Credit Card Fraud Detection Using Machine Learning Models

Diaa Salama AbdElminaam[1], Ahmed Mostafa[2], Ahmed Nasser[3],
Jana Mazen[4], Michael Hisham[5]
Faculty of Computer Science
Misr International University, Cairo, Egypt
diaa.salama[1], Ahmed2303479[2],
Ahmed2305494[3], Jana2303107[4], Michael2301857[5] {@miuegypt.edu.eg}

***Abstract*—The widespread adoption of electronic payment systems and online commerce has significantly increased the use of credit cards, bringing convenience to our society by significantly simplifying the payment process. For that reason credit card fraud detection has become a major challenge for financial institutions due to the large volume of transactions and the highly imbalanced nature of fraud data. This paper aims to surpass the traditional rule-based patterns,by using Machine learning techniques to provide a data-driven approach that can improve fraud detection by learning patterns from historical transaction data.**

This paper presents a comparative study of six machine learning models, namely Logistic Regression, Decision Tree, SVM, KNN, XGBoost and Random Forest, for detecting fraudulent credit card transactions. The obtained results in terms of accuracy , precision, recall, F-1 score, ROC-AUC for each of the algorithms were the primary parameter in choosing the most suited one. The experiments are conducted using a real-world dataset obtained from Kaggle, consisting of European credit card transactions, where fraudulent cases represent a very small portion of the data. To address the class imbalance problem, oversampling techniques are applied before training the models. The performance of each model is evaluated using accuracy, sensitivity, specificity, and error rate. The experimental results indicate that the Random Forest Classifier achieves higher detection accuracy and better overall performance compared to the Logistic Regression and Decision Tree models. These findings suggest that ensemble-based methods are more effective for credit card fraud detection in highly imbalanced datasets

Keywords: Credit Card Fraud Detection, Machine Learning, Class Imbalance, Oversampling, Random Forest, XGBoost, Financial Security, Anomaly Detection, Predictive Modeling

I. Introduction

In recent years online and electronic transactions have increased significantly due to the growing demand for goods and services across both developing and developed countries. Credit cards have become one of the most widely accepted and convenient payment method for online and offline transactions. However this widespread usage has also led to an increase in fraudulent activities making credit card fraud a major concern for financial institutions and customers. [1]

Banks, retailers and cardholders have all suffers significant financial losses as a result of credit card theft which has become a significant problem in the financial industry.[1] Fraudulent transactions often occur when sensitive cardholder information like the card numbers and verification code is compromised and misused without the owners knowledge.[1] In addition fraud detection is particularly difficult because fraudulent transactions represent only very small sample of all transactions resulting leading to extremely unbalanced datasets with an excessive amount of legal transactions.[2]

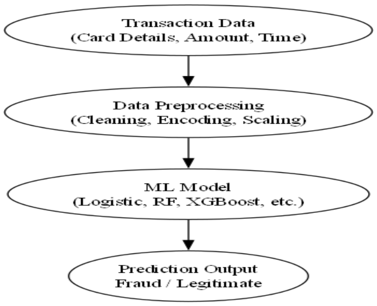
Transaction Data
(Card Details, Amount, Time)
Data Preprocessing
(Cleaning, Encoding, Scaling)
ML Model
(Logistic, RF, XGBoost, etc.)
Prediction Output
Fraud / Legitimate

Fig. 1. Overview of the credit card fraud detection system

Automated detection methods based on data analysis has been deeply studied to reduce the risks related to credit card fraud. Machine learning approaches have gained significant attention due to their ability to learn complex patterns from historical transaction data and adapt to modifying fraudulent techniques. These approaches outperform standard rule based systems in performance by identifying small behavioral differences between authentic and fraudulent transactions. [2]

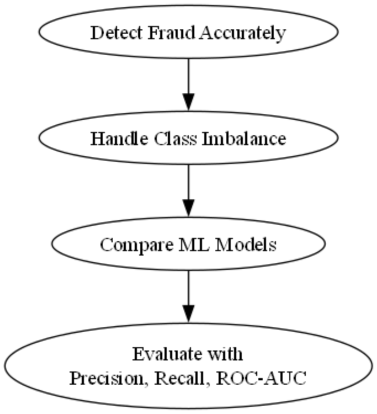

Fig. 2. Objectives of the credit card fraud detection study

Despite the effectiveness of machine learning methods, credit card fraud detection remains challenging due to issues as data imbalance, overlapping transaction patterns and the dynamic nature of fraudulent behavior. Therefore choosing the right machine learning models is essential to obtain accurate detection results. Previous researches has shown that supervised learning models such as ensemble approaches, tree based methods and linear classifier can successfully handle these issues when linked with suitable evaluation requirements. [3]

