Proactive Fraud Detection in Financial Transaction using Generative AI

INTRODUCTION

- Digital transactions have grown rapidly, increasing financial fraud and associated security risks.
- In FY2024, India experienced a 400% increase in cyber fraud, resulting in over $20M in losses, with 77% of cybercrimes linked to online fraud. UPI frauds accounted for 47% of the cases, with daily fraud incidents reaching 800.
- This rise in cyber fraud has increased security risks and caused significant financial losses, making it harder to protect individuals and businesses from fraud.
- Accurate fraud detection is essential to protect money and maintain trust in digital payments.

LITERATURE SURVEY

S.NO	Title	Year of Publication	Journal/ Conference	Algorithms used	Limitations
1.	Credit Card Fraud Detection using GAN	2024	Mar Athanasius College of Engineering	CTGAN/ Tabular GAN/ Feature Enginnering	The model performed well initially but showed reduced efficiency on CTGAN-generated data, highlighting the need for fine-tuning or ensembling to handle distribution shifts.
2.	An Improved Variational Autoencoder Generative Adversarial Network with Convolutional Neural Network for Fraud Financial Transaction Detection.	2024	IEEE	VAEGAN/ SMOTE/CNN	Although the approach achieved high accuracy, it needs to be tested on different datasets and timeframes to ensure reliable performance in various scenarios.
3.	Generating Synthetic Data For Credit Card Fraud Detection	2022	Bournemouth University, UK	WGAN/ KCGAN	While K-CGAN outperformed other oversampling methods, its effectiveness needs to be validated on different anomaly types and datasets to ensure broader applicability.
4.	Synthetic Data Generation for Fraud Detection using GANs	2021	University Of London	SDGGAN & SMOTE	The approach improved identification rates but showed varying performance across datasets, requiring further validation for consistency.

Table:1 Literature Survey

EXISTING SYSTEM

- Existing models depend on rule-based methods (e.g., Logistic Regression) using fixed rules from historical data.
- Often result in high false positives and false negatives.
- They have difficulty handling imbalanced data since fraudulent transactions are extremely rare compared to legitimate ones.

PROPOSED SYSTEM

- The proposed solution is an AI-driven fraud detection system that leverages balanced transaction data, utilizing GAN-based methods (such as CTGAN, WGAN, or SDG-GAN).
- Advanced machine learning techniques (Random Forest Classification or Autoencoders) are utilized to classify transactions and detect anomalies.
- The model is trained on the Fraud Detection Dataset to enhance accuracy and recognize emerging fraud patterns.

PROBLEM STATEMENT

- Digital transactions are growing fast, which increases the risk of fraud
- Fraudsters constantly evolve their methods to bypass detection and take advantage of gaps in security systems.
- Traditional fraud detection systems struggle to adapt to new, evolving fraud tactics, leading to increased vulnerability.
- This can lead to missing actual fraud or wrongly flagging good transactions, affecting financial security.

O B J E C T I V E S

- Collect and preprocess the Fraud Detection Dataset.
- Generate synthetic fraud data to balance the dataset using GAN models.
- Develop a ML model to classify transactions as fraudulent or legitimate.
- Evaluate the model’s performance using metrics like accuracy, precision, recall, and F1-score.

