Dissolving the Issues and Challenges in Credit Card Fraud Detection: A Machine Learning Based Approach

[1] J. Pattnayak; [2] D. Biswas; [3] K. Acharjee; [4] P. Nandi
[1]Department of Information Technology, JIS College of Engineering, Kolkata, India
[2]Department of Computer Application, JIS College of Engineering, Kolkata, India
[3]Department of Information Technology, JIS College of Engineering, Kolkata, India
[4]Department of Computer Application, JIS College of Engineering, Kolkata, India

Abstract: At present, credit card fraud (CCF) is a significant issue for financial institutions and consumers in similar manner. Fraudulent transactions are detected by Machine learning (ML) by analyzing patterns and anomalies in data. The study is designed to deliver the presentation of AI comprehension in accordance to powered fintech solution in delivering the best product to detect, understand, predict by alerting the card transactions undergone in individual's credit card. This study proposes a machine learning-based approach to detect credit card fraud using multiple algorithms including logistic regression, XGBoost and others were used to detect various activities that are flagged as suspicious and they significantly helps in detection and alerting the customer. Optimized algorithms like Xgboost and random forest provides better percentage in F1 scores and also varied the real time responses to a greater extent and trends through GUI. Synthetic Minority Over-sampling Technique (SMOTE) feature is crucial in this process that is undergone by handling for class imbalances and it shows various analytics to users in classifying the trends of classification Ensuring smooth transaction is endured by altering immediate notifications, dashboard using linked Multi Factor Authentication (MFA). Performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC. Henceforth the need to create the burden of financial risk overloading is prevented.

Keywords: CCF, CCFD, fintech, F1 Score, SMOTE, XGBoost

1. Introduction:

In today's digitized culture e-payment system plays a pivotal role. Majority of the payment are now being done by internet banking system. The customers are provided with cards by the financial institutions for making it simple for them to buy goods when they don't have cash on hand. This revolution in digital technology industry, have variably increased the tendencies of fraud and it is more

vulnerable in case of large number of transactions. It is one of the most prevalent cybercrimes that is being encountered. With the rapid expansion of digital payments, the incidence of CCF has also increased. CCF is the unauthorized use of someone's card details for making transactions. Migrating to the Internet and the electronic financial transactions that take place in fostering the cashless economy, accurate fraud detection has become essential in protecting such transactions (1). The conventional fraud detection techniques mostly rely on static rules which face difficulties in adapting changing fraud patterns. For security and trust point of view, payment systems should provide fast and reliable authentication mechanisms for security purpose in order to ensure easy access for legitimate users by identifying fraudulent transactions attempts by others [2]. ML provides dynamic and adaptive capabilities, and becomes a powerful tool for detecting anomalies and fraudulent behaviour in large datasets. Things are very subtle and handle challenges as in class imbalance as well as alert system mechanism. Incorporation of mitigation premises of fraud features mostly deploys supervised models on trained transactional data. The research focuses strongly on SMOTE to handle class imbalance to improve model performance on minority fraud cases. Extraction and use of techniques dealing in feature engineering applies to cater a good range of diversification in variables creating higher impact ratio like transactions in principal component, amount if transaction along with behaviors in temporal space. Fine tuning of models is done using optimized hyper parameter to achieve maximum accuracy and F1 across different classifiers. Enhancement of operational validation on legitimate data points makes sure that integration with notification is done thoroughly thereby gap between proactive and intelligent system is minimizing the Cyber security threat. This study proposed methods to build a model to accurately classify fraudulent or legitimate transactions by comparing the performance of different ML algorithms. It also addresses challenges such as data imbalance and model interpretability.

2. Related Works:

Below are some similar contributions, different perspectives are compared, the framework is established on which the current research is built:

ML is committed to centralize the financial frauds mainly in credit cards as its in built capability to handle data in various large sources and it also take care of the patterns that are happening in the backend and can take ensemble learning in its addition to commemorate the root cause of recovery. The study compared six Supervised ML algorithms: Logistic Regression, K-Nearest Neighbours, Naive Bayes, Decision Trees, Random Forest and Linear Support Vector Machine (SVM) on a huge imbalanced dataset in order to evaluate their performance [3]. The preprocessing technique such as Feature extraction is used to extract a richer set of compact dataset and data sampling is used to solve class imbalance. In this paper, these two preprocessing techniques are investigated, using a credit card fraud

dataset and four ensemble classifers (Random Forest, Cat Boost, Light GBM, and XGBoost). The study encompasses two feature extraction methods, the Principal Component Analysis (PCA) and Convolutional Autoencoder (CAE) along with three resampling techniques i.e., the Random Under sampling (RUS), SMOTE and SMOTE Tomek to know their effectiveness in fraud detection. To handle that SMOTE is used and challenges were addressed more carefully in addition to evaluate the classification. Similar considerations followed the rules of leevy and Khoshgoftaar in samples of data sampling and extracting the features of identifying the frauds and identify those effectively [4]. In the next study, methods were suggested to evaluate the usage of a convolutional Variational Auto Encoder (VAE) on a Credit Card Fraud Detection (CCFD) dataset. The research analyzed how anomalous transactions are scattered in the latent space of the model and the how performance is influenced by scaling this space. This study focuses on modeling cardholders' spending behavior and regarding anomalous data as a statistical outlier comparing the normal spending inliers [5]. The next study focused on the natural imbalance in credit card transaction data, explored new technical methods to improve fraud detection accuracy and reliability. It has been observed from the findings that the integration of Neural Network (NN) and SMOTE produced better precision, recall, and F1-score in comparison to traditional models, showing its efficacy better to handle imbalanced datasets. The suggested technology is not significance not only for improving accuracy but also for developing a more comprehensive and advanced method to detect CCF. The F1-Score is a useful measure to balance precision and recall especially when not evenly distributed. A higher F1-Score means a well-balanced trade-off between precision and recall [6]. The next study uses various ML algorithms to analyze customer's data based on specific proposed approach. The analysis of the data set along with current dataset is supported by the classification algorithms, which help in improving model's performance based on training and testing. Performance evaluation was done by using common metrics such as accuracy, precision, recall, and F1 score. The model that performs best identifying fraud within a particular set of transactions is considered as the most effective one. In this study, CCF is detected using the ML models below.