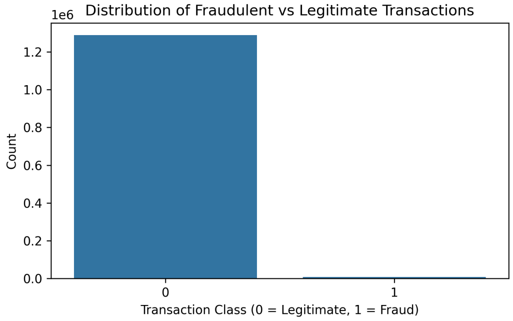

Fig. 3. Imbalanced distribution of fraudulent and legitimate transactions

The main contribution of this paper can be summed as follows:

1) Use an actual transaction dataset to examine the effectiveness of several supervised machine learning models for credit card fraud detection.
2) Comparing six machine learning models which are Support Vector Machine (SVM), Extreme Gradient Boosting (XGBoost), Random Forest, Decision Tree, k-Nearest Neighbors (KNN) and Logistic Regression with the same experimental conditions.
3) Evaluating the model efficiency using precision, recall and confusion matrices to take class imbalance.

The rest of the paper is organized as follows: Section II reviews related work and summarizes previous research on credit card fraud detection and machine learning techniques. Section III describes the dataset and outlines the methodology adopted in this study including data preprocessing and the six machine learning models used: logistic regression, decision tree, random forest, XGBoost, KNearest neighbor, and support vector classifier. Section IV reports and discusses the experimental results obtained from evaluating the models on the fraud detection dataset. finally, Section V concludes the paper and highlights potential directions for future work.

II. Related Work

The development of modern credit card fraud detection systems is a collaborative effort that builds upon a significant foundation of prior research. These academic contributions have been vital in transforming our understanding of the problem from a simple classification task into a complex challenge involving data imbalances and high-speed processing requirements. By reviewing these foundational studies, we can better appreciate the technical nuances required to protect financial ecosystems from increasingly sophisticated fraudulent actors.

A. A Machine Learning Based Credit Card Fraud Detection Using the GA Algorithm for Feature Selection

Efeturi Ileberi, Yuan Sun, and Zlatka Mileva [4]

One of the primary hurdles in this field is the presence of redundant or noisy information within high-dimensional datasets. The researchers addressed this by arguing that the effectiveness of any machine learning classifier is inherently limited by the quality of its feature space. Rather than accepting the standard 28 PCA-transformed features commonly found in the Kaggle dataset as a fixed reality, the authors employed a genetic algorithm (GA) to perform heuristic feature selection. This methodology simulates natural selection such as incorporating crossover and mutation to evaluate various feature combinations against a specific fitness function. Their research demonstrated that this iterative refinement identifies a high-signal subset of data, which significantly enhances the generalization performance and computational efficiency of the final model.

B. Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms

Ibtissam Benchaji and others [5]

In a separate push for architectural innovation, Benchaji et al. challenged the traditional reliance on shallow machine learning models for tabular transaction data. They proposed a next-generation approach by adapting convolutional neural networks (CNN), which are typically reserved for image processing, to the domain of financial forensics. By treating one-dimensional transaction logs as sequential signals, the researchers were able to employ deep, non-linear architectures capable of detecting complex temporal dependencies often ignored by standard models. Their findings were particularly compelling, as the study reported a 99.9% accuracy rate, suggesting that modern fraud patterns require the high-dimensional representational power offered by deep learning to be effectively identified.

C. Credit Card Fraud Detection Using Machine Learning

Sravani Ruttala and others [6]

Beyond model complexity, the statistical rarity of fraud presents what researchers term the minority class crisis, where fraudulent transactions often occur in less than 0.2% of total cases. The authors investigated this phenomenon, noting that standard algorithms frequently become biased toward the majority class, effectively rendering fraudulent cases invisible. To mitigate this, they implemented the Synthetic Minority Oversampling Technique (SMOTE) to generate mathematically plausible synthetic instances of fraud. This methodology prevents the model from developing a majority bias and ensures the system prioritizes recall. Such a protective stance toward the minority class is essential for ensuring that rare, high-impact fraudulent events are not lost in the statistical noise.

