iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70866

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Threat Guard AI: A Multi-Domain System for Intelligent Fraud Detection Using Machine Learning

Pranjal Bhinge[1], Ranjeeta Chikkaparappa[2], Shruti Basarikatti[3], Rohan Pawar[4]
Department of Computer Science and Engineering, S.G. Balekundari Institute of Technology, Karnataka, India

Abstract: ***In the era of rapid digital transformation, cyber threats such as phishing, QR code manipulation, UPI fraud, and PDF malware have become increasingly sophisticated. Threat Guard AI is a comprehensive, multi-domain fraud detection system designed to counter these threats using artificial intelligence and machine learning. The platform integrates specialized modules to analyze UPI transactions, QR codes, PDF documents, and phishing links. Through a user-centric Streamlit interface and a robust Flask backend, it provides real-time analysis, prediction, and alerts. Each module is powered by trained models using Scikit-learn and OpenCV, with MongoDB for scalable data storage. Threat Guard AI not only enhances the precision of fraud detection but also ensures actionable insights, unified visualization, and extensibility to new threat domains. This paper presents the architecture, implementation methodology, and system performance analysis of Threat Guard AI, demonstrating its capability to provide intelligent, real-time digital threat mitigation.***

Keywords: Fraud Detection, UPI Scam Prevention, QR Code Security, PDF Malware Analysis, Phishing Detection, Machine Learning, Artificial Intelligence, Cybersecurity, Streamlit Dashboard, Flask API, MongoDB Database, Multi-Domain Detection, Real-Time Threat Analysis, AI Security Platform, Secure Digital Transactions

I. INTRODUCTION



Figure 1.1 Modules of Threat Guard AI

In today's rapidly evolving digital landscape, the use of online platforms for commerce, financial transactions, and communication has become indispensable. However, this shift has simultaneously elevated the risks of cyber threats and financial frauds, which are increasingly sophisticated and difficult to detect.

©IJRASET: All Rights are Reserved

The rapid proliferation of Application Programming Interfaces (APIs), digital wallets, and UPI-based payments has provided cybercriminals with new avenues to exploit vulnerabilities through methods such as fake QR codes, malware-infected files, and socially engineered phishing attacks.

Traditional security systems often address threats in isolation and are primarily reactive, detecting malicious activity only after an attack has occurred. As these threats diversify across multiple platforms and formats, the need for an integrated, proactive detection system has become more urgent than ever.

Threat Guard AI emerges as a strategic response to these modern threats. It is a multi-domain fraud detection system that leverages artificial intelligence (AI) and machine learning (ML) techniques to detect fraudulent activities across four major threat vectors:

- UPI fraud: Unauthorized or suspicious digital payment transactions.
- QR code fraud: Maliciously crafted QR codes redirecting users to harmful domains.
- PDF malware: Documents embedded with malware capable of compromising system integrity.
- Phishing attacks: Deceptive communication attempts aimed at extracting sensitive user credentials.

The project aims to bring these threat detection capabilities into a unified platform that not only identifies suspicious behavior but also delivers real-time alerts and actionable insights to users through a simple, accessible dashboard. At the core of this platform is a Streamlit-powered user interface that integrates seamlessly with a Flask-based backend. The backend coordinates the execution of trained ML models, manages data flows, and interacts with a scalable MongoDB database for secure and flexible storage of user data and results.

What sets Threat Guard AI apart is its design emphasis on scalability, extensibility, and real-time interaction. By using libraries such as OpenCV and pyzbar, the system can process visual data including QR codes via both image uploads and live webcam feeds. PDF malware detection is facilitated by examining structural and metadata features of uploaded documents, while phishing URLs are analyzed using heuristic and ML-based models for real-time classification.

The fraud detection models are primarily developed using the Scikit-learn framework, with methods like Random Forest Classifier for transaction-based anomaly detection. These models are trained using historical fraud data and evaluated on accuracy, recall, and confusion matrix metrics to ensure performance robustness. For data preprocessing and model training, libraries such as pandas and numpy are used, and joblib is employed to serialize trained models for deployment.

The result is a responsive and highly interpretable dashboard that not only detects threats but explains the rationale behind its classifications, empowering both end-users and security professionals to respond effectively.

Through its comprehensive architecture and multi-layered design, Threat Guard AI addresses a critical need in modern cybersecurity. It bridges the gap between domain-specific threat detection and integrated security intelligence, providing a holistic defense platform against the emerging spectrum of online fraud.

II. SYSTEM DESIGN & ARCHITECTURE

A. Scope of Project

The focus of this project includes the overall design, development, and deployment of a multi-domain system for fraud detection. The system is intended to utilize machine learning and artificial intelligence (AI) methods to offer a powerful and integrated platform. The main objective of the project is to adequately solve and prevent different types of online fraud, which have been rapidly growing in number and complexity. The system is intended to target the following types of fraud:

- UPI fraud: Illegal activities performed using the Unified Payments Interface, including unauthorized transactions and tampering with payment procedures.
- QR code fraud: Misleading practices that involve the utilization of malicious or tampered QR codes to mislead users and trigger fraudulent processes.
- PDF malware detection: Detection and prevention of malicious software contained in Portable Document Format (PDF) files, which can weaken system security.
- Phishing attacks: Identification and blocking of fraud attempts to acquire sensitive data like passwords, usernames, and credit card numbers by pretending to be a legitimate entity via electronic communication.

B. Core Components and Features

1) Frontend Development

The frontend is developed using the Streamlit library, chosen for its simplicity in creating interactive Python-based web applications. It enables the fraud detection system to be easily usable and navigable by users. The frontend includes specialized modules for each type of fraud—UPI, QR, PDF, and phishing. A combined dashboard acts as a central console to visualize prediction outcomes, helping users make fast and informed decisions.

2) Backend Development

A Flask API server receives requests from the frontend and processes them. Flask is selected due to its lightweight and scalable architecture. Backend logic integrates fraud module processing and interacts with MongoDB and Firebase, which stores user credentials, input logs, and prediction results.

3) Database Management

A NoSQL MongoDB database is utilized for storing various types of structured and unstructured data. This includes UPI transaction records, PDF content metadata, QR decode outputs, and phishing URL logs. MongoDB's flexibility supports scalable growth and performance.

III. PROPOSED METHODOLOGY

Threat Guard AI begins with a comprehensive requirement analysis to identify both functional and non-functional needs.

A. Functional Requirements

- Secure data ingestion from varied sources (image, text, document).
- Multi-layered detection for UPI, QR, PDF, and phishing.
- Real-time threat monitoring.
- Dashboard and logging system with audit trails.

B. Non-Functional Requirements

- Performance: Minimal latency for predictions.
- Scalability: System must handle high volume data inputs.
- Security: Strong encryption and access control for sensitive data.

The system addresses digital transaction vulnerabilities such as financial fraud, malware dissemination, and phishing schemes using advanced machine learning and AI techniques.

C. Multi-Layered Detection Framework

At the core lies the implementation of multi-layered detection mechanisms combining:

- Rule-based logic for detecting known patterns.
- Anomaly detection using ML for uncovering unknown threats.
- Advanced techniques like:
  - Predictive modelling
  - Image analysis for QR code and PDF scanning
  - Natural language processing (NLP) for phishing URLs.

Random Forest classifiers are widely used across modules for their accuracy and interpretability.

