A Mini-Project 1 Report on

Proactive Fraud Detection in Financial Transaction using Generative AI

Submitted to the Department of Computer Science & Engineering, GNITS in the partial fulfillment of the academic requirement for the award of B. Tech (CSE) under JNTUH

By

M. Rishitha (23251A05J3)
CH. Devendra (24255A0513)
S. Akhila (24255A0514)

under the guidance of

Mrs. M. Lalitha
Assistant Professor

Department of Computer Science & Engineering
G. Narayanamma Institute of Technology & Science
(Autonomous) (for Women)
Shaikpet, Hyderabad- 500 104.

Affiliated to
Jawaharlal Nehru Technological University Hyderabad
Hyderabad – 500 085
May, 2025

G. Narayanamma Institute of Technology & Science
(Autonomous) (For Women)
Shaikpet, Hyderabad – 500 104.
Department of Computer Science & Engineering

Certificate

This is to certify that the Mini-Project 1 report on "**Proactive Fraud Detection in Financial Transaction using Generative AI"** is a bonafide work carried out by **M.Rishitha (23251A05J3), CH.Devendra (24255A0513), S.Akhila (24255A0514)** in the partial fulfillment for the award of B. Tech degree in Computer Science & Engineering, G. Narayanamma Institute of Technology & Science, Shaikpet, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide

Mrs. M. Lalitha
Assistant Professor

Head of the Department

Dr. A.Sharada
Professor & Head
Department of CSE

G. Narayanamma Institute of Technology & Science
(Autonomous) (For Women)
Shaikpet, Hyderabad – 500 104.

Department of Computer Science & Engineering

Research Center for Data Analytics

Certificate

This is to certify that **M. Rishitha (23251A0J3), CH. Devendra (24255A0513), S. Akhila (24255A0514)** II B. Tech, has successfully completed project work in **"Research Center for Data Analytics"** CSE Department.

The project titled **"Proactive Fraud Detection in Financial Transactions using Generative AI"** that is being submitted in partial fulfillment for the award of B. Tech, Computer Science and Engineering, G. Narayanamma Institute of Technology & Science affiliated to Jawaharlal Nehru Technological University is a record of bonafide work carried out by her in our guidance and supervision.

Supervisor

M. Lalitha
Assistant Professor
Department of CSE

RC Incharge

Dr.G.Malini Devi
Associate Professor
Department of CSE

Head of the Department

Dr. A. Sharada
Professor and Head Department of CSE

Acknowledgements

We would like to express our sincere thanks to **Dr. K. Ramesh Reddy**, Principal, GNITS, for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. K. Rama Linga Reddy**, Professor ETM Dept& Dean Academics, **Dr. M. Seetha**, Professor & Dean R&D,**Dr. N. Kalyani**, Professor CSE Dept. & Dean Innovation and Incubation, GNITS, for all the timely support and valuable suggestions during the period of our project.

We extend our heartfelt gratitude to **Dr. A. Sharada**, Professor & Head, Department of Computer Science and Engineering, GNITS, for unwavering support and invaluable guidance throughout our project, providing timely assistance and insightful suggestions.

We are extremely thankful to **Dr.K.L.S.Soujanya**, Associate Professor and **K.Sindura,** Asst. ProfessorMini-Project 1 coordinators, Department of CSE, GNITS for all the valuable suggestions and guidance during the period of our project.

We are also extremely thankful to our project coordinators **Mrs Ch. Radhika,** Asst.Profand **Mrs G.Amulya,** Asst. Prof., Dept. of CSE, GNITS, for their encouragement and support throughout the project

We are extremely thankful and indebted to our internal guide, **Mrs. M. Lalitha**Assistant Professor, Department of CSE,GNITS forher constant guidance, encouragement, and moral support throughout the project.

Finally, we would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

M. Rishitha (23251A05J3)

CH. Devendra (24255A0513)

S. Akhila (24255A0514)

Abstract

The rise in digital transactions has led to increased financial fraud, causing security risks and monetary losses. Traditional methods struggle to adapt to evolving fraud techniques, making accurate fraud detection essential. Traditional models like Logistic Regression rely on predefined rules and historical data but fail to detect complex fraud patterns. They often produce high false positives, blocking legitimate transactions, or false negatives, missing actual fraud.

The proposed system introduces an AI-driven fraud detection system on a credit card dataset that leverages anomaly detection techniques using Autoencoders and Random Forest classifiers to improve accuracy. Autoencoders can effectively learn normal transaction behaviors and detect anomalies, while Random Forest classifiers provide robust fraud classification. Additionally, Generative AI tools are utilized to generate synthetic fraudulent transactions, enhancing the model's ability to detect new and rare fraud patterns. The model is trained on the Credit Card Fraud Detection Dataset, enabling it to analyze transaction history, frequency, and amount to flag suspicious activities in real-time. The expected outcome enhances fraud detection accuracy, reduces false positives, and strengthens financial security in digital transactions.

Table of contents

SI. No	Topic	Page. No
	Abstract	v
	List of Figures	vii
	List of Tables	viii
1	Introduction	1
	1.1 Background of Study	1
	1.2 Problem Statement	1
	1.3 Existing System	2
	1.4 Challenges in the Existing System	2
	1.5 Proposed System	3
	1.6 Methodology	4
	1.7 Objectives	5
	1.8 Hardware & Software Requirements	5
	1.9 Organization of the Project	5
2	Literature Survey	6
3	Fraud Detection System	14
	3.1 Architecture of the System	14
	3.2 Description of Modules	15
4	Implementation	17
	4.1 Dataset Information	17
	4.2 Technologies Used	18
	4.3 Data Preprocessing	19
	4.4 Synthetic Data Generation	21
	4.4.1 Conditional GAN(CTGAN)	21
	4.4.2 Wasserstein GAN(WGAN)	22
	4.4.3 Synthetic Data Generation GAN (SDG-GAN)	23
	4.4.4 Evaluation of GANs	24
	4.5 Fraud Detection Models	25
	4.5.1 Random Forest	25
	4.5.2 Autoencoders	28
5	Results	30
	5.1 Statistical Comparative Analysis	30
	5.4 Performance Analysis Metrics	33
6	Conclusion and Future Enhancements	38
	References	39

List of Figures

SI. No	Figure No	Figure Name	Page no
1	1.6	Methodology	4
2	3.1	The Architectural Diagram	14
3	4.1	Original Dataset	17
4	4.2	Fraud vs Non-Fraud Class Distribution Bar Chart	20
5	4.3	Preprocessed Dataset	20
6	4.4	Synthetic Data generation Architecture	22
7	4.5	Architecture of Random Forest Classifier	27
8	4.6	Architecture of Auto Encoders	29
9	5.1	CTGAN Statistics graph	30
10	5.2	WGAN Statistics graph	31
11	5.3	SDGGAN Statistics graph	31
12	5.4	User Interface	35
13	5.5	Fraud Transaction Result	36
14	5.6	Safe Transaction Result	37

List of Tables

SI. No	Table No	Table Name	Page No
1	2.1	Literature Survey	11
2	5.1	KS Test	32
3	5.2	Chi-Square Test	33
4	5.3	Performance of CTGAN Balanced Data	33
5	5.4	Performance of Original Processed Data	34

1. INTRODUCTION

The increase in digital payments in India has led to a significant spike in online cheating and security issues. The rate of cyber fraud increased by 400% over the past year, resulting in losses of more than $20 million. 77% of these crimes occurred online, 47% of UPI payments. On average, approximately 800 cases of digital payment fraud are reported daily across the country. It has become increasingly challenging to protect individuals and companies from becoming victims of cheating. It poses the risk of losing a large sum of money. Therefore, it is essential to identify and prevent fraud at its early stages to protect the involved money and uphold trust in online payments. This study will focus on new technologies such as Generative AI to prevent credit card fraud in transactions.

1.1 Background of the study

As more people are using online transaction, there has been a corresponding increase in fraud cases, causing financial losses and security risks. One of the primary challenges the current systems face is handling imbalanced datasets. In financial transactions, fraudulent activities contain a small percentage compared to legitimate ones. This imbalance can lead to models that are biased towards the majority class, resulting in a high rate of false negatives where fraudulent transactions go undetected. Moreover, these systems frequently operate in batch processing modes, leading to delayed detection and response times. The reactive approach not only increases operational costs but also risks significant financial losses and damage to customer trust. It's therefore crucial to have more intelligent and faster-acting protection. New AI tools can study how fraud happens, spot warning signs, and stop cheaters before they succeed.

1.2 Problem Statement

The rapid growth of digital transactions is increasing significantly, presenting a massive challenge to financial institutions fraud detection techniques. Fraudsters continually develop new ways of circumventing detection techniques over time. As a

result, older fraud detection systems cannot effectively handle emerging forms of fraud. Consequently, they may fall short in identifying new fraud patterns, making institutions more vulnerable to modern fraud techniques.

The most unfortunate outcome is that failure to detect fraud may allow genuine fraudulent activities to go unnoticed, leading to financial losses, or may result in legitimate transactions being flagged as fraudulent, thereby inconveniencing customers or diminishing confidence in financial transactions.

1.3 Existing System

Current fraud detection systems usually depend on traditional methods and algorithms such as Logistic Regression to prevent it. These models analyze the transactions history to come up with the fixed patterns that indicate fraud. A key characteristic of these traditional models is their heavy dependence on manual feature engineering, where expert must carefully design feature using domain knowledge to improve model's predictive performance. Although these systems perform reasonably well, they often face challenges when encountered with new types of fraud, or highly imbalanced datasets where instances of fraud are much less frequent than legitimate transactions.

- **Rule-based systems** work using manually defined conditions such as "if a transaction exceeds ₹50,000 from a new location, mark it as suspicious." These systems are simple and fast but cannot adapt to new fraud patterns unless updated manually.
- **Logistic Regression models** are used to calculate the probability of a transaction being fraudulent based on historical data. They require manual feature selection and may not perform well with complex fraud patterns.
- **Imbalanced data structures** are also a major issue. In most datasets, fraudulent transactions are very rare compared to genuine ones. This imbalance makes it difficult for models to learn and identify fraud correctly, often leading to false positives or missed fraud cases.

1.4 Challenges in the Existing System

- Rule-based systems do not automatically learn new fraud patterns. They require regular manual updates and retraining to adapt to new ways of committing fraud. As a result, they are less flexible when confronting new threats. Time can allow for easier instances of fraud.
- The rapid growth of digital transactions is substantial. Each transaction is going to be evaluated through a growing list of rules. Each time, we are going to need more resources to understand the rules and adjust the rules for new fraud.
- They struggle to accurately detect fraud due to the rarity of fraudulent transactions compared to legitimate ones, leading to biased models.
- Manual feature engineering is usually necessary with traditional methods, which takes time and does not always obtain all the required information for detecting fraud.
- As stated in the initial context, these systems often result in a significant number of both false alarms (flagging legitimate transactions) and missed fraudulent activities.