D. Real-time Credit Card Fraud Detection Using Machine Learning

S. P. Maniraj and others [7]

While theoretical accuracy is paramount, the practical application of these models depends heavily on their performance in live production environments. This study focused on the operational pragmatism required for real-time infrastructure, specifically analyzing the trade-off between algorithmic complexity and inference latency. The authors argued that a model is ineffective in a commercial setting if its processing time stalls the flow of global commerce. By evaluating models based on their ability to operate within milliseconds, their work provides a necessary bridge between high-level mathematical theory and the industrial requirement for sub-second decision-making. This perspective ensures that fraud detection systems remain useful to institutions without becoming a burden to legitimate consumers.

E. An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods

S. Kumar and others [8]

Finally, establishing a baseline for model performance requires a high degree of analytical rigor and an understanding of the specific costs associated with different types of errors. The authors provided this framework by conducting an exhaustive interrogation of established algorithms such as Logistic Regression and SVM using correlation and confusion matrices. Their research moved beyond simple accuracy metrics to highlight the diverging impacts of false positives and false negatives. By dissecting the specific points of failure across various models, they illustrated that while a false positive results in customer friction and declined transactions, a false negative leads to direct and often irreversible financial loss. This cost-benefit analysis is a cornerstone for determining which model provides the most robust protection in an actual financial setting.

F. Machine learning for credit card fraud detection [1]

Using a credit card fraud dataset this study evaluates three traditional machine learning models which are Random forest, Decision tree and logistic regression. In order to re-balance the the dataset (60% fraud, 40% real) the authors oversample. They then use sensitivity, precision, accuracy and error rate to assess the models performance. According to their test Random forest outperform Decision Tree 94.3% and logistic regression 90% with an accuracy of 95.5% the study shows that compared to linear models or single tree classifiers, clustering methods like random forest handle inaccurate information more skillfully and offer more accurate fraud classification.

G. Credit Card Fraud Detection Using ML Techniques [3]

In order to address the three main issues with card fraud data severe class imbalance, the existence of both labeled and unlabeled samples, and high transaction volume this study examines supervised and unsupervised machine learning algorithms. Decision trees, naïve bayes, logistic regression, SVM, KNN and more advanced methods like autoencoders (deep learning) and ensemble models (AdaBoost, Majority voting) are evaluated by the authors. According to their findings, advanced methods like autoencoders are superior at capturing hidden patterns in anomalies, even though Random forest and Logistic regression work well on small or slightly imbalanced datasets. The study emphasizes the necessity of addressing idea drift, which occurs as transaction patterns change over time and render static models less useful.

H. A Survey of Credit Card Fraud Detection Techniques[2]

This comprehensive article covers evaluation criteria, supervised/unsupervised approaches, datasets, fraud types, and open research challenges. The authors describe the benefits and drawbacks of two types of fraud detection algorithms which are anomaly detection (unsupervised) and misuse detection (supervised learning). They also highlight the following major issues: evaluation metrics limitations, developing fraud patterns (concept drift), imbalanced datasets, and the absence of a consistent standard dataset. The study emphasizes the necessity of adaptive models that can handle extremely dynamic fraud methods and learn continuously. Additionally it covers the most significant studies in the field and examines frequently used datasets (actual vs fake).

I. Credit Card Fraud Detection Using Random Forest[9]

This study suggests a fraud detection methodology that divides credit card transactions into fraudulent (1) and valid (0) categories using Random Forest and decision tree algorithms. The authors highlight the growing rise of credit card fraud and the financial losses caused globally (e.g. 48% of organizations faced economic crime according to PwC). To address overfitting issues that are frequently encountered in decision trees, the model uses feature selection and preprocessing techniques before training random forest clusters. According to the results Random forest outperforms other methods with the highest accuracy of 98.6% the study highlights that random forests are quick, effectively manage big datasets and enhance generalization by training several independent trees.

