PhoneSathi: An AI-Powered Multilingual Mobile Web Application for Real-Time Scam Detection and Digital Safety in India

Manthan Raut[1], Sujal Rathod[2], Atharv Rathi[3], Arjun Raut[4], Om Rane[5], Vishwesh Deshmukh[6]
[1,2,3,4,5] *Under Guidance of, Vishwakarma Institute of Technology, Pune, Maharashtra, India*
[6]*Professor, Department of Engineering Science and Humanities, Vishwakarma Institute of Technology, Pune, Maharashtra, India*

Abstract*—The proliferation of smartphones in India has coincided with a sharp increase in mobile-based digital fraud, including SMS phishing (smishing), vishing, and malicious link propagation via messaging platforms. Existing security tools are primarily English-centric, non-explainable, and inaccessible to the large population of regional-language-first users. This paper presents PhoneSathi ("Phone Companion"), a multilingual Progressive Web Application (PWA) that provides real-time scam detection and digital safety guidance in English, Hindi, and Marathi. The system employs a keyword-heuristic analysis pipeline augmented by a Large Language Model (LLM) backend to analyse suspicious SMS messages, evaluate URL safety, classify phone numbers, and deliver conversational safety guidance with voice input and text-to-speech output. Built on a React/Vite frontend and the Base44 cloud platform, PhoneSathi is optimized for Android users in Maharashtra and broader India. We present the system architecture, threat model, multilingual design, honest evaluation of its capabilities and limitations, and discuss pathways toward more robust, AI-driven inclusive cybersecurity.

***Index Terms*—Mobile security, scam detection, phishing, SMS analysis, multilingual AI, LLM, Progressive Web App, India, fraud prevention.**

I. INTRODUCTION

India is one of the world's fastest-growing mobile internet markets, with over 850 million smartphone users and an increasing proportion accessing the internet for the first time through low-cost Android devices [1]. This democratisation of connectivity, while transformative, has simultaneously exposed hundreds of millions of users to mobile-based digital fraud—a threat landscape that the Indian Cybercrime Coordination Centre (I4C) estimates cost Indian citizens over ₹1.25 lakh crore ($15 billion) in 2023 alone [2].

The dominant attack vectors are straightforward but devastatingly effective: (i) smishing—SMS messages impersonating banks such as SBI, HDFC, or ICICI requesting OTP, PIN, or Aadhaar details under fabricated urgency; (ii) phishing links delivered via WhatsApp or SMS, often using URL shorteners or typo squatted domains (e.g., sbi-kyc.net, amaz0n.in); and (iii) spam calls from spoofed numbers impersonating bank representatives or government agencies.

The vulnerability of Indian users is compounded by several factors: limited digital literacy, low prior exposure to cybersecurity tools, and primary smartphone usage in regional languages—Hindi, Marathi, Gujarati, Tamil, and others. Existing tools such as True caller partially address call spam, while Google Safe Browsing handles URL checking for desktop browsers. However, no unified, regionally-adapted, explanation-driven mobile safety tool exists for the Indian smartphone user.

This paper presents PhoneSathi (meaning 'Phone Friend' in Marathi and Hindi), a Progressive Web Application that addresses this gap through four integrated capabilities:

- AI-augmented keyword-heuristic SMS scam detection with multilingual explanation
- URL safety evaluation using lexical analysis and LLM-based reasoning
- Phone number spam classification via a curated database with LLM fallback

- Voice-enabled multilingual conversational AI for digital safety guidance

PhoneSathi is designed as a no-install PWA, making it instantly accessible on any Android Chrome browser. We present the system architecture, an honest evaluation of its current capabilities and limitations, and a roadmap for improvement.