1.5 Proposed system

To address the challenges posed by traditional methods of fraud detection, the solution present hypothesis of an AI-based system. The system addresses the imbalance in transaction data by creating synthetic fraudulent samples using GANs (Generative Adversarial Networks). Subsequently, more advanced machine learning techniques like Random Forest (RF) Classification and Autoencoders (AE) will be used as classifying models to detect anomalies with greater precision and accuracy. The model will be trained with the Credit Card Fraud Detection Dataset so it can detect patterns of fraud in financial transactions regardless of whether they are new or old.

The approach provides several crucial benefits. It enables more effective dealing with imbalanced datasets with the incorporated GAN-based synthetic data generation, which improves the classification models' skills. Moreover, the

application of advanced algorithms of machine learning increases the accuracy of identification of fraudulent transactions and also enhances the ability to identify new and developing patterns of Scams.

The built-in Anomaly detection features of autoencoders and random forest classifiers can reduce the incidence of both false positives and negatives, increasing the reliability of fraud detection systems, especially when applied to detecting new and unseen types of fraud.

To counter the ineffective strategies offered for automated fraud detection, an intelligent framework using Generative AI is proposed for this project. It uses Generative Adversarial Networks (GANs) to produce fraudulent transaction data in order to mitigate the imbalanced biases which exist in datasets. This modified data is then used to train advanced classification models such as autoencoders and random forest classifiers so they can accurately detect sophisticated patterns of fraud within financial transactions. The enhancement provides better agility and resilience for credit card fraud detection and prevention systems.

1.6 Methodology

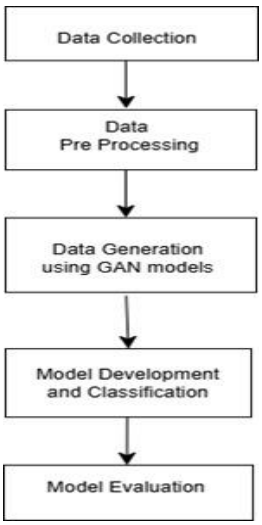

Fig 1.6: Methodology

The methodology, as shown in Figure 1.6**,** begins with preprocessing the Fraud Detection Dataset by removing noise, handling missing values, and normalizing features to ensure data consistency for effective modeling. Next, Generative Adversarial Networks (GANs) are employed to learn the distribution of fraudulent

transactions and generate synthetic data, addressing the class imbalance problem by creating a balanced dataset. After balancing the data, advanced machine learning algorithms like Random Forest Classification and Autoencoders are trained on the combined dataset of genuine and synthetic fraudulent samples to detect fraud patterns. The performance of the models is measured and compared with metrics like Accuracy, Precision, Recall, and F1-Score to ascertain the best method for fraud detection after addressing data imbalance using GANs

1.7 Objectives

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

1.8 Hardware & Software Requirements

- **Hardware Requirements:** Intel Core i7 Processor, Hard Disk of Capacity 1024GB, 16GB RAM, Windows 11
- **Software Requirements:** Python, pandas, numpy, matplotlib, seaborn, scikit-learn, TensorFlow, SDV, Flask, HTML, CSS, JavaScript.

1.9 Organization of the Project

The project is designed to address the increasing issue of online payment fraud by creating an intelligent, proactive system that detects fraud early. First off, we are concerned with gathering and cleaning the data from a real-life Credit Card Fraud Detection Dataset. This data will be used to create our progressive fraud detection models. At the same time, we will utilize GANs to develop fake transaction data for fraudulent transactions. This will help to address the unbalance in our data set. After we have a balance data set, the final classification models we will create will be Random Forest Classification models and Autoencoders that will classify difficult fraudulent patterns. After our classification models are built, we will explore various metrics that will accurately outline their performance.

2. LITERATURE SURVEY

Credit card fraud detection has become more challenging because of the increased number of online transactions. Machine learning and deep learning methods are becoming more common because traditional rule-based systems are unable to adapt to evolving fraud schemes. The survey provides an overview of the latest techniques that generate synthetic fraud data, including Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) as a solution to class imbalance. It assesses the effectiveness of key models such as the (CTGAN),(SDGGAN), and hybrid models for improving detection accuracy. The objective is to formulate generative models for effective fraud detection while addressing key challenges such as validation of real-world data and on-going stream data.

Afeef Ashraf, Ajaz Ali, Anand D, Shabiya M I, Rini T Paul (2024) proposed a GAN-based method for credit card fraud detection [1] to address the data class imbalance problem using CTGAN for synthetic data generation. The overall method includes three main components: the feature engineering to discover an important transactional pattern; the anomaly detection that coded the fraud as either anomalous, and projecting the polarities of the data so that useful synthetic generation could occur for both classes**.** CTGAN synthesized data that was more real than other traditional models(i.e. TabGAN) but kept the relevant statistical characteristics. Supportive results indicated that better quality features had improved model performance which indicated having better feature engineering would improve model performance even when using generative models. The paper also suggested a framework for future research in anomaly detection that would be helpful for other researchers especially in order to be successful with feature engineering and synthetic data augmentation.

Yogesh W Bhowte Arundhati Roy; K. Bhavana Raj, Megha Sharma,K. Devi Prem LathaSoundarraj investigates the effectiveness of supervised machine learning models in identifying fraudulent financial transactions. Utilizing a real-world dataset, the authors developed five models Logistic Regression, Support Vector Machine (SVM), Random Forest, Neural Network, and Boosted Tree following extensive data cleaning, variable creation, and feature selection [13]. Among these models, the Boosted Tree outperformed the others, achieving a Fraud Detection Rate (FDR) of

54.3% on the test set at a 3% cut-off, showcasing its ability to manage imbalanced datasets effectively. This study emphasizes the value of using robust machine learning models, particularly ensemble methods, in constructing reliable and efficient fraud detection systems.

Yinan Cheng, Chi-Hua Wang, Vamsi K. Potluru, Tucker Balch, Guang Cheng, (2024) examines the selection of generative models used in training fraud detection systems with respect to their performance on synthetic data. [14] It analyzes Neural Network models CTGAN and TVAE as well as Data Synthesizer, a Bayesian Network model, for different downstream fraud detection tasks. The research aims at solving two major problems: one focuses on model utility-driven selection (e.g. maximizing accuracy, AUROC, recall, precision, F1-score) while the other focuses on interpretability selection (model choice on the basis of classifier like Logistic Regression or Decision Trees examples). The findings suggest NN models are superior in recall and AUROC, both crucial in identifying scarce fraud cases, while BN models excel in F1 and precision, which minimizes false positives and is supportive of explainability. Classifiers employed include Random Forest, XGBoost, Logistic Regression, NAM while among the paper's contribution is synthetic-augmented training in which real and synthetic data blends impact performance across models differently. In summary, the paper offers defaults strategies targeted towards the design of generative models which align with the requirements crafted by fraud detection tasks.

B.R.Gudivaka, M. Almusawi, P.M.S. Priyanka, and M.R. Dhanda(2024) introduced an improved VAEGAN-CNN hybrid model for credit card fraud detection with the objective of overcoming class imbalance typical of financial transactions.[4] The main advancement is the redefined architecture of VAEGAN that enhances the VAEGAN process by including two encoders (E1 and E2) in parallel, allowing the generative processes to produce more realistic synthetic fraud samples by probabilistically fusing from multiple Gaussian distributions. The improved VAEGAN, which integrates the best of both VAE and GAN, for withholding better minority-class oversampling, followed the dimensionality reduction and normalization preprocessing provided as part of the model. The CNN architecture

incorporated in the model utilizes convolutional and pooling layers that extract discriminative features for classification.

Achieving 99.78% accuracy, 88.97% precision, 95.24% recall, and 95.00% F1-score, the results presented here demonstrate substantial improvement compared to existing solutions. In addition to comparing performance against industry benchmarks including a stand-alone CNN, LightGBM, and explicitly an ensemble of LSTMs, this paper provides descriptive mathematical formulations regarding the mechanics of the probabilistic fusion process. The chronicled and quantified results outline the model's ability of uncovering complex fraud signatures while cautiously reviewing the limitations of traditional approaches such as SMOTE.

Y.M. Ding, W. Kang, J.X. Feng, B. Peng, and A. Yang (2023) [15] address the issue of unbalanced datasets in credit card fraud detection (CFD),proposes better VAEGAN (Variation Auto encoder Generative Adversarial Network) that oversamples minority-class fraudulent data. To improve both the quality and diversity of the synthetic fraud samples the authors propose a dual-encoder frame work intended to provide richer latent representations. The improved VAEGAN model improves on the standard VAEGAN and traditional oversampling methods (e.g., SMOTE, GAN, VAE) when applied to an extreme imbalanced dataset using the following metrics: Precision, Recall and F1-scoreWhileitprovides only limited improvements in recall, it achieves a decrease in underperformance of Precision (0.9478 versus 0.9197 baseline) and F1-score (0.884 versus 0.863 baselines). The use of dual-encoder fusion for better sampling of the latent space improved recall and the robust augmentation demonstrates not big changes in the classifier performance, but improved information diverseness about the same majority-class, provides two significant advancements that continues to take advantage of hybrid generative modes (like VAE + GAN) to be useful in detection of financial fraud.

Aishwarya Arora, Arun Prakash Agrawal, presents a comparative study of supervised machine learning algorithms for the purpose of credit card fraud detection. The authors evaluate models such as Decision Tree, Random Forest, Support Vector Machine (SVM), and Neural Networks on a real-world dataset containing credit card transaction data. Each algorithm is assessed based on performance metrics including accuracy, precision, recall, and F1-score [2]. The

experimental results demonstrate that while all models show potential, certain algorithms perform better in minimizing false positives and detecting fraudulent patterns efficiently. The analysis helps inform the selection of suitable models for deployment in financial fraud detection systems.

Emilija Strelcenia and Simant Prakoonwit (2023) discuss research on synthetic data generation for credit card fraud detection through Generative Adversarial Networks (GANs). The study tackles the problem of imbalanced datasets in fraud detection, where fraud cases are much fewer than genuine transactions. The authors introduce a new K-CGAN model, which uses KL divergence as part of the generator's loss function to improve synthetic data quality [6]. The work compares K-CGAN with the conventional oversampling methods such as SMOTE, B-SMOTE, and ADASYN, and other variants of GAN. Experiments are conducted by training the classifiers such as Random Forest, Logistic Regression, and MLP on original and augmented datasets. Experiments show that K-CGAN performs better than other approaches by registering higher F1 scores. The work indicates the efficiency of GANs in addressing class imbalance and enhancing fraud detection performance. The work concludes with potential future uses of K-CGAN in anomaly detection.