SYSTEM ARCHITECHTURE

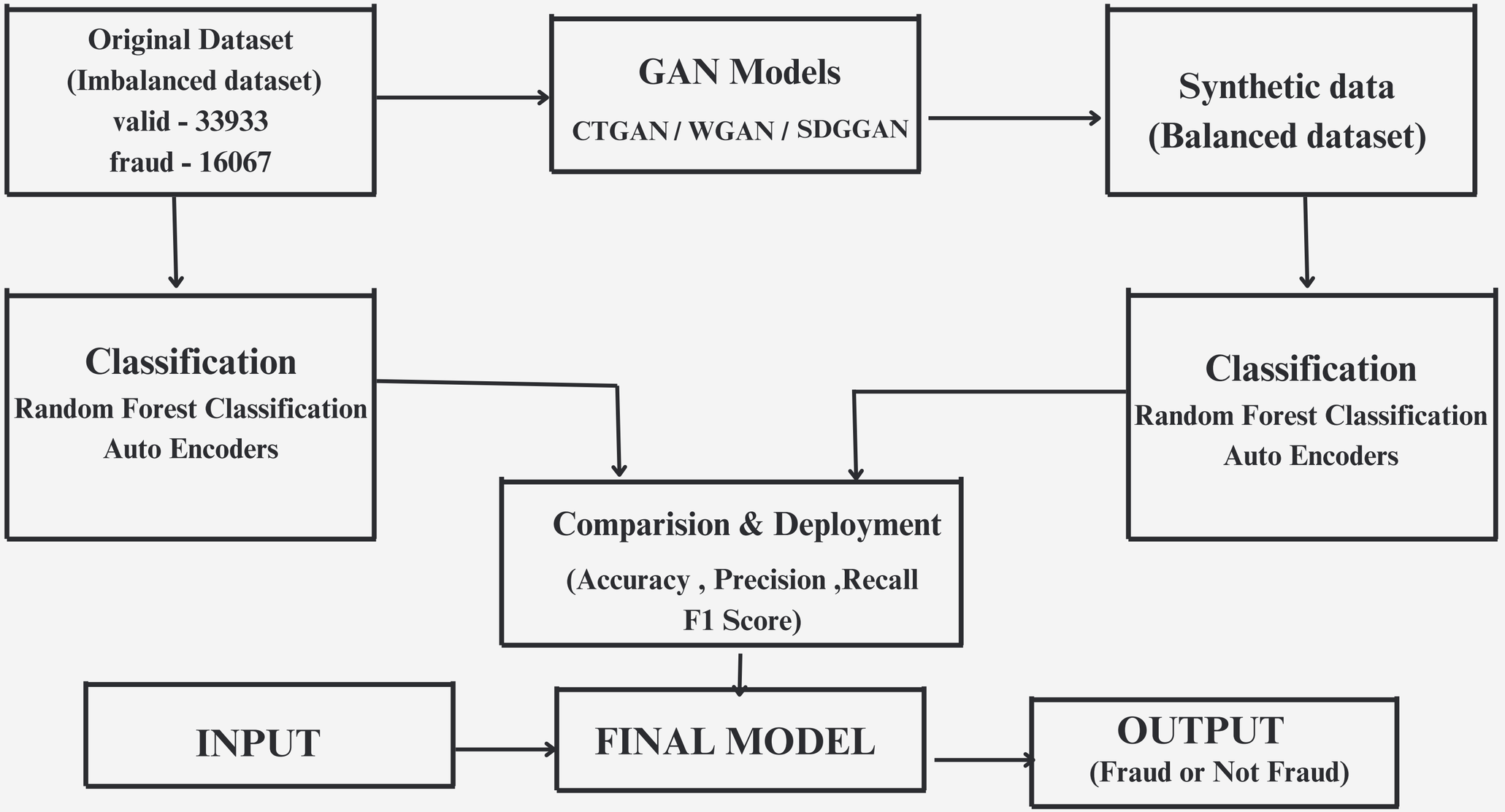

Fig 1: System Architecture

METHODOLOGY

Module 1: Data Collection and Preprocessing

- Data cleaning and handling missing values.

Module 2: Synthetic Data generation using GAN models

- Generate synthetic fraud data to balance the dataset.

Module 3: Model Development and Classification

- Training the model on the balanced dataset.

Module 4: Model Evaluation and Performance Comparision

- Evaluate the trained AI model using relevant metrics.
- Compare the model's performance with other fraud detection techniques.

DATA SET INFORMATION

- The dataset used was obtained from Kaggle
- It contains a total of 50,000 entries, indexed from 0 to 49,999.
- The target variable is the "is_Fraud" column, which categorizes each transaction as either Fraud or Non-Fraud.
- The dataset consists of 21 columns, which include both categorical (5) and continuous (16) features.
- The distribution of classes in the dataset is as follows:
- Non-Fraud: 33,933 entries
- Fraud: 16,067 entries

DATA COLLECTION AND PREPROCESSING

Data Cleaning :

- **Missing Value Handling** :Checked for null values using df.isnull().sum().
- **Duplicate Removal**: Identified and removed duplicates using df.duplicated().sum() and df.drop_duplicates().
- **Dropped non-essential columns**: Transaction_ID, User_ID, Merchant_Category, Card_Age, and Timestamp (after extraction).
- **Class Distribution Check**: Checked the count of fraudulent vs. non-fraudulent transactions in the target variable (Fraud_Label).
- **Correlation Analysis**: Used df.corr() and visualized with a heatmap to analyze feature relationships.

DATA COLLECTION AND PREPROCESSING

Feature Engineering:

- **Timestamp Decomposition:**Extracted useful time-based features from Timestamp (Year, Month, Day, Hour, Minute, Second)
- Dropped original Timestamp, Transaction_Date, and Transaction_Time after decomposition.

Label Encoding:

Applied LabelEncoder() on categorical features to convert them into numerical values:

- Device_Type
- Card_Type
- Authentication_Method
- Transaction_Type

GENERATIVE ADVERSARIAL NETWORK(GANS)

- GAN (Generative Adversarial Network) is a deep learning framework that uses two neural networks, a Generator and a Discriminator to produce realistic synthetic data by learning the patterns of real data.
- Our dataset contains 50,000 credit card transactions, with only 16,067 labeled as fraudulent highlighting a severe class imbalance.To address this, we use GANs to generate realistic synthetic fraud samples, helping to balance the dataset and improve the detection accuracy of rare fraud cases.

Types of GANs Used:

- **CTGAN (Conditional Tabular GAN)**:Tailored for tabular data, effectively handles mixed data types and imbalanced datasets using conditional sampling.
- **WGAN (Wasserstein GAN):**Uses Wasserstein distance for more stable training and better-quality data generation.
- **SDGGAN (Synthetic data generation GAN):**Combines self-supervision and differential privacy to generate synthetic data while preserving data privacy.

CTGAN IMPLEMENTATION

Step 1: Loaded the processed fraud dataset

Step 2:Installed and imported libraries like sdv, CTGAN Synthesizer

Step 3:Split into features X and target y

Step 4:Trained CTGAN Synthesizer on the imbalanced dataset to generate synthetic data for minority class.

Step 5:Generated a balanced synthetic dataset using CTGAN

Step 6:Combined real and synthetic data to create a balanced dataset.