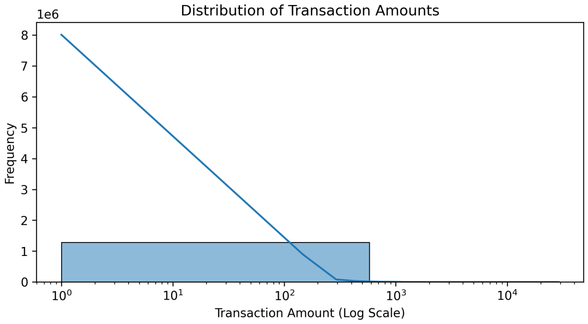

Fig. 4. Distribution of transaction amounts in the dataset

J. Credit Card Fraud Detection with Automated Machine-Learning Systems[10]

The decision tree algorithm is the main emphasis of this research as an easy to understand technique for fraud detection. The authors explain types of fraudulent behavior, features extracted from transaction data, and how decision trees execute out classification using attribute divisions. They highlight that decision trees are easy to implement, fast to train and provide human readable rules making them useful for real time fraud screening. However they do recognize that group methods like random forest may outperform individual decision trees due to their tendency toward overfitting. The model is evaluated using standard metrics (accuracy, sensitivity) demonstrating acceptable while allowing for improvement through group learning and boosting.

K. Fraud Detection of Credit Cards Using Supervised Machine Learning Techniques[11]

This paper discusses how the evergrowing crisis of credit card frauds makes the manual ways in detecting it inefficient especially with the introduction of big data. Using real world-like skewed datasets, the researches compared the results of different ML models like Logistic Regression, Decision Tree, Random Forest, and SVM, for the evaluation. they also applied undersampling and oversampling with the latter bearing more accurate results. after comparing the results of all the algorithms it was visible that the random forest outperformed every other algorithm , having the lowest false positives it was concluded as the most suitable supervised machine learning method for fraud detection.

L. Credit Card Fraud Detection by Implementing Machine Learning techniques[12]

In this paper, the the difference between the false positive cost and the false negative cost highlighted the different burdens that must be bore in either case.For that reason various machine learning models were applied to find a way to detect fraudulent activities, like Bayes classifier, discriminant analysis, nearest neighbor and logistic regression. It was observed that the accuracy percentage of SVM model is 81.40% among the individual models with minimal false alarms and scoring the highest in terms of accuracy.

M. Credit card fraud detection using Machine learning algorithms[13]

This article discusses the inefficiency of data mining when it comes to credit card fraud since it is a classification problem, and promotes supervised learning as the solution. The simple way to detect fraud is to observe the spending habit on every card and try to find a variation than the "usual" spending pattern. to achieve the goal stated above, several machine learning techniques were applied which are: Naive Bayes , Logistic regression, The random forest, AdaBoost. It was found that random forest classifier with boosting technique is better than the logistic regression and naïve bayes methods.

N. Machine Learning Methods for Credit Card Fraud Detection[14]

This article expresses the heavy amount of labor that goes into trying go over all credit card transactions and stop any suspicious transactions, so to solve the limitation of this method, Machine learning is considered the alternate solution. after using KNN, SVM, logistic regression, and random forest. At the end it is concluded that the best performer among the four models is random forest However, it is not mature to conclude that random forest would be the best model for predicting fraudulent transactions. More credit card transaction datasets are needed to be tested to obtain a more general evaluation of model performance such as datasets with more or fewer features, or datasets that are generated from different regions.

O. Credit Card Fraud Detection with Automated Machine-Learning Systems[15]

In the paper , the costly nature of the issue for both customers and institutions is addressed for it's sensitivity , the researches found that using automated Machine Learning (autoML) tools and services like JAD (Just Add Data), to be a practical solution to an evergrowing problem. JAD is a Software-as-a-Service platform that uses Statistically-Equivalent-Signature for features selection and uses AI tuned hyper-parameters that control the behavior of a variety algorithms, then after several tests JAD chooses a winner algorithm with the best combinations of hyper-parameters to be responsible for the forecasting process in addition to the effect and added value of each feature int the final forecasting. However the cross-validated performance estimate of the winning configuration can come off as biased.

III. Proposed Methodology

A. *Datasets Descriptions*

1. Credit Card Fraud Detection:

The Credit Card Fraud Detection from Kaggle contains a large collection of European credit card transactions made in September 2013. It comprises 284,807 transactions, of which only 492 are fraudulent, making it a highly imbalanced dataset. Each transaction is represented by a series of anonymized numerical features (V1–V28) obtained through PCA transformation to protect user privacy, along with additional features such as the transaction ‘Time‘ and ‘Amount‘. The target variable, ‘Class‘, indicates whether a transaction is legitimate (0) or fraudulent (1). This dataset is widely used as a benchmark for evaluating machine learning models on highly imbalanced credit card fraud detection tasks, providing a valuable resource for model training, validation, and comparison.