Rashi Jaiswal and Brijendra Singh (2022) discuss how Generative Adversarial Networks (GANs) can be utilized to generate artificial financial data as an answer to deficit and privacy problems of fraud detection. [7] The article describes how common fraud data sets suffers with class imbalance since there is little fraudulent case data and plenty of real instance data, thereby inhibiting the model to function smoothly. The authors use GANs to produce realistic synthetic samples, in this case, for the minority class, and then use them to rebalance a dataset and improve the accuracy of machine learning classifiers. The technique adds synthetic data to model training pipelines, which improves accuracy and boosts fraud detection rates compared to those trained on real, imbalanced datasets. From comparative analysis to performance benchmarks, the research identifies the foremost advantage of utilizing GAN-augmented datasets towards enhancing fraud detection systems. Beyond addressing an essential financial security issue, the work lays the foundation for more extensive application within privacy-sensitive contexts.

Tushar Patil (2021) examined the use of Conditional Tabular GAN (CT-GAN) together with machine learning models like Logistic Regression, Random Forest and XGBoost in a class imbalance context to identify credit card fraud. The study illustrated that CT-GAN was able to generate realistic synthetic minority-class samples and shed light into areas where traditional methods (such as SMOTE) have their limitations. [11] The study utilized the CRISP-DM framework and implemented a conditional generator with WGAN-GP loss so that mode collapse would be avoided with the synthetic minority class that was generated by CT-GAN. Of the models examined, Random Forest performed the best with CT-GAN-enhanced data achieving a 100% recall and also achieving a 100% F1-score. Additional benefits of this approach included that the data probe functionality leveraged open-source Python libraries that allow for scaling. However, since the study was restricted to numerical data, any efforts to process categorical features was left for future use. Overall, the contribution summarized the possibilities of GAN-based augment using CT-GAN, as a technique for generating (augmented) data to help support model performance when dealing with imbalanced datasets.

Charitos Charitou, Simo Dragicevic, Artur d'Avila Garcez (2021) introduced SDG-GAN (Synthetic Data Generation GAN)[5]to address a problem of class imbalance associated with potential fraud detection applications. It is well established that traditional oversampling methods, including the "synthetic minority over-sampling technique" (SMOTE), are not able to sufficiently represent the richness of complex data distributions; SDG-GAN adopted a conditional GAN (cGAN) framework which utilized feature matching loss, with the aim of stabilizing training and avoiding the mode collapse, which is a serious limitation when using GAN methods. The most compelling component of the model is its innovation in a hybrid two-point objective function that utilized binary cross-entropy with feature matching loss, to improve the synthetic data quality. The experimental results using credit card fraudulent detection datasets showed positive results with random forest showing 91.31% F1-score and XGBoost 89.73% accuracy. This method SDG-GAN when implemented in the context of a rule-based system for fraud detection, successfully flagged fraudulent applications and transactions at a much lower rate of false-positives at 5%, which outperformed SMOTE-second and vanilla cGAN.

Table 2.1: Literature Survey

S. No	Title	Publication and Year	Algorithms/Techniques Used	Limitations
1	Credit Card Fraud Detection using GAN and Feature Engineering	2024, Research Square	CTGAN, Feature Engineering, Anomaly Detection	Focuses only on numerical data
2	Advanced Fraud Detection Using Machine Learning Techniques in Accounting and Finance Sector	2024, IEEE	Logistic Regression, SVM, Random Forest, Neural Network, Boosted Tree; Feature selection and variable creation	.Focuses on supervised methods only; lacks exploration of unsupervised or hybrid techniques; performance evaluation limited to a single cutoff (3%) and real-time applicability not assessed.
3	Downstream Task-Oriented Generative Model Selections on Synthetic Data Training for Fraud Detection Models	2024, JP Morgan AI Research	CTGAN, TVAE (Neural Networks), Bayesian Networks, RF, XGBoost, LR and NAM.	Excludes ML-based generative models due to privacy-accuracy trade-offs; focuses on fraud detection training, limiting broader applicability.
4	An Improved Variational Autoencoder Generative Adversarial Network with Convolutional Neural Network for Fraud Financial Transaction Detection	2024, IEEE	Improved VAEGAN (dual parallel encoders), CNN	Discusses limitations of traditional approaches like SMOTE.

5	Credit Card fraud Detection based on improved Variational Autoencoder Generative Adversarial Network	2023, IEEE	Improved VAEGAN (dual-encoder framework)	Limited improvement in recall.
6	Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison	2023, Stanford university	Decision Tree, Random Forest, SVM, Neural Networks; Evaluated using accuracy, precision, recall, F1-score	Limited to traditional supervised models; no use of ensemble or hybrid techniques; does not address class imbalance mitigation strategies explicitly.
7	Generating Synthetic Data for Credit Card Fraud Detection using GANs	2023, Bournemouth university	SDG-GAN (CTGAN with feature matching loss), Random Forest, XGBoost	Discusses computational overhead and the need for hyper parameter tuning. Does not extensively cover categorical features.
8	Financial Fraud Prevention with Synthetic Data Generation using GAN	2022, Arya Bhatta Journal of Mathematics and Informatics	GANs for synthetic data generation.	The research addresses the challenge of analyzing user's real data, which is risky and difficult to access, by generating synthetic temporal data using GANs for fraud detection.

9	Credit Card Fraud Detection Using CT-GAN and Supervised Machine Learning Techniques.	2021, National College of Ireland	Addressing class imbalance in CCF using CT-GAN and supervised ML	Only considers numerical data; treatment of categorical features is for future work.
10	Synthetic Data Generation for Fraud Detection using GANs	2021, University of London	GANs (SDG -GAN)	Addresses the class imbalance problem in fraud detection.

As shown in table 2.1, the survey reviews the shift from static rule-based fraud detection to machine learning (ML) approaches, as rule-based methods struggle with evolving fraud patterns. ML models like Neural Networks, Random Forests, SVMs, and Genetic Algorithms have shown potential in addressing challenges such as class imbalance and the need for real-time detection, with hybrid models suggested for improved performance.

GAN-based methods, including CTGAN and SDG-GAN, effectively generate synthetic minority-class samples to overcome imbalance. Enhanced frameworks like improved VAEGANs and VAEGAN-CNN hybrids demonstrated significant gains in precision, recall, and F1-scores. Validation studies highlighted that while GANs capture marginal distributions well, modelling multivariate relationships remains difficult.

Comparative analysis showed that neural network-based models are better for recall and AUROC, while Bayesian network-based models perform well in precision and F1. The survey concludes that combining synthetic data generation, effective validation, and advanced hybrid models can significantly strengthen fraud detection systems

3. PROPOSED SYSTEM

The proposed system is designed not only to detect fraudulent transactions with greater accuracy but also to act as a proactive tool in mitigating financial fraud. By continuously synthesizing updated fraudulent transaction data through GANs, the system can adapt to emerging fraud techniques. This balanced data is then utilized to train advanced classification models, including Random Forest Classification and Autoencoders, enabling them to learn intricate fraud patterns and improve detection accuracy in financial transactions. The system aims to provide a more robust and adaptive solution for proactively identifying and mitigating credit card fraud.

3.1 Architecture of the System

As shown in Fig 3.1**,** the architectural diagram outlines a step-by-step workflow where an initial imbalanced dataset undergoes augmentation using GAN models (CTGAN/WGAN/SDGAN) to produce a balanced synthetic dataset. Both the original and synthetic datasets are then used to train and evaluate classification models (Random Forest, Auto Encoders, Logistic Regression). The performance of these models is compared using metrics like accuracy, precision, recall, and F1 score to determine the final fraud detection model for outputting a prediction of "Fraud" or "Not Fraud".

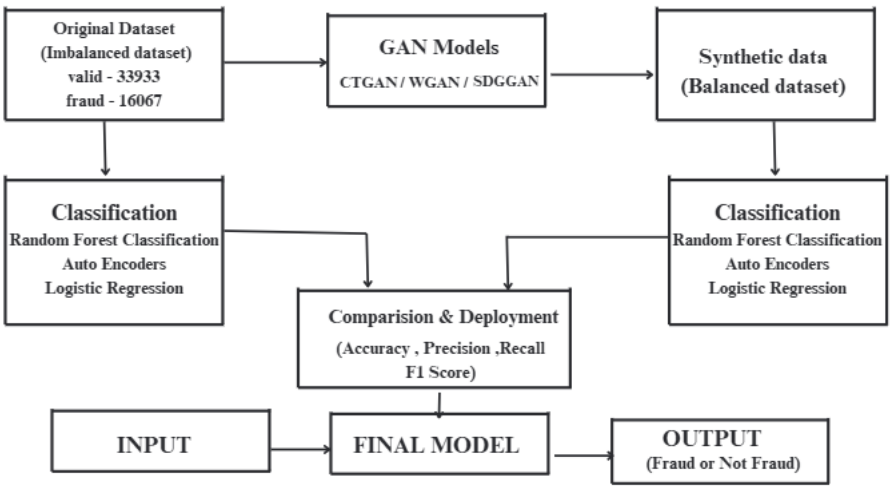

Fig 3.1: The Architectural Diagram

3.2 Description of Modules

The project is divided into four key modules, each addressing a crucial step in the fraud detection pipeline—from data collection and preprocessing to model training, synthetic data generation, and performance evaluation.

3.2.1 Collect and preprocess the Fraud Detection Dataset.

In the initial phase, we gathered raw transaction data, for instance, a dataset comprising 50,000 transactions obtained from Kaggle. Following the collection, preprocessing phase involved critical steps to make the data for ready for analysis and model training. This included data cleaning by handling any missing values or outliers which might affect the model results.

Furthermore, we performed feature engineering to generate more useful features from the existing ones, possibly extracting new insights relevant to fraud detection. Finally, data scaling or normalization techniques was used to guarantee that all the features contribute equally to the model training process so that features with larger ranges do not dominate. The original dataset tends to have a high-class imbalance, which necessitates the following module.

3.2.2 Generate synthetic fraud data to balance the dataset using GAN models.

For solving the class imbalance problem highlighted in Module 1, the module uses GAN models, i.e., CTGAN, WGAN, and SDGAN. These advanced generative models learn the hidden patterns of the minority class (fraudulent transactions) and generate synthetic data points that mimicking actual fraud cases. The goal is to generate a more balanced dataset, which can improve the training and performance of the classification models in the next module.

