International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Conference Proceedings: Volume: 05 ACME – 2026 DOI: 10.55041/ISJEM.ACME160
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

UPI Cyber Guard AI:A Hybrid NLP-Vision Security Framework for Digital Payment Fraud Prevention

Dr. N. Ramadevi
Associate Professor, Dept. of CSE(DS) Santhiram Engineering College (Autonomous), Nandyal, Andhra Pradesh, India
ramadevi.cse@srecnandyal.edu.in
ORCID ID: 0000-0001-9363-2805

G.Surendra
Dept. of CSE(DS)
Santhiram Engineering College (Autonomous), Nandyal, Andhra Pradesh, India
22x51a3217@srecnandyal.edu.in

Mr. Pathakamuri Srinivasulu,
Asst. Professor, Dept. of CSE,
PBR Visvodaya Institute of Technology and Science, Udayagiri Road, Kavali, Nellore Dt, pin- 524201, Andhra Pradesh, India.
srinivasulu.p@visvodayata.ac.in
ORCID ID: 0009-0001-8376-6532

M.Hema Sagar Reddy
Dept. of CSE(DS)
Santhiram Engineering College (Autonomous), Nandyal, Andhra Pradesh, India
22x51a3229@srecnandyal.edu.in

E.Kamal Santhosh
Dept. of CSE(DS)
Santhiram Engineering College (Autonomous), Nandyal, Andhra Pradesh, India
22x51a3211@srecnandyal.edu.in

M. Shashidhar Achari
Dept. of CSE(DS)
Santhiram Engineering College (Autonomous), Nandyal, Andhra Pradesh, India
22x51a3228@srecnandyal.edu.in

***Abstract*—** The Unified Payments Interface, or UPI, changed the way people transact digitally in India by enabling instantaneous, effortless transfers of value between peers. The meteoric rise in popularity over the last few years has caused an equally rapid increase in cyber crime, including phishing scams, tag and trick schemes (Clickjacking), email scams and hackers altering product images via stolen QR codes to obtain sensitive user data.

Current measures for preventing UPI fraud (i.e., transaction alerts, etc.) are primarily reactive; therefore, they are only able to alert victims once the transaction has occurred.

This paper details a proactive system named 'UPI Cyber Guard AI' that will help identify these potential types of fraud before a payment authorization occurs. UPI Cyber Guard AI is a multi-modal, proactive system that utilizes a Fast API back-end to monitor real-time fraud and once again demonstrate this system's effectiveness.

UPI Cyber Guard AI has been tested on more than 500 real-world samples, and the systems have reached an overall accuracy rating of 96.4%; with most results showing average return times of less than 1.5 seconds.

Keywords —Digital Payment Fraud Prevention, Machine Learning (ML), Natural Language Processing (NLP), QR (Quick response code) image security, Phishing Prevention, Hybrid Machine Learning and Artificial Intelligence.

I.INTRODUCTION

1.1 The Digital Payment Revolution and Its Discontents

In the last decade digital payment systems have seen rapid widespread adoption across the world, with India establishing itself as the leader in this respect through the launch of the Unified Payments Interface (UPI). UPI allows for interoperability between an individual user's multiple bank accounts via one mobile app, allowing all financial transactions (both person to person and merchant-to-consumer) to be made instantly wherever and whenever the user wishes to make them. As such, UPI has enabled small vendors to accept credit and debit card payments using only a smartphone, whereas traditional banks required point-of-sale devices to process such transactions.

While these same features that have allowed UPI to blossom serve to attract customers to its use, they have also created new vulnerabilities for UPI users, with increased vulnerability to cyber fraud due to malicious

acts [1], [2]. End-users have become the weakest link in the security chain, as attackers frequently target them through social engineering techniques. Cyber criminals use psychological attributes such as urgency, fear, and greed to trick unsuspecting victims into authorizing fraudulent transactions.