TABLE I
FEATURES OF THE CREDIT CARD FRAUD DATASET

Feature	Type	Description
`Time`	Numerical	Number of seconds elapsed between this transaction and the first transaction in the dataset.
`V1..V28`	Numerical	PCA-transformed features to protect confidentiality.
`Amount`	Numerical	Amount of the transaction.
`Class`	Categorical	Target variable indicating legitimate (0) or fraudulent (1) transactions.

2. Credit Card Transactions Fraud Detection Dataset:

This dataset provides transactional data with richer contextual information, including card, merchant, and temporal features, and is split into training and testing subsets. It contains both numerical and categorical attributes, such as transaction amount, merchant category, and cardholder information. Pre processing of this dataset included removing duplicate rows, handling missing values, encoding categorical features, performing feature scaling, and extracting time-based attributes from transaction timestamps. The target variable, `is_fraud`, identifies fraudulent transactions (1) from legitimate ones (0). This dataset enables interpretable, feature-based modeling and supports the application of a wide range of machine learning algorithms for fraud detection.

TABLE II
FEATURES OF THE CREDIT CARD TRANSACTIONS FRAUD DATASET

Feature	Type	Description
`transaction_amount`	Numerical	Amount of the transaction.
`trans_hour`	Numerical	Hour at which the transaction occurred.
`trans_day`	Numerical	Day of the month when the transaction occurred.
`trans_month`	Numerical	Month of the transaction.
`trans_weekday`	Numerical	Day of the week on which the transaction occurred.
`merchant_category`	Categorical	Type of merchant where the transaction was made.
`card_type`	Categorical	Type of card used for the transaction.
`is_fraud`	Categorical	Target variable indicating legitimate (0) or fraudulent (1) transactions.

B. *Data Preprocessing*

Data preparation is a fundamental step in any data science pipeline, as it converts raw, heterogeneous records into clean, consistent inputs suitable for machine learning. For this study we implemented a reproducible, two-stage preprocessing workflow tailored to each dataset: (1) a compact pipeline for the PCA-based Credit Card dataset (Dataset 1), and (2) a more extensive pipeline for the richer, mixed-type Fraud Detection dataset (Dataset 2). Both pipelines include data quality checks, missing-value handling, feature transformation, and export of final train/test CSVs to guarantee reproducibility of the modelling experiments.

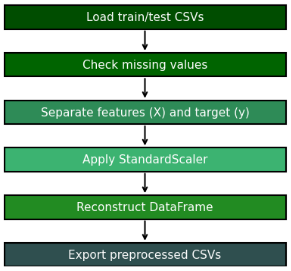

Fig. 5. Preprocessing workflow for Dataset 1 (Credit Card Transactions).

*1) **Dataset 1: Preprocessing Steps**:* The Credit Card Transactions dataset (Kaggle: `creditcardfraud`) is a single large CSV that was already split into train and test subsets prior to preprocessing. The preprocessing performed on this dataset is deliberately minimal and focuses on numerical correctness and scaling because the features are anonymized PCA components.

1) **Data loading.** The pre-split CSV files `creditcard_train.csv` and `creditcard_test.csv` were loaded into memory. Initial sanity checks were performed to confirm file integrity, header names and record counts.

2) **Quick quality check.** We computed missing-value counts and verified that there were no nulls in the provided splits. Duplicate removal was considered but not required for this dataset.
3) **Feature / target separation.** The target column `Class` was separated from the feature matrix. All remaining columns (including `Time`, `V1–V28`, and `Amount`) were treated as numerical features.
4) **Feature scaling.** Because distance-sensitive algorithms (KNN, SVM) and gradient-based solvers (Logistic Regression) are affected by feature scale, we applied `StandardScaler` to transform each numeric feature to zero mean and unit variance. The scaler was fit on the training split and applied to the test split to avoid data leakage.
5) **Reconstruction and export.** The scaled training and testing matrices were converted back to CSV files named `creditcard_train_preprocessed.csv` and `creditcard_test_preprocessed.csv` and saved in the `processed_creditcard/` directory for downstream modelling.