3.2.3 Develop a Machine Leaning (ML) model to classify transactions as fraudulent or legitimate.

In the module, ML models are developed to differentiate between fraudulent and genuine transactions. The workflow tries out many classification algorithms like RF classifier, AEs, and Logistic Regression. These models train on both original imbalanced dataset as well as newly generated balanced synthetic datasets. By training on a more balanced dataset, the models are expected to learn the patterns of fraudulent transactions more effectively, which will result in improved classification performance.

3.2.4 Evaluate the model's performance using evaluation metrics.

The final module deals with evaluating performance of trained classification models. The performance is evaluated using key metrics such as accuracy (overall correctness), precision (ability to correctly identify fraudulent transactions), recall (ability to identify all actual fraudulent transactions), and F1-score (the harmonic mean of precision and recall). These metrics provide a comprehensive understanding of the model's ability to detect fraud while minimizing false alarms. The best-performing model, based on these evaluations, is then selected as the final model for deployment.

4. IMPLEMENTATION

4.1 Dataset Information

The dataset used in this project is a synthetic financial transactions dataset. The dataset is considerably imbalanced; there are very few instances of fraud compared to legitimate transactions. Each record has several features regarding the transaction itself (i.e., amounts; customer information; risk features). In the target variable, the label Fraud_Label is used where 0 is a legitimate transaction - or non-fraud - and 1 is a fraudulent transaction. When using financial datasets, privacy issues are of utmost importance, and synthetic data generation means no real customer information was compromised and still using realistic behavioral patterns.

- The dataset was split into 80% for training and 20% for testing to create and evaluate models.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

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Transaction_ID	User_ID	Transaction	Transaction_Type	Timestamp	Account_B	Device_Ty	Location	Merchant_	IP_Address	Previous_F	Daily_Tran	Avg_Trans	Failed_Tra	Card_Type	Card_Age	Transactio	Authentica	Risk_Score	Is_Weeker	Fraud_Label
2	TXN_33553	USER_183	39.79	POS	14-08-2023 19.30	93213.17	Laptop	Sydney	Travel	0	0	7	437.63	3	Amex	65	883.17	Biometric	0.8494	0	0
3	TXN_9427	USER_787	1.19	Bank Transfer	07-06-2023 4.01	75725.25	Mobile	New York	Clothing	0	0	13	478.76	4	Mastercar	186	2203.36	Password	0.0959	0	1
4	TXN_199	USER_273	28.96	Online	20-06-2023 15.25	1588.96	Tablet	Mumbai	Restauran	0	0	14	50.01	4	Visa	226	1909.29	Biometric	0.84	0	1
5	TXN_12447	USER_261	254.32	ATM Withdrawal	07-12-2023 0.31	76807.2	Tablet	New York	Clothing	0	0	8	182.48	4	Visa	76	1311.86	OTP	0.7935	0	1
6	TXN_39489	USER_201	31.28	POS	11-11-2023 23.44	92354.66	Mobile	Mumbai	Electronics	0	1	14	328.69	4	Mastercar	140	966.98	Password	0.3819	1	1
7	TXN_42724	USER_685	168.55	Online	05-06-2023 20.55	33236.94	Laptop	Tokyo	Restauran	0	0	3	226.85	2	Discover	51	1725.64	OTP	0.0504	0	0
8	TXN_10822	USER_505	3.79	POS	07-11-2023 1.18	86834.18	Tablet	London	Restauran	0	0	2	298.35	2	Mastercar	168	3757.19	Password	0.0875	0	0
9	TXN_49498	USER_466	7.08	ATM Withdrawal	25-02-2023 3.43	45826.27	Tablet	London	Restauran	0	0	3	164.38	4	Discover	182	1764.66	Biometric	0.5326	0	1
10	TXN_4144	USER_158	34.25	ATM Withdrawal	09-03-2023 22.51	94392.35	Tablet	Tokyo	Clothing	0	0	7	90.02	3	Visa	24	550.38	Biometric	0.1347	1	0
11	TXN_36958	USER_949	16.24	POS	20-09-2023 17.27	91859.97	Mobile	Mumbai	Travel	0	0	6	474.42	1	Mastercar	124	720.91	PIN	0.3394	0	0
12	TXN_43106	USER_284	367.5	POS	11-04-2023 7.11	14640.09	Laptop	Mumbai	Electronics	0	0	4	397.58	0	Amex	136	292.36	PIN	0.643	0	0
13	TXN_38695	USER_686	50.44	ATM Withdrawal	06-08-2023 8.22	19962.22	Laptop	London	Travel	0	0	14	278.55	1	Discover	131	3993.62	PIN	0.4582	0	0
14	TXN_6188	USER_672	55.5	ATM Withdrawal	20-07-2023 3.10	89664.63	Laptop	London	Groceries	0	0	6	483.58	4	Discover	192	3721.54	OTP	0.5837	0	1
15	TXN_1414	USER_665	54.09	Bank Transfer	17-11-2023 20.13	51287.15	Mobile	Sydney	Electronics	0	0	11	152.63	2	Amex	91	1061.27	PIN	0.1454	0	0
16	TXN_18471	USER_765	9.66	Online	20-06-2023 17.15	12420.17	Tablet	New York	Electronics	0	0	6	173.97	0	Mastercar	5	762.21	OTP	0.0793	1	0
17	TXN_29282	USER_124	64.78	POS	03-07-2023 9.20	23487.76	Laptop	Mumbai	Restauran	0	0	1	17.85	2	Amex	3	3378.44	Password	0.2245	0	0
18	TXN_15177	USER_934	37.11	POS	21-12-2023 8.48	92977.91	Tablet	Sydney	Electronics	0	0	12	457.42	3	Visa	135	395.36	Biometric	0.1136	1	0
19	TXN_34304	USER_941	1.58	Bank Transfer	08-04-2023 9.13	63076.36	Laptop	New York	Clothing	0	0	13	309.49	1	Amex	47	2074.77	Biometric	0.3433	0	0
20	TXN_12609	USER_586	178.56	POS	16-08-2023 10.01	62359.52	Tablet	Tokyo	Groceries	0	0	4	491.21	4	Visa	29	2159.31	Password	0.444	1	1
21	TXN_12144	USER_593	19.37	Bank Transfer	29-04-2023 4.34	34416.01	Tablet	London	Travel	0	0	9	163.5	3	Discover	129	4436.61	PIN	0.2924	0	0
22	TXN_6113	USER_556	22.02	ATM Withdrawal	28-12-2023 4.45	55851.38	Mobile	Sydney	Electronics	0	0	13	179.73	3	Visa	204	633.99	PIN	0.1305	1	0
23	TXN_15908	USER_461	203.97	ATM Withdrawal	18-06-2023 23.36	73616.91	Mobile	New York	Restauran	1	0	4	482.2	4	Mastercar	173	4723.22	Password	0.8633	1	1
24	TXN_821	USER_383	55.35	POS	11-05-2023 16.15	88292.27	Tablet	Sydney	Electronics	0	0	12	283.43	0	Visa	80	625.67	PIN	0.7883	1	0
25	TXN_15118	USER_975	17.17	ATM Withdrawal	25-11-2023 7.35	71983.33	Mobile	Sydney	Electronics	0	0	2	36.46	2	Mastercar	165	3402.37	PIN	0.0233	0	0
26	TXN_13466	USER_977	72.33	ATM Withdrawal	14-08-2023 21.55	29954.64	Mobile	Sydney	Clothing	0	0	3	233.6	4	Discover	120	3956.54	Password	0.408	1	1
27	TXN_26497	USER_672	188.75	Online	16-07-2023 7.49	58078.66	Mobile	Sydney	Travel	0	1	4	495.1	0	Amex	85	3935.53	Password	0.1379	0	0
28	TXN_42111	USER_308	2.61	Bank Transfer	11-01-2023 0.41	51803.62	Tablet	Tokyo	Groceries	0	0	1	382.67	2	Visa	86	4963.49	Biometric	0.5355	0	0
29	TXN_30188	USER_798	195.76	Online	28-11-2023 19.13	85000.68	Tablet	Mumbai	Travel	0	1	14	226.9	1	Mastercar	65	4045.87	PIN	0.2553	1	0

Fig 4.1 Original Dataset

4.2 Technologies Used

Python

Python was the sole programming language used in the entire project. All data preprocessing tasks, including data preprocessing and synthetic data generation using machine-learning models as well as evaluation and visualization of machine-learning models were completed using Python because it is a simple yet highly effective programming language that offers an abundance of libraries, which have strong statistical and data science components.

Pandas

Pandas was heavily used for reading the dataset (read_csv), handling any null values, inspecting the structure of the data set, and manipulating the data for things like feature selection; renaming columns and feature engineering, and preparing the data for training and testing purposes.

NumPy

Used NumPy when we needed to apply numpy operations on arrays and matrices. NumPy is especially handy and relevant to ensure that we efficiently handle manipulations and preparations to the data before feeding the data into models or generating Learner-GAN's.

Matplotlib and Seaborn

Matplotlib and Seaborn were used to complete visualizations on the data. Several data visualizations were employed, including confusion matrices, bar plots, and correlation heat maps. These helped us visually evaluate our features, determine if there were any visual anomalies with the distribution of our data, and evaluate models with similar visualizations.

Scikit-learn

Utilized train_test_split to separate the dataset into training (80%) and testing (20%) set and computed evaluation metrics (accuracy, precision, recall, F1-score, precision/recall, ROC-AUC score, confusion matrix), all from the relevant metric functions that Scikit-learn gave us. We got feature scaling from scikit-learn too (e.g., standard scaler) if we decided to normalize the data.

Google Colab

Google Colab was utilized as a development environment, which allowed free access to GPU/TPU support to add faster training for GAN models, in addition to speed up the processes from preprocessing to processing to deployment of the model.

Flask

Flask is a light-weight framework for web applications that empowered the fraud model to operate as a web service. Therefore, when the model is finished it can be executed in real applications and detect fraud immediately for companies.

4.3 Data Preprocessing

Data Loading and Exploration

- The dataset was imported using Pandas.
- Exploratory Data Analysis (EDA) was performed to observe feature distributions, missing values, class imbalances, and basic descriptive statistics.

Addressing Missing Entries

- Missing values were checked using df.isnull(). sum().
- Since no missing entries were detected, no imputation was necessary.

Handling Duplicated Entries and Removing Unwanted Columns

- Duplicate records were identified and removed from the dataset.
- Non-essential columns such as Transaction_ID, User_ID, Merchant_Category, Card_Age, Risk_Score, Location, and Timestamp (after feature extraction) were dropped to reduce noise and redundancy.