- Random Forest
- Decision trees
- Ada Boost algorithm

The research compared and analyzed the ML techniques such as Random Forest, Decision Tree, and AdaBoost to identify the best. The model with the highest is then used to classify transactions as either honest or fraudulent. The system architecture outlines the system's development step-by-step highlighting the major interactions and a high-level explanation of the method of problem solving

(1). The following research is based on real-world transaction data from an international credit card operation. It suggests two data mining techniques, support vector machines (SVM) and random forests, along with logistic regression, as part of an effort to better identify (and thus manage and investigate) CCF. Real-life data from an international credit card operation is used. It introduced a new innovation that became a reference for analyzing of different types of fraud. In Supervised fraud detection, model training is done by using labeled datasets of fraudulent and legitimate transactions while in unsupervised methods anomalies or outliers are detected which may indicate fraud [7]. Some insights from a practitioner's perspective were offered by focusing on three complex issues: data imbalance, non-stationarity and evaluation. The analysis were made possible using two real credit card datasets provided by the industrial partner. CCFD problems are generally tackled in two different ways. Models are retained periodically (e.g., monthly or yearly) in the static learning approach with the help of complete dataset, where in online learning, models are continuously updated with the arrival of new transaction. In the online learning setting, the detection model is updated as soon as new data is there. Another problematic issue in CCFD is the limited data due to confidentiality concerns which makes it hard for the research community to share real datasets and evaluate existing techniques. This paper aims to address this gap by conducting an extensive comparison of a many algorithms and modeling techniques on two real datasets. It focuses on key questions like: Which ML algorithm is best? Is it enough to retain the model once a month or must it be updated daily? How many transactions are required to train the model? Should the data be analyzed in their original unbalanced form or rebalancing is required? What performance metrics is the most suitable to evaluate results? These questions were explored with the aim of understanding their importance on real world data and from a practical standpoint. The paper starts by analyzing the fraud problem and provides a formal definition of the detection task [8]. A predictive framework was proposed to help the credit bureau by modeling/assessing the risk of credit card delinquency. Risk assessment is supported by ML, identifying legitimate from fraudulent transactions within highly imbalanced datasets. In case of any suspicious transaction, the financial institution is issued with alert system. Alert can be sent to the relevant financial institution to suspend payment for the transaction in case of suspicious transaction. Numbers of evaluation metrics were used such as sensitivity, specificity, precision, F scores, ROC-AUC and PR-AUC. Datasets used for training and testing of the models have been taken from kaggle.com. The research investigates the challenge of classifying imbalanced data by combining data-level and algorithm level techniques to detect the fraudster from the log files generated for credit cards used at IoT-enabled terminals. Furthermore, an appropriate alert message can be sent to either the credit card holder or the issuer for reverting/ blocking the transaction [9]. Recent Deep Learning (DL) methods are compared such as convolutional neural network

(CNN), simple recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) to find their effectiveness in CCF. It explores suitable performance metrics, common issues faced during the training CCFD models using DL architectures and potential solutions, which are not covered in previous studies. This is beneficiary for researchers and practitioners working with DL. It also reviews the most recent advancements in CCFD using DL techniques. A brief overview of DL techniques is provided used in CCFD along with the comparison of performance. A detailed analysis is conducted on the performance evaluation metrics used in CCFD and their suitability for this domain, focusing on their suitability for CCFD. A thorough examination is made on existing challenges in using DL for CCFD is done along with potential solutions, and research directions [10]. Analyzing and pre-processing of data and also the deployment of multiple anomaly detection algorithms such as Local Outlier Factor and Isolation Forest algorithm on the PCA transformed transaction data has been suggested in this study. ML algorithms are used to analyze all authorized transactions and flag any suspicious activities. These flagged transactions are then investigated by professionals who contact the cardholders to confirm the legitimacy of the transaction. The feedback from these investigators is incorporated back into the automated system which is used to train and update the algorithm thereby improving fraud-detection performance over time [11]. A GA-driven feature selection framework for CCFD was proposed, followed by classifiers including decision tree (DT), RF, logistic regression (LR), ANN, and Naïve Bayes (NB). The GA was implemented with the RF as its fitness function. Further application of the GA to the European cardholder's credit card transactions dataset resulted in generation of optimal feature vectors. The experimental results using the GA selected attributes demonstrated that the GA-RF (using v5) achieved a high accuracy of 99.98%. Additionally, the GA-DT achieved a high accuracy of 99.92% using v1. The results were better than those obtained using existing methods [12]. A novel fraud detection method for streaming transaction data was proposed with the objective, to analyze customer transaction history and extract the behavioral patterns. Classifiers were then applied to three different groups and rating scores were generated for each classifier. These dynamic changes in parameters enabled the system to adapt to new cardholder's behaviors in time. A feedback mechanism was introduced to address the issue of concept drift. It was observed that the Matthews Correlation Coefficient (MCC) performed in handling imbalance dataset. However, MCC was not the only solution. The researchers attempted to balance the dataset using SMOTE, and the classifiers showed better performance than before. Another approach to handle imbalance dataset was the use one-class classifiers such as one-class SVM. The result showed that LR, decision tree and random forest performed best [13]. A comprehensive approach was taken by integrating nine distinct ML techniques and three sampling techniques to address the challenges posed by highly imbalanced datasets. This approach was necessary