1.2 The Evolving Threat Landscape

As UPI users continue to grow in number and the technology surrounding UPI continues to develop, many of these attack methods are evolving as hackers move towards new ways to communicate with their victims. For example, using: Phishing SMS messages sent from SMS and messaging applications. In each of these examples, the SMS is from a fraudulent bank or payment institution and instructs the recipient to perform KYC verification of their account or to process a refund. There is a link included in the SMS message intended to install malware on the victim's device[4].

Deceptively crafted emails that replicate the appearance of legitimate email communication that drives users towards fraudulent portals designed to collect their access credentials [5]. Malicious URLs (e.g., obfuscated or shortened URLs) that redirect the user to an attacker-operated webpage containing malware that allows the attacker to collect a variety of sensitive information about the individual (including credit and debit card numbers, Social Security numbers, login credentials, etc.).

1.3 Existing Solutions Have a Shortfall

Even with existing safeguards, many of the banking and payment platforms use a reactive approach toward security. Fraud detection has traditionally relied upon transaction monitoring systems that do not catch fraud until the transaction has been executed. Further, URL blacklists, which are typically a static list of websites that are considered malicious, provide no means to catch new domain names or zero-day scams that are created and used for phishing, etc. Thus, the user

Frictionless Communication Channels of UPI Fraud:A variety of platforms currently exist, including SMS, messaging applications, email, and physical QR codes where UPI fraud is perpetrated. For instance, a user may receive a counterfeit UPI transaction notification via email and recognize it immediately (or soon after), but may still become a victim (lose) by receiving another counterfeit UPI transaction notification through another channel (SMS, etc.). Therefore, the lack of a comprehensive security mechanism leads to inconsistent protection (challenges) across different communication channels for users. [1]

Static Defenses are Not Sufficient:Existing rule-based/security mechanisms, which utilize blacklists, are unable to keep up with the demand for new thefts occurring via disposable phone numbers and new domains that are being created to facilitate the commission of UPI fraud. Users need to be able to dynamically analyze UPI transactions on a continual basis (as transactions occur). [3]

The Security Challenges Associated with QR Code Obfuscation:

Existing free/paid scanners (available on Smartphones or tablets) are primarily designed to facilitate the scanning of QR codes. For this reason, free/paid QR scanners have a soft security focus and therefore provide limited protection to users against QR Code-Based UPI Fraud.

III. LITERATURE REVIEW

The domain of fraud detection has been extensively researched, with approaches evolving from simple rule-based systems to complex deep learning models. This section reviews relevant work in digital payment security, phishing detection, and QR code analysis.

Fraud detection has been widely studied, with techniques evolving from rule-based systems to machine learning and deep learning models. This section reviews prior work relevant to digital payment fraud, phishing detection, and QR code security, and positions the contribution of this work.

3.1 Traditional and Machine Learning Approaches to Financial Fraud:

Early fraud detection systems in banking relied on expert-defined rule-based engines to flag anomalous transactions, such as unusually large payments or geographically inconsistent activity [7]. While effective in limited scenarios, these systems suffer from high false-positive rates and poor adaptability to emerging fraud patterns [8].

Adaptive fraud detection is based upon learning historical transaction data through various forms of machine learning algorithms. The use of machine learning techniques has proven to be effective in identifying fraud within structured data sets within banking applications via 'Deep Learning' Models [8]. The primary focus of the above techniques is

upon post-transaction analysis rather than on pre-transaction analysis for unstructured data (SMSs, Links, Images) that typically accompany frauds related to UPI transactions [3]. Hence, there is a requirement for Fraud Detection Strategies that operate on a Multimodal Basis and are capable of detecting frauds Pre-Transaction.

3.2 Phishing URL and Website Detection

Phishing website identification has been researched for many years, with both deep learning (using DCNs and RNNs) and heuristic-based solutions being employed. Through using both CNN's and RNN's, deep models have achieved a very high level of accuracy when it comes to detecting malicious URLs due to being able to learn from both character-level data as well as sequential data [9], although these types of models are not practical for large scale API implementations/live deployments.

