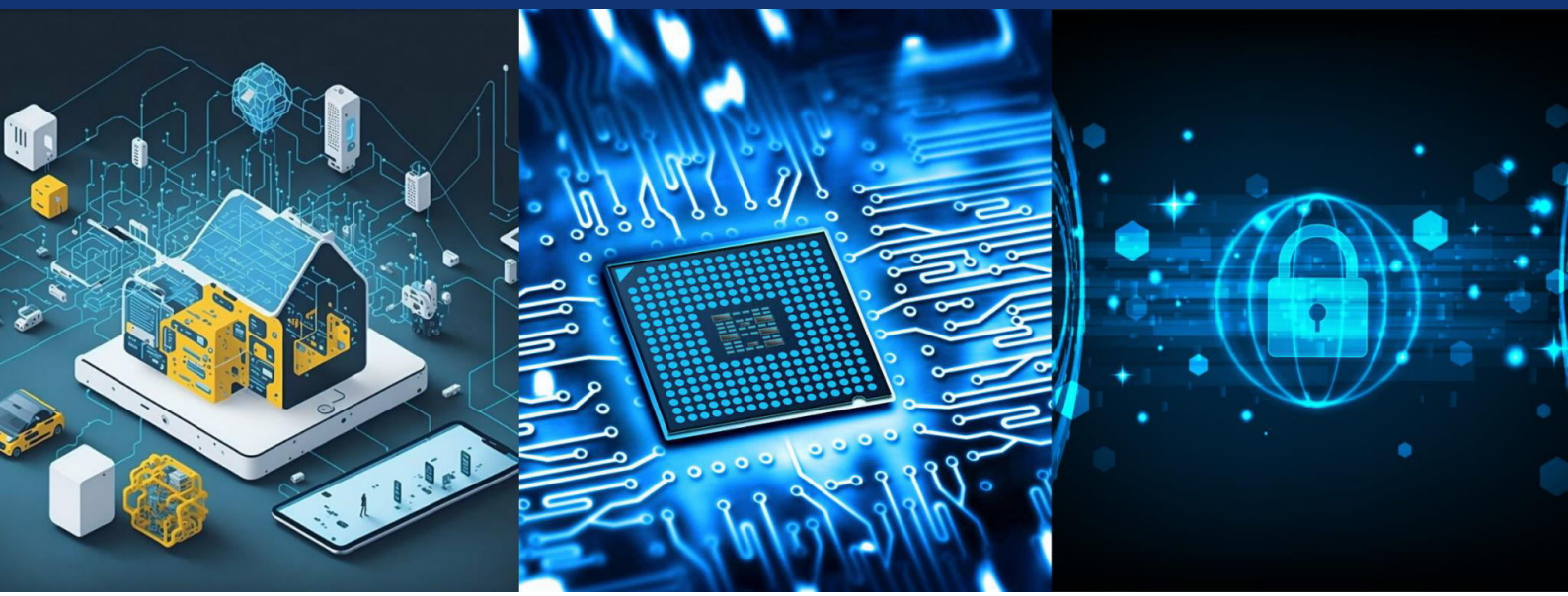
IJIRCCE

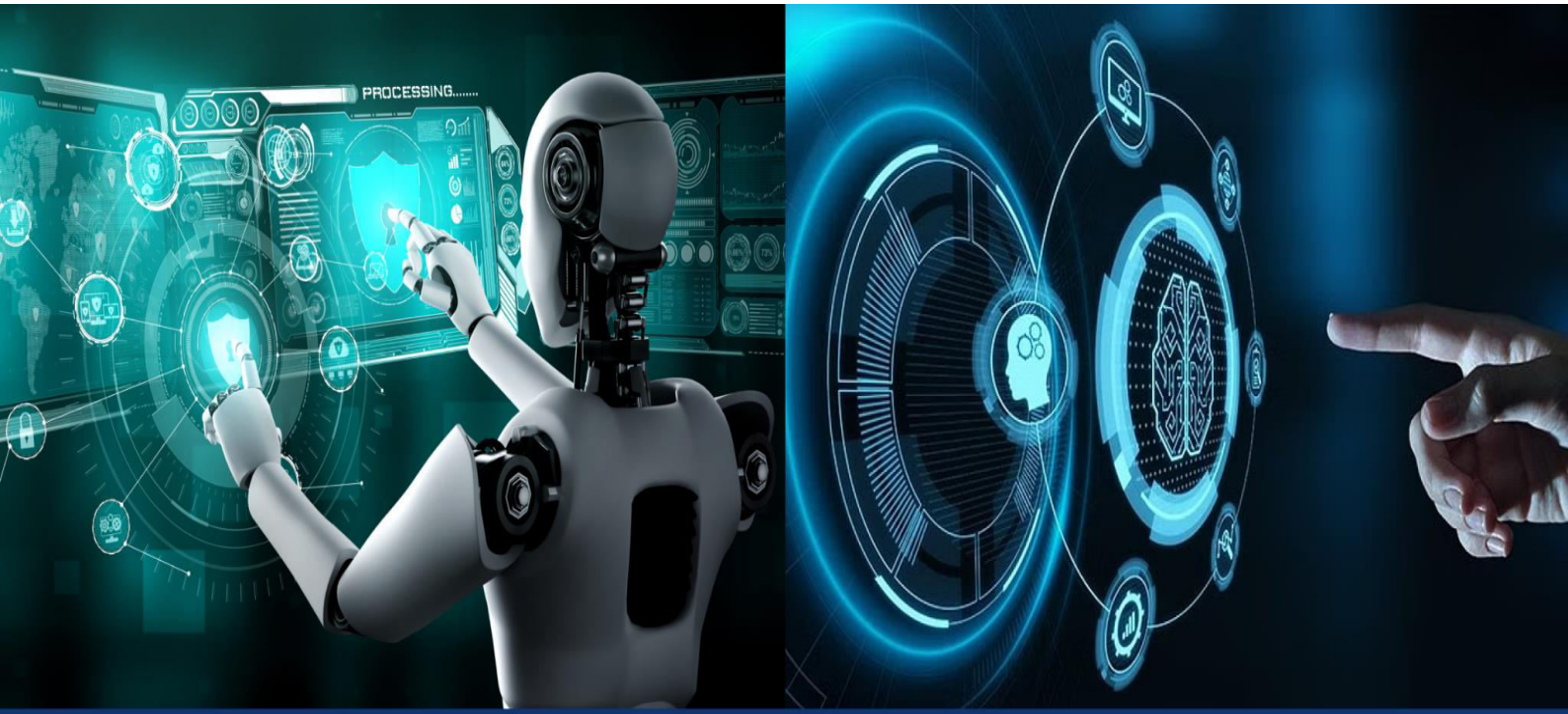
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 10, October 2026

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410050

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

UPI Fraud Detection System Using Convolutional Neural Network (CNN)

Misba Tabassum[1], Shaheen Magi[1], Naziya Banu[1], Mehek S[1], Prof. Manjunath K. G.[2]

UG Students, Dept. of CSE, Jain Institute of Technology, Davangere, Karnataka, India[1]

Assistant Professor, Dept. of Computer Networks & Communication, Jain Institute of Technology, Davangere, Karnataka, India[2]

ABSTRACT: The rapid growth of the Unified Payments Interface (UPI) has made digital transactions fast and convenient. However, the increasing use of digital payments has also created opportunities for fraudulent activities such as phishing, impersonation, and misuse of user-behavior patterns. Conventional fraud detection approaches that depend on fixed rules and thresholds may not identify changing or complex transaction patterns effectively. This paper presents UPI Guard, a UPI fraud detection system that uses a Convolutional Neural Network (CNN) to classify transaction activity as valid or fraudulent. The system is developed as a Flask-based web application and includes OTP-based login, role-based portals for users, merchants, and administrators, merchant QR-code generation, and a QR-based payment simulation. During payment initiation, transaction information is represented using pre-scaled features and passed to the CNN model for prediction. The application displays the result as "VALID" or "FRAUDULENT"; transactions identified as suspicious are intended to be blocked, and the user is notified. Transaction records are maintained for administrative monitoring. The proposed system aims to support proactive fraud detection within the project's simulated UPI payment workflow. The project presentation does not provide dataset details or measured evaluation values; therefore, no numerical performance claims are made in this paper.