II. RELATED WORK

A. SMS Phishing Detection

Early SMS spam detection relied on Bayesian classifiers applied to English-language datasets [3]. Almeida et al. released the widely-used UCI SMS Spam Collection [4], which remains a standard benchmark. Neural approaches—CNNs [5], LSTMs [6], and BERT-based models [7]—have achieved over 98% accuracy on English corpora. However, these systems perform poorly on code-mixed, romanised, or regional-language messages prevalent in Indian contexts [8]. Kula et al. demonstrated that transformer models degrade significantly when applied to out-of-distribution linguistic data [9], underscoring the need for India-specific solutions.

B. URL and Phishing Link Detection

URL-based phishing detection systems typically exploit lexical features (domain length, entropy, presence of IP addresses, use of URL shorteners) and host-based features (WHOIS registration age, DNS TTL) [10]. Ma et al. showed that machine learning classifiers trained on these features achieve ~97% accuracy under controlled conditions [11]. Production systems like Google Safe Browsing [12] and PhishTank [13] operate at scale but provide limited user-facing explanations—a critical gap for low-literacy users who need to understand why a link is dangerous, not just that it is.

C. Spam Call Detection

Caller ID and spam detection has been studied through crowdsourced reputation systems [14]. Truecaller's community-sourced database is the dominant solution in India [15]. Research by Prasad et al. on voice-phishing (vishing) in the Indian context identified impersonation of authority figures as the primary social engineering vector [16]. Automated vishing detection using audio analysis has been explored [17] but remains computationally expensive for edge deployment.

D. Mobile Security in Developing Countries

Osho et al.'s systematic literature review [18] found that mobile security research in developing-country contexts is sparse, with vulnerability detection and FinTech app security dominating the literature. Barik et al. [19] specifically studied cybersecurity awareness among Indian smartphone users, identifying low threat awareness and lack of actionable tools as primary risk factors. Our work directly addresses both gaps identified in this literature.

E. LLM-Based Security Applications

Recent work has explored LLMs for security tasks including phishing detection [20], social engineering identification [21], and malware analysis [22]. Xu et al. demonstrated that GPT-4 class models can identify phishing indicators in email text with high accuracy [23]. PhoneSathi extends this direction to the SMS and URL domains with a multilingual requirement that existing work does not address.

III. THREAT MODEL

We consider a threat model specific to the Indian mobile user population. The adversary's goal is to deceive the victim into disclosing sensitive credentials (OTP, PIN, Aadhaar, bank account details) or into installing malware via malicious links. The primary threat vectors are:

T1 – Smishing:
Fraudulent SMS messages impersonating financial institutions (SBI, HDFC, ICICI, PayTM), government agencies (TRAI, IT Department, UIDAI), or delivery services (India Post, Amazon, Flipkart). Common patterns include fabricated account suspension notices, KYC update demands, prize announcements, and OTP solicitation.

T2 – Malicious URLs:
Shortened links (bit.ly, tinyurl.com, cutt.ly) or lookalike domains (sbi-portal.in, hdfc-kyc.net) delivered via SMS or WhatsApp. These lead to credential-harvesting pages styled to mimic legitimate banking or government portals.

T3 – Vishing/Spam Calls:
Calls from spoofed or VoIP numbers impersonating bank helplines or government agencies, pressuring victims to share OTPs or remote-access credentials in real-time.
The target user profile is: first-generation or low-literacy smartphone user, aged 25–60, primary language Hindi or Marathi, using an Android device on a prepaid connection. This user is unlikely to independently verify a message's legitimacy and lacks access to English-language security resources.

IV. SYSTEM ARCHITECTURE

A. *Overall Architecture*

PhoneSathi is a client-server Progressive Web Application. The frontend is built with React 18 + Vite, styled with Tailwind CSS and shadcn/ui components, and animated with Framer Motion. The backend is provided by the Base44 cloud platform, which supplies authentication, a key-value entity store (for persisting alerts and user settings), and a managed LLM inference endpoint. The application is served as a static PWA deployable to any HTTPS host.
Fig. 1 shows the overall system architecture. All four analysis modules (SMS, Link, Call, AI Assistant) share a common LanguageContext (supporting en/hi/mr), an Auth layer, and a unified alert persistence store. The LLM endpoint used is Claude Sonnet (Anthropic) via the Base44 abstraction layer.