Feature Engineering

- The Timestamp column was decomposed into separate temporal features: Year, Month, Day, Hour, Minute, and Second.
- After feature extraction, the original Timestamp column was removed.

Encoding Categorical Features

- Label Encoding was applied to convert categorical variables (Device_Type, Card_Type, Authentication_Method, and Transaction_Type) into numeric format, making them suitable for machine learning models.

Scaling Numerical Features

- Numerical features were scaled using appropriate techniques like StandardScaler to normalize the data and improve model performance.

Analyzing Class Imbalance

- The distribution of the target variable Fraud_Label was examined, revealing a significant imbalance between fraudulent and legitimate transactions.

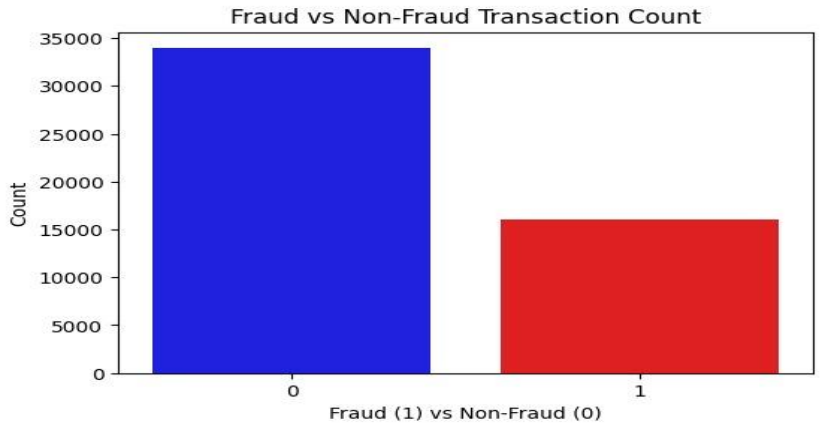

Fig4.2: Fraud vs. Non-Fraud Class Distribution Bar Chart

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Transaction_Amount	Transaction_Type	Account_Balance	Device_Type	IP_Address	Previous_Fr	Daily_Tran	Avg_Trans	Failed_Tra	Card_Type	Transactio	Authentica	Is_Weeker	Fraud_Label	Year	Month	Day	Hour	Minute	Second
2	39.79	3	93213.17	0	0	0	7	437.63	3	0	883.17	0	0	0	2023	8	14	19	30	0
3	1.19	1	75725.25	1	0	0	13	478.76	4	2	2203.36	3	0	1	2023	6	7	4	1	0
4	28.96	2	1588.96	2	0	0	14	50.01	4	3	1909.29	0	0	1	2023	6	20	15	25	0
5	254.32	0	76807.2	2	0	0	8	182.48	4	3	1311.86	1	0	1	2023	12	7	0	31	0
6	31.28	3	92354.66	1	0	1	14	328.69	4	2	966.98	3	1	1	2023	11	11	23	44	0
7	168.55	2	33236.94	0	0	0	3	226.85	2	1	1725.64	1	0	0	2023	6	5	20	55	0
8	3.79	3	86834.18	2	0	0	2	298.35	2	2	3757.19	3	0	0	2023	11	7	1	18	0
9	7.08	0	45826.27	2	0	0	3	164.38	4	1	1764.66	0	0	1	2023	2	25	3	43	0
10	34.25	0	94392.35	2	0	0	7	90.02	3	3	550.38	0	1	0	2023	3	9	22	51	0
11	16.24	3	91859.97	1	0	0	6	474.42	1	2	720.91	2	0	0	2023	9	20	17	27	0
12	367.5	3	14640.09	0	0	0	4	397.58	0	0	292.36	2	0	0	2023	4	11	7	11	0
13	50.44	0	19962.22	0	0	0	14	278.55	1	1	3993.62	2	0	0	2023	8	6	8	22	0
14	55.5	0	89664.63	0	0	0	6	483.58	4	1	3721.54	1	0	1	2023	7	20	3	10	0
15	54.09	1	51287.15	1	0	0	11	152.63	2	0	1061.27	2	0	0	2023	11	17	20	13	0
16	9.66	2	12420.17	2	0	0	6	173.97	0	2	762.21	1	1	0	2023	6	20	17	15	0
17	64.78	3	23487.76	0	0	0	1	17.85	2	0	3378.44	3	0	0	2023	7	3	9	20	0
18	37.11	3	92977.91	2	0	0	12	457.42	3	3	395.36	0	1	0	2023	12	21	8	48	0
19	1.58	1	63076.36	0	0	0	13	309.49	1	0	2074.77	0	0	0	2023	4	8	9	13	0
20	178.56	3	62359.52	2	0	0	4	491.21	4	3	2159.31	3	1	1	2023	8	16	10	1	0
21	19.37	1	34416.01	2	0	0	9	163.5	3	1	4436.61	2	0	0	2023	4	29	4	34	0
22	22.02	0	55851.38	1	0	0	13	179.73	3	3	633.99	2	1	0	2023	12	28	4	45	0
23	203.97	0	73616.91	1	1	0	4	482.2	4	2	4723.22	3	1	1	2023	6	18	23	36	0
24	55.35	3	88292.27	2	0	0	12	283.43	0	3	625.67	2	1	0	2023	5	11	16	15	0
25	17.17	0	71983.33	1	0	0	2	36.46	2	2	3402.37	2	0	0	2023	11	25	7	35	0
26	72.33	0	29954.64	1	0	0	3	233.6	4	1	3956.54	3	1	1	2023	8	14	21	55	0
27	188.75	2	58078.66	1	0	1	4	495.1	0	0	3935.53	3	0	0	2023	7	16	7	49	0
28	2.61	1	51803.62	2	0	0	1	382.67	2	3	4963.49	0	0	0	2023	1	11	0	41	0

Fig 4.3: Preprocessed Dataset

4.4 Synthetic Data Generation

4.4.1 Conditional GAN (CTGAN)

CTGAN is a type of Generative Adversarial Network designed specifically for generating realistic synthetic tabular data, especially effective in handling imbalanced datasets with discrete and continuous features.

The following steps were followed to build and train the CTGAN model

- Loaded the Fraud dataset which has already been prepared (cleaned and ready for balancing).
- Installed and imported libraries needed:
    - Import sdv.tabular.CTGAN from sdv tabular, so we can generate synthetic data.
    - Pandas for data handling.
    - Numpy for math and numerical operations.
    - Used other helper libraries for data handling and evaluation.
- Split the dataset into:
    - Features (X): All columns except the target
    - Target (y): The indicator for fraud (Class).
- Trained the CTGAN Synthesizer:
    - Focused on the minority class (the fraud samples), Learned the distribution of the minority class features to generate synthetic samples across the features in a similar fashion to how fraudulent transactions might appear.
- Synthetically generated samples:
    - Used the trained CTGAN to create an adequate number of synthetic fraud samples to achieve balance in the dataset (i.e., make the fraud and non-fraud samples approximately equal).
- Combined/merged new synthetic data with the existing real-world data:
    - Combined original real data (both fraud and non-fraud) with new synthetic fraud samples to create a balanced dataset. Using this now will provide effective model training with a resultant model not biased toward the majority class**.**

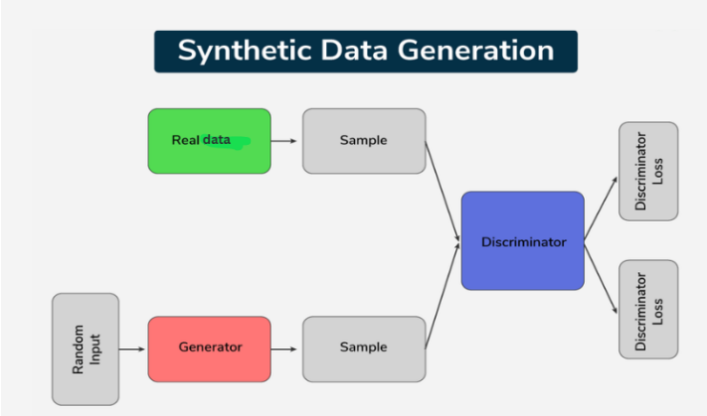

Fig 4.4: Synthetic Data generation Architecture

4.4.2 Wasserstein GAN (WGAN)

WGAN is an improved GAN variant that uses the Wasserstein distance as a loss metric to generate more stable and higher-quality synthetic data, especially helpful in handling imbalanced datasets like fraud detection.

The following steps were followed to build and utilize the WGAN model

- Generated latent vectors
- Random noise vectors have been created by sampling random noise vectors sampled from a standard normal distribution. These latent vectors are the inputs to the WGAN generator for generating synthetic fraud-like data.
- Passed the noise to the trained WGAN Generator as the input values
- The WGAN generator model had been trained for fraud detection. The generator mapped these latent vectors into the feature space of a fraud transaction and produced synthetic fraud samples.
- Reformatted generated data to a Pandas Data Frame
- The synthetic data was generated as a NumPy array in the previous steps. We reformatted this NumPy array into a Pandas Data Frame by using the original fraud feature names to keep the column names the same.
- Split continuous and categorical columns From the generated synthetic data we created columns for continuous features (numerical values).Categorical features (such like labels or discrete values) were assigned as well, but different treatments were needed.

- Specifically, there was a need to reverse transformation for each column type.
- Applied inverse transformations
- For the continuous features, we used the original feature scaler (either a StandardScaler or MinMaxScaler) that had previously been fitted to the real data, and applied an inverse_transform to the continuous features to undo the normalization and scale the synthetic data back to a realistic value.
- Recombined the continuous and categorical features
- After inverse scaling, we were able to recombine the continuous and categorical features back together. This procedure maintained the real data in structure and distribution style for every synthetic sample.
- Merged the synthetic fraud transactions generated with the original real data, which is now the file with synthetic fraud with the real non-fraud transactions. Dataset has been generated with equal number of frauds and non-frauds.

4.4.3 Synthetic Data Generation GAN (SDG-GAN)

SDGAN is a GAN variant that leverages both labeled and unlabeled data in a semi-supervised manner with denoising autoencoders. It can effectively generate synthetic fraud samples, even when labeled data is scarce.