KEYWORDS: UPI Fraud Detection, Convolutional Neural Network, CNN, Digital Payments, Flask, OTP Authentication, QR Code Payment, Transaction Classification, Financial Security.

I. INTRODUCTION

Digital payment platforms have changed the way people transfer money and make purchases. The Unified Payments Interface (UPI) supports convenient account-to-account transactions through mobile applications and QR-code-based payment methods. The ease and speed of UPI have encouraged widespread use of digital payments. Along with these benefits, the increase in transaction activity has made the detection of fraudulent transactions an important concern.

Fraudulent activity in digital payment environments can involve phishing, impersonation and attempts to exploit user behaviour. Such activities may result in unauthorized payments and financial loss. A fraud detection mechanism must therefore examine transaction information and identify suspicious activity as early as possible in the payment process.

Traditional systems often use predefined rules, such as flagging transactions that cross a fixed amount threshold. Although rule-based methods can identify known conditions, they may not adapt well to complex or changing patterns. This creates a need for data-driven approaches that can learn patterns from transaction information.

The project presented in this paper, UPI Guard, uses a CNN-based classification model within a Flask web application. It combines transaction prediction with OTP-based authentication and separate portals for users, merchants and administrators. The system is designed to provide a prediction when a payment is initiated and to block transactions classified as suspicious.

II. RELATED WORK

Research in financial fraud detection has considered rule-based techniques, conventional machine learning algorithms and ensemble learning methods. Rule-based approaches are relatively simple to implement but depend on predefined conditions and may have difficulty identifying patterns that are not represented by existing rules.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The literature survey in the research paper refers to Liu et al. (2025), who discuss fraud identification using machine learning methods including CatBoost, XGBoost and LightGBM [1]. These gradient-boosting methods demonstrate the use of data-driven models for identifying fraudulent activity.

Sharma et al. (2023) are cited in the research paper for machine-learning-based financial fraud detection using Decision Tree and Logistic Regression [3]. These methods provide established approaches for classification and can be used as baseline techniques in fraud detection studies.

III. PROBLEM STATEMENT

UPI payment systems are vulnerable to fraudulent transactions that may cause financial loss to users. Static rule-based detection methods depend on predefined thresholds and may not identify dynamic or unfamiliar fraud patterns. In addition, delayed alerts may not prevent a suspicious transaction at the time it is initiated. The problem addressed in this project is to develop a CNN-based system that analyses transaction information during payment initiation, classifies the transaction as valid or fraudulent, and supports immediate blocking and user notification when suspicious activity is detected

IV. PROPOSED SOLUTION

UPI Guard is proposed as a Flask-based web application that connects user authentication, merchant QR-code functionality, transaction processing and CNN-based fraud prediction. The application provides separate access for users, merchants and administrators. A user initiates a payment through the QR-code-based payment simulation. Transaction information is prepared in the feature format expected by the model and supplied to the CNN for classification.

The model prediction is returned to the application and displayed as “VALID” or “FRAUDULENT”. If the transaction is classified as suspicious, the proposed workflow blocks the transaction and immediately alerts the user. Transaction information is also logged so that authorized administrators can monitor activity.

4.1 Data Input and Feature Preparation

The project presentation states that the CNN model uses more than ten behavioural and transactional features. Examples mentioned include transaction amount, time, location and category. The presentation also specifies that pre-scaled features are used as model input. The complete feature list, the source of the data and the exact scaling procedure are not specified in the supplied presentation and should be documented from the implemented project.