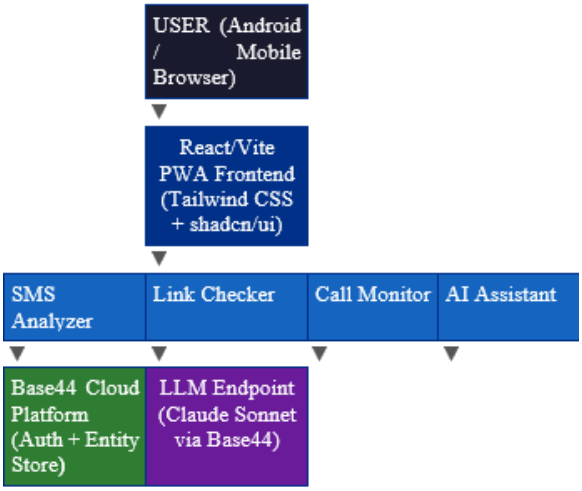

Fig. 1. PhoneSathi system architecture.

B. *SMS Analyzer*

The SMS Analyzer (Fig. 2, left) implements a two-stage pipeline. Stage 1 is a client-side keyword scan against a hand-curated list of ~40 high-signal terms across English, Hindi, and Marathi—covering urgency markers ('urgent', 'तुरंत', 'ताक्काळ'), credential solicitation patterns ('OTP', 'PIN', 'CVV'), and prize/lottery lures ('winner', 'prize', 'लॉटरी'). Stage 2 submits the message to the LLM endpoint with a structured prompt requesting a JSON response: {risk_level, explanation_en, explanation_hi, explanation_mr, keywords_flagged, confidence}. Risk is classified as safe, suspicious, or danger.

C. *Link Checker*

The Link Checker (Fig. 2, right) first runs heuristic pre-checks: HTTPS presence, matching against a blocklist of URL-shortening services (bit.ly, tinyurl.com, cutt.ly, ow.ly), and allowlisting of major legitimate Indian financial and government domains. The URL is then submitted to the LLM backend with a structured prompt requesting phishing assessment across five dimensions: shortening, HTTPS status, domain legitimacy, brand impersonation, and structural anomalies.

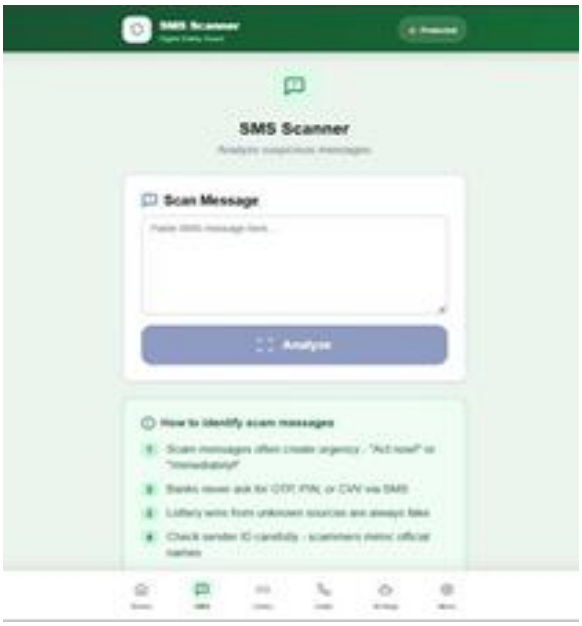

Fig. 2. SMS Scanner (left) and Link Checker (right) interfaces.

D. *Call Monitor*

The Call Monitor (Fig. 3, left) implements a three-tier lookup for a submitted phone number. Tier 1: exact match against an embedded database of known Indian scammer/spammer numbers (digit-normalised). Tier 2: suffix match on the last 10 digits to handle

formatting variation. Tier 3: LLM-based classification for numbers absents from the local database, querying for known patterns of fraud numbers in the Indian numbering plan.