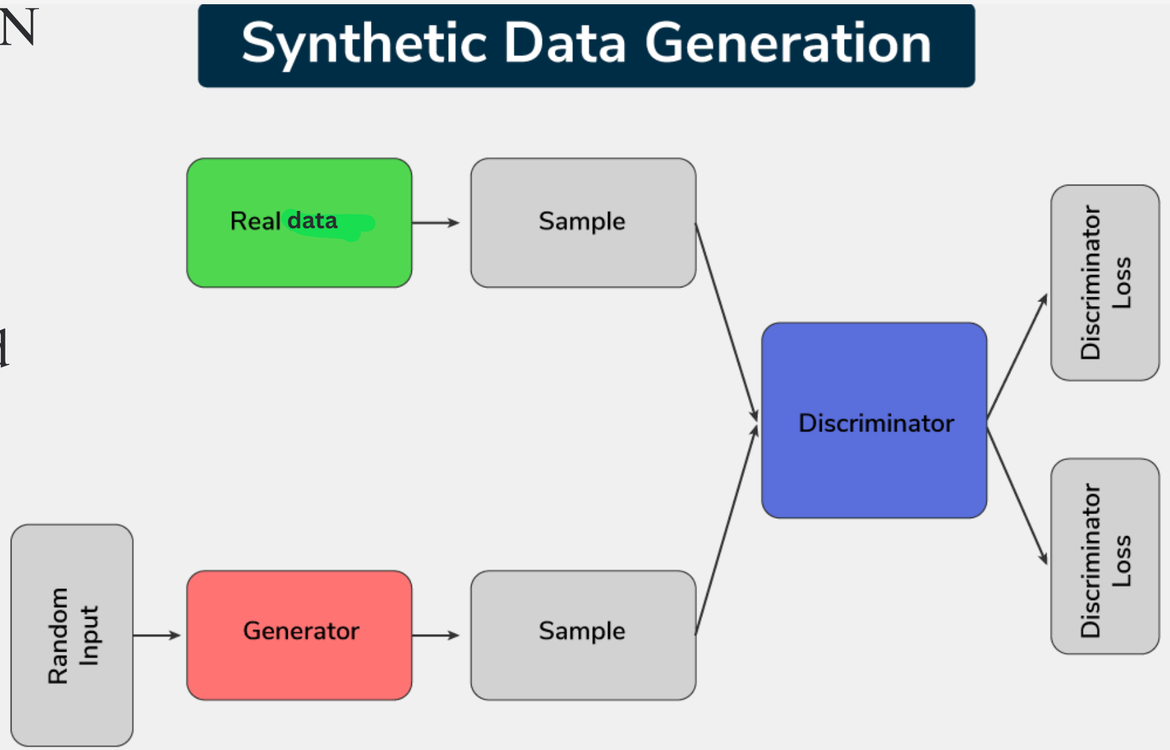

Fig:2 CTGAN Architecture

WGAN IMPLEMENTATION

Step 1:Generated latent vectors from a standard normal distribution

Step 2:Passed the noise vectors into the trained WGAN generator to generate synthetic fraud data.

Step 3:Converted the generated NumPy array to a Pandas DataFrame using the original fraud data's feature names.

Step 4:Separated the continuous and categorical columns from the generated data.

Step 5:Applied inverse transformation to the continuous features using the original scaler to restore the data to its real-world scale

Step 6:Combined the rescaled continuous features with the categorical features

Step 7:Merged the synthetic fraud samples with the original dataset to create a balanced dataset.

SDG-GAN IMPLEMENTATION

Step 1: Loaded the processed fraud dataset

Step 2:Normalized data to [-1, 1] and split into fraud/non-fraud.

Step 3:Created synthetic fraud samples from noise

Step 4:Classified real and fraud samples

Step 5:Performed alternate training (5,000 epochs) to balance generator/discriminator.

Step 6:Produced synthetic fraud cases to fix class imbalance.

Step 7:Denormalized and merged with original data for final balanced dataset

COMPARISION OF GAN'S (EVALUATION METHODS)

- Upon comparing the **absolute log mean** and **standard deviation plots** of real vs. synthetic data, **CTGAN** shows the closest alignment to the diagonal with minimal deviation, indicating strong statistical stability and accurate replication of the original data distribution.
- CTGAN showed the best alignment in **table evaluation graphs**, where real and synthetic data distributions merged almost perfectly.
- SDGGAN, while generally effective, shows slight deviations suggesting moderate discrepancies in some features.
- WGAN displays more pronounced outliers, especially in standard deviations, making it the least consistent among the three.
- **KS Test**: CTGAN shows the lowest KS statistics, proving it generates continuous features most similar to the original data distribution.
- **Chi-Square Test**: CTGAN has the lowest chi-square values across features, indicating it best replicates the distribution of real categorical data compared to WGAN and SDGGAN.
- Therefore, CTGAN emerges as the most reliable GAN model in this evaluation for generating statistically accurate synthetic data.