due to the limitation of static rules in identifying fraudulent patterns. The effectiveness of nine machine learning techniques RF, gradient boost classifier, MLP classifier, extra tree classifier, naive bayes, Ada-boost classifier, k-nearest Classifier, Decision Tree, and Gradient Boost Classifier was evaluated within the proposed framework. Performance metrics, including recall, precision, F1-score, F2-score, and accuracy, were used to assess the effectiveness of these techniques [14]. Most of the proposed methods just keep only recent instances for model training, but do not consider the adaptability. To address these issues transaction behaviors of a cardholder were analyzed using both his/her historical transaction data and the data of some similar cardholders. A feedback mechanism was introduced to adapt to changing transaction behaviors seasonally [15]. A protocol or a model was proposed to detect the fraud activity in credit card transactions. This system is designed to provide essential features for identifying fraudulent and legitimate fraudulent and legitimate transactions. As technology evolves, tracking the modeling and pattern of fraudulent transactions become challenging. With the rise of ML, AI and other relevant fields of information technology, it becomes possible to automate this process and to reduce the labour intensive nature of detecting CCF. This proposed module is applicable for the larger dataset and provides more accurate results. Better performance is achieved by RF algorithm with many training data, but speed during testing and application will still suffer. Usage of more pre-processing techniques would also assist [16]. The next research proposed a biologically inspired technique in feature engineering phase for handling imbalance data of a small number of classes. One-point crossover used to generate the new data of minority classes. Crossove plays a major role in searching based on the genetic algorithm (GA). The data set of this research was provided by Vesta Corporation. Vesta Corporation is the predecessor to assured payment solutions for e-commerce. Founded telecommunications indutry's fully generated card not present (CNP) payment tansactions. The data comes from real-world e-commerece purchases and provides a wide range of features ranging from device type to product features [17].

3. Research Questions:

Following are the research questions:

- AI/ML approaches provide better accuracy and efficiency in comparison to traditional fraud detection methods?
- Dissolving the challenges arise in detecting fraud in highly imbalanced datasets where fraudulent transactions are rare?

4. Methodology:

4.1 Dataset Description:

The dataset used in this system is collected by European cardholders over a two-day period in September 2013 and is publicly available credit card transaction dataset. It comprises a total of 284,807 transactions, among which only 492 are fraudulent, making the dataset highly imbalanced with fraud cases representing just 0.172% of the total data. This significant imbalance poses a major challenge in training effective fraud detection models, as traditional classifiers may become biased towards predicting only the majority class (non-fraudulent transactions).
The dataset contains 30 numerical features, all of which are the result of a PCA transformation to ensure confidentiality and protect sensitive customer information. Except for the 'Time' and 'Amount' features, the remaining 28 features are labeled as V1 through V28, which are anonymized linear combinations of the original features. The 'Time' feature represents the seconds elapsed between each transaction and the first transaction in the dataset, while the 'Amount' feature indicates the transaction value in Euros. The target variable, labeled ‘Class’, is a binary indicator where 0 denotes a legitimate transaction and 1 denotes a fraudulent one. This dataset reflects real-world transactional behavior and is particularly valuable for research in fraud detection because it mirrors the complexities and nuances of actual financial systems. However, the anonymization and transformation of features through PCA mean that interpretability of individual features is limited, requiring more focus on model performance and pattern detection rather than feature semantics. Despite this, effective use of the dataset involves applying techniques like data normalization, exploratory data analysis (EDA), resampling with SMOTE and model training using ensemble and supervised learning algorithms. The dataset's characteristics make it ideal for benchmarking fraud detection methods in highly skewed environments, thus offering a realistic and challenging scenario for machine learning applications.

4.2 Data Preprocessing

Data pre-processing is a critical phase in any ML systems, particularly in financial transactions, where data quality directly impacts the performance of predictive models. The dataset used here contains anonymized transaction records with 30 numerical features derived through PCA transformation, alongside the transaction amount and class label indicating whether a transaction is fraudulent or not. Several pre-processing steps were employed to ensure the data was clean, structured, and suitable for training machine learning models. The first step involved handling missing or duplicate records; although the dataset was relatively clean, checks were performed to eliminate any redundant entries or anomalies. Following this, feature scaling was applied using the Min Max Scaler to normalize the range of numerical values, especially for the 'Amount' feature, so that no feature would disproportionately influence model training due to scale differences.

To address the extreme class imbalance, the SMOTE was implemented. SMOTE generates synthetic examples of the minority class by interpolating between existing fraud cases, thereby enhancing the model's ability to recognize fraudulent patterns without simply duplicating data. Additionally, EDA was conducted to understand the feature distributions, detect outliers, and identify correlations among variables using visual tools like heat maps and pair plots. This helped in selecting the most relevant features and in mitigating noise that could degrade model performance. The pre-processed data was then split into training and testing sets in a stratified manner, ensuring that both sets preserved the original class distribution. Overall, the pre-processing pipeline established a robust foundation for the subsequent phases of model development, ensuring data consistency, representativeness, and fairness in learning across both majority and minority classes.

4.3 Feature Engineering and Smote

Feature engineering is a vital component in fraud detection techniques, where subtle patterns must be extracted from complex and often noisy transactional data. In this study, feature engineering was employed to enhance the model's ability to distinguish between legitimate and fraudulent transactions. Initially, the dataset consisted of anonymized numerical features derived from PCA, along with features such as transaction amount and time. These features were carefully analyzed using EDA techniques including correlation heat maps, pair plots, and time-series visualizations. Through this analysis, it was possible to identify which features contributed most significantly to the detection of anomalies. New custom risk scores were also deve loped by combining multiple weak indicators into aggregated features that capture behavioral nuances over time.

