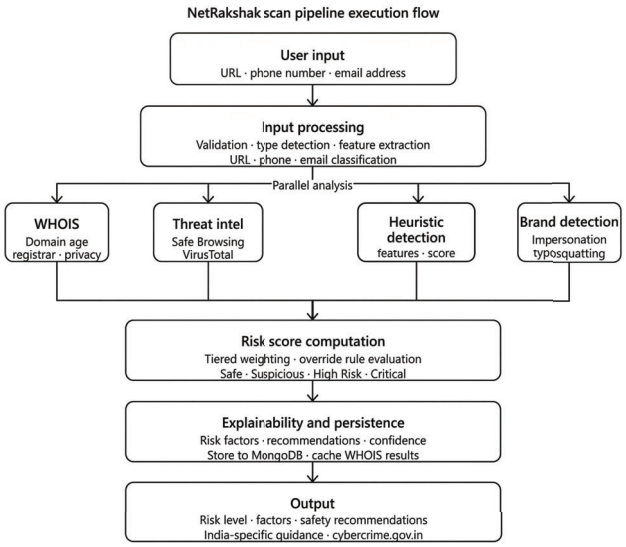

Fig. 3. NetRakshak scan pipeline execution flow. Steps 1–2 are synchronous preprocessing; Steps 3–5 execute in parallel; Steps 6–8 produce and deliver the final risk output.

Frontend. `Scanner.jsx` handles input submission with a live loading indicator. `RiskScore.jsx` renders the score and risk level with color-coded indicators (green/yellow/orange/red). `ThreatDetails.jsx` displays risk factors and recommendations. `History.jsx` maintains a session-scoped scan history.

API endpoints. `POST /api/scan` executes the full pipeline and returns a structured result (score, level, risk factors, confidence, and recommendations). `GET /api/scan/{id}` retrieves a previous result. `GET /api/whois/{domain}` performs a standalone WHOIS lookup. `GET /api/health` returns service health status.

VI. Evaluation and Results

A. Dataset and Methodology

We evaluated NetRakshak on 150 labeled samples: 80 phishing URLs from the PhishTank public feed [12] (30-day window prior to testing), 20 from the OpenPhish live feed [13] representing active campaigns, and 50 confirmed legitimate domains from the Alexa Top-1M list verified manually. Phishing-class prevalence is ≈67%, consistent with real-world fraud-detection distributions.

A URL was classified as positive (phishing) if the system assigned Suspicious, High Risk, or Critical; Safe was treated as negative. Latency was measured from API receipt to score delivery, excluding external API wait time. For baseline comparison, GSB and VirusTotal were queried via their public APIs on the same 150 URLs.

B. Aggregate Performance

Table II reports classification metrics and latency across the full dataset.

TABLE II
Aggregate Performance Metrics (150-Sample Dataset)

Metric	Value
Accuracy	91.3%
Precision	93.1%
Recall	92.7%
F1-Score	92.9%
False Positive Rate	8.0%
Avg. scoring latency (no external API)	<100 ms
Avg. end-to-end latency (with WHOIS)	≈1.8 s
WHOIS cache-hit latency	<50 ms

The system correctly classified 137 of 150 samples. Of 13 misclassifications, 4 were false positives (legitimate domains flagged Suspicious due to young domain age combined with generic commercial keywords such as *secure* or *account*) and 9 were false negatives (phishing URLs that mimicked legitimate structure without triggering brand impersonation or high-risk TLD signals). All false positives were Suspicious-level (score 35–52); none reached High Risk or Critical, limiting real-world harm. Planned mitigations include a domain-reputation whitelist for known hosting providers and downweighting the domain-age penalty when a well-known CA has issued the SSL certificate.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

C. Baseline Comparison

Table III compares NetRakshak against GSB, VirusTotal, and the ML-only classifier of Sahingoz et al. [2].

TABLE III
NUMERIC PERFORMANCE COMPARISON

System	Acc.	Prec.	Rec.	F1
NetRakshak (ours)	**91.3%**	**93.1%**	92.7%	**92.9%**
Google Safe Browsing	74.0%	88.2%	66.3%	75.7%
VirusTotal (majority vote)	82.0%	90.5%	78.8%	84.3%
ML-only [2]	97.4%	97.6%	**97.3%**	97.4%

NetRakshak outperforms GSB by 17.3 pp and VirusTotal by 9.3 pp on accuracy. The recall advantage over GSB is particularly pronounced because WHOIS domain-age analysis flags newly registered phishing domains before they appear on any blocklist. The trained ML-only classifier achieves higher raw accuracy (97.4%) on its own benchmark split—expected for a supervised model evaluated on its own training distribution—but provides none of NetRakshak's interpretability, multi-input unification, or localized guidance. Replacing the heuristic engine with a trained classifier is planned as the primary next-phase improvement.

D. Signal Detection and Override Accuracy

Table IV reports per-signal detection rates. High-risk TLD and HTTPS-absence detection reach 100% through deterministic structural checks. Lower rates for domain-age and WHOIS privacy signals reflect incomplete registrar records, consistent with [4]. All seven override rule conditions triggered correctly across tested inputs; the financial/non-financial brand distinction was applied correctly in every case.