4.2 CNN-Based Transaction Classification

The CNN is the central prediction component of the proposed system. It receives the prepared transaction features and classifies the transaction into one of two categories: valid or fraudulent. The project presentation describes the CNN as learning patterns from transaction data and extracting relevant information for classification. The precise layer configuration, activation functions, optimizer and training parameters are not provided in the presentation; these implementation-specific details should be taken from the model code.

4.3 Transaction Decision and Alert

After classification, the Flask application presents the model output to the user. A transaction identified as valid is displayed with the corresponding valid status. A transaction identified as fraudulent is treated as suspicious, and the proposed workflow blocks it and displays an alert. This decision stage is intended to provide proactive control rather than relying only on post-transaction alerts.

4.4 Authentication and Role-Based Access

The application includes OTP-based login as an additional authentication step. It provides distinct portals for users, merchants, and administrators. User functions are associated with payment initiation, merchant functions include account creation and QR-code generation, and administrator functions include monitoring users and transactions. The exact permissions and access-control rules should be described according to the implemented application.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410050

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. SYSTEM IMPLEMENTATION AND WORKFLOW

The system is implemented as a web application using Flask. The application coordinates the user interface, authentication process, QR-based payment simulation, CNN prediction and transaction logging. The project presentation identifies an SQL database for maintaining application data and transaction records.

5.1 System Workflow

The user registers or logs in to the application using the OTP-based authentication process.
The merchant creates an account and generates a QR code through the merchant portal.
The user scans the QR code and initiates a payment in the project's payment simulation.
The application prepares the transaction information as pre-scaled input features.
The prepared features are passed to the CNN model for classification.
The model returns a valid or fraudulent prediction to the Flask application.
The application displays the result; a suspicious transaction is blocked, and the user is alerted.
Transaction information is recorded for authorized monitoring through the administrator portal.

5.2 System Advantages

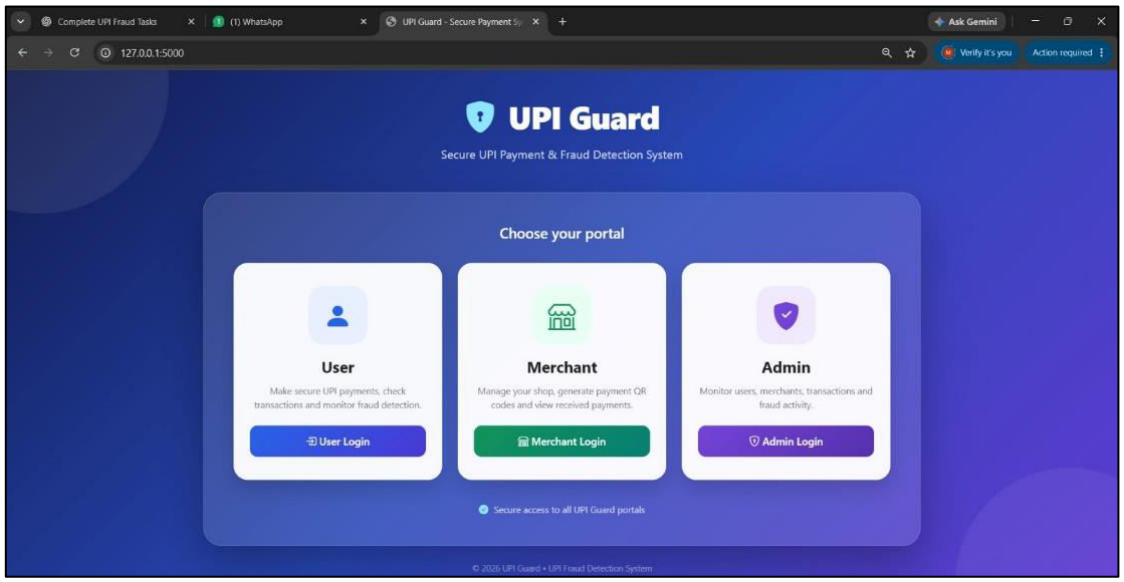
- Enables real-time UPI fraud detection.
- Identifies suspicious transaction patterns using CNN.
- Helps prevent potentially fraudulent payments.
- Supports behavior-based transaction analysis.
- Provides secure OTP-based authentication.
- Supports QR-code-based payment processing.
- Generates alerts for suspicious transactions.