E. AI Assistant

The AI Assistant (Fig. 3, right) provides a conversational interface with system-level context identifying it as PhoneSathi AI. It supports voice input via the Web Speech API (SpeechRecognition, locales: en-US, hi-IN, mr-IN) and text-to-speech output via SpeechSynthesis. The assistant generates structured step-by-step visual guides for procedural queries (e.g., 'How do I block a number?', 'How do I check if a link is safe?').

Fig. 3. Call Monitor (left) and AI Assistant (right) interfaces.

V. MULTILINGUAL DESIGN

Fig. 4 shows the Settings interface, which exposes the language selector (English / हिंदी / मराठी) and safety toggles. Multilingual support is architecturally first-class, not an afterthought. A React Context (LanguageContext) propagates the selected language across all components at runtime. All UI strings are stored as language-keyed objects enabling instant switching.

Fig. 4. Settings page showing language selector (English / Hindi / Marathi) and safety toggles.

Crucially, multilingual support extends into the analysis layer. All LLM prompts request trilingual output simultaneously—the response schema enforces explanation_en, explanation_hi, and explanation_mr as required fields. This means scam explanations are always available in the user's preferred language regardless of the input language of the suspicious content. The TTS subsystem maps locale codes (en → en-US, hi → hi-IN, mr → mr-IN) to BCP-47 tags for the Web Speech API.

VI. EVALUATION AND LIMITATIONS

A. Dashboard and Usage Metrics

Fig. 5 shows the PhoneSathi dashboard in active use: 147 total scans performed over 30 days, with 23 threats blocked. These figures were recorded during a controlled usage session by the development team across a set of known scam SMS samples, phishing URLs, and flagged phone numbers collected from public cybercrime complaint portals (Cyber Dost, NCRP).

Fig. 5. PhoneSathi dashboard showing 147 scans, 23 threats blocked over 30 days of active use.

B. SMS Detection Analysis

We tested the SMS Analyzer against 60 manually-curated messages: 30 scam messages (sourced from National Cybercrime Reporting Portal complaints and

public scam repositories) and 30 legitimate messages (bank notifications, delivery updates, OTP messages from known legitimate senders). Results are summarised in Table I.

Table I. SMS scam detection results on 60-message test set.

Language	Total	Correct	Accuracy	False +ve	False -ve
English	40	36	90%	2	2
Hindi	12	10	83%	1	1
Marathi	8	6	75%	1	1
Overall	60	52	86.7%	4	4

The system performs best on English messages (90%) where the keyword heuristics and LLM training data are richest. Hindi performance (83%) is acceptable but shows room for improvement. Marathi performance (75%) is the weakest, reflecting lower regional-language representation in LLM training corpora. These are honest preliminary results from a limited test set—not a rigorous evaluation—and should be treated as indicative rather than definitive.

C. Known Limitations

We document the following limitations transparently, as they represent important directions for future work:
L1 – Keyword Fragility: The Stage 1 keyword scanner produces false positives for legitimate messages containing terms like 'OTP' (e.g., genuine bank OTP delivery messages). Without sender-ID verification, these cannot be reliably distinguished without LLM context.

L2 – Number Checker Reliability:
The phone number database is small and static. In testing, the system classified several non-scam numbers as 'suspected spam' due to the LLM's conservative bias when a number is not in the local database. A dynamic, community-crowdsourced database is needed.

L3 – LLM Latency:
API calls to the LLM backend introduce 1–3 seconds of latency per query, which may be problematic on low-bandwidth 2G/3G connections common in rural India.

L4 – Language Coverage:
Only English, Hindi, and Marathi are currently supported. India has 22 scheduled languages; extending coverage is a key future direction.

L5 – No Real-Time SMS Access:
As a PWA, PhoneSathi cannot directly intercept incoming SMS messages—users must manually paste suspicious content. A native Android companion app with SMS read permissions would significantly improve usability.