Extreme class imbalance, with fraudulent transactions comprising less than 0.2% of the total data is a significant challenge a significant challenge. Training a model on such skewed data would bias it toward predicting the majority class (legitimate transactions), resulting in poor detection of actual frauds. To address this, the SMOTE was applied. SMOTE generates synthetic samples of the minority class (fraud) by interpolating between existing minority class instances. Unlike simple oversampling, which merely duplicates rare cases, SMOTE creates new and plausible data points that help the model generalize better. This not only increases the representation of fraudulent transactions in the training dataset but also reduces the risk of over fitting.

The integration of feature engineering and SMOTE played a pivotal role in improving model performance. With well-constructed features and a more balanced training set, machine learning algorithms such as XGBoost and Random Forest were able to better discriminate between normal and suspicious transactions. As a result, performance metrics like F1-score, precision, and recall showed significant improvement, ensuring that the model remained both sensitive

to detecting fraud and robust against false positives. This comprehensive approach to feature design and class balancing may be regarded as core strength of the proposed fraud detection system.

4.4 Handling Class Imbalance Using Smote

Severe class imbalance inherent in real-world datasets is one of the most significant challenges in the domain of CCFD, during model development. Fraudulent transactions constitute less than 0.2% of the total transactions, which leads to a highly skewed class distribution. This imbalance causes most traditional ML algorithms to become biased toward the majority class, resulting in misleadingly high accuracy while failing to detect the minority class—the fraudulent cases—that are of actual interest. To address this problem and improve the model's ability to learn discriminatory features from both classes, SMOTE is integrated into the data pre-processing pipeline. SMOTE is an advanced over-sampling algorithm that works by generating synthetic examples rather than simply duplicating existing ones. It is done by selecting samples from the minority class and creating new instances along the line segments between them and their nearest neighbors in the feature space. This not only helps in expanding the representation of the minority class but also preserves the underlying structure and distribution of the original data, reducing the risk of overfitting.

SMOTE is applied after normalization and feature engineering, to ensure that the synthetic samples reflect the same scale and patterns as the real data. Special attention is given to ensure that SMOTE does not distort the distribution of high-risk features, such as transaction amount, frequency, and behavioral anomalies. The technique is used in conjunction with cross-validation to prevent over-sampling the test data, and care is taken to preserve the chronological integrity of transaction sequences during synthetic data generation. Once the class balance is achieved, the models—such as Random Forest, XGBoost, and Logistic Regression—are trained on the augmented dataset, resulting in significant improvements in recall, F1-score, and precision, particularly for the fraud class. Furthermore, the use of SMOTE is validated through confusion matrices and ROC-AUC curves, confirming its positive impact on fraud detection accuracy. Overall, SMOTE plays a **pivotal role** in transforming an otherwise unreliable and biased learning setup into a robust, fair, and fraud-sensitive detection system, enabling real-time systems to make better-informed decisions in high-risk financial environments.

4.5 Model Selection and Training

The model selection and training phase determines the predictive accuracy and real-world usability of the solution. Baseline modeling is done using simple algorithms like Logistic Regression, which provides interpretability and serves as a benchmark for more complex models. Then more sophisticated models such as

Decision Trees, Random Forests, and XGBoost are evaluated. XGBoost, in particular, is highly favored due to its ability to handle non-linear feature interactions, regularization capabilities, and resistance to overfitting. These models are trained on a pre-processed dataset that has been balanced using SMOTE to ensure the model can learn meaningful patterns from the minority (fraudulent) class.

To assess the effectiveness of each model, cross-validation is used in combination with performance metrics such as Precision, Recall, F1-Score, and AUC-ROC, with special attention paid to minimizing false negatives, as these represent missed fraud cases. Models are iteratively refined using hyper parameter tuning via Grid Search CV, which tests combinations of parameters like tree depth, learning rate, and the number of estimators to identify the optimal configuration. For unsupervised anomaly detection, models like Isolation Forest and Auto encoders are also explored. These models are particularly useful when labeled data is scarce or incomplete, as they learn to identify outliers based on reconstruction error or feature isolation. Once trained, the best-performing model—usually an ensemble-based classifier like XGBoost—is serialized using joblib or pickle and integrated into the real-time prediction engine. This trained model is then deployed as part of a live fraud detection system capable of evaluating transactions in real-time. The chosen model's performance is further validated against unseen test data to confirm its generalization capability before final deployment. This comprehensive and methodical approach to model selection and training ensures the system is both accurate in detection and resilient to evolving fraud patterns.

4.6 Performance Evaluation Metrics

In a CCFD system, evaluating the performance of ML models goes far beyond simple accuracy due to the **extreme class imbalance**—where fraudulent transactions make up less than 1% of the data. The system employs a set of comprehensive evaluation metrics specifically chosen to measure the effectiveness of the models in detecting rare, yet critical, fraudulent cases. The primary metrics include Precision, Recall, F1-Score, and Confusion Matrix, all of which provide a more meaningful interpretation of the model's ability to correctly identify fraud. Precision measures the proportion of correctly predicted fraudulent transactions out of all transactions that were flagged as fraud. It answers the question: "Of all the transactions predicted as fraud, how many were actually fraud?" This is crucial in minimizing false positives, which can lead to customer dissatisfaction and unnecessary investigations. Recall, on the other hand, measures the proportion of actual fraudulent transactions that were correctly identified by the model. It helps assess the model's sensitivity to fraud detection and is particularly important to avoid false negatives, where fraudulent activity goes undetected.

The F1-Score provides a balanced harmonic mean between precision and recall, making it the most reliable single metric for performance evaluation in imbalanced

classification problems. A high F1-Score indicates that the model maintains a good trade-off between catching fraud (recall) and avoiding misclassification of legitimate transactions (precision). Additionally, the Confusion Matrix is used to visualize the count of true positives, true negatives, false positives, and false negatives, offering a detailed overview of the model's classification performance. For multi-model comparison, the project also tracks Area under the Receiver Operating Characteristic Curve (AUC-ROC), which provides insight into the model's ability to distinguish between fraudulent and legitimate transactions across various threshold settings. Models like XGBoost and Random Forest consistently performed well across these metrics, with ensemble approaches demonstrating superior F1-scores and recall rates. These evaluation metrics not only help in selecting the best-performing model but also ensure that the deployed fraud detection system is robust and reliable in real-world scenarios, e missing a single fraudulent transaction can have significant financial consequences.