IV. IMPLEMENTATION

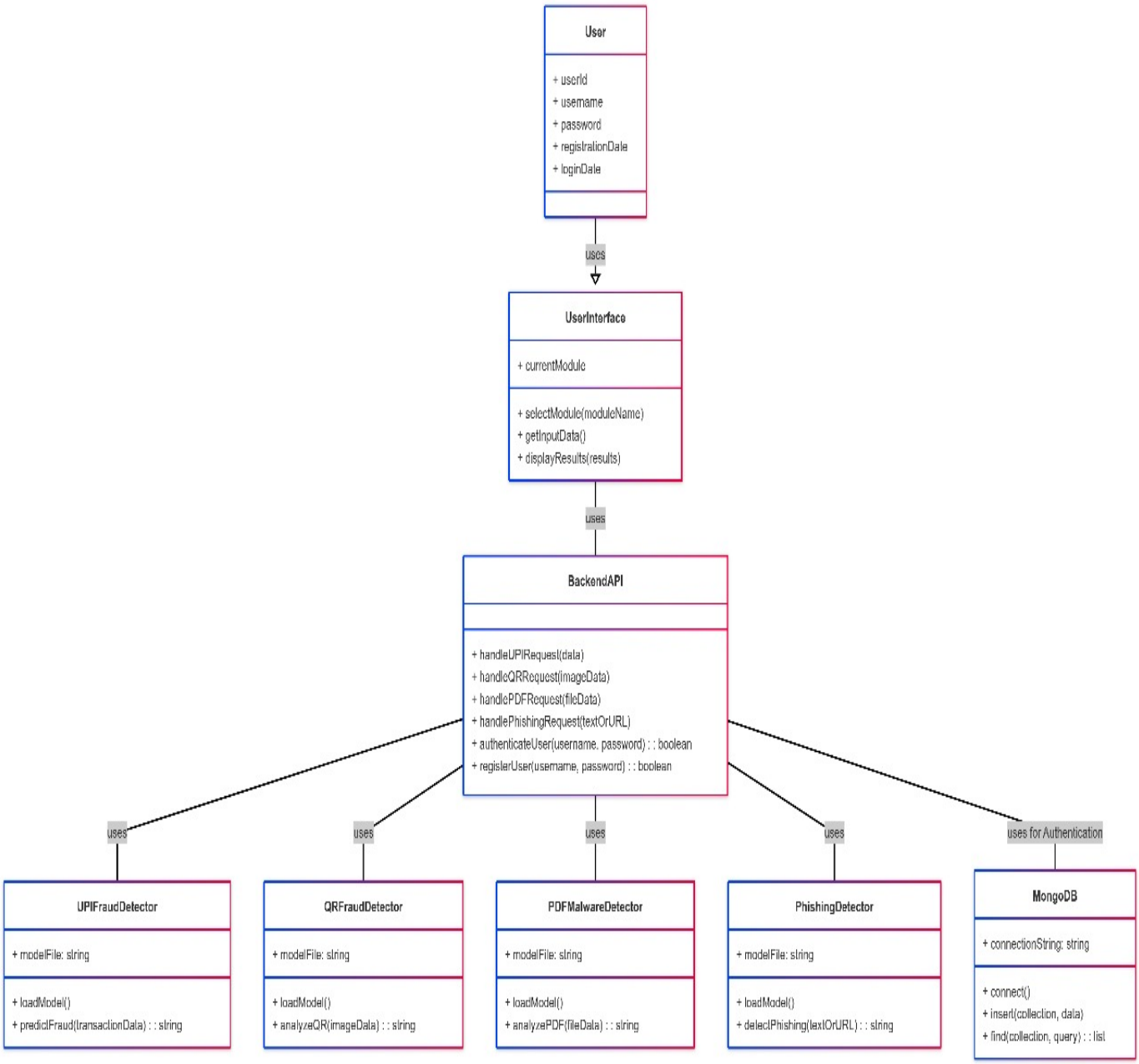


Figure 1.2 Class Diagram

A. Class Diagram

The class diagram represents the structure of the system in terms of class objects and their interactions:

- User: Handles login credentials and access history.
- UserInterface: Manages user interaction via modules.
- BackendAPI: Routes requests to relevant detection classes.
- UPIFraudDetector, QRFraudDetector, PDFMalwareDetector, and PhishingDetector: Individual module handlers.
- MongoDB: Stores system logs, results, and user actions.

B. Activity Diagram

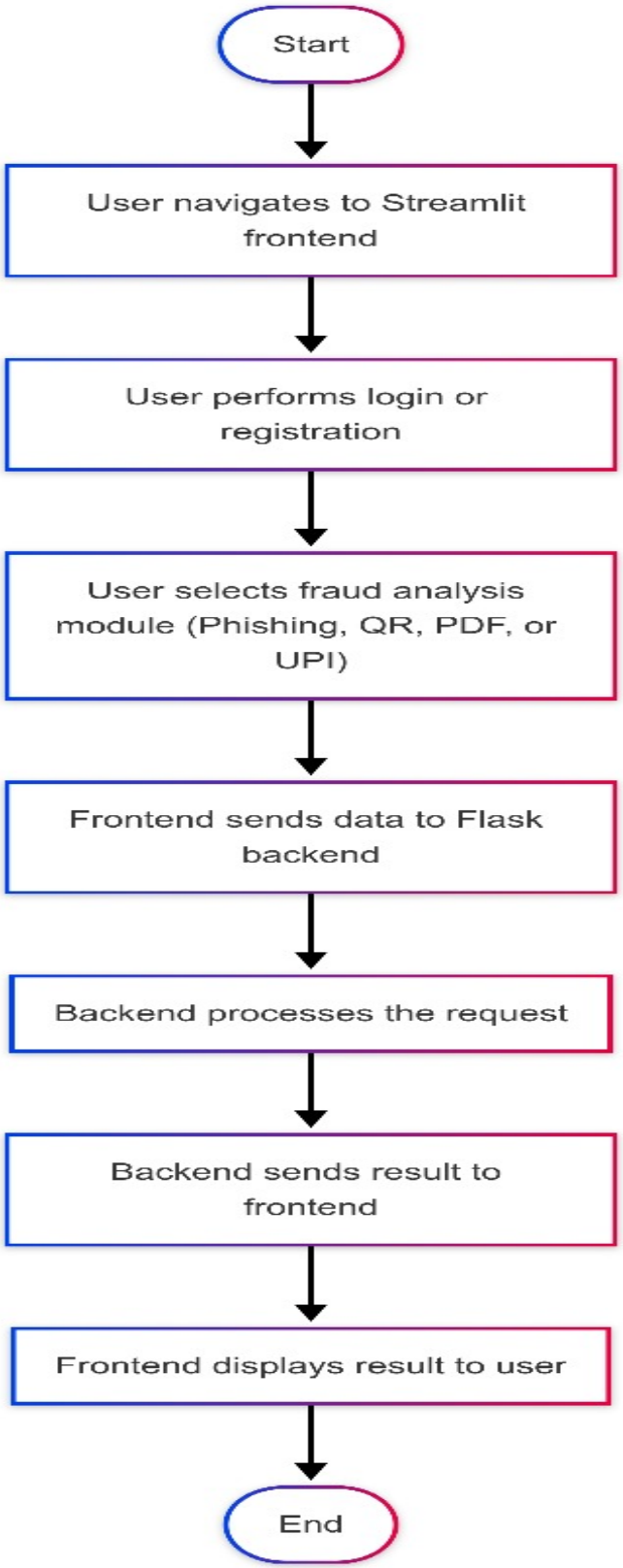


Figure 1.3 Activity Diagram

V. TEST CASES

Below are selected test scenarios validating each module:

Test Case	Input Details	Expected Output	Pass/Fail
User Registration	Email, Password, UPI ID	Data saved in Firebase + MongoDB	☑ Pass
Malicious PDF Upload	ransomware.pdf	Red alert: Detected ransomware via entropy & MIME type	☑ Pass
Malicious QR via Upload	QR	Red alert: Phishing tags, suspicious domain	☑ Pass
UPI Fraud Detection	Suspicious ID + Large Amount	"Fraud Detected" (90% confidence)	☑ Pass
Phishing URL	http://fakemail-login.ru	Red alert, Risk Score 85	☑ Pass
Admin Fraud Filter	Amount range 10,000-50,000 INR	Filtered results displayed	☑ Pass
Model Retraining	Button click	New model trained, accuracy updated	☑ Pass

VI. RESULTS AND ANALYSIS

A. Login and Home Page Design:

Before home page , there is sign in and sign up page.