CTGAN statistics graph

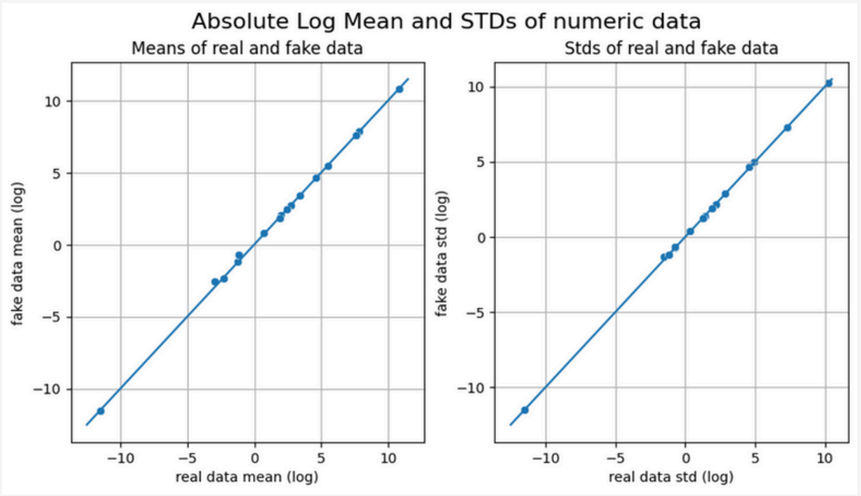

Fig 3: CTGAN statistics graph

WGAN statistics graph

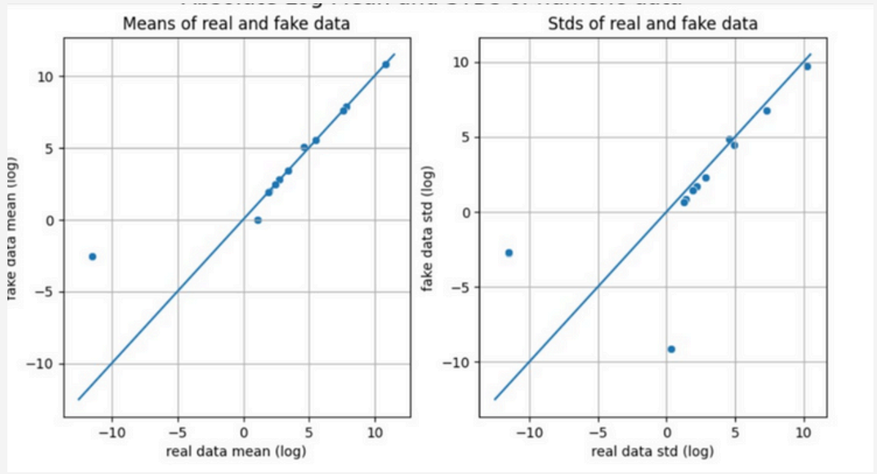

Fig 4:WGAN statistics graph

SDG GAN statistics graph

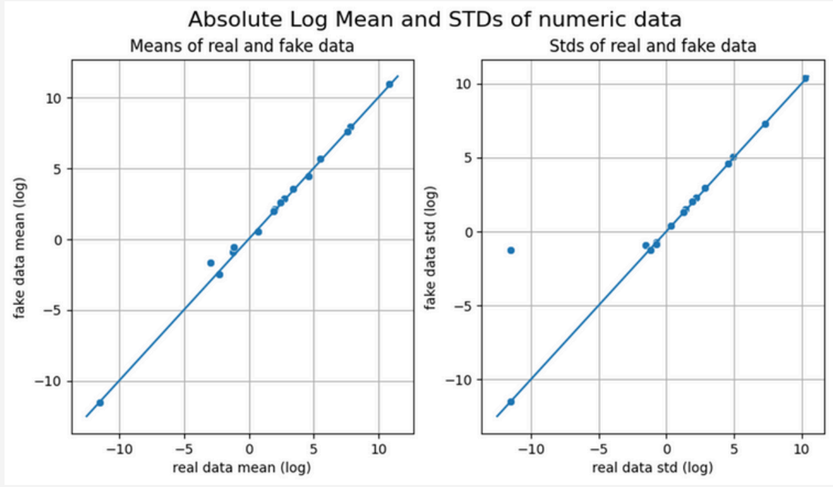

Fig 5: SDG GAN statistics graph

KS Test Comparison (Numerical Features):

Model	Transaction Amount	Account balance	Transaction Distance	Average Transaction Amount
CTGAN	0.0288	0.0364	0.0162	0.0225
WGAN	0.0514	0.0490	0.0462	0.0431
SDGGAN	0.1452	0.1452	0.1223	0.1452

Table 2: KS Test Comparison

Chi-Square Test Comparison (Categorical Features):

Model	Transaction Type	Device Type	Previous Fraudelent Activity	Card Type	Authentication Method	Is Weekend	Fraud Label	Ip Address Flag
CTGAN	23.30	7.64	0.35	13.30	6.65	0.11	0.13	3.28
WGAN	150000	100000	49988.72	150000	150000	49995.23	49995.41	49979.02
SDGGAN	14100.18	4132.84	2984.64	13.30	13070.59	3945.35	28392.96	6039.24

Table 3: Chi-Square Test Comparison

FRAUD DETECTION MODELS

To classify transactions as fraudulent or genuine, we used two key machine learning approaches:

- **Random Forest Classifier :**
  This predicts whether a transaction is fraudulent by analyzing features like amount, time, type, device, and user behavior. It uses multiple decision trees, making it accurate, reliable, and robust.
- **Auto Encoders:**

  Autoencoders detect suspicious transactions by learning patterns of normal activity. If a transaction deviates from these patterns, it's flagged as potentially fraudulent based on reconstruction error. This helps catch new or unusual frauds.

- **To improve model performance and ensure fairness, we performed classification on both:**
  Imbalanced dataset (original dataset with fewer fraud cases)
  Balanced dataset (augmented using GAN-based synthetic data)

IMPLEMENTATION OF RANDOM - FOREST CLASSIFIERS

Step 1: Loaded and prepared the dataset

Step 2:Splited into features (X) and target (y), then further divided into training and testing sets with an 80/20 ratio.

Step 3: Initialized the Random Forest model and Configured with 500 estimators, maximum depth of 30, and balanced class weights.

Step 4: Fit the Random Forest classifier on the training data.

Step 5: Predicted fraud labels and probabilities on the test set.

Step 6:Evaluated model performance with accuracy,precision,R1 score,F1,score,ROC-AUC

Step 7:Plotted confusion matrix and ROC curve to validate performance.