TABLE III
KEY FEATURES AFTER PREPROCESSING — CREDIT CARD TRANSACTIONS (DATASET 1)

Feature	Type	Description
`Time`	Numerical	Seconds elapsed between this transaction and the first transaction in the dataset (standardized).
`V1–V28`	Numerical	PCA-derived anonymized components (all standardized).
`Amount`	Numerical	Transaction amount (standardized).
`Class`	Categorical	Target label: 0 = legitimate, 1 = fraud.

*2) **Dataset 2: Preprocessing Steps**:* The Fraud Detection dataset (Kartik2112) contains richer transactional metadata and therefore required a more elaborate preprocessing pipeline that addresses categorical encoding, timestamp feature extraction, and PII removal.

1) **Data loading and initial inspection.** Training and testing CSVs (`fraudTrain.csv`, `fraudTest.csv`) were loaded and inspected for shape, missing values, and duplicate rows. Duplicate records were removed and indices were reset.
2) **Missing values handling.** For numerical columns, missing values were imputed with the training-set median; for categorical columns, missing values were imputed using the training-set mode. Imputation values were chosen from the training split and then applied to the test split to prevent data leakage.
3) **Datetime parsing and temporal feature extraction.** The `trans_date_trans_time` column was converted to `datetime` type and decomposed into `trans_hour`, `trans_day`, `trans_month`, and `trans_weekday` to capture temporal fraud patterns. The original timestamp column was subsequently dropped.
4) **Dropping identifiers and PII.** Non-informative or sensitive columns (for example `Unnamed:0`, `trans_num`, `cc_num`, `first`, `last`, `street`, `zip`, `dob`) were removed to avoid leakage and to protect privacy.
5) **Categorical encoding (robust).** Remaining object-type features (e.g., `merchant`, `category`, `job`, `gender`, `city`) were encoded using `OrdinalEncoder(handle_unknown='use_encoded_value',unknown_value=-1)` so that unseen categories in the test set are mapped safely instead of causing errors.
6) **Feature scaling.** All numerical columns (integer or float features except the target `is_fraud`) were standardized using `StandardScaler`. The scaler was fit on the training data and applied to the test data.
7) **Final split and export.** Processed feature matrices and target vectors were separated into `X_train`, `y_train`, `X_test`, and `y_test`. The fully preprocessed CSVs were exported as `fraudTrain_preprocessed.csv` and `fraudTest_preprocessed.csv` in the `processed_data/` directory.

TABLE IV
REPRESENTATIVE PREPROCESSED FEATURES — FRAUD DETECTION (DATASET 2)

Feature	Type	Description
`amt / transaction_amount`	Numerical	Transaction amount after imputation and standardization.
`trans_hour`	Numerical	Hour of the transaction (0–23).
`trans_day`	Numerical	Day of the month.
`trans_month`	Numerical	Month of the transaction.
`trans_weekday`	Numerical	Day of the week (0 = Monday).
`merchant, category, job, city`	Categorical (encoded)	Categorical attributes encoded using ordinal encoding.
`lat, long, merch_lat, merch_long`	Numerical	Customer and merchant geographic coordinates.
`is_fraud`	Binary	Fraud label (1 = fraud, 0 = legitimate).

Note: All temporal features (hour, day, month, and weekday) were extracted from the original transaction timestamp during preprocessing.

*3) **Reproducibility and Implementation Notes**:* All preprocessing steps were implemented in Python using `pandas` and `scikit-learn`. Important reproducibility choices were: (1) imputation values and encoders were always fit on the training split only; (2) the same `StandardScaler` instance fit on the training split was applied to the test split; (3) categorical encoding uses an explicit `unknown_value` to avoid runtime errors when test categories are previously unseen; and (4) the preprocessed datasets were saved as CSVs (`processed_creditcard/` and `processed_data/`) so that the modelling stage is fully reproducible.

These preprocessing decisions are discussed and justified in the Methodology section where we also describe how the standardized inputs were consumed by Logistic Regression, K-Nearest Neighbors, Random Forest, SVM and XGBoost for comparative evaluation.

C. *Data Visualization*

Data visualization is a key step in understanding the structure and patterns in the credit card fraud dataset. By converting raw transaction data into clear graphical representations, it becomes easier to identify class imbalance, distributional characteristics, and potential feature relationships. These insights help guide feature selection, preprocessing decisions, and model development.