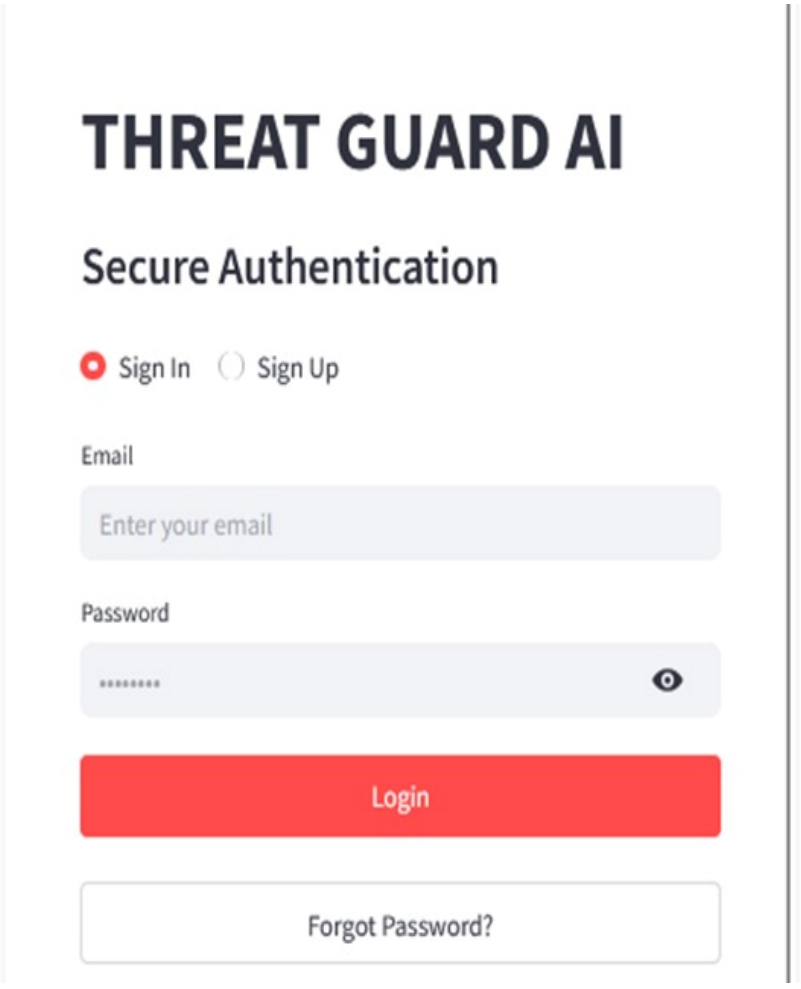


Figure 1.4 Sign-in/Sign-up Page

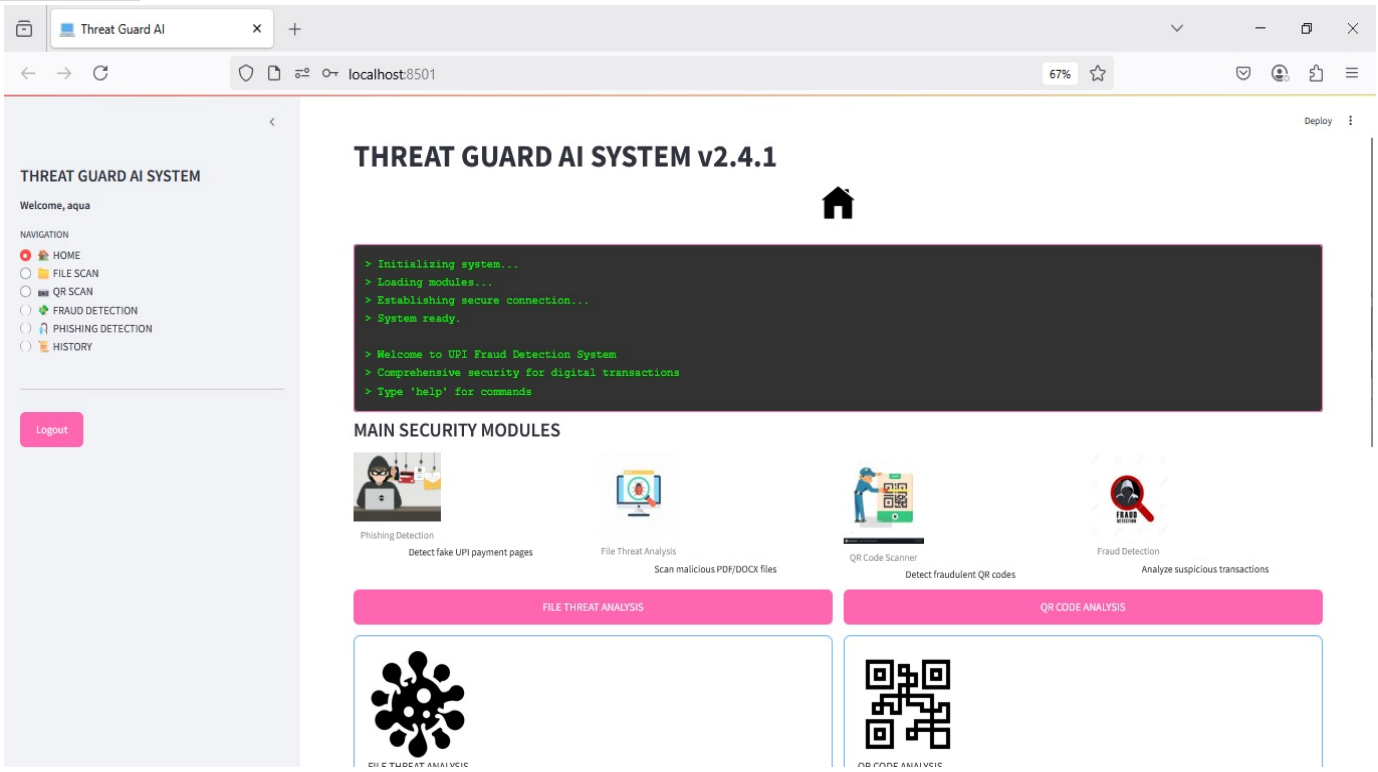


Figure 1.5 Home page with Header

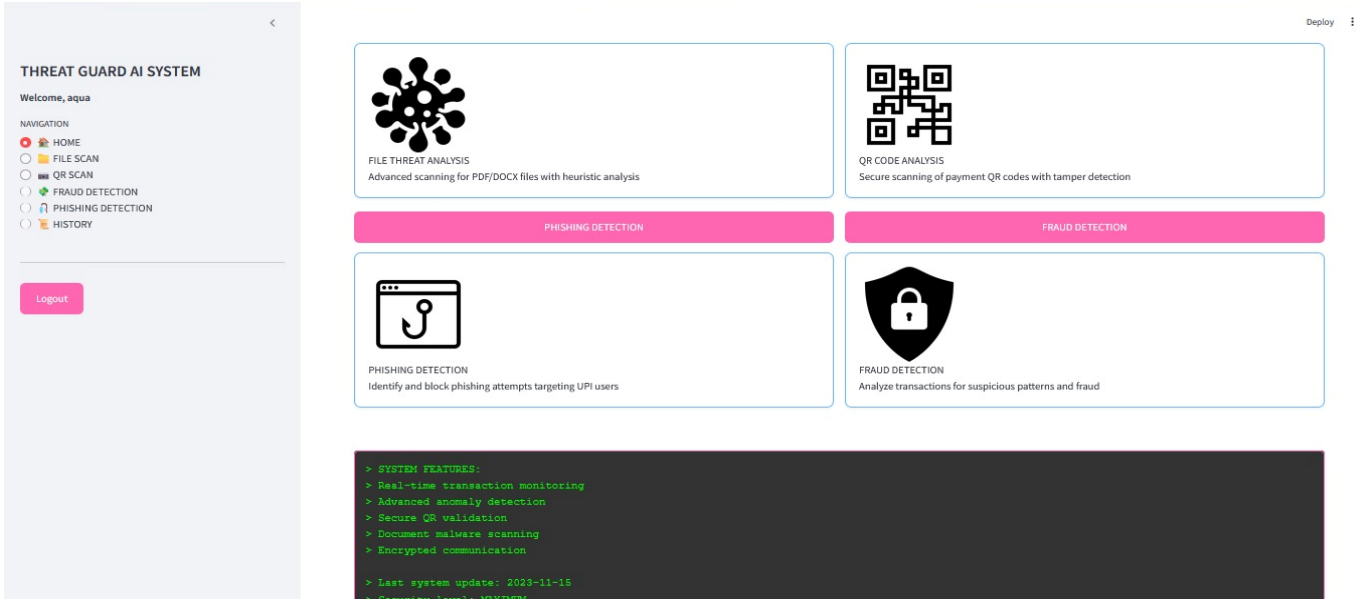


Figure 1.6 Home page with Footer

The home page welcomes users with a command-line style terminal interface, showing real-time system status in monospace font. A glowing digital shield pulses at the center to represent system defense status. Four interactive modules are arranged in a grid:

- File Threat Analysis
- QR Code Scanner
- Fraud Detection
- Phishing Protection

B. UPI Fraud Detection Module

This module analyzes transaction parameters such as sender UPI ID, phone number, shop name, transaction amount, frequency, and location. The input is analyzed against a trained machine learning model to predict whether the transaction is fraudulent.