The following steps were followed to build and utilize the SDGAN model

- Loaded the processed fraud dataset
- Imported the cleaned, processed dataset for fraud detection, the features were already selected and scaled. Verified the dataset.
- Normalized data from [-1, 1] and Partitioned into Fraud/Non-fraud
- Normalized the data by applying all the feature values between -1 and 1. This scaling is ideal for training the GAN.
- Parted the dataset into two portions:
    1. Fraudulent transactions (minority class)
    2. Non-fraudulent transactions (majority class)
- Created synthetic fraud samples from noise

- Created random noise vectors to input into the SDGAN Generator. The generator was able to learn to generate realistic synthetic fraud transactions using noise vectors.
- Classify real and fraud samples
- Used the Discriminator network to classify real (authentic) from fake (synthetic) fraud samples. This classification allowed the Generator to improve its sample style over time.
- Engaged in alternative training (5,000 epochs) for about 5000 epochs iteratively trained the generator and discriminator networks.
- The idea is to train the generator and discriminator in a balanced way that the generator is not dominating the training or vice-versa, which is important for relatively stable training of GANS.
- Generated synthetic fraud cases; dealing with the class imbalance
- After having trained the GAN, produced a large amount of synthetic fraud samples to characterize the fraud cases. These synthetic cases were then used to balance the fraud to nonfraud ratio in the dataset.
- Denormalized and combined with original data to create a balanced final dataset. Combined the synthetic fraud samples with the original data,

4.4.4 Evaluation of GANS

After conducting a complete analysis of the synthetic datasets obtained from each CTGAN, SDGAN, and WGAN approach, it is possible to formulate the following in-depth observations:

Distribution Similarity

The analysis of the absolute log means and standard deviation plots for both the real and synthetic datasets highlighted that CTGAN has distributions and means that were most accurately aligned to the diagonal line indicating very little difference; thereby confirming a high level of statistical stability and outstanding replication of the original dataset's distribution patterns.

TableEvaluator Evaluations

Through the distributions and correlations graphing that was provided in TableEvaluator it was possible to visualize and determine that CTGAN had the most

successful aligned distributions, where the real and synthetic distributions and features almost perfectly overlapped.

SDGAN was good, though it had slight variations in some features indicating that there was still a small amount of lack of quality in the synthetic dataset. WGAN had substantial deviations, especially in the standard deviation plots - ultimately displaying a relatively low ability to systematically replicate the feature spread of the real data for WGAN.

Kolmogorov-Smirnov (KS) Test

The KS Test results solidified the preceding visualizations proving that CTGAN had lower KS statistics for continued features thus supporting that the statistical distance of the CTGAN synthetic data was equivalent to the statistical distance of the actual data.

Chi-Square Test

As with the chi-square analysis with categorical feature distributions, CTGAN also had the lowest chi-square values. This demonstrates that CTGAN demonstrated compliance against continued feature distributions and modeled categorical relationships far greater than WGAN and SDGAN.

4.5 Fraud Detection Models

4.5.1 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees. It’s a powerful model for classification tasks, such as fraud detection, as it can handle large datasets with complex relationships.

The following steps were followed to implement and train the Random Forest model:

- Loaded and Prepare the Dataset

- At first, Loaded the balanced dataset created with CTGAN for modeling. The features were properly prepared (continuous vs. categorical, no missing values) to ensure they were in a suitable condition for the model to perform well.
- Create features (X) and target (y) and split dataset for training and testing
- In the second step, the dataset was separated in order for models to utilize independent variables (features, X) and dependent variable (target label, y, fraud/non-fraud). Then the dataset was split into a training and test set in an 80:20 ratio so that the model would be able to generalize well to unseen data while still have enough data to learn complex patterns.
- Specify and Initialize the Random Forest Model
- The third step was to specify and initialize the Random Forest Classifier. As well as some sweet tuning for hyper parameters to maximize performance:
    - Number of estimators (trees) = 500
    - Max depth of trees = 30
    - Class weights = balanced (to minimize any residual class imbalance so that we can have fair fraud detection).
- Fit the Random Forest Classifier
- The Random Forest model was fit to the 80% training data. At this stage, many decision trees are constructed on bootstrapped subsets of the dataset. With the ensemble of trees, the Random Forest model is able to create a strong model that avoids over fitting and is suited for fraud detection by capturing fraud patterns of importance in the model training.
- Making Predictions

    When the model was trained, the next step was to use the model to predict fraud labels (classes) and corresponding probabilities for the unseen 20% test dataset. In this process, we acquired both hard classifications (fraud vs. non-fraud) and prediction probabilities.
- Performance Evaluation of Model

    The model performance was fully evaluated using a number of classification metrics that included:
    - Accuracy Score (overall accuracy)

- Precision Score (correct positive predictions / total positive predictions)
- Recall Score (correct positive predictions / actual positives)
- F1 Score (the harmonic means of precision and recall)
- ROC-AUC Score (perfectly distinguish between fraud and non-fraud

- Visualization and Validation

To visualize the model's performance and validate the effectiveness of the model a confusion matrix was plotted which allowed the visualization of the correct and incorrect classifications.

The reason to plot the ROC curve was to understand the tradeoffs between true positive rate and false positive rate, to show argue that the model was solid.

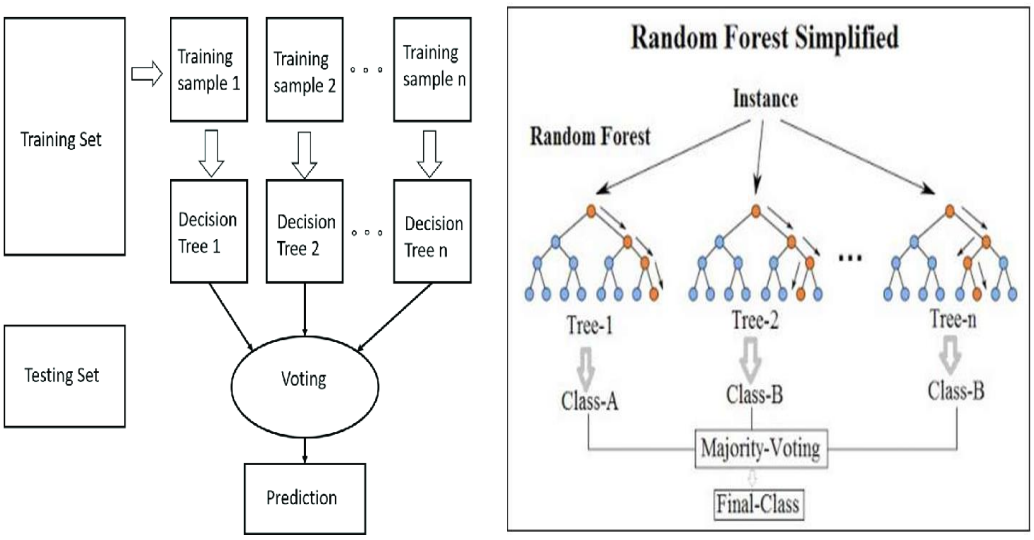

Fig 4.5: Architecture of Random Forest Classifier

4.5.2 Autoencoder

Autoencoders are unsupervised neural networks used for anomaly detection. They learn a compressed representation of input data and are useful for detecting outliers, such as fraudulent transactions, by reconstructing the input data and measuring the reconstruction error.

The following steps were followed to implement and train the Autoencoder model

- Loading and Pre-processing the Dataset
- Before training the anomaly detection model, the balanced dataset was loaded. The model is a neural network (meaning that is sensitive to the magnitudes of the features) so care was taken to scale all feature columns appropriately.
- Splitting into Feature (X) and Target (y)
- Next, we split the dataset into feature variables (X) and the target labels (y). The split was at 80:20 so that the model could be trained and evaluated on unseen data (to reducing the chances of over fitting).
- Create and Fit the Autoencoder

  An autoencoder neural network was created to learn the structure of normal (meaning not fraudulent) transactions. The autoencoder model was trained using Mean Squared Error (MSE) loss in order to see how well it could reconstruct normal transactions. The reasoning behind this approach was to have the autoencoder do well on normal data but poorly on funds involved in fraudulent activity. The logic being that the model had never seen anomalous data, or fraud, in the past.
- Reconstructing Error Calculation and Threshold Establishment

  The reconstruction errors have then been calculated for each transaction through the testing phase of the autoencoder. At the 95th percentile for the reconstruction error calculated from normal transactions (without fraudulent activities), it was set as the threshold.
- Transactions that incurred reconstruction errors exceeding this agreed-upon threshold are considered potential fraud.

- Evaluating the Efficiency of the Autoencoder

Overall, this is illustrative of how well the autoencoder does in recognizing definite frauds but still not catching a larger number of fraudulent cases.

Though the autoencoder model showed very high precision scores in fraud detection, it's extremely low recall values indicated a dismal coverage in detecting all fraud cases. However, Random Forest trained on CTGAN balanced dataset did show a better balance of precision, recall and accuracy which gave it greater reliability and practicality for fraud detection in this project.

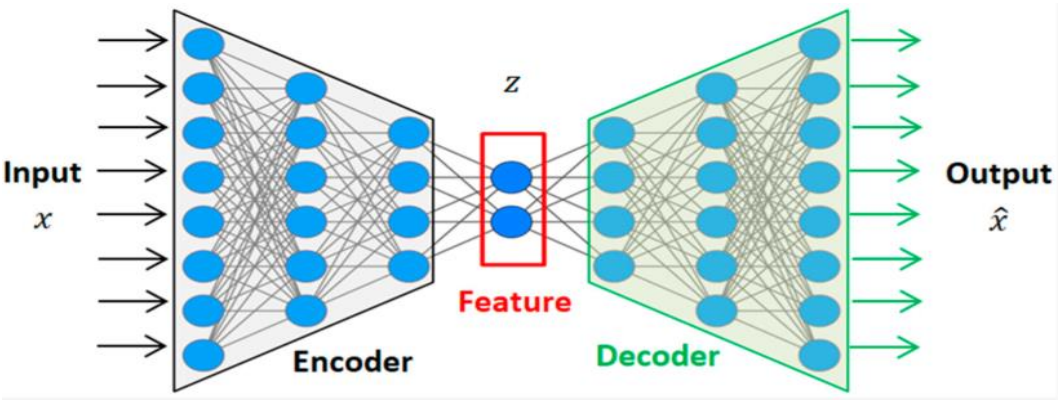

Fig 4.6: Architecture of Auto Encoders.

5. RESULTS

5.1 Statistical Comparative Analysis

The visualization graphs allow for qualitative evaluation among the three GAN models, CTGAN, WGAN, and SDG-GAN, in terms of statistical representation of the real data. By plotting the log means and log standard deviations of the synthetic data versus the real data, the plots showed how well each of the models preserved the central tendency and variability of the data. The diagonal line (y=x) represents perfect reproduction and serves as a baseline for judgment. The closer the estimation is to the line, the better the statistical fit. The more that the estimation diverges, the more limited the fit.