4.7 Real Time Integration and Alert System

This system enables instantaneous response to suspicious activities and bridging the gap between prediction and user action. Once the fraud detection model is trained and deployed, it is integrated into a **real-time transaction monitoring environment**, where incoming transaction data is continuously streamed and evaluated. This setup mimics real-world financial systems, where decisions must be made in milliseconds to prevent unauthorized or malicious activities. Lightweight APIs are employed and background services that listen to transaction triggers—such as swipes, online purchases, or large withdrawals—and process them through the machine learning model hosted on a backend server or cloud environment. If the model flags a transaction as potentially fraudulent, the alert mechanism is activated immediately. This mechanism operates with minimal latency and high reliability, leveraging message queues or event-driven architectures (e.g., Kafka or WebSockets) to ensure scalability and speed.

The alert system supports multiple communication channels for notifying stakeholders. Alerts are generated in real-time and can be configured to send SMS, email, or push notifications to the customer and/or bank administrators. The system also includes a Graphical User Interface (GUI) that displays flagged transactions, risk scores, and decision rationale, allowing financial analysts or users to verify or dispute the alert promptly. For enhanced security, the system can optionally invoke Multi-Factor Authentication (MFA) to confirm the legitimacy of high-risk transactions before approval. The architecture also supports logging of flagged transactions into a secured audit trail for compliance and forensic analysis. By integrating real-time detection with user-facing alert mechanisms, the system ensures proactive fraud prevention, significantly reducing financial loss and improving customer trust. The modularity of the system allows it to be deployed in existing fintech infrastructures with minimal modifications, making it both

practical and scalable for real-world applications.

4.8 Real Time Integration and Alert System

The development and deployment of the CCFD system rely on a robust and scalable technology stack built primarily on Python, due to its versatility and the availability of rich data science libraries. The core data manipulation and pre-processing tasks are handled using Pandas, which provides powerful data structures for cleaning, transforming, and managing tabular data. For numerical operations, Num Py is utilized to support high-performance matrix computations and statistical analysis. To visualize distributions, correlations, and evaluation metrics, libraries such as Matplotlib and Seaborn are employed, aiding in insightful Exploratory Data Analysis (EDA). For model development, the project integrates several machine learning frameworks, notably Scikit-learn, which support a wide range of supervised and unsupervised algorithms, including logistic regression, decision trees, and evaluation utilities like precision, recall, and F1-score. For handling extreme class imbalance in the dataset, Imbalanced-learn is used to implement SMOTE, which generates synthetic samples of the minority class to improve classifier sensitivity. Advanced classification and ensemble models such as XGBoost are leveraged for their high accuracy and gradient-boosting capabilities, making them particularly effective in detecting nuanced fraud patterns.

For real-time system integration and API development, Flask is used as a lightweight web framework, allowing the trained models to be exposed as RESTful endpoints. These APIs enable seamless interaction between the prediction engine and the front-end or transaction interface. On the front-end side, a basic HTML and JavaScript-based interface can be deployed to interact with users and display alerts, while SMTP libraries are used for sending real-time email notifications. The complete stack is containerized and can be deployed using Docker, making it scalable and easily integrable into existing banking infrastructures or fintech platforms. This cohesive blend of libraries and technologies ensures the production-readiness, scalability, and responsiveness of the system in real-world scenario.

The system design flowchart is give below:

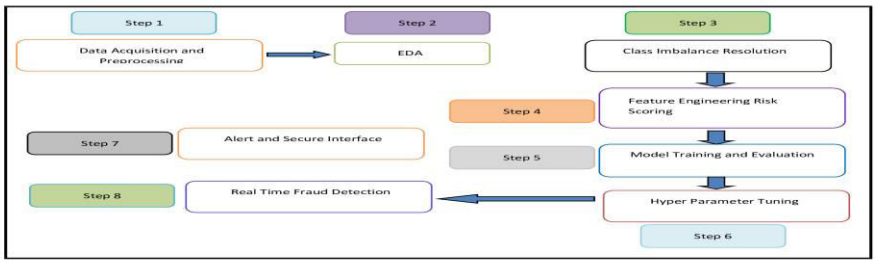

Fig: 1 The System Design Flow chart

5. Result Analysis and Discussion:

The result shows that ensemble models such as XGBoost and Random Forest outperformed traditional algorithms in terms of F1-Score, precision, and recall, effectively handling the imbalanced nature of the dataset.

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9802	0.9916	0.9688	0.9801
Decision Tree	0.9991	0.9989	0.9994	0.9991
Random Forest	0.9998	0.9995	1.0000	0.9998
Gradient Boosting	0.9990	0.9987	0.9992	0.9990
Support Vector Machine (SVM)	0.6928	0.6981	0.6858	0.6919
Naïve Bayes	0.9519	0.9823	0.9211	0.9507
K-Nearest Neighbors	0.9706	0.9693	0.9724	0.9709
XGBoost	0.9996	0.9995	0.9997	0.9996
Isolation Forest (Unsupervised)	N/A	0.9183	0.9462	0.9310

Table: 1 Comparative Analysis of Performance Metrics of Different Algorithms

After applying SMOTE to address class imbalance and optimizing hyper parameters, the XGBoost model achieved the highest F1-score, indicating a strong

balance between detecting actual fraud cases and minimizing false alarms. Visual tools like confusion matrices and performance comparision graphs were used to assess the predictive capability of each model confirming that models trained with engineered features and balanced data significantly improved fraud detection accuracy.