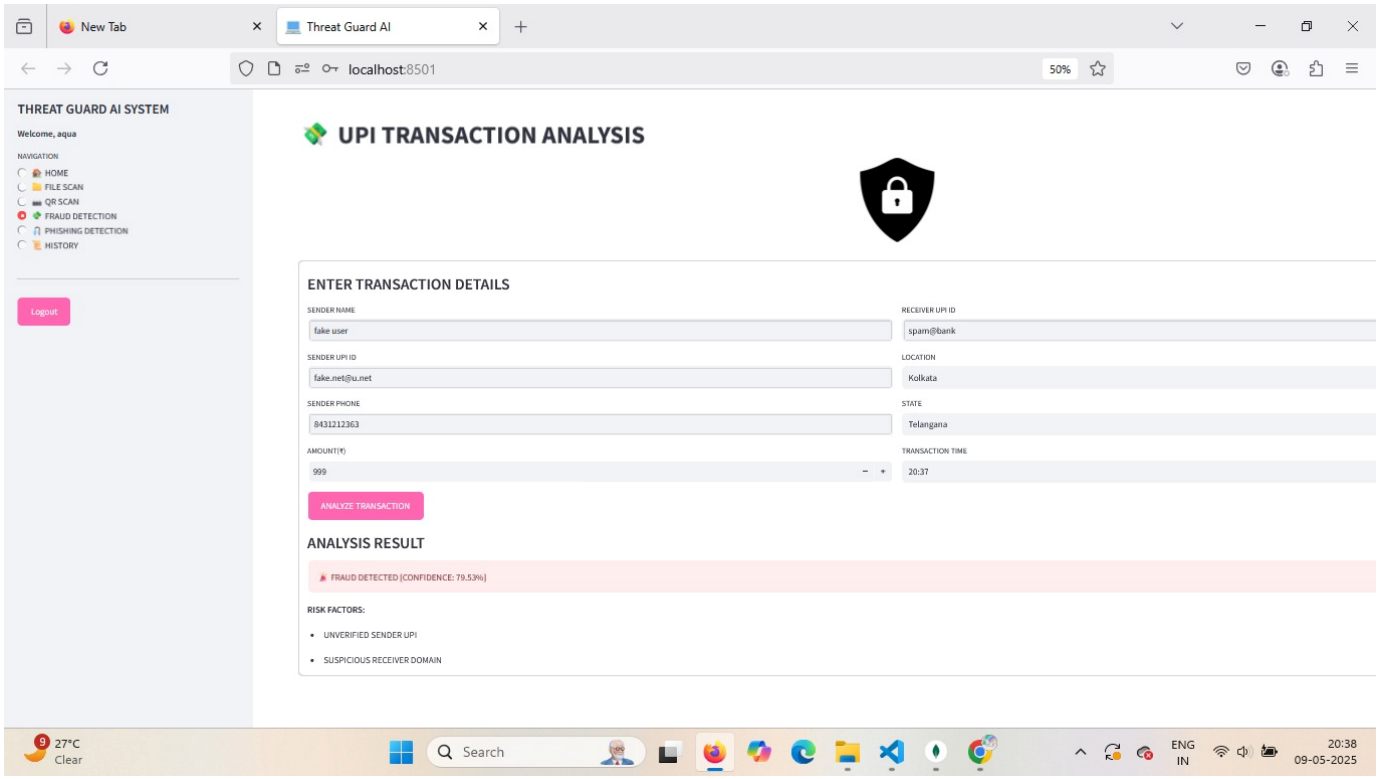


Figure 1.7 UPI Fraud Detection - Unsafe

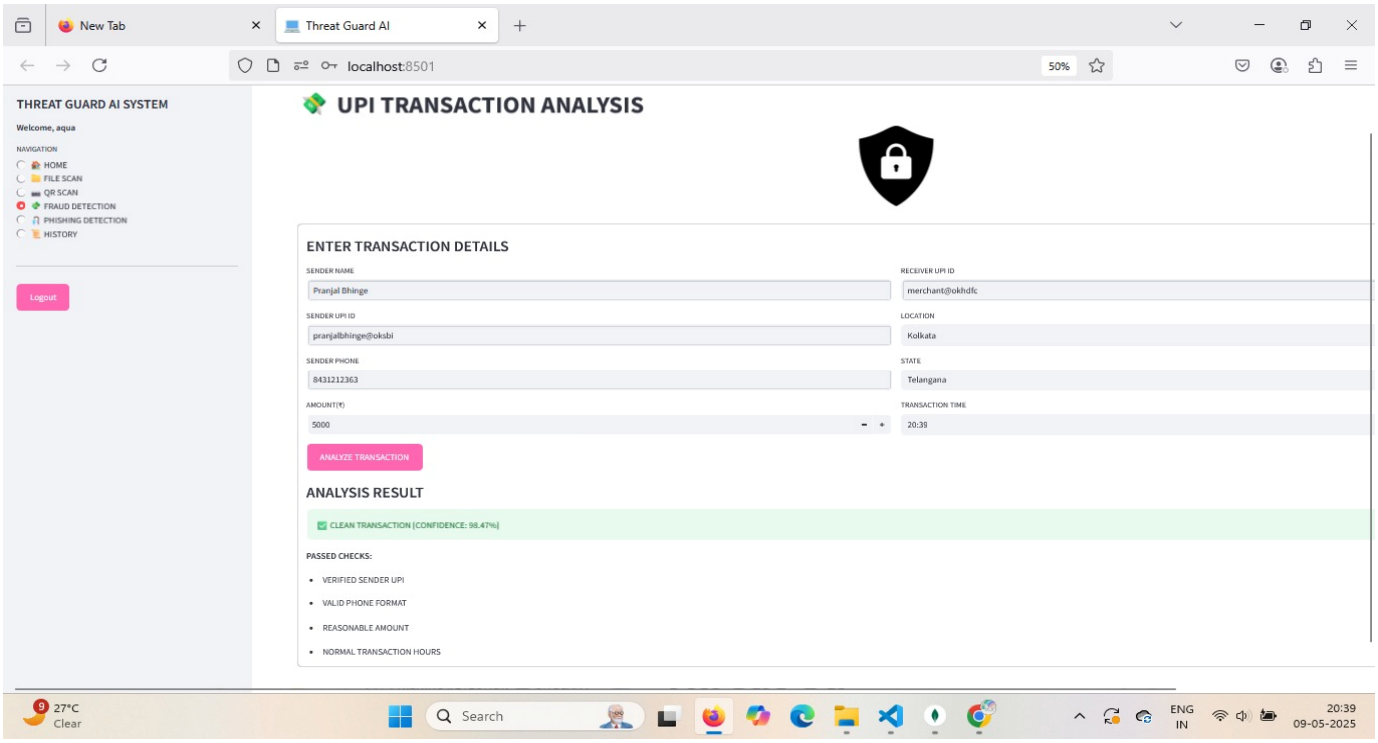


Figure 1.8 UPI Fraud Detection – Safe

C. QR Code Fraud Detection Module

In this module, users can upload an image of a QR code. The system decodes the QR using OpenCV and checks for indicators of tampering, redirection, or links to suspicious domains.