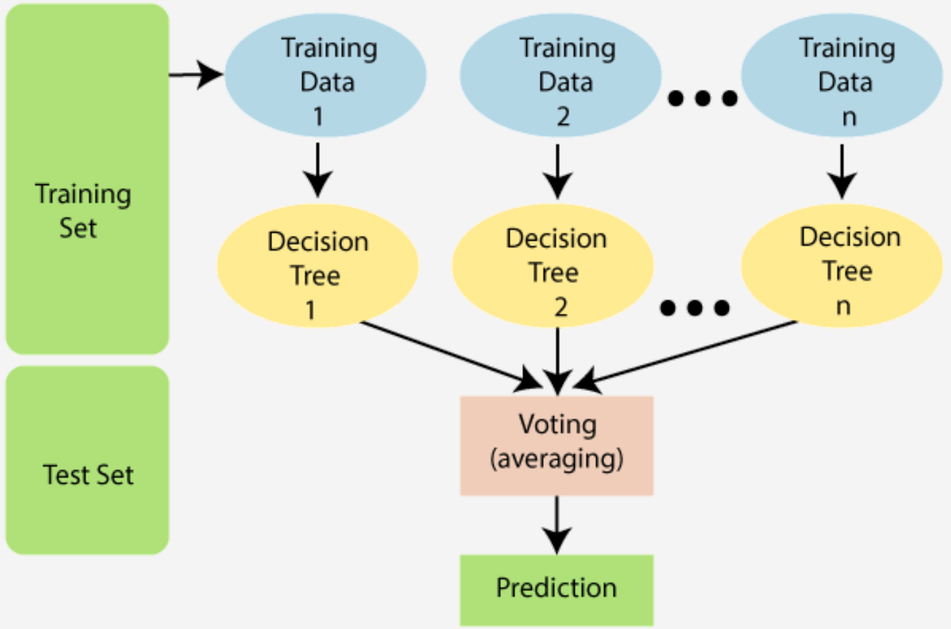

Fig 6: Random forest Classifiers Architecture

AUTO ENCODERS IMPLEMENTATION

Step 1: Loaded and prepared the dataset
Step 2: Splited into features (X) and target (y), then further divided into training and testing sets with an 80/20 ratio.
Step 3: Built an autoencoder and trained it to reconstruct normal transactions using MSE loss.
Step 4: Calculated reconstruction errors on test data and set a threshold (95th percentile of normal errors) to flag fraud.
Step 5: Achieved 50% accuracy and 47% precision, but low recall (4.6%), indicating precise but incomplete fraud detection.

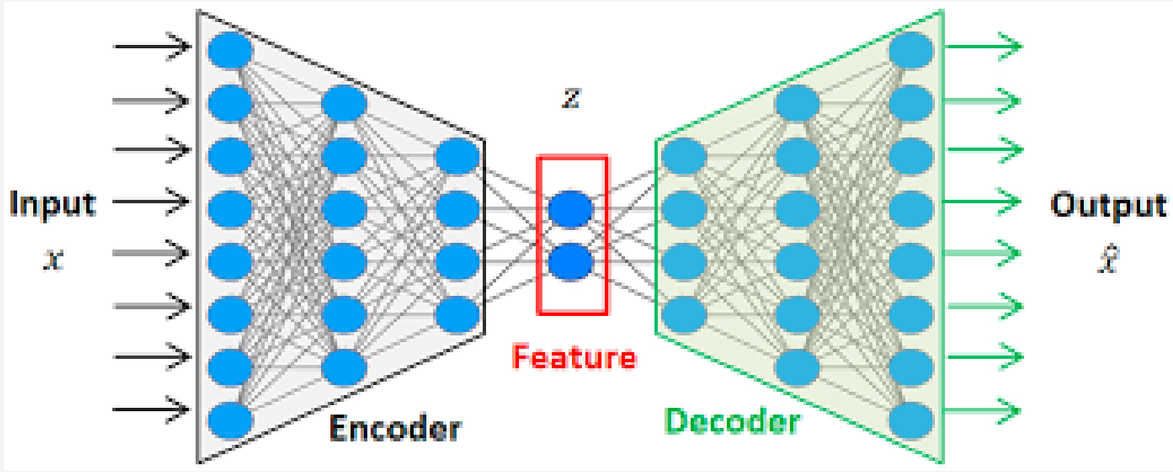

Fig 7: Autoencoders Architecture

MODEL PERFORMANCE METRICS

- **Accuray:**The percentage of correct predictions (both fraud and non-fraud) out of all predictions.

$$\textbf{Accuracy=TP+TN/(FP+FNTP+TN)}$$

- **Precision:** The proportion of predicted fraud cases that are actually fraud.

$$\textbf{Precision=TP/(FP+TP)}$$

- **Recall:**The proportion of actual fraud cases that are correctly identified by the model.

$$\textbf{Recall=TP/(FN+TP)}$$

- **F1-Score:**The harmonic mean of Precision and Recall, balancing both metrics.

$$\textbf{F1 Score=2×((Precision+Recall)/(Precision×Recall))}$$

- **True Positive (TP)**: Correctly predicted positive cases
- **True Negative (TN)**: Correctly predicted negative cases
- **False Positive (FP)**: Incorrectly predicted positive cases
- **False Negative (FN)**: Incorrectly predicted negative cases

MODEL PERFORMANCE ANALYSIS

Split Ratio	Model	Accuracy	Precision	Recall	F1-Score
80-20	Random Forest	88	0.90	0.88	0.87
80-20	Auto encoders	67.79	0.62	0.68	0.59
70-30	Random Forest	88	0.90	0.88	0.87
70-30	Auto Encoders	66.6	0.58	0.67	0.57
90-10	Random Forest	87.34	0.89	0.87	0.86
90-10	Auto encoders	68.46	0.65	0.68	0.60

Table 4 :Model Trained on Unbalanced Dataset

MODEL PERFORMANCE ANALYSIS

Split Ratio	Model	Accuracy	Precision	Recall	F1-Score
80-20	Random Forest	0.89	0.90	0.89	0.89
80-20	Auto encoders	0.50	0.49	0.50	0.38
70-30	Random forest	0.84	0.85	0.85	0.85
70-30	Auto encoders	0.79	0.80	0.79	0.79
90-10	Random Forest	0.84	0.85	0.85	0.85
90-10	Auto Encoders	0.79	0.80	0.79	0.79