Table: 2 Accuracy of Different Models

Classification Model	Accuracy (%)
Random Forest	100
XGBoost	99.97
Logistic Regression	99.97
Decision Tree	99.97
Classification Model	Accuracy (%)

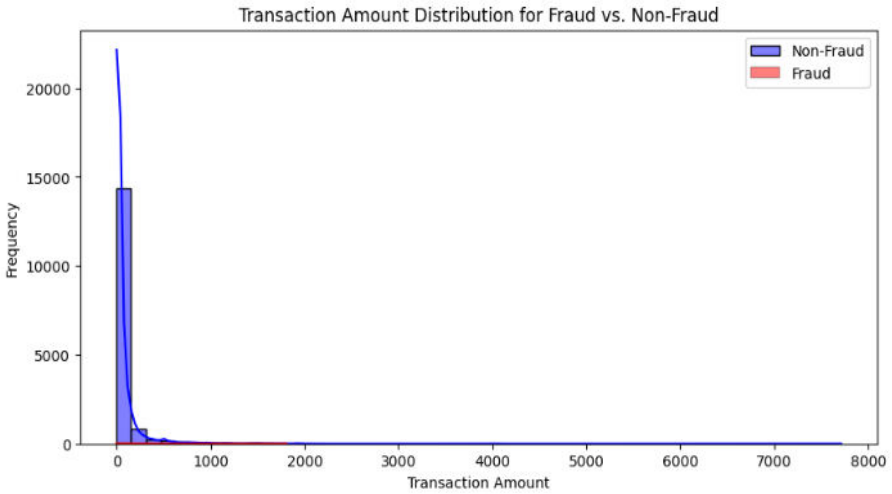

Fig: 2 Classification from Dataset

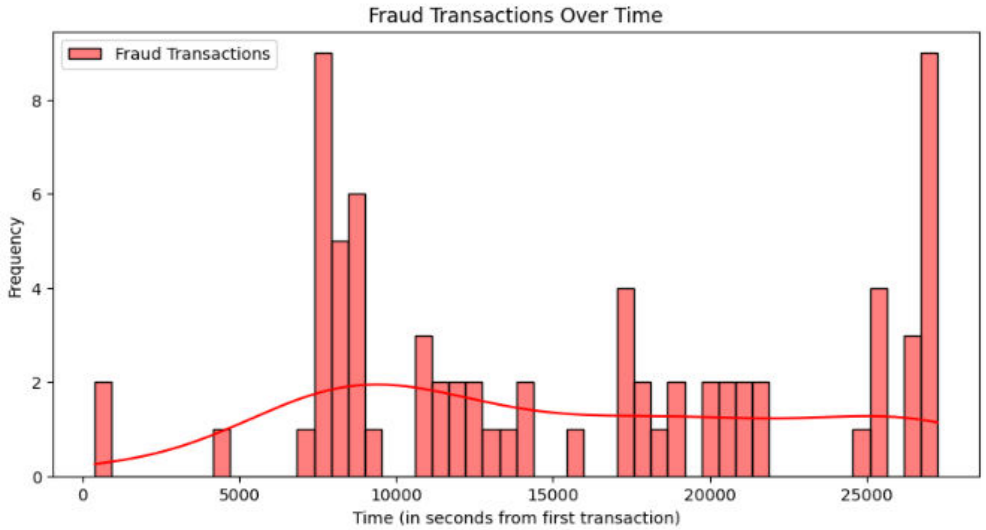

Fig: 3 Time Series Analysis

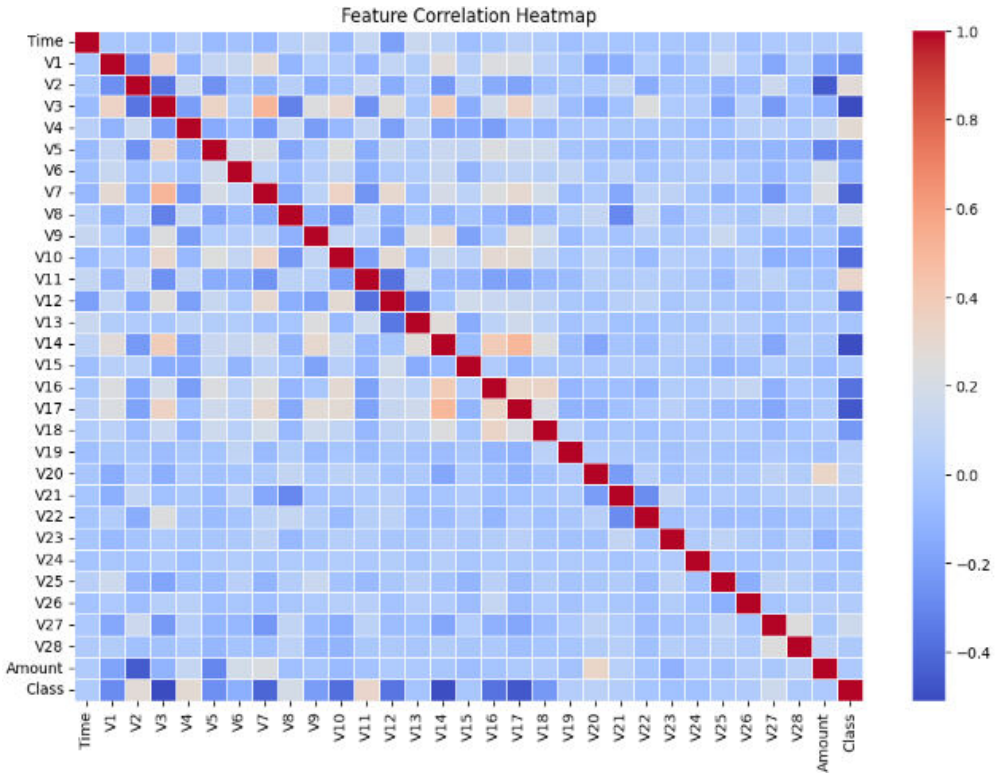

Fig: 4 Correlation Heatmap

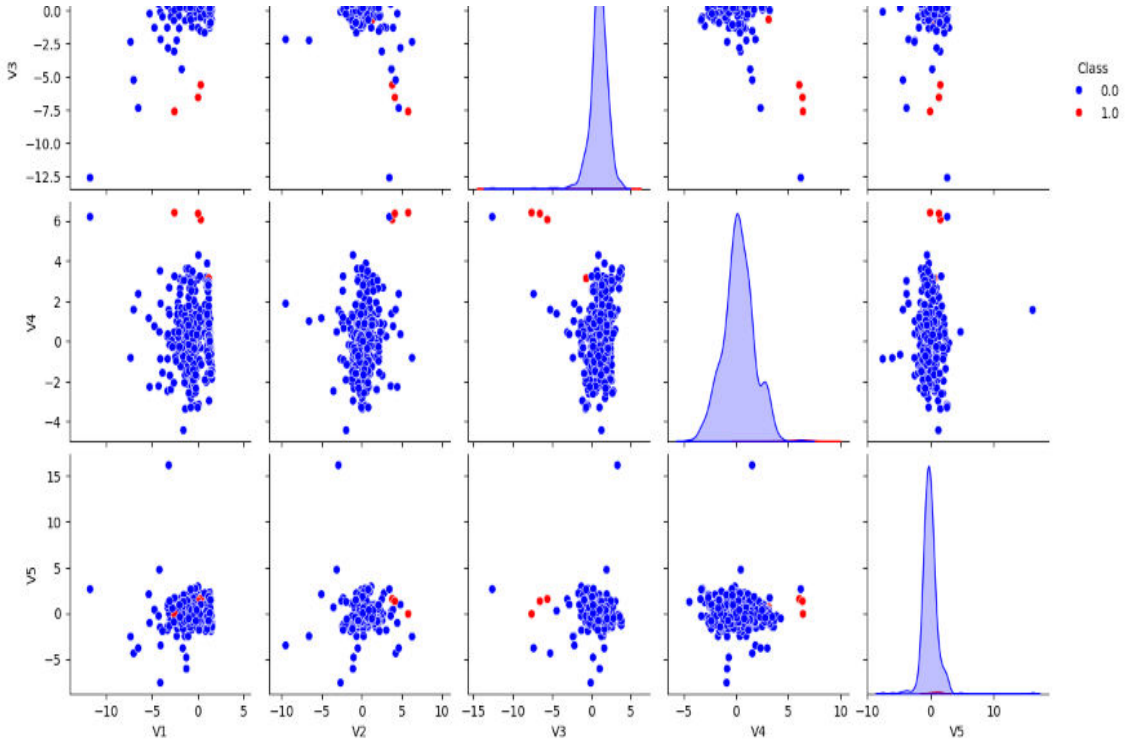