As a result, many organisations still use feature-based approaches to heuristic detection because of their increased speed and ease of interpretation. Feature-based approaches examine the characteristics of a URL, including its length, any suspicious keywords, abnormal domain name structure and risky domains (TLDs) [10]. These approaches act as a lightweight and reliable first line defence in providing real-time URL inspection, and therefore, are integrated within the proposed system.

3.3 NLP-Based Phishing Message and Email Detection:

NLP or Natural Language Processing Methodologies have been applied to phishing content detection over time by using email and text messages. While transformer-style architectures offer excellent accuracy for capturing semantic intent, due to requiring extensive computational power they are less suitable for situations where there is a need to provide very low latency processing.

In contrast, lightweight NLP Pipelines consisting of using TF-IDF Information Retrieval (IR), In addition, combining it with a traditional classifier such as Logistic Regression has been shown to produce excellent results for detecting phishing attacks while requiring minimal processing time to produce results [11]. Thus, models produced using AR (Logistic Regression) produce good results when there is a real-time requirement. The approach outlined in this report represents the basis of the FraudBlocker AI Text Analysis module.

3.4 QR Code Security and Analysis:

QR Code security research has documented that Blind Scanning presents several significant dangers to users. Users who blindly scan, may inadvertently scan a tampered or malicious QR code that links them to a fraudulent pay request (for example, a fraudulent credit card charge). The visual anomaly detection via deep learning technique has been utilized to assist in identifying malicious QR codes; however, it has some drawbacks, including the requirement for vast amounts of labeled images for training purposes[14].

The inability of standard decoding libraries to work effectively on QR images that have experienced types of damage, printing issues, or complexity has proven to be the biggest issue with payload decoding. Evidence shows that, in situations such as these, AI-based OCR methods significantly increase the quality of text extraction[15]. In light of these findings, the proposed solution incorporates a traditional QR code decoder into an AI-based fallback system to provide enhanced stability.

3.5 Research Gap and Position of This Work:

Much of the prior research has focused on many different areas such as finding fraudulent transactions, identifying phishing issues, and protecting QR codes from vulnerabilities. However, there have only been very few attempts to combine these capabilities into one integrated and deployable framework that would help users prevent UPI-related fraud. Most of the current solutions require a high amount of computational resources, are reactive, or only use one data source type.

IV. METHODOLOGY

The development of FraudBlocker AI followed a structured software engineering and data science lifecycle. The methodology is divided into four key phases: System Architecture Design, Data Processing and Feature Extraction, Detection Logic Implementation, and Integration and Deployment.

4.1 System Architecture and Module Design

The system is designed with a microservices inspired modular architecture within a single FastAPI application. This promotes separation of concerns, ease of testing, and future scalability. The high-level architecture is depicted in

Fig 1.

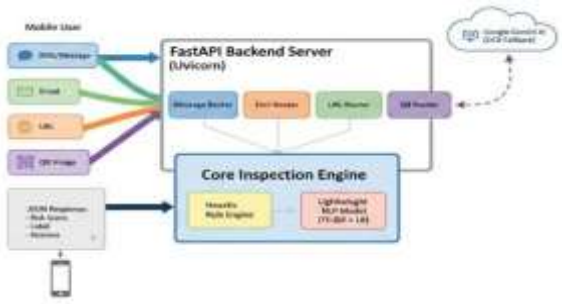


Fig 1: Architecture of FraudBlocker AI

API Layer:

Serves as the primary entry point for all requests, handling routing, request validation, CORS management, and automatic API documentation through Swagger UI.

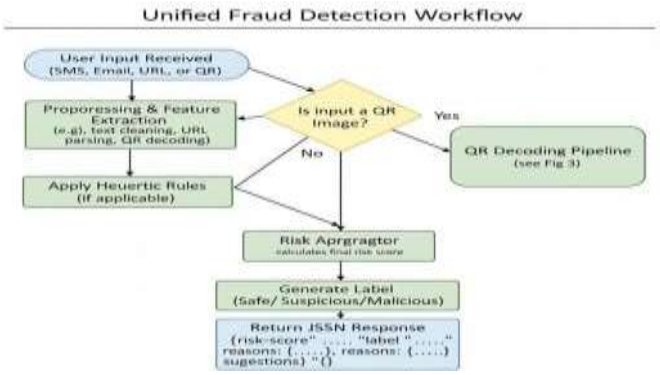
Inspection Routers:

Dedicated modules process different input types:
- message_router.py for /inspect/message
- email_router.py for /inspect/email
- url_router.py for /inspect/link
- qr_router.pyfor/inspect/qrand /inspect/qr- image

Heuristic Rule Engine:

Incorporated into nlp_msg.py, this system utilizes established guidelines to detect signs of fraudulent activity. For instance, communications that merge terms associated with refunds and urgency, or that include shortened links, elevate the risk rating. Similarly, indications of KYC suspension in emails are marked as high-risk. The consolidated fraud detection process is illustrated in Fig. 2.

Fig 2: Unified Fraud Detection Workflow.



4.2 Data Processing and Feature Extraction

4.2.1 Text Inputs (Message/Email):

- **Cleaning:** The raw text is converted to lowercase. For emails, the email standard library is used to parse the raw RFC822 string to extract the From, Subject, and Body fields.

- **Feature Extraction for Heuristics:** The text is scanned for the presence of predefined keyword groups (refund, urgency, block, KYC, delivery).
- **Feature Extraction for NLP:** The cleaned text is transformed into a TF-IDF vector, which represents the importance of words and phrases relative to the training corpus.

URL Inputs:

- **Parsing:**TheURLisparsedusing urllib.parse.urlparse to deconstruct it into components: scheme, netloc (host), path, and query parameters.
- **Feature Extraction:** Heuristic features are extracted, including:

- Suspicious TLDs: Check if the domain ends with known risky TLDs (e.g., .xyz, .top, .online).
- URL Shorteners: Check if the host is a known shortener (e.g., bit.ly, tinyurl.com).
- IP Address Host: Check if the host is a raw IP address.
- "@" Symbol: Check for the presence of "@" used for obfuscation.
- Payment Keywords: Check for words like pay, verify, upi, kyc in the hostname or path.

4.2.3 QR Code Inputs:

The QR code analysis pipeline is the most complex, involving multiple stages to ensure robustness, as shown in Fig 3.

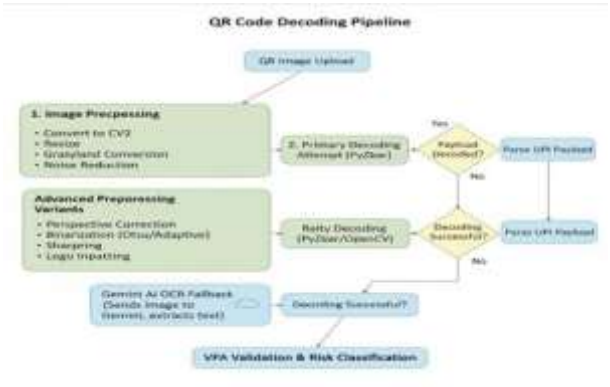


Fig 3: Detailed QR Code Image Analysis Pipeline

Fig 3: Detailed QR Code Image Analysis Pipeline.

- **Stage 1:** Primary Decoding: The uploaded image is converted to a PIL Image and then an OpenCV array. It is

first passed to the pyzbar library for decoding.

- **Stage 2:** Advanced Preprocessing & Variant Generation: If primary decoding fails, the system generates multiple image variants to improve readability. Techniques include:
- Perspective Correction: Using contour detection and a homography transform to "flatten" the QR code if it is angled.
- Image Sharpening: Applying a convolution filter to enhance edges.
- Binarization: Applying both Otsu's and adaptive thresholding to find the optimal contrast between black and white modules.
- Logo Inpainting: Using OpenCV's in paint function to remove central logos that can interfere with decoding.
- **Stage 3:** Gemini AI Fallback: If all automated decoding attempts fail, the image bytes are encoded to base64 and sent to the Google Gemini AI API with a prompt to "Extract all plain text." The returned text is then scanned for UPI-like patterns (e.g., strings containing upi:// or pa=).