Table 5 :Model Trained on balanced Dataset

RESULTS

Fig 8: User Interface

RESULTS

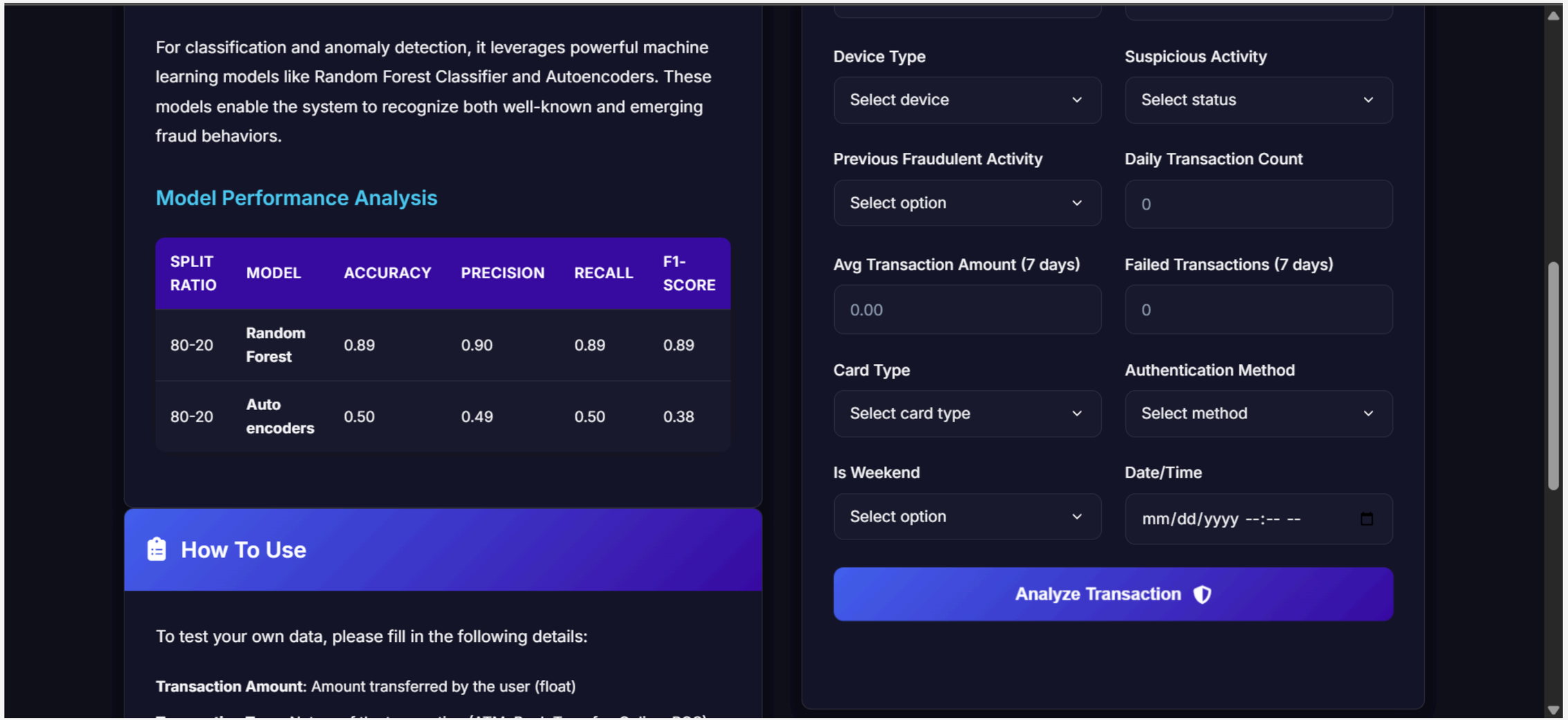

Fig 9 : User Interface

RESULTS

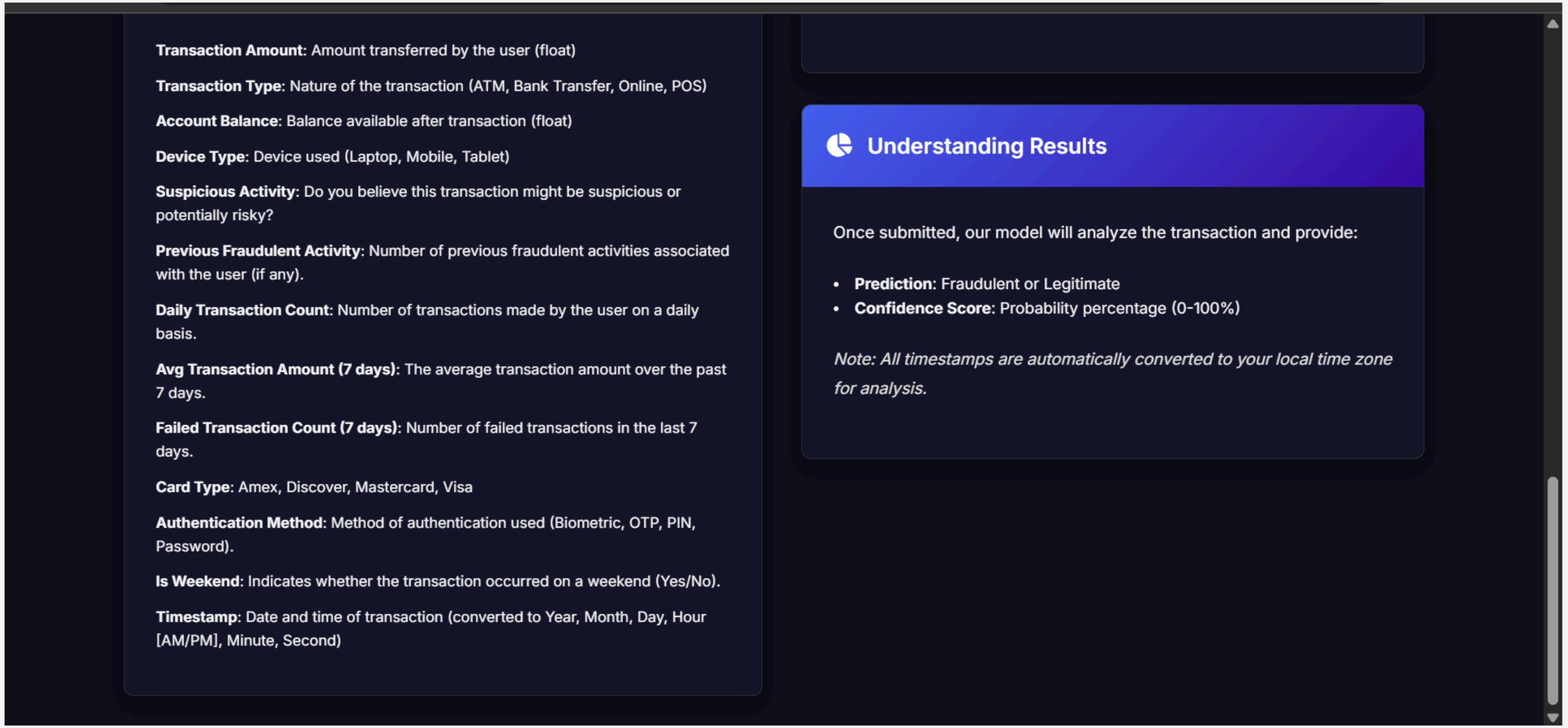

Fig 10 : User Interface

RESULTS