VI. SIMULATION RESULTS

The UPI Guard system was implemented and tested using the developed web application. The CNN model was used to analyze transaction details and classify transactions as VALID or FRAUDULENT.

The simulation demonstrated that valid transactions could proceed, while suspicious transactions were identified and blocked. The system also displayed fraud alerts and maintained transaction records for monitoring.

The results demonstrate the working of the proposed CNN-based UPI fraud detection system and its ability to identify suspicious transactions during the payment process.

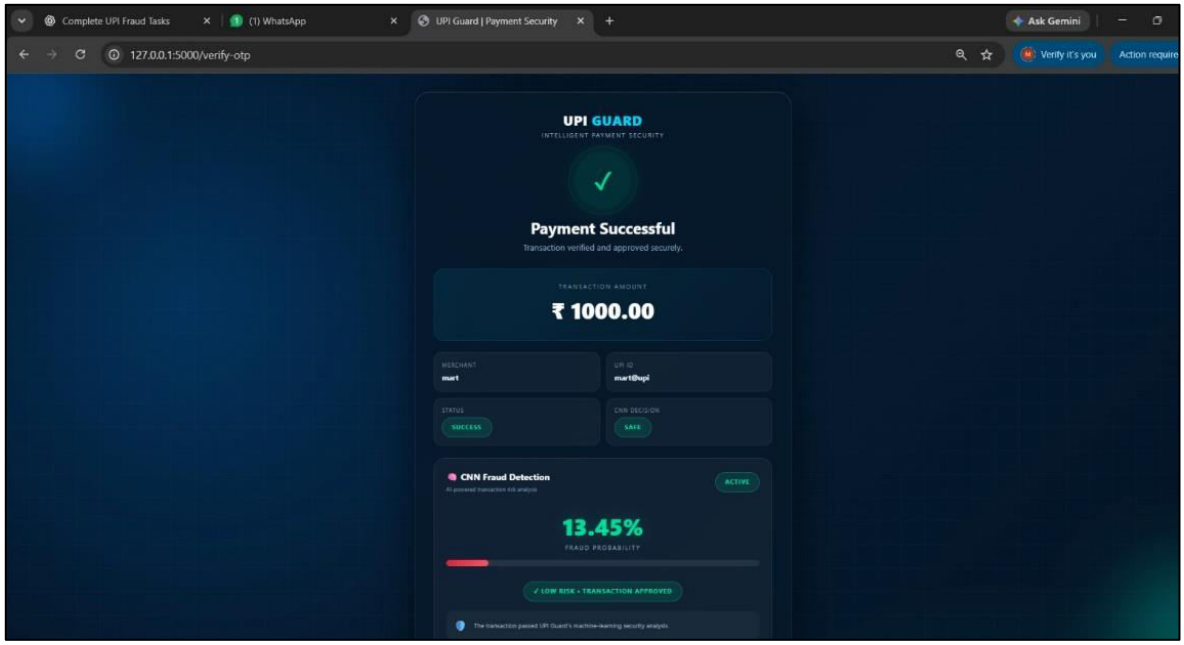


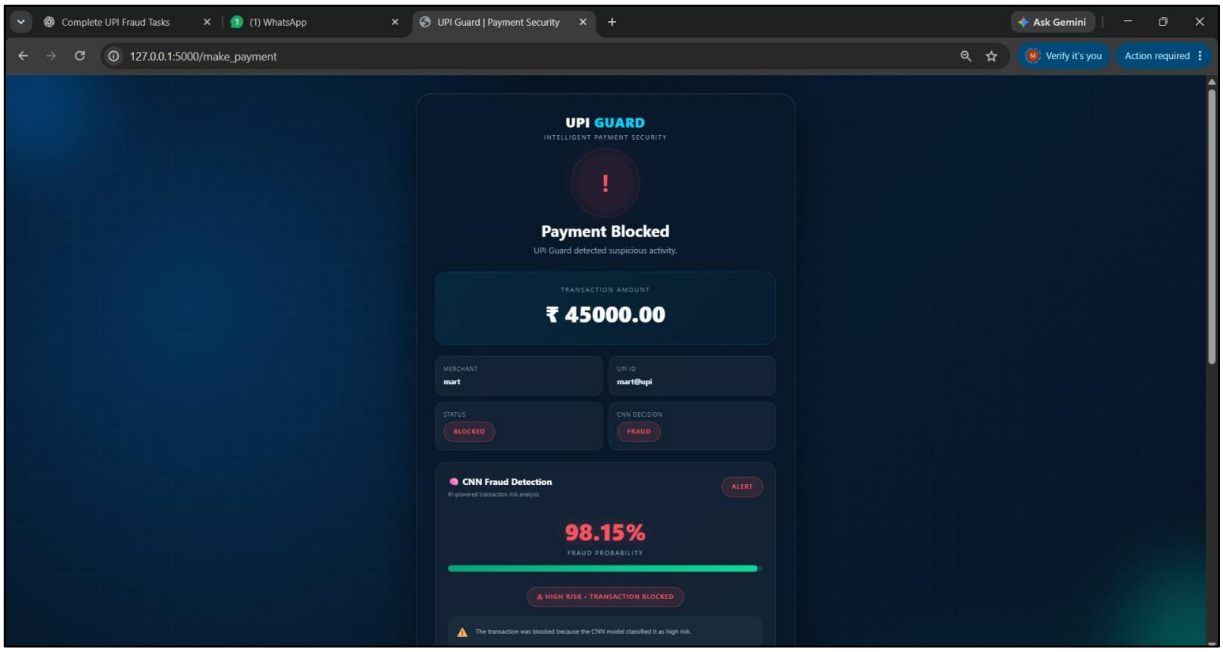
© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410050

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





VII. CONCLUSION AND FUTURE WORK

This paper presents UPI Guard, a CNN-based UPI fraud detection system developed as a Flask web application. The proposed system combines OTP-based authentication, role-based portals, merchant QR-code generation, QR-based payment simulation, transaction feature preparation and CNN-based transaction classification. The application displays the prediction as valid or fraudulent and is designed to block suspicious transactions and notify users. Transaction logging and an administrator dashboard support monitoring of application activity.