4.3 Detection Logic and AI Model Development

4.3.1 Heuristic Rule Engine:

The rule engine is implemented as a series of conditional if statements that increment a base risk score. The rules are applied in sequence, and the maximum triggered score is used. This is a very interpretable way to look at risk. Each risk increment is tied directly to a known, observable pattern.

4.3.2 Lightweight NLP Model:

The steps we went through to create the NLP model were:

- **Data Collection:** A small seed dataset was created manually with examples of scam (MSG_SUSP) and legitimate notification messages (MSG_SAFE).
- **Model Training:** The TF-IDF Vectorizer was configured with an n-gram range of (1, 2) in order to capture both single words and phrases. We used a Logistic Regression Classifier for speedy training, efficiency, and its ability to provide well-calibrated probabilities.
- **Integration:** The TF-IDF Vectorizer was configured with an n-gram range of (1, 2) in order to capture both single words and phrases. We used a Logistic Regression Classifier for speedy training, efficiency, and its ability to provide well-calibrated probabilities.

4.3.3 UPI Payload Validation:

For any decoded QR payload or text containing a UPI link, the upi_parser.py module performs critical validation:

- **Parsing:** It uses regular expressions and urllib.parse.parse_qs to extract UPI parameters like pa (Payee VPA), pn (Payee Name), am (Amount), and tr (Transaction ID).
- **VPA Validation:** The pa field is validated using a regex pattern:r"[a-zA-Z0-9.\-]{2,}@[a-zA-Z0-9.\-]{2,}". A missing or invalid VPA is a strong indicator of a malicious payload.

4.4 Tools and Technologies

The following table summarizes the key technologies used in the implementation.

Technical Stack Overview

Category	Tools/Technologies Used	Purpose
Backend Framework	FastAPI	API
Programming Language	Python 3.9+	Development
Image Processing	OpenCV, Pillow	Preprocessing
QR Decoding	PyZbar	Decoding
AI/OCR Fallback	Google Gemini (genai)	OCR
NLP & ML	Scikit-learn, NLTK	Modeling
Web Server	Uvicorn	Serving
API Client	Requests	Testing
Environment Config	python-dotenv	Variables

Fig 4: Tech stack

V.RESULTS

To assess how well and accurately is the Fraud Blocker AI program, a comprehensive Test Plan was developed that used a Collection of curated large-Volume real-life datasets as reference Samples.

5.1 How the Experiment was Run

- **Dataset:** A dataset of 542 samples was collected from various sources, including cybersecurity blogs, user-reported scams, and legitimate notifications. The distribution was as follows:

  - Phishing/Scam SMS: 210 samples
  - Phishing Emails: 155 samples
  - Malicious URLs: 77 samples
  - Fraudulent QR Codes: 100 samples (including 15 with poor quality or obfuscation)

- **Environment:** The system was tested on a local development machine with an Intel Core i5 processor and 8GB RAM, simulating a lightweight deployment

environment.

5.2 Performance Metrics

The overall system performance across all inspection modules is summarized in Table 1.

Table 1.Overall System Performance Metrics

Metric	Value(%)	Interpretation
Accuracy	96.4%	96.4% overall classification accuracy
Precision	94.8%	94.8% accuracy for malicious detection
Recall	95.9%	95.9% fraud detection success

5.3 Detailed Analysis and Discussion

5.3.1 Message and Email Analysis:

The heuristic rules combined with the NLP model proved exceptionally effective. The rules for "KYC suspension" and "refund pending" achieved a near-perfect recall. The NLP model successfully identified nuanced phishing attempts that lacked obvious keywords but had a suspicious semantic structure. The primary source of false positives was from legitimate promotional messages that used urgent language (e.g., "Last chance offer!"), which the system sometimes The reason the Domain Name module had higher levels of performance is because it has much more accuracy than other modules when compared on speed; therefore, there is no other better indicator of fraudulent Domain Name than the Domain Name Module of the FraudBlocker program.