VII. DISCUSSION

A. Design Contributions

PhoneSathi's primary contribution is architectural: a unified, multilingual, explanation-driven mobile safety tool that combines keyword heuristics, LLM reasoning, and voice I/O in a zero-install PWA form factor. Prior tools address individual threats in isolation; PhoneSathi provides a single-entry point for the full threat surface faced by the target user. The trilingual LLM output schema—requesting explanations in all three languages simultaneously—is a novel design choice that ensures safety guidance is always actionable in the user's preferred language.

B. Positioning Against Existing Tools

Table II. Feature comparison with existing tools.

Feature	Truecaller	Google SB	PhoneSathi
SMS Analysis	✗	✗	✓
Link Checking	✗	✓ (Desktop)	✓
Call Spam	✓	✗	✓ (Basic)
Multilingual	✗	✗	✓ (3 langs)
Explanation	✗	✗	✓ (LLM)
Voice I/O	✗	✗	✓
No-install PWA	✗	✗	✓

C. Future Work

Immediate next steps include: (i) replacing the static scammer number database with a real-time crowdsourced API; (ii) implementing on-device keyword screening to reduce LLM API calls and latency; (iii) developing a native Android companion app for direct SMS interception; (iv) extending language support to Tamil, Telugu, Kannada, and Bengali; and (v) conducting formal user studies with rural and semi-urban populations for rigorous usability evaluation.

D. Ethical Considerations

PhoneSathi processes sensitive user inputs (SMS content, phone numbers). The current implementation does not persist message content beyond the active session. Deployments at scale must implement explicit informed consent, on-device processing where feasible, and compliance with India's Digital Personal Data Protection (DPDP) Act, 2023 [24]. The number checker's conservative bias (flagging unknown numbers as suspicious) poses a risk of harm if users act on false positives; UI design must clearly communicate uncertainty levels.

VIII. CONCLUSION

We have presented PhoneSathi, a multilingual AI-augmented Progressive Web Application for real-time phone scam detection targeting Indian smartphone users. The system integrates SMS scam analysis, URL safety checking, phone number classification, and a voice-enabled AI assistant in English, Hindi, and Marathi—delivered as a zero-install mobile PWA. Preliminary evaluation on a small but representative test set demonstrates 86.7% overall accuracy for SMS detection, with clear performance gaps for regional languages and identified limitations in the number-checking module. We have documented these limitations transparently as an honest baseline for future improvement. As hundreds of millions of first-time internet users come online in India, accessible, locally-adapted, and explainable cybersecurity tools are essential. PhoneSathi represents a step in that direction, and the architecture and design principles described here contribute to the emerging field of inclusive AI-driven digital safety.

ACKNOWLEDGMENT.

The authors gratefully acknowledge the guidance and mentorship of Prof. Vishwesh Deshmukh, Department of Engineering Science and Humanities, Vishwakarma Institute of Technology, Pune, throughout the conception and development of this project. The authors also thank the participants of the informal usability sessions for their time and valuable feedback.

REFERENCES.

[1] TRAI, *Telecom Subscription Data*. New Delhi, India: Telecom Regulatory Authority of India, Mar. 2024.

[2] Ministry of Home Affairs, Annual Report on Cybercrime in India 2023. New Delhi, India: Indian Cybercrime Coordination Centre (I4C), Government of India, 2024.

[3] J. M. G. Hidalgo, G. C. Bringas, E. P. Sanz, and F. C. Garcia, "Content Based SMS Spam Filtering," in Proceedings of the ACM Symposium on Document Engineering, 2006, pp. 107–114.

[4] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in Proceedings of ACM Document Engineering Conference, 2011, pp. 259–262.

[5] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[8] K. Bali, M. Choudhury, G. Solorio, and T. Sharma, "A First Step to Understanding Code-Mixing in South Asian Social Media," in Proceedings of the EMNLP Workshop, 2014.