The project provides a prototype workflow for proactive fraud detection in a simulated UPI payment environment. The supplied presentation does not establish integration with a live banking or UPI payment network, nor does it provide numerical model evaluation results. Future work should document the dataset and input features, describe the CNN

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

architecture and training configuration, and evaluate the model using accuracy, precision, recall, F1-score, confusion matrix and prediction latency. Further improvements may include testing on larger and more diverse transaction datasets, reducing false alerts, examining changing fraud patterns and assessing deployment scalability and security.

REFERENCES

[1] Liu et al., “Fraud Identification Using Machine Learning,” 2025. (Full publication details were not supplied in the project presentation; verify the title, publication venue, pages and DOI.)
[2] Patel et al., “Fraud Detection using Random Forest,” 2024. (Full publication details were not supplied in the project presentation; verify the title, publication venue, pages and DOI.)
[3] Sharma et al., “ML-Based Financial Fraud Detection,” 2023. (Full publication details were not supplied in the project presentation; verify the title, publication venue, pages and DOI.)
[4] Python Software Foundation, “Python Documentation.” https://docs.python.org/
[5] Pallets Projects, “Flask Documentation.” https://flask.palletsprojects.com/
[6] TensorFlow Team, “TensorFlow Documentation.” https://www.tensorflow.org/



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details