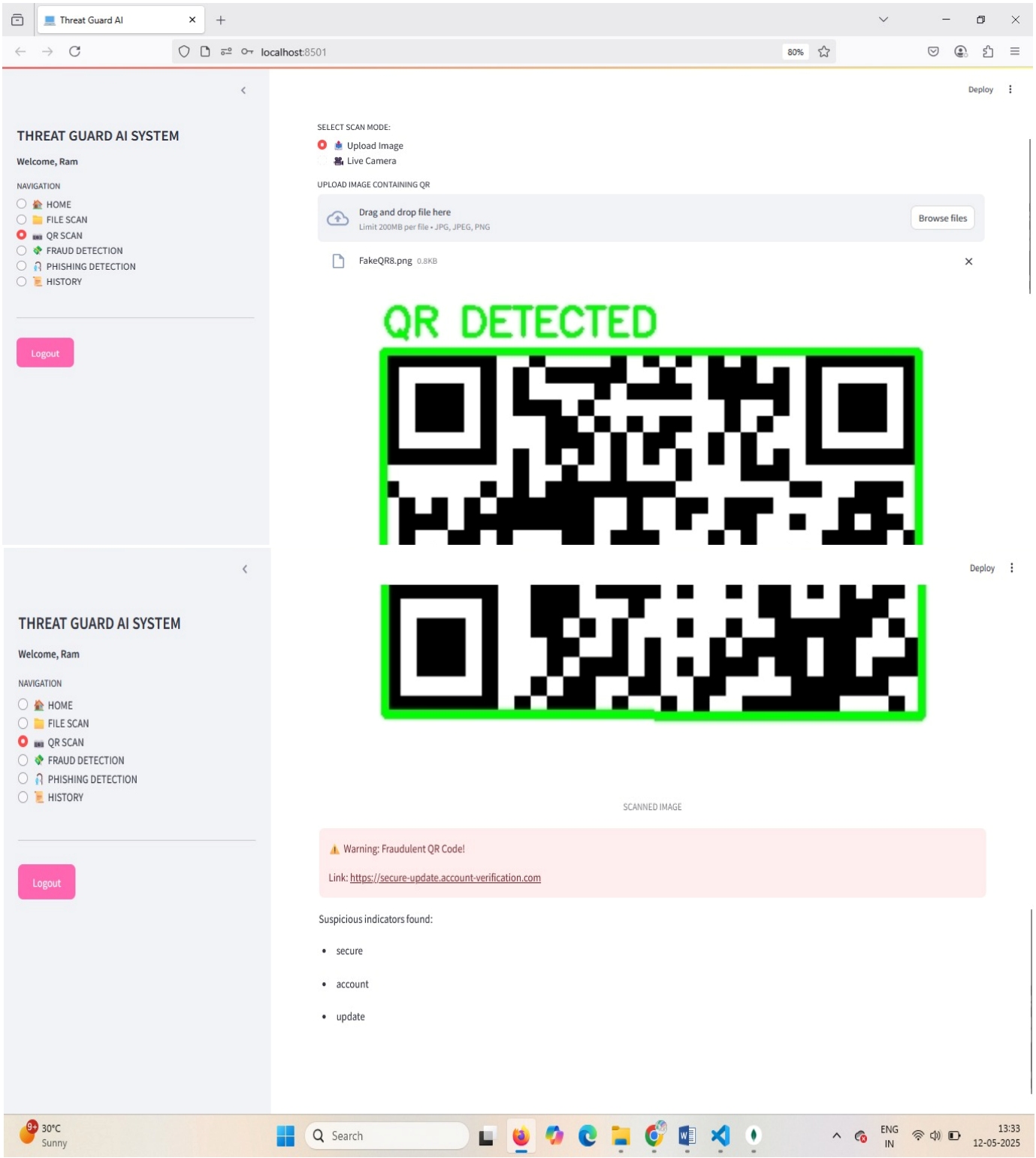


Figure 1.9QR Fraud Detection - Unsafe

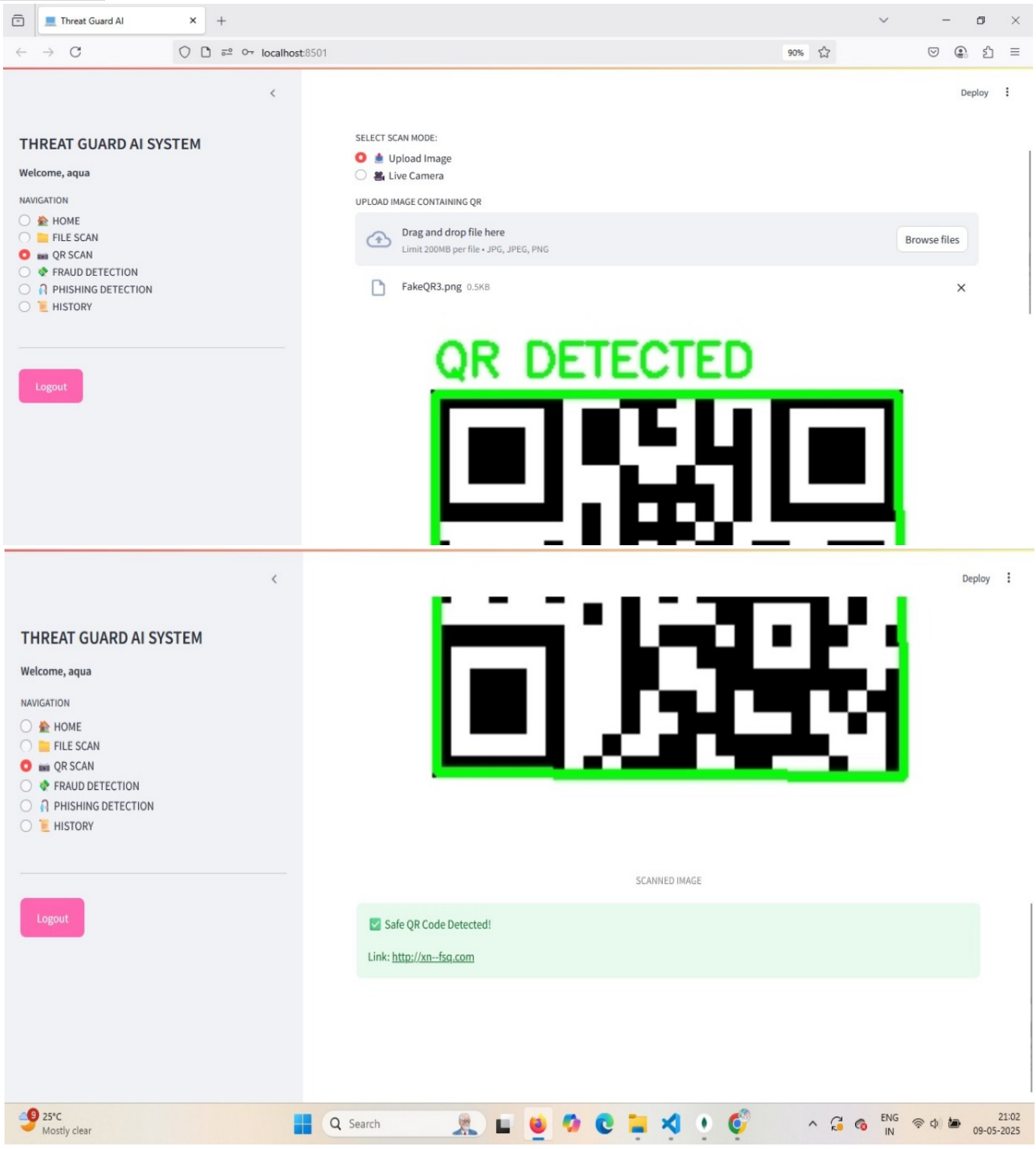


Figure 1.10QR Fraud Detection - Safe

D. PDF Malware Detection Module

Users can upload a .pdf file, which is scanned using static malware analysis features such as embedded JavaScript, suspicious metadata, and encryption flags. The backend model predicts whether the file is malicious.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