Fig: 5 Pair wise Scatter Plot Matrix

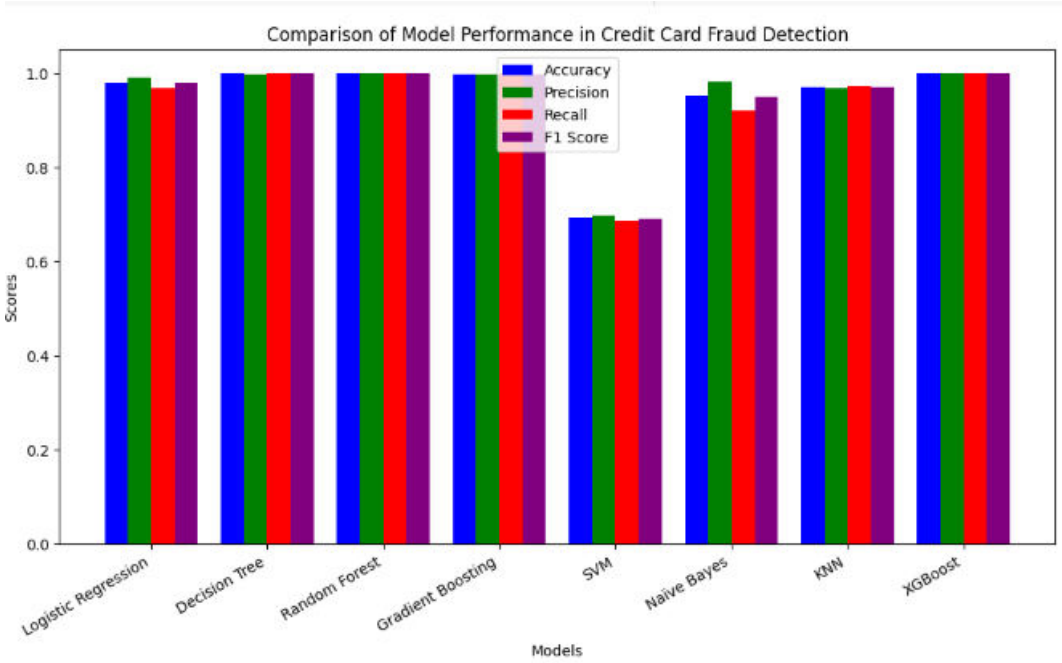

Fig: 6 Comparision of Different Algorithms

```
Enter your 16-digit credit card number: 7898123456781234

✅ Card Verified! Proceeding with OTP authentication...

📲 Sending OTP to 8745...
✅ OTP Sent! (Simulated OTP: 417519)

🔑 Enter the OTP received on your phone: 9876

❌ Authentication Failed! Transaction blocked for security reasons.
```