CTGAN Statistical Graph

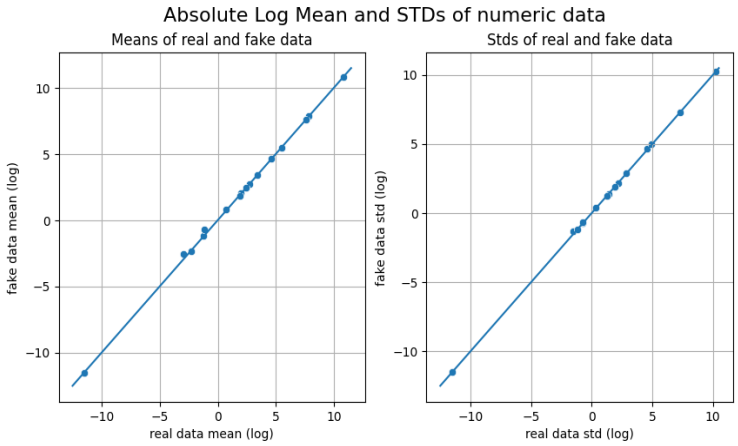

Fig 5.1: CTGAN statistics graph

The plot Fig 5.1 shows near-perfect alignment between synthetic (fake) and real data means/STDs, with points tightly clustered along the diagonal (y = x). This indicates CTGAN excels at replicating both central tendency (mean) and variability (STD) of real data, making its synthetic data statistically indistinguishable from the original.

WGAN statistics graph

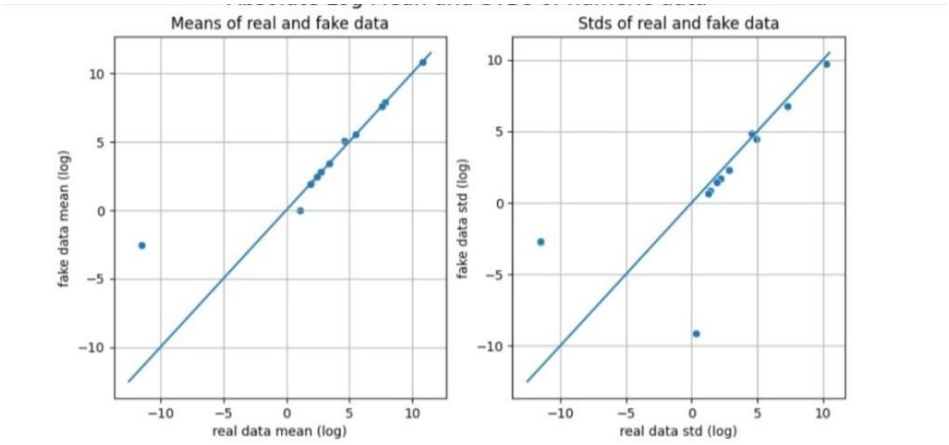

Fig5.2:WGAN statistics graph

From the Fig 5.2, Values deviate noticeably from real data, with inconsistent means/STDs (e.g., mismatched −5/−10 entries). The scatter plot (if reconstructed) would show points drifting from the diagonal, confirming WGAN's weaker performance in preserving statistical fidelity compared to CTGAN.

SDG GAN statistics graph

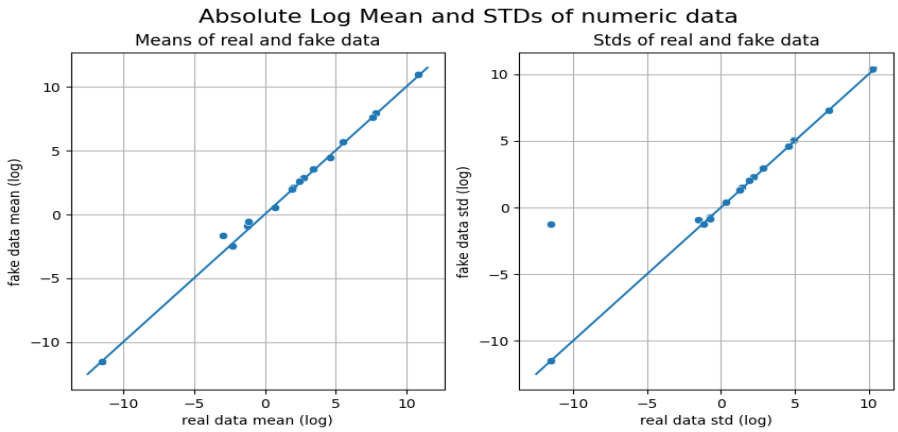

Fig5.3: SDG-GAN statistics graph

Fig 5.3 shows SDG-GAN's results are closer to real data than WGAN's, the plot reveals persistent minor deviations. The logarithmic means and standard deviations show slight curve misalignments, particularly at extreme values

(-10 and +10 ranges).These small but consistent offsets indicate that while SDG-GAN captures the overall distribution better than WGAN, it lacks CTGAN's precision in replicating fine-grained statistical patterns.

Comparative analysis of Numerical and Categorical Features

Table 5.1 summarizes the results of the three generative models: CTGAN, WGAN, and SDGGAN by the KS test. The KS test measures how close synthetic data is to real data in terms of Transaction Amount, Account Balance and Transaction Distance and Average Transaction Amount. Smaller KS values (ideally in the range of 0.01–0.1) indicate greater similarity to real data. It is indicated that the best performance is for the CTGAN as can be seen the lowest KS e.g., 0.0227 for Transaction Amount, 0.0162 for Transaction Distance. WGAN performs fair with KS value 0.0416 and SDGGAN is the maximum with the KS value 0.1452 for all the features, indicating least similarity. Overall, CTGAN performs best at generating realistic synthetic financial data.

Table 5.1: KS Test

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 5.2 depicts Chi-Square test results to compare how much synthetic categorical data produced by CTGAN, WGAN and SDGGAN corresponds to the real data. The analyzed features are Transaction Type, Device Type, Previous Fraudulent Activity, Card Type, and Authentication Method. The smaller the Chi-Square value is, the better is the fit to the observed data. CTGAN generated values of 0.35 for Previous Fraudulent Activity and 0.13 for Fraud Label, it therefore most closely replicates the original distributions. On the other hand, WGAN and SDGGAN had larger values, meaning they were further from the real data. Overall, these results

demonstrate that of the tested models, CTGAN provides the best approximation to the true distribution of the categorical features.

Table 5.2: Chi-Square Test

Model	Transaction Type	Device Type	Previous Fraudulent 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	13076.59	3945.35	28392.96	6039.24

5.2 Performance Analysis Metrics

Below Table 5.3 compares how well Random Forest and Autoencoders work on data created by CTGAN, tested with various train and test split ratios. The Random Forest usually does pretty well, hitting around 89% accuracy most of the time, and getting solid scores in precision, recall, and F1 across the board. Autoencoders, on the other hand, don't work quite as well. Their performance drops, especially when we split the data 80-20, the accuracy falls to about 50%. Basically, this shows that Random Forest tends to do a lot better with data generated by CTGAN, especially in cases like fraud detection where accuracy really matters.

Table 5.3: Performance of CTGAN-Balanced Data

Split Ratio	Model	Accuracy	Precision	Recall	F1-Score
80-20	**Random Forest**	**89.10**	**0.90**	**0.89**	**0.89**
80-20	Auto encoders	50	0.49	0.50	0.38
70-30	Random forest	84.50	0.85	0.85	0.85
70-30	Auto encoders	79.37	0.80	0.79	0.79
90-10	Random Forest	84.67	0.85	0.85	0.85
90-10	Auto Encoders	79	0.80	0.79	0.79

The performance of Random Forest and Autoencoders when trained on the original processed data with various train-test splits is illustrated in Table 5.4. Random Forest consistently gives strong results, with high accuracy and well-balanced precision, recall, and F1 scores. Autoencoders, meanwhile, don't perform as well across all the metrics. Overall, this suggests that Random Forest is the better choice for working with this dataset.

Table 5.4: Performance of Original Processed Data

Split Ratio	Model	Accuracy	Precision	Recall	F1-Score
80-20	Random Forest	0.88	0.90	0.88	0.87
80-20	Auto encoders	0.6779	0.62	0.68	0.59
70-30	Random Forest	0.88	0.90	0.88	0.87
70-30	Auto Encoders	0.666	0.58	0.67	0.57
90-10	Random Forest	0.8734	0.89	0.87	0.86
90-10	Auto encoders	0.6846	0.65	0.68	0.60

The fraud detection system shown in Figure 5.4 may look into suspicious financial transactions. Each feature has a description provided so the user is aware of what information to enter, and the interface is extremely straightforward. The system will be easy for a novice to use. To demonstrate how the system will evaluate each transaction in real-time for fraud detection, users will input the transaction type, amount, and card information. To identify patterns in data and anticipate questionable activity, the system is equipped with machine learning algorithms like Random Forest and Autoencoders. Both synthetic and real data are introduced for improved outcomes, with CTGAN being used to create the synthetic data. To help users predict the system's performance, the system displays key performance indicators such as accuracy, precision, recall, and F1-score. A quick overview of how everything works is provided to new users, thus helping them grasp the system better and start answering questions with confidence.