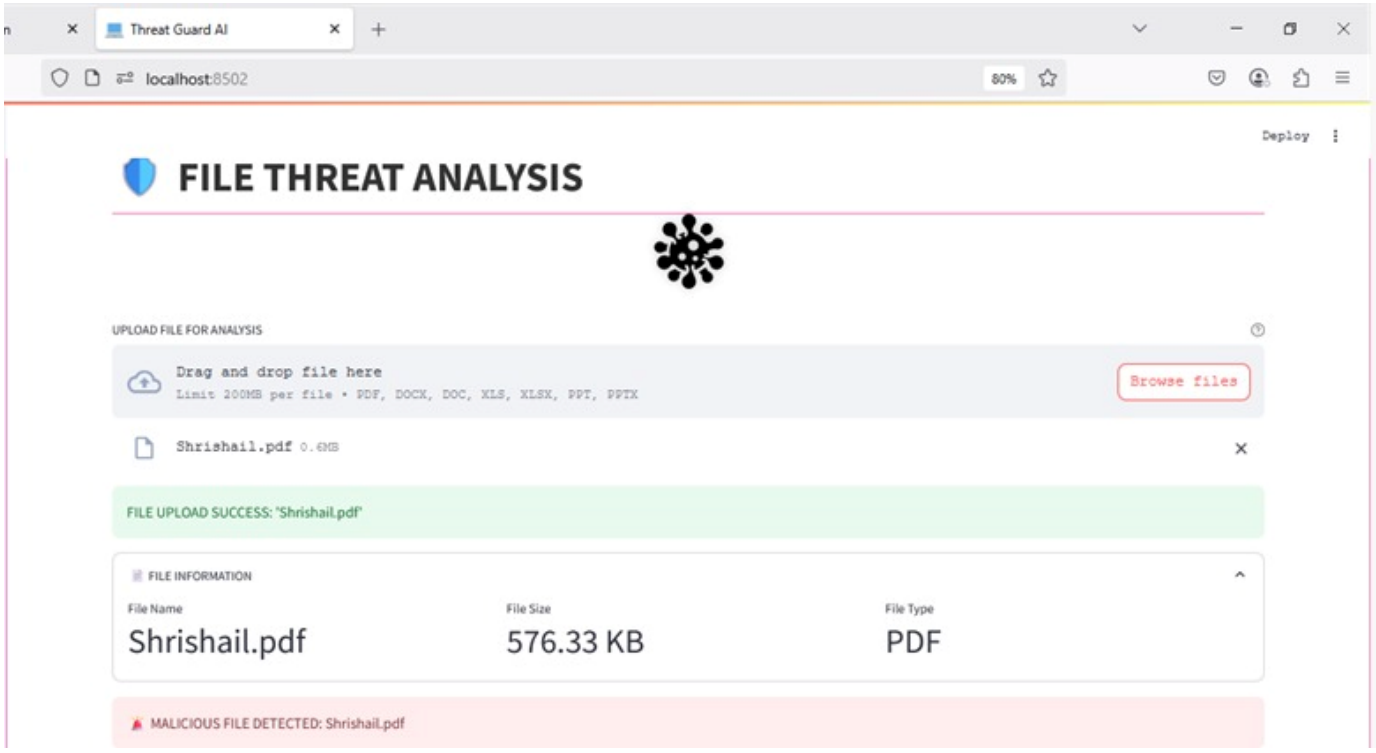


Figure 1.11 PDF Malware Detection – Unsafe

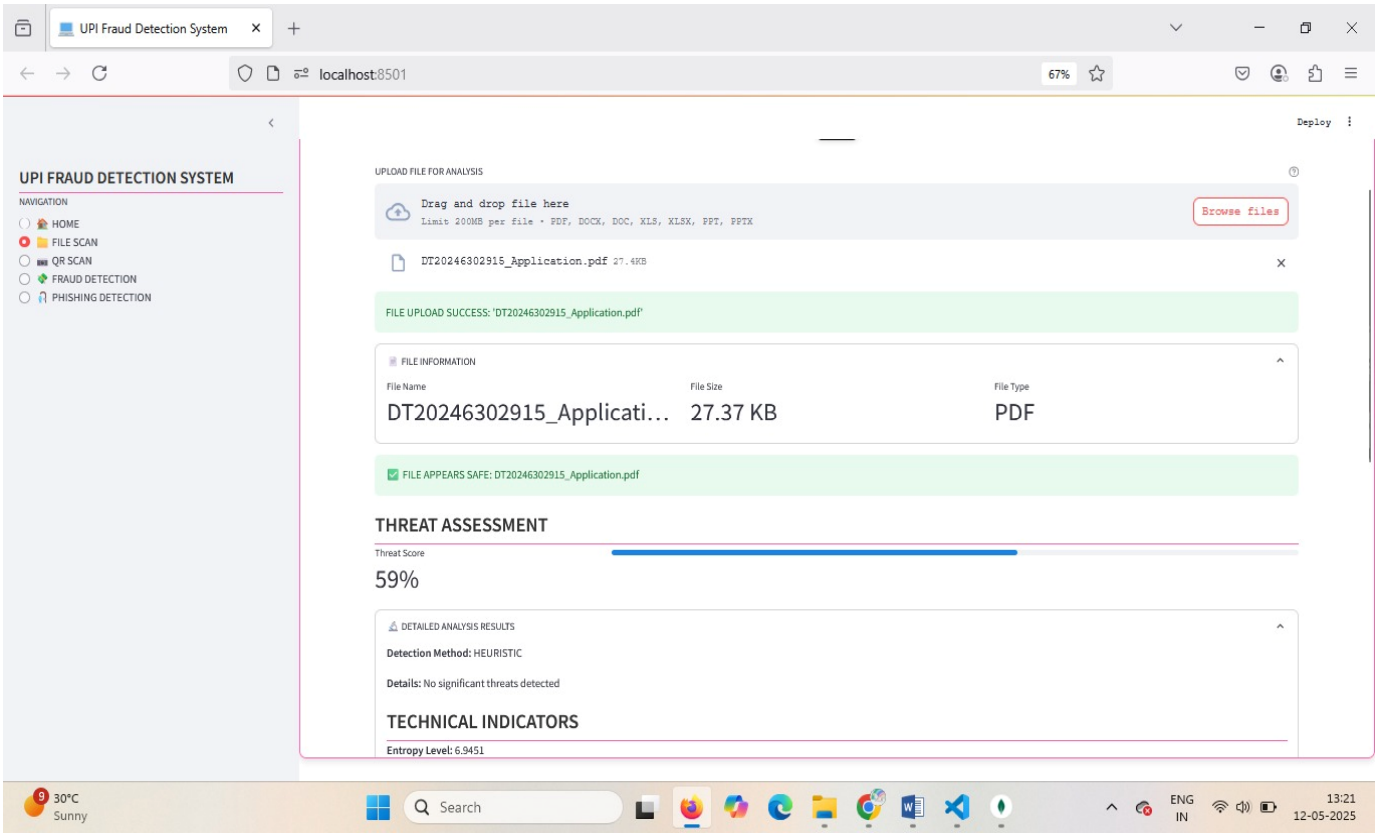


Figure 1.12 PDF Malware Detection –Safe

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

E. Phishing Link Detection Module

This module allows users to input a URL. The system extracts features such as presence of '@' symbols, subdomains, URL length, and domain age, feeding them into a model to detect phishing attempts