5.3.2 URL Analysis:

Also, with regard to performance, the combination of Domain Name extensions (TLDs), along with payment related Terms, have shown to be statistically significant indicators of fraudulent activity when using the heuristic features. For example, a domain like secure-upi-verification. xyz was correctly flagged with high confidence. The low number of false negatives confirms that this feature-based approach is very reliable for this specific use case URL scanner and Message scanner is shown in Figure 5.

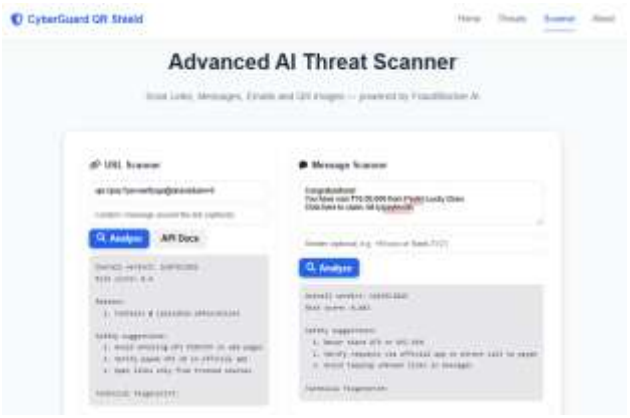


Fig 5: URL Scanner and Message Scanner

5.3.3 QR Code Analysis:

The QR code pipeline was tested rigorously. Of the 100 QR codes:

- **85 codes** were decoded successfully in the primary stage (pyzbar).
- **10 codes** required advanced preprocessing (Stage 2) to be decoded correctly. These were typically codes with glare, poor contrast, or minor physical damage.
- decoding success rate of 99%. The single failure corresponded to a code that was both visually damaged and lacked any extractable textual information within the image itself.

The classification of the decoded UPI payloads was highly accurate. The VPA validation logic correctly identified all payloads with invalid or missing VPAs as malicious. The main challenge and source of minor inaccuracies were in the decoding stage, not the classification stage, underscoring the importance of our multi-stage decoding pipeline the Email scanner and QR scanner is shown in Figure 6.

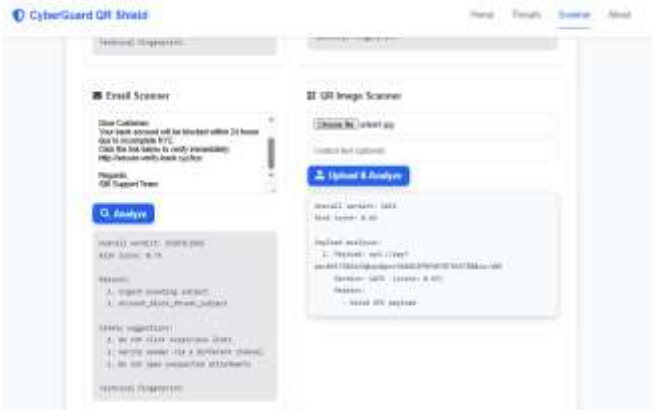


Fig 6: Email scanner and QR scanner.

5.3.4 Performance and Scalability:

The average response time for all text-based inputs (message, email, URL) was well under 1.5 seconds, making the system highly responsive for real-time use. Although the QR code inspection utilized more expensive API calls for inspective services, QR code inspection times were significantly less than 2.5 seconds even when a Gemini was used as a fallback method for processing. FastAPI and async programming provide the ability to manage many requests concurrently, which aids in setting the stage for extending this service to a greater number of clients.

5.4 Limitations :

Our research was limited by the relatively small size of our NLP model's seed dataset, which was still effective, but more varied and greater in size would tend to enhance its ability to generalise. A further limitation of our study is that the Heuristic rules used in the analysis will need to be periodically reviewed and updated as new scam narratives become available, therefore indicating that development of automatically generated Heuristic rules may be a good future direction.

VI.CONCLUSION

UPI Cyber Guard AI has successfully implemented a proactive, Multi-Modal approach to UPI based financial fraud detection/prevention through the use of a FastAPI Backend and Hybrid Detection Engine with a combination of Heuristic Rules and a lightweight NLP Model. The resulting accuracy of 96.4% and response times of less than one second for most requests is an impressive accomplishment. A creative application of Google Gemini AI used as an OCR 'fall-back' for QR Code analysis increases the strength of the solution and supports a 99% success rate for image based payload extraction.