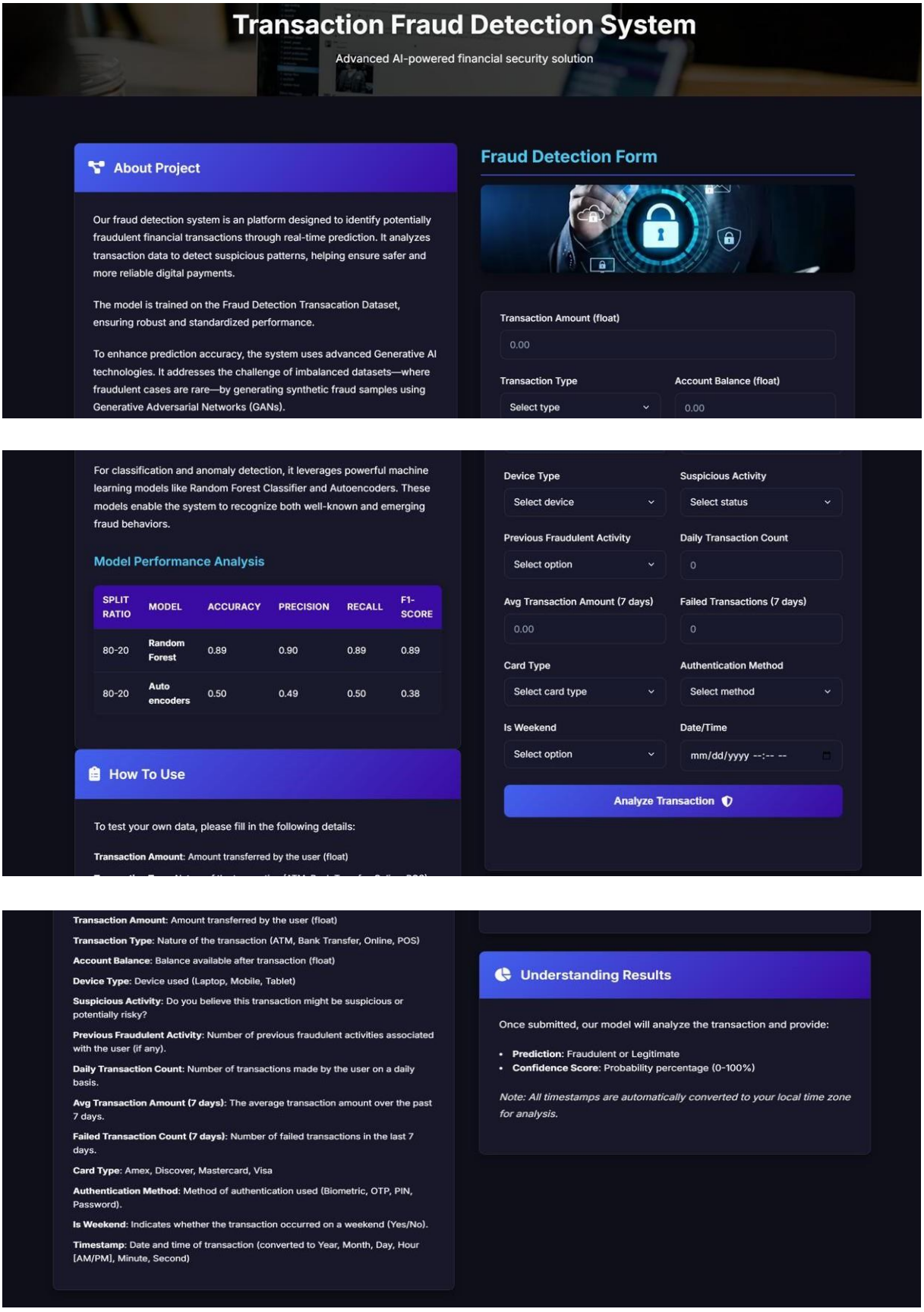

Fig 5.4 User Interface

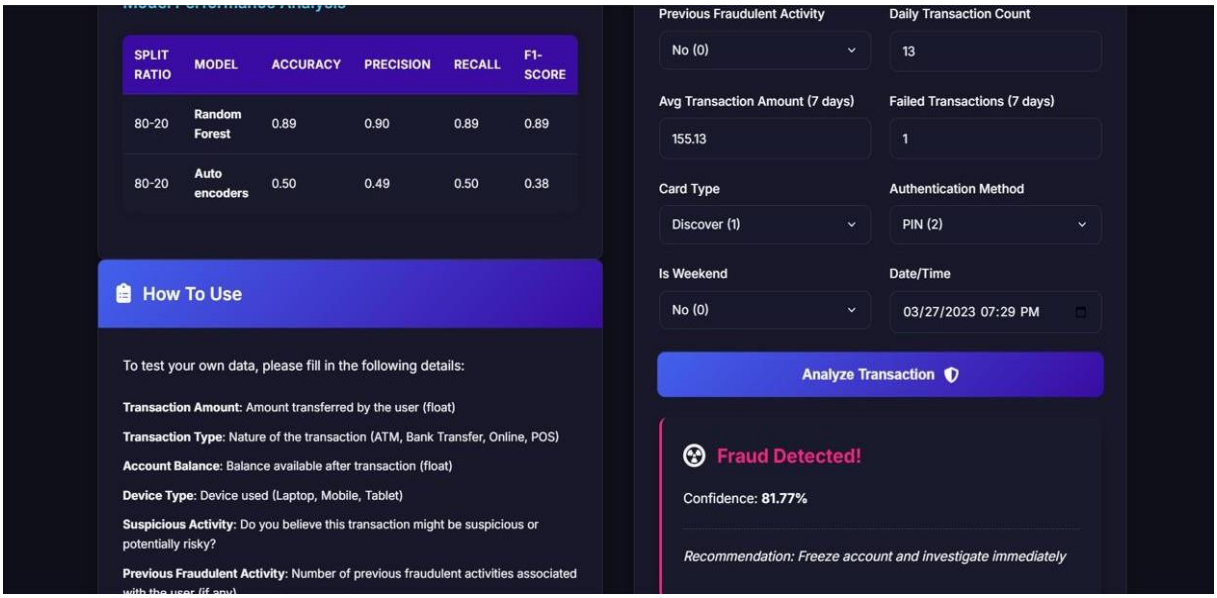

Fig 5.5 Fraud Transaction Result

In the above Figure 5.5, the user enters transaction details, and the system decides that the transaction is fraudulent. The inputs include a transaction amount of 155.13, 13 daily transactions, 1 failed transaction in the past 7 days, and use of a Discover card with PIN authentication. There is a clear history of no prior fraudulent activity, and the transaction is made through a weekday. After analysis through the machine learning models, the system gives a clear-cut output of 81.77 percent confidence in high probability of fraud. This provides an easy and informative way for users to understand the risk of the given transactions.

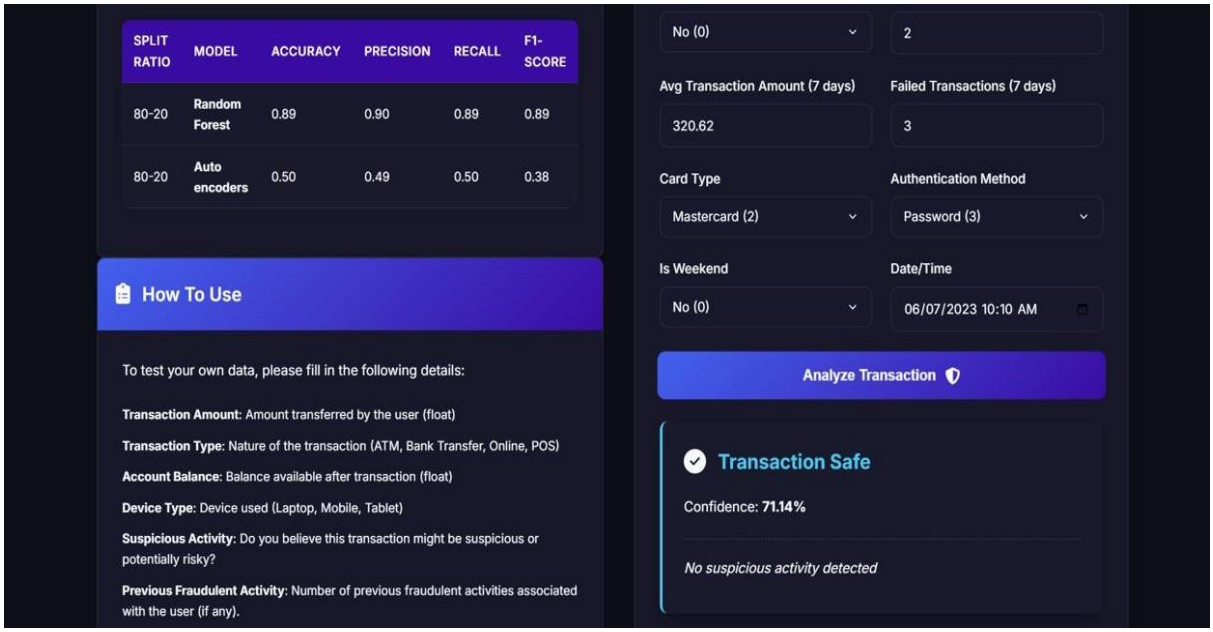

Fig 5.6 Safe Transaction Result

In Figure 5.5, the system reviews the entered transaction details and marks it as safe. The transaction amount is $320.62, with 2 daily transactions and 3 failed attempts in the past 7 days. The payment was made using a MasterCard with password authentication. There is no history of fraud, and the transaction occurred on a weekday morning. After analyzing this information with the trained machine learning model, the system confirms the transaction is safe with a confidence level of 71.14%. The message "No suspicious activity detected" provides reassurance to the user, confirming that the transaction does not generate any fraud issues.

6. Conclusions and Future Enhancements

The proposed solution looks at fraudulent transaction detection using a Random Forest Classifier that was fitted on a balanced dataset. The original dataset had a much higher-class imbalance and very few fraudulent records compared to legitimate transactions. The classes of transactions were then balanced through synthetic fraud sample generation using CTGAN, which allowed for more accurate modeling of the patterns present in fraud and non-fraud transactions. Once trained, the model achieved an accuracy of 89% and thus, would classify those transactions much more accurately. The use of advanced preprocessing and synthetic data generation techniques were applied to gain more of an insight into fraudulent behaviors and to improve upon model accuracy.

In the direction of further development, a system that uses real time data could enhance the model's responsiveness to changing fraud circumstances. The alternative to historical data is identifying the indicator of fraud in real time with a model that uses live data. Continuous training could also be researched so that the model can learn about recent transactions without having to retrain from first principles. Overall, these developments would make the fraud detection system more live and real for real situations.

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.

[11] T. Patil,*Credit Card Fraud Detection Using Conditional Tabular Generative Adversarial Networks (CT-GAN) and Supervised Machine Learning Techniques*, National College of Ireland, 2021.

[12] Vaishnavi T, Krishnaveni S, Aravindprakash N, Akilesh S, Hari Prakash A.C,Detection of Credit Card Fraud Using Machine Learning, Kongu Engineering College, Perundurai, Erode, 2024.

[13] Yogesh W Bhowte; Arundhati Roy; K. Bhavana Raj; Megha Sharma; K. Devi; Prem LathaSoundarraj,*Advanced Fraud Detection Using Machine Learning Techniques in Accounting and Finance Sector*,IEEE,April 2024

[14] Y. Cheng, C-H. Wang, V. K. Potluru, T. Balch, and G. Cheng,*Downstream Task-Oriented Generative Model Selections on Synthetic Data Training for Fraud Detection Models*,J.P. Morgan AI Research Department of Statistics and Data Science., 2024.

[15] Y. M. Ding, W. Kang, J. Feng, B. Peng, and A. Yang,*Credit Card Fraud Detection Based on Improved Variational Autoencoder Generative Adversarial Network*, IEEE Access, 2023