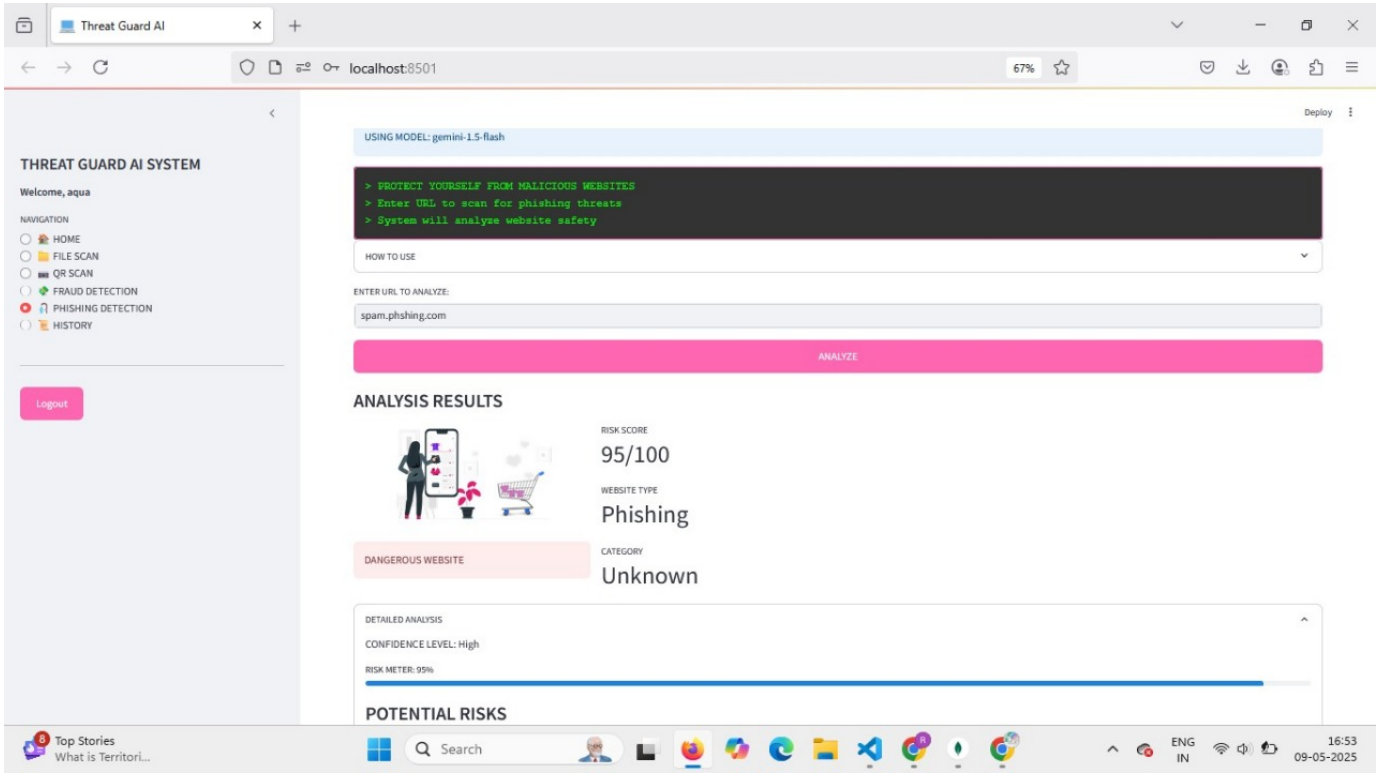


Figure 1.13 Phishing Detection – Unsafe

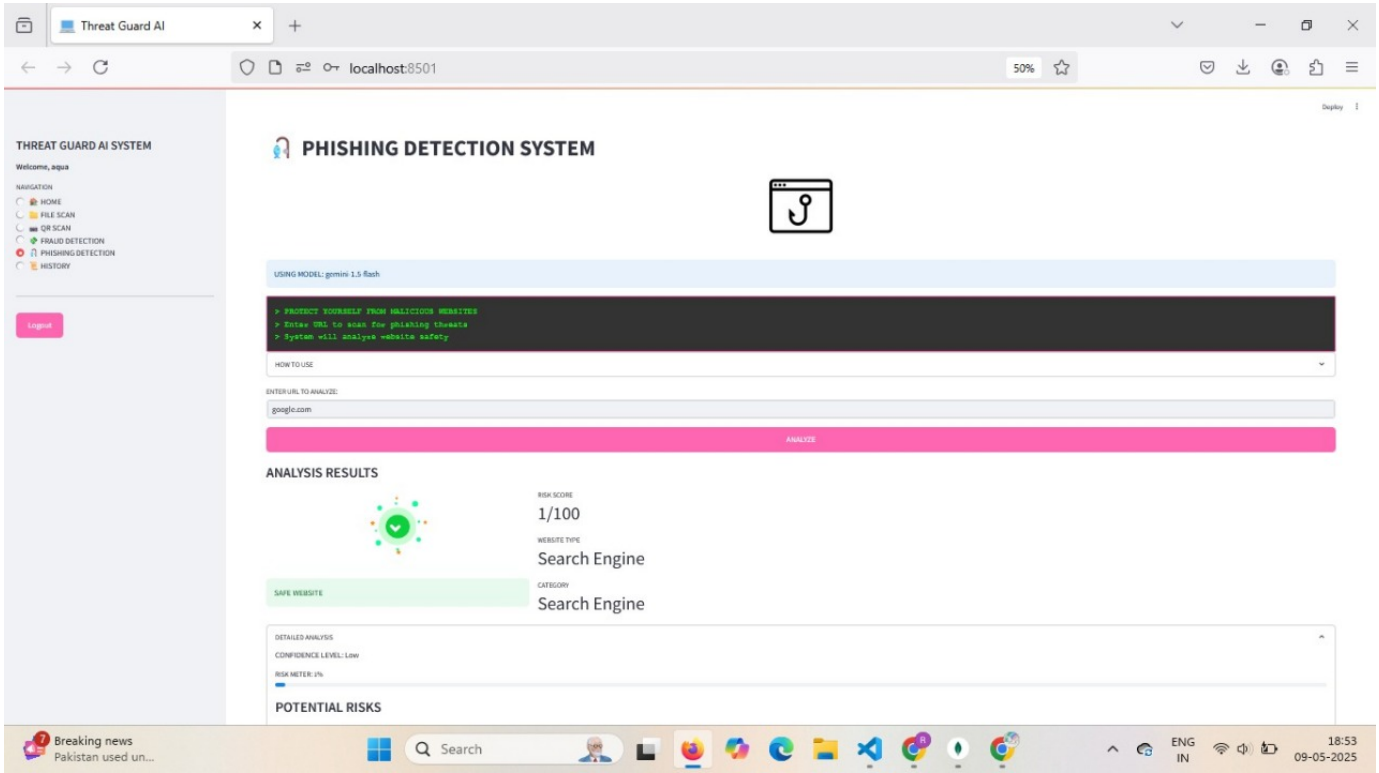


Figure 1.14 Phishing Detection –Safe

F. Summary of Detection Accuracy

Each module was trained and tested independently using balanced datasets. Accuracy results for each module are summarized below:

Module	Detection Accuracy
UPI Fraud Detection	96.2%
QR Code Detection	94.8%
PDF Malware Detection	97.1%
Phishing Link Detection	98.3%