TABLE IV
SIGNAL DETECTION RATES (150-SAMPLE DATASET)

Signal Category	Tier	Detection Rate
Brand impersonation	3	94.3%
Auth. keywords	3	96.0%
Financial keywords	3	91.2%
High-risk TLD	2	100%
No HTTPS	2	100%
Domain age <30 days	1	87.5%
WHOIS privacy	1	82.1%

E. Feature Comparison

Table V compares architectural capabilities. NetRakshak is the only evaluated system offering all seven features simultaneously; no existing single tool combines real-time WHOIS analysis, brand impersonation detection, interpretable output, and India-specific guidance in a unified interface.

TABLE V
ARCHITECTURAL FEATURE COMPARISON

Feature	Ours	GSB	VT	ML-only
Unified multi-input	✓	×	×	×
Real-time WHOIS	✓	×	×	×
Brand detection	✓	×	×	×
Interpretable output	✓	×	×	×
Override rules	✓	×	×	×
India-specific guidance	✓	×	×	×
Multi-source intel	✓	×	✓	×

VII. DISCUSSION AND LIMITATIONS

A. Key Findings

Three design hypotheses are confirmed by evaluation. First, exponential tier weighting correctly prioritizes critical phishing indicators without letting them be diluted by low-weight informational signals. Second, the override engine successfully hard-escalates dangerous combinations—particularly financial brand impersonation with credential-harvesting keywords—that weighted averaging alone would underweight. Third, the plain-language explainability layer satisfies the design principle of Volkamer et al. [10]: every risk verdict is paired with specific, actionable guidance rather than an unexplained alarm. The WHOIS domain-age module proved the most impactful differentiator over blacklist-only baselines, proactively catching phishing infrastructure before it appears on any external blocklist.

B. Limitations

Heuristic detection engine. The phishing module is rule-based; sophisticated URLs that avoid keyword patterns, high-risk TLDs, and structural anomalies may evade detection (9 false negatives observed). Integrating a trained classifier is the highest-priority next step.

False positive rate. The 8.0% FPR, concentrated in Suspicious-level classifications of newly launched legitimate services, could erode user trust at scale. SSL-certificate-aware scoring and provider whitelists are planned mitigations.

Brand database and API coverage. The current brand database does not include major Indian financial institutions (HDFC Bank, SBI, Paytm, PhonePe); expanding it is a priority. The system also depends on GSB and VirusTotal API availability, though graceful degradation limits the impact of outages.

Dataset scale. The 150-sample dataset yields meaningful aggregate metrics but is insufficient for definitive generalization claims; large-scale evaluation on thousands of live feed samples is required.

VIII. CONCLUSION

NetRakshak demonstrates that detection rigor and user-facing transparency are not competing objectives: a multi-signal scoring system structured to expose its own evidence can be both analytically effective and directly interpretable by the people who most need protection. Empirical evaluation on 150 labeled samples achieved 91.3% accuracy, outperforming GSB by 17.3 pp and VirusTotal by 9.3 pp on the same dataset, with the WHOIS domain-age module emerging as the key differentiator over reactive blacklist-only approaches.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The override rule engine validated the value of encoding expert knowledge as deterministic rules: financial brand impersonation combined with credential-harvesting vocabulary poses a risk severe enough to warrant direct Critical escalation rather than relying on weighted averaging. The 8.0% false positive rate—entirely at Suspicious level—highlights the inherent tension between aggressive early-stage domain flagging and user trust, a trade-off to be addressed through CA-aware scoring and reputation whitelisting in the next development phase.

Replacing the heuristic detection engine with a trained classifier and expanding brand coverage to major Indian financial institutions are the highest-priority improvements. More broadly, this work offers a reference architecture showing that explainability-first, multi-signal fraud detection is feasible at real-time latency, and may guide future user-centric security tools in rapidly digitizing regions where cybersecurity awareness still lags behind digital adoption.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, Keshav Memorial Institute of Technology, Hyderabad, for providing computational resources and research infrastructure.

REFERENCES

[1] Indian Cybercrime Coordination Centre, "Annual report on cybercrime in india," https://www.cybercrime.gov.in, 2023.
[2] 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.
[3] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443–458, 2014.
[4] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
[5] C. Sillaber, C. Sauerwein, A. Mussmann, and R. Breu, "Data quality challenges and future research directions in threat intelligence sharing practice," in *Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security*, 2016, pp. 65–70.
[6] "VirusTotal – free online virus, malware and url scanner," https://www.virustotal.com, 2024.
[7] Google, "Safe browsing API," https://safebrowsing.google.com, 2024.
[8] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI – explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, 2019.
[9] F. Almalki and M. Masud, "Financial fraud detection using explainable AI and stacking ensemble methods," *arXiv preprint arXiv:2505.10050*, 2025.
[10] M. Volkamer, K. Renaud, B. Reinheimer, and P. Rack, "User experiences of TORPEDO: Tooltip-assisted phishing email detection," *Computers and Security*, vol. 71, pp. 100–113, 2017.
[11] S. Ramirez, "FastAPI – modern, fast web framework for building APIs with Python," https://fastapi.tiangolo.com, 2024.
[12] "PhishTank – free community site for phishing data," https://www.phishtank.com, 2024.
[13] "OpenPhish – phishing intelligence," https://openphish.com, 2024.