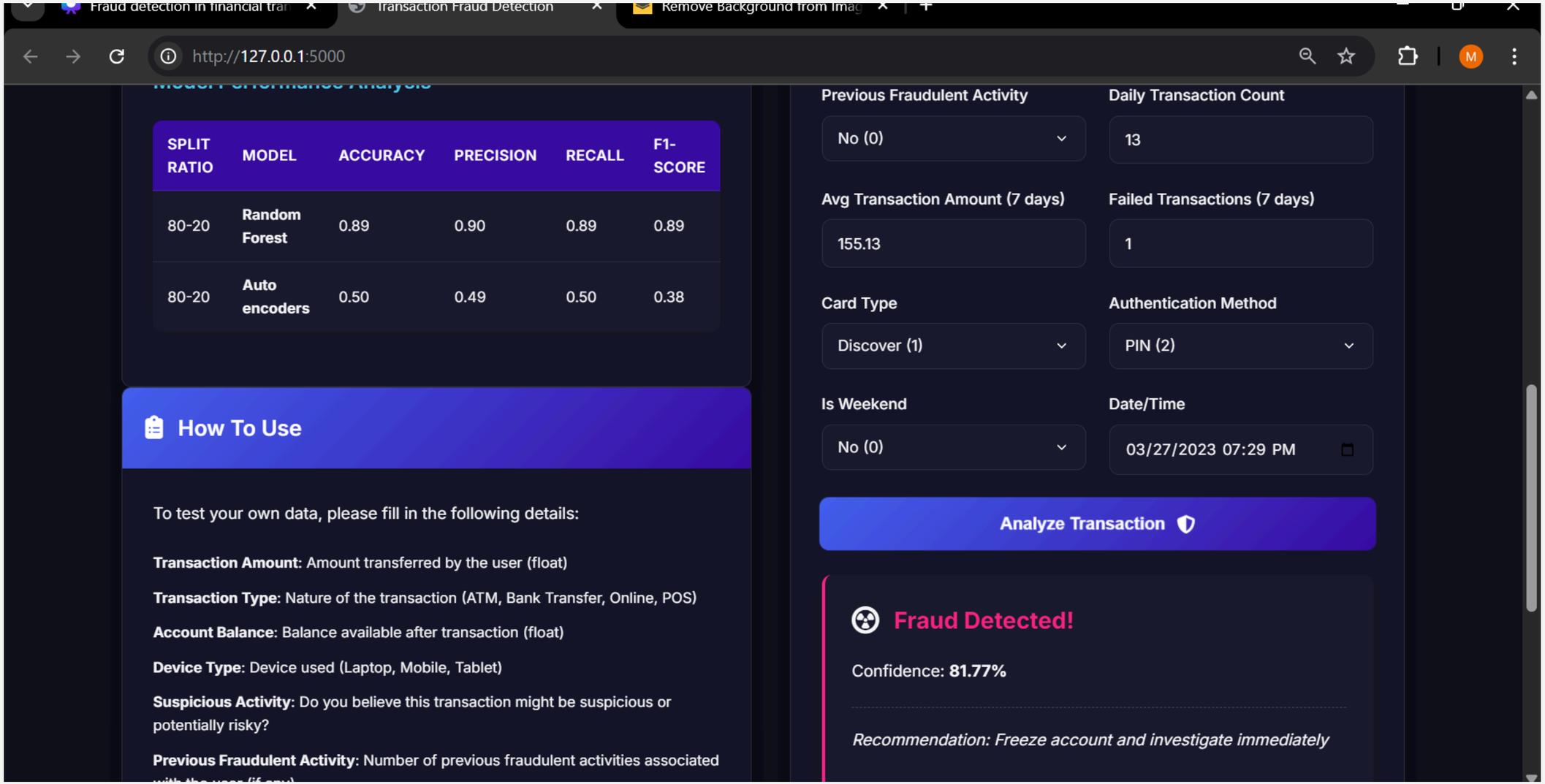

Fig 11 : Fraud Transaction result

RESULTS

SPLIT RATIO	MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
80-20	**Random Forest**	0.89	0.90	0.89	0.89
80-20	**Auto encoders**	0.50	0.49	0.50	0.38