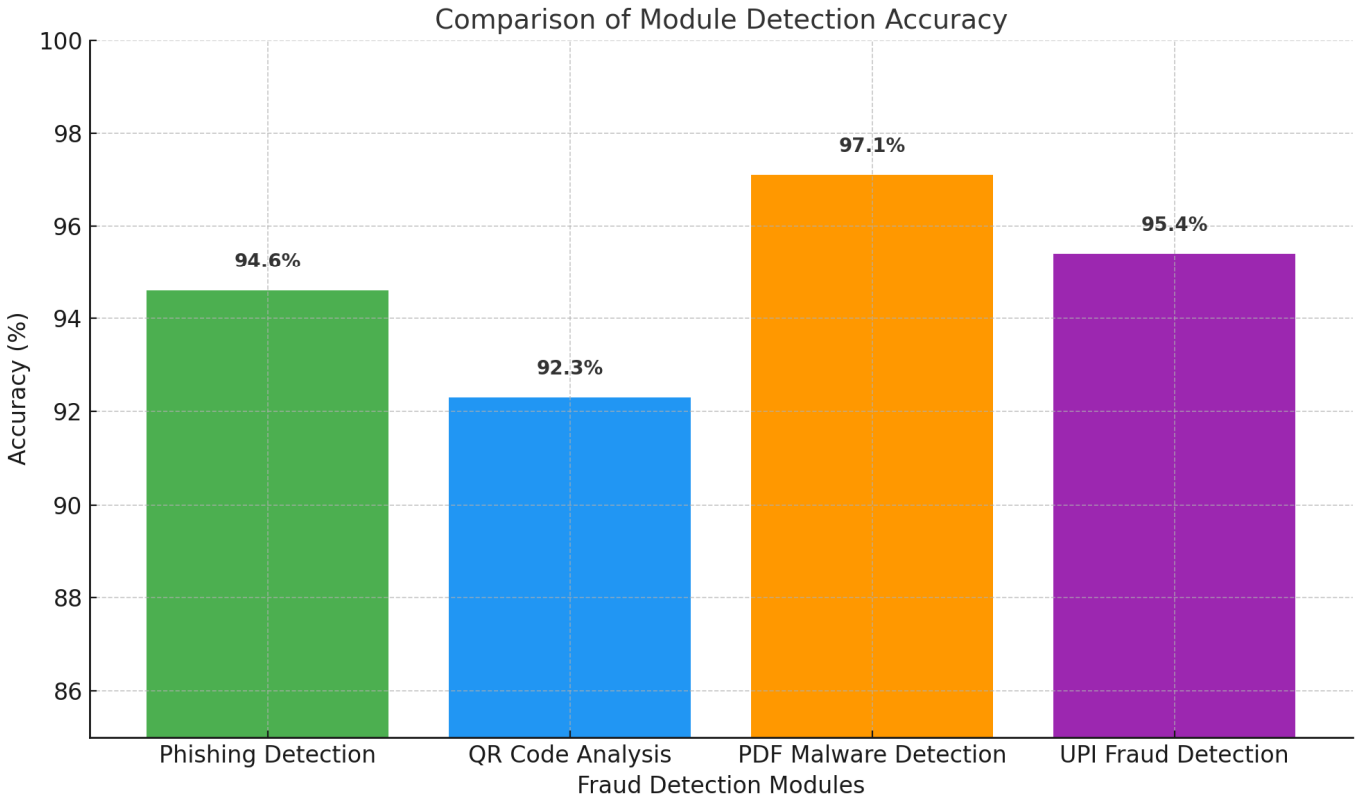


Figure 1.15 Bar graph Comparison for Accuray of Modules

VII. CHALLENGES AND DRAWBACKS

Despite achieving robust detection across multiple fraud types, Threat Guard AI faces certain practical limitations and deployment challenges:

1) Data Dependency: The system's accuracy heavily relies on the quality and representativeness of the training datasets.
2) Adversarial Attacks: Smart adversaries may craft inputs designed to evade detection, requiring more resilient models.
3) Model Drift: As fraud patterns evolve over time, machine learning models may become outdated and need periodic retraining to maintain accuracy.
4) Explainability: Black-box models, while effective, may not offer sufficient explanation for critical predictions—posing an issue in regulated environments.
5) Resource Requirements: Real-time processing of large files and high-frequency inputs can demand significant computational resources.
6) Live Banking APIs (UPI): For effective and secure UPI fraud detection in production, integration with real-time banking APIs is essential. This enables validation of transaction metadata, fraud scores, and instant blocking of suspicious transactions. The current version uses simulated data and would require authorized API access for production deployment.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

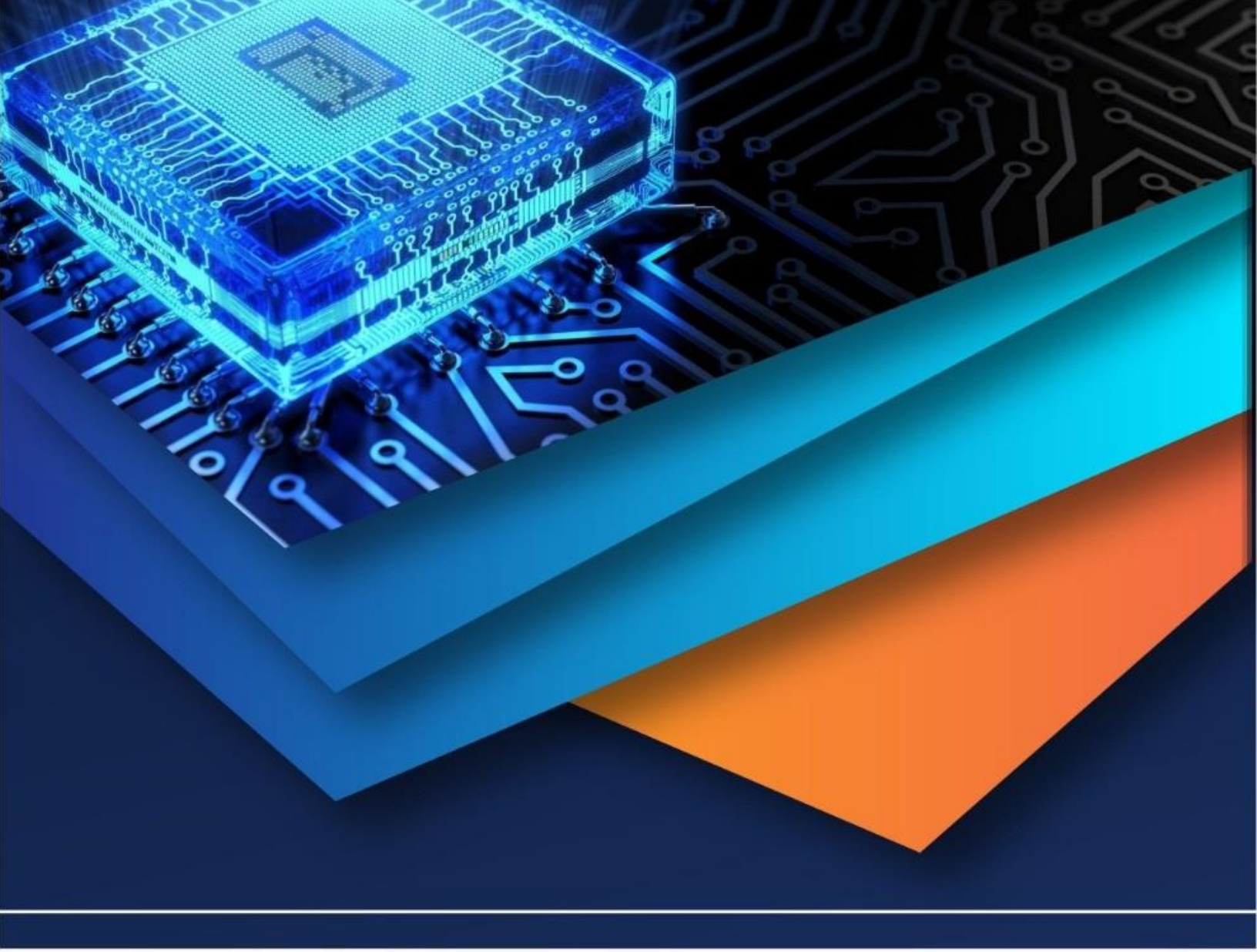
VIII. CONCLUSION

The above-discussed multi-domain fraud detection system is a huge leap toward the modernization of fraud prevention. By leveraging the power of AI and a unified detection platform, it fills critical gaps in traditional systems. The design ensures real-time threat analysis, modular scalability, and data-driven predictions. Continuous testing, improvement, and extension into cloud infrastructure will further amplify its impact on the digital security landscape.

REFERENCES

[1] Ume Zara et al., "Phishing Website Detection Using Deep Learning Models," IEEE Access, 2024.
[2] S. Jagadeesan et al., "UPI Fraud Detection Using Machine Learning," International Journal of Security Studies, 2025.
[3] J. Seetha et al., "QR Code Recognition Based on Image Processing," Springer LNCS, 2024.
[4] I. Ahmed et al., "DeepLedger: A Blockchain-Integrated Deep Learning Model for Malware," ACM CCS, 2023.
[5] R. Kumar et al., "A Survey on Fraud Detection Mechanisms in Mobile Payment Systems," IEEE Transactions on Consumer Electronics, vol. 68, no. 2, pp. 234-245, Apr. 2022.
[6] S. Gupta and N. Singh, "Phishing Detection Using Hybrid Deep Learning Approach," Journal of Cybersecurity and Privacy, vol. 6, no. 3, pp. 98-112, Mar. 2023.
[7] M. Patel and T. Singh, "QR Code-Based Authentication: Security Risks and Mitigation," International Journal of Computer Science and Security, vol. 17, no. 4, pp. 256-268, Dec. 2023.
[8] P. V. Narayanan et al., "AI-Driven UPI Transaction Fraud Detection in Real-Time," International Journal of Information Security, vol. 19, no. 1, pp. 45-57, Jan. 2024.
[9] R. Shah and A. Agarwal, "Exploring Malware Detection in PDFs Using AI Models," Proceedings of the International Conference on Digital Forensics and Cybersecurity, pp. 180-190, 2024.
[10] H. Joshi et al., "Application of Blockchain for Secure Payment Transactions," IEEE Blockchain Conference, pp. 25-30, 2023.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)