Fig: 7 Data Validation

```
    print("\n🚨 Transaction Flagged as Suspicious!")
    for reason in fraud_reasons:
        print(reason)
    print("❌ Transaction Blocked! Contact Customer Support.")
else:
    print("\n✅ Transaction Approved! Your payment has been processed.")
```

```
Enter transaction amount: $80000
Enter transaction location: Sylamr
Enter transaction time (HH:MM AM/PM): 06:09 AM

🚨 Transaction Flagged as Suspicious!
⚠️ Unusual High Amount
❌ Transaction Blocked! Contact Customer Support.
```

Fig: 8 Filtering Transaction

```
🔍 AI-Based Fraud Detection Model Accuracy: 0.9500

📊 Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        95
           1       0.00      0.00      0.00         5

    accuracy                           0.95       100
   macro avg       0.47      0.50      0.49       100
weighted avg       0.90      0.95      0.93       100


📌 Enter Transaction Details for Fraud Detection:
Enter transaction amount: $90000
Is the location high-risk? (1 = Yes, 0 = No): 0
Is the transaction at an unusual time? (1 = Yes, 0 = No): 1

✅ Transaction Approved! You may proceed.
```

Fig: 9 Transaction Checking

Fig: 10 Creation of Web Application

6. Conclusion and Future Scope:

The growing volume and complexity of digital transactions have made CCFD a critical concern for financial institutions worldwide. In this study, a robust and intelligent fraud detection system has been suggested using various ML algorithms and techniques, with a strong focus on addressing class imbalance—a major challenge in real-world fraud datasets. Through extensive data preprocessing, feature engineering, and model experimentation, it was found that ensemble methods such as XGBoost and Random Forest delivered superior performance in terms of F1-score, precision, and recall, making them more suitable for identifying rare fraudulent activities. Again the use of SMOTE is a key aspect to handle the class imbalance problem, which significantly enhanced the model's ability to recognize minority-class instances (fraud cases) without compromising the overall

accuracy. The implementation of EDA and visualization techniques further enriched the understanding of transaction patterns, feature relationships, and anomalies. Also a real-time alert mechanism is incorporated and the simple user interface, enabling timely notifications via email or SMS for suspicious transactions, thereby bridging the gap between detection and action. The resulting system may be scalable, interpretable, and suitable for deployment in a fintech environment. While current results are promising, future work can focus on integrating deep learning models to learn complex patterns and relationships in large datasets of transactions, enabling them to identify both known and novel fraudulent activities more accurately than traditional methods, behavioral analytics, and compliance frameworks to further enhance the system's adaptability and resilience against evolving fraud patterns.

References:

- Sri D P, Babu G P. (2023). Comparative Study of Machine Learning Algorithms for Credit Card Fraud Detection. International Journal for Research in Applied Science & Engineering Technology (IJRASET) 2 (VIII): 2252-2259.
- Zareapoor M, Shamsolmoali P. (2009). Application of Credit Card Fraud Detection: Based on Bagging Ensemble classifier. In 2009 International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015). Procedia Computer Science 48: 679-685.
- Kumar A, Anand K, Jha S, Gupta J. (2021). Online Credit Card Fraud Analysis Using machine Learning techniques. Data Analytics and Innovation. Advances in Intelligent Systems and Computing. Springer.
- Salekshahrezaee Z, Leevy J L, Khoshgoftaar T M. (2023). The effect of feature extractionand data sampling on credit card fraud detection. Journal of Big Data, 10 (6), 1-17.
- Alshameri F, Xia R. (2023). Credit Card Fraud Detection: an evolution of SMOTE resampling and machine learning model performance. International Journal of Business Intelligence and Data Mining, 23(1): 1-13.
- Zhu M, Zhang Y, Gong Y, Changxin X, Xiang Y. (2014). Enhancing credit card fraud detection: A Neural network and SMOTE integrated approach. Journal of Theory and Practice of Engineering Science, 4(2): 23-30.
- Bhattachryya S, Jha S, Tharakunnel K, Westland J C. (2011). Data mining for credit card: A comparative study. Decision Support Systems, 50(3), 602-613.
- Pozzolo A D, Caelen O, Borgne Y L, Bontempi G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10): 4915-4928.

- Arora V, Leekha R S, Lee K, Kataria A. (2020). Facilitating user authentication from imbalanced data logs of credit cards using artificial intelligence, Hindwai Mobile Informations System, 1-13.
- Miyena I, Jere N. (2016). Deep learning for credit card fraud detection: A review of algorithms, challenges and solutions, IEEE Access, 4: 1-18.
- Maniraj S P, Saini A, Sarkar S D, Ahmed S. (2019). Credit card fraud detection using machine learning and data science. International Journal of Engineering Research and Technology (IJERT), 8(9), 110-115.
- Illeberi E, Sun Y, Wang Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(24), 1-17.
- Dornadula V N, Geetha S. (2019). Credit card fraud detection using machine learning algorithms. In:2019 International Conference on Recent Trends in Advanced Computing (ICRTAC 2019). Procedia Computer Science, 165(2019): 631-641.
- Idress A M, Elhusseny N S, Ouf S. (2024). Credit card fraud detection model-based machine learning algorithms. International Journal of Computer Science, 51(10): 1649-1662.
- Jiang C, Song J, Liu G. (2018). Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism, IEEE Internet of Things Journal, 5(5): 3637-3647.
- Thirunavukkarasu M, Nimisha A, Jyothsna A. (2021). Credit card fraud detection using machine learning. International Journal of Computer Science and Mobile Computing, 10(4), 71-79.
- Soleh M, Djuwitaningrum E. R, Ramli M, Indriasari M. (2020). Feature engineering strategies based on a one-point crossover for fraud detection on big data analytics. Journal of Physics: Conference Series; **1566** (2020); 012049; 2020, 1-8.