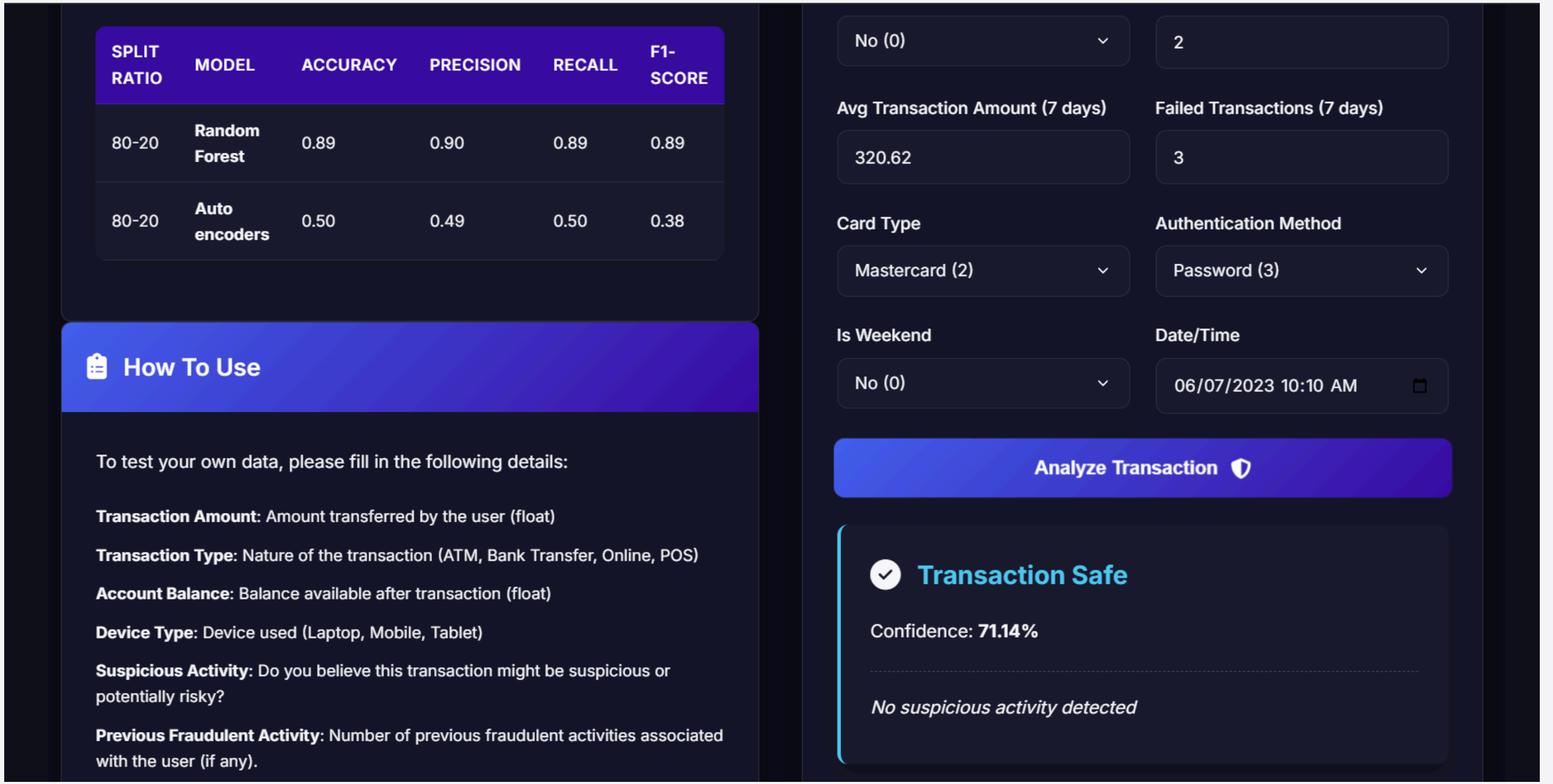

Fig 12 : Safe transaction result

CONCLUSION AND FUTURE SCOPE

- The proposed solution focuses on detecting fraudulent transactions using a robust Random Forest Classifier, trained on a balanced dataset enhanced through CTGAN-generated synthetic data.
- After balancing the highly imbalanced dataset, the model achieved an improved accuracy of **89%**.
- The model shows great potential in identifying fraud patterns effectively.

Future Scope

- Incorporate real-time transaction data training to adapt to evolving fraud patterns dynamically.
- Enhance model accuracy with continuous learning and feature optimization.

REFERENCES

[1] A. C. Ashraf, A. Ali, D. Anand, M. I. Shabiya, and R. T. Paul,Credit Card Fraud Detection using GAN and Feature Engineering, Mar Athanasius College Of ; Engineering, May 2024.

[2]Aishwarya Arora, Arun Prakash Agrawal, Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison,IEEE,January 2023.

[3] Asma Cherif, Arwa Badhi, Heyfa Ammar, Suhair Alshehri, Manal Kalkatawi, Abdessamad Imine, Credit card fraud detection in the era of disruptive technologies: A systematic review, Journal of King Saud University – Computer and Information Sciences, November 2022.

[4] B. R. Gudivaka, M. Almusawi, M. S. Priyanka, and M. R. Dhanda, An Improved Variational Autoencoder Generative Adversarial Network with Convolutional Neural Network for Fraud Financial Transaction Detection, Second International Conference on Data Science and Information System (ICDSIS), 2024.

[5] C. Charitou, S. Dragicevic, and A. d'Avila Garcez, Synthetic Data Generation for Fraud Detection using GANs,University Of London, 2021.

[6] Emilija Strelcenia and Simant Prakoonwit,Generating Synthetic Data for Credit Card Fraud Detection using GANs, Bournemouth University, UK, 2024.

[7] Rashi Jaiswal and Brijendra Singh,Financial Fraud Prevention with Synthetic Data Generation using GAN, AryaBhatta Journal of Mathematics and Informatics,September 2022.

[8] Sourav Verma, Joydip Dhar, Credit Card Fraud Detection: A Deep Learning Approach,ABV-Indian Institute of Information Technology and Management
Gwalior,September 2024.

[9] Sumaya S. Sulaiman, Ibraheem Nadher, Sarab M. Hameed, Credit Card Fraud Detection Using Improved Deep Learning Models, Tech Science Press,January 2024.

[10] Syeda Farjana Farabi, Mani Prabha, Mahfuz Alam, Md Zikar Hossan, Md Arif, Md Rafiqul Islam, Aftab Uddin, Maniruzzaman Bhuiyan, Md Zinnat Ali Biswas, Enhancing Credit Card Fraud Detection: A Comprehensive Study of Machine Learning Algorithms and Performance Evaluation, AL-KINDI Center for Research and Development, London, 2024.

Thank you