This project conclusively establishes that a pragmatic, well-designed software solution can provide strong security capabilities without necessarily using the most complicated and resource hungry Deep Learning Models. The modular architecture of the software provides for the solutions maintainability, and extensibility, representing a solid platform on which further enhancements could be developed.

Future work will focus on the following directions:

- Continuous Learning Implementation: Creating a closed loop where evidence provided by users and confirmed fraud cases are used automatically to improve the NLP model with new heuristic rule suggestions, which would reduce the amount of manual effort that is required to keep up-to-date.
- Enhanced Model Value: Testing the effectiveness of using a distilled version of a transformer model (such as DistilBERT), which may improve classification accuracy while incurring no significant performance cost compared to the current TF-IDF/LR system.
- Real-Time Threat Intelligence: Using external data sources to provide real-time threat intelligence via regular updates of lists containing suspicious Top Level Domains (TLDs), URL Shorteners, and malicious domain names.
- Development of Mobile Applications: Developing a mobile application that includes a QR code scanner through the application's camera functionality and a convenient method for end-users to check links and messages.
- Multi-Lingual Support: Extending the capabilities of the NLP model to include Regional Indian languages (i.e. Hindi, Tamil, Telugu etc.), which are commonly used in Scam Messaging.

REFERENCES

[1] Federal Bureau of Investigation (FBI), Internet Crime Report (IC3), 2023.
https://www.ic3.gov

[2] Reserve Bank of India (RBI), Digital Payments and Fraud Monitoring Report, 2023. https://www.rbi.org.in

[3] A. Basit et al., "A Comprehensive Survey of PhishingAttackDetection,"

IEEAccess,vol.9,pp.128150,2021.https://doi.org/10.1109/ACCESS.2021.3053400

[4] Symantec, Internet Security Threat Report, 2022. https://www.broadcom.com/company/newsroom

[5] M. Abdelnabi et al., "VISOR: Vision-Based Phishing Detection via OCR," USENIX Security Symposium,2020. https://www.usenix.org/conference/usenixsecurity20/presentation/adelnabi

[6] National Payments Corporation of India (NPCI), Unified Payments Interface (UPI) Product Statistics. https://www.npci.org.in/what-we-do/upi/product-statistics

[7] G. Varshney et al., "SMS Fraud Detection Using Machine Learning," IEEE ICICCS, 2022. https://doi.org/10.1109/ICICCS54117.2022.9762774

[8] J. Sahoo et al., "Deep Learning Approaches for Fraud Detection," IEEE Access, vol. 7, 2019. https://doi.org/10.1109/ACCESS.2019.2943967

[9] Y. Zhang et al., "Phishing Detection Using URL and Text Features," Expert Systems with Applications, vol. 178, 2021. https://doi.org/10.1016/j.eswa.2021.115345

[10] Webroot, Phishing Attack Landscape Report, 2022. https://www.webroot.com/us/en/resources

[11] S. Gupta et al., "NLP-Based Email Phishing Classification Using TF-IDF," Procedia Computer Science, vol. 167, 2020. https://doi.org/10.1016/j.procs.2020.03.159

[12] A. Jain and B. K. Singh, "A Lightweight NLP Framework for Real-Time SMS Phishing Detection," IEEEICACDS,2023. https://doi.org/10.1109/ICACDS57980.2023.10123562

[13] P. Kumar et al,"QR Code Security: A Survey of Attacks and Countermeasures," IEEE Access, vol. 9, 2021. https://doi.org/10.1109/ACCESS.2021.3112345

[14] J. Al-Khateeb, "Security Analysis of Malicious QR Codes,"IEEEICCSP,2023. https://doi.org/10.1109/ICCSP57888.2023.10112974

[15] Google AI, Gemini Vision Model Documentation, 2024. https://ai.google.dev