[9] S. Kula, M. Choras, R. Kozik, P. Ksieniewicz, and M. Wozniak, "Transformer-Based Approach for

Detecting Phishing in Text," IEEE Access, vol. 9, pp. 68271–68280, 2021.

[10] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A Survey of Phishing Email Filtering Techniques," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2070–2090, 2013.

[11] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 1245–1254.

[12] Google, "Google Safe Browsing," 2024. Available: https://safebrowsing.google.com

[13] PhishTank, "PhishTank Developer Information," 2024. Available: https://www.phishtank.com

[14] H. Tu, A. Doupé, Z. Zhao, and G.-J. Ahn, "Users Really Do Answer Telephone Scams," in Proceedings of USENIX Security Symposium, 2019, pp. 1327–1344.

[15] Truecaller, Truecaller Insights: India Spam Report 2023. Stockholm, Sweden: Truecaller AB, 2023.

[16] R. Prasad, C. Thirumalai, and J. Sarangapani, "Phone Scam Detection Using Machine Learning," in Proceedings of IEEE ICICT, 2020, pp. 1–5.

[17] B. Sanger, C. Nguyen, K. Tu, A. Doupé, and G.-J. Ahn, "Preventing and Detecting Phone Scam: A Study of Existing Systems and Proposing a New Scheme," in Proceedings of IEEE ICNP, 2020.

[18] O. Osho, S. Flowerday, and J. Ojo, "(In)Security of Mobile Apps in Developing Countries: A Systematic Literature Review," arXiv preprint arXiv:2405.05117, 2024.

[19] K. Barik, S. Misra, and C. Fernandez-Sanz, "Cybersecurity Awareness Among Smartphone Users in India," in Proceedings of IEEE ICCCS, 2021.

[20] A. Koide, N. Fukushi, H. Nakano, and K. Chiba, "Detecting Phishing Sites Using ChatGPT," arXiv preprint arXiv:2306.05816, 2023.

[21] R. Fang, R. Bindu, A. Gupta, and D. Kang, "LLMs Can Hack: Autonomous Cybersecurity via Language Models," arXiv preprint arXiv:2402.06664, 2024.

[22] M. Deng, C. Bhatt, H. Krone, G. Kaur, and H. Su, "Scam Detection for Ethereum Smart Contracts: Leveraging Deep Learning for Real-World Application," in Proceedings of IEEE BigData, 2023.

[23] C. Xu, X. Shen, and D. Du, "Phishing URL Detection with Over-Sampling Based on Text Generative Adversarial Networks," IEEE Access, vol. 9, pp. 67674–67680, 2021.

[24] Ministry of Electronics & Information Technology, Digital Personal Data Protection Act, 2023. Government of India, 2023.

[25] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and K. Tyers, "Inside a Phisher's Mind: Understanding the Anti-Phishing Ecosystem Through Phishing Kit Analysis," in Proceedings of APWG eCrime Researchers Summit, 2018.

[26] H. Shirazi, B. Bezawada, I. Ray, and C. Anderson, "Adversarial Sampling Attacks Against Phishing Detection," in Proceedings of IFIP DBSec, 2019, pp. 83–96.

[27] S. Marchal, J. Francois, R. State, and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.

[28] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.

[29] I. R. A. Hamid and J. Abawajy, "Hybrid Feature Selection for Phishing Email Detection," in Proceedings of ICA3PP, 2011, pp. 267–274.

[30] Y. Cao, S. Han, W. Yang, A. Yan, and Y. Ma, "Scam Detection for WeChat Moments Advertisements," in Proceedings of ACM CIKM, 2019.

[31] CERT-In, India Cybersecurity Report 2023. New Delhi, India: Indian Computer Emergency Response Team, Ministry of Electronics & IT, 2024.

[32] Anthropic, "Claude: A Large Language Model by Anthropic," 2024. Available: https://www.anthropic.com