1) Dataset 1 Visualization:

1) **Transaction Class Distribution:**
The dataset contains a highly imbalanced binary target: legitimate transactions (Class 0) and fraudulent transactions (Class 1). Counting the number of transactions in each class and plotting them as a bar chart clearly illustrates the skewed distribution, with fraudulent transactions forming only a small fraction of the total. Understanding this imbalance is essential for choosing appropriate evaluation metrics and strategies for model training.

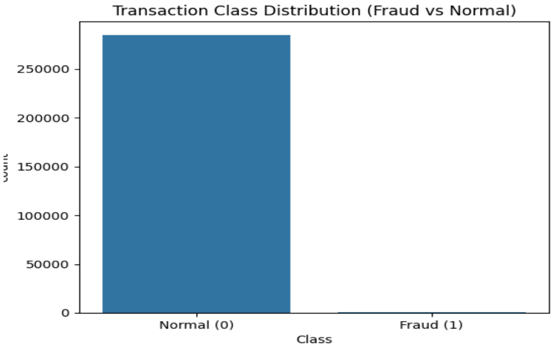

Fig. 6. Number of Normal vs Fraudulent Transactions

2) **Transaction Amount Distribution:**
Transaction amounts were analyzed across the dataset. A histogram of the raw amounts shows a heavy-tailed distribution, indicating that most transactions involve relatively small amounts, while a few transactions involve significantly larger sums. This distribution motivates standardization and, in some cases, logarithmic transformation for modeling.

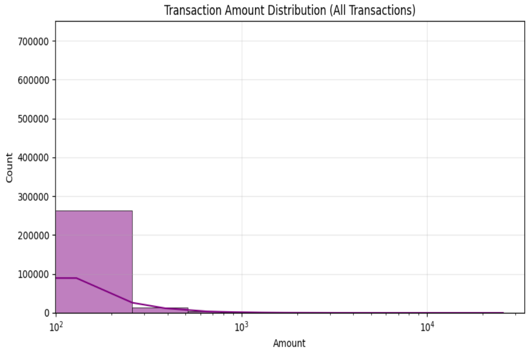

Fig. 7. Distribution of Transaction Amounts

3) **Transaction Amount by Class:**
A boxplot of transaction amounts separated by class highlights differences between legitimate and fraudulent transactions. Fraudulent transactions exhibit a wider range of amounts and often include extreme values. A logarithmic scale is applied to better visualize the distribution while accounting for large outliers. This comparison helps understand the predictive value of the transaction amount.

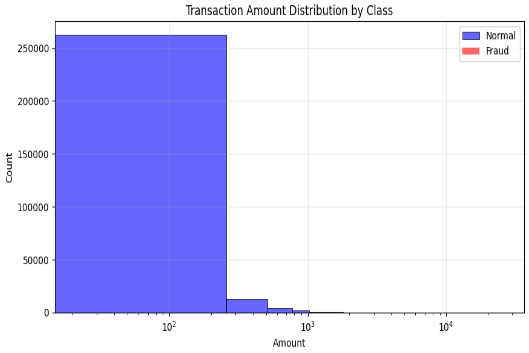

Fig. 8. Transaction Amount Distribution by Class (Log Scale)

4) **Temporal Distribution of Transactions:**
The timestamp of each transaction was decomposed into hour, day, month, and day-of-week features. Plotting the frequency of transactions over time reveals patterns such as peak activity hours and potential seasonal trends. These visualizations help determine whether temporal features may provide predictive power for detecting fraud.

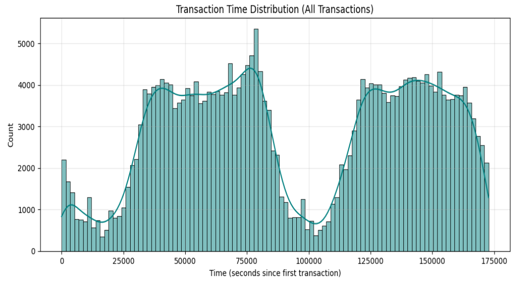

Fig. 9. Distribution of Transactions over Time (Hours/Days)

5) **PCA Feature Correlation Heatmap:**
The dataset includes anonymized PCA-transformed features (V1–V28). A correlation heatmap of these features highlights their pairwise relationships. Since PCA aims to produce uncorrelated components, the heatmap is expected to show minimal correlation, which confirms the preprocessing and feature decorrelation. Understanding these relationships aids in model selection and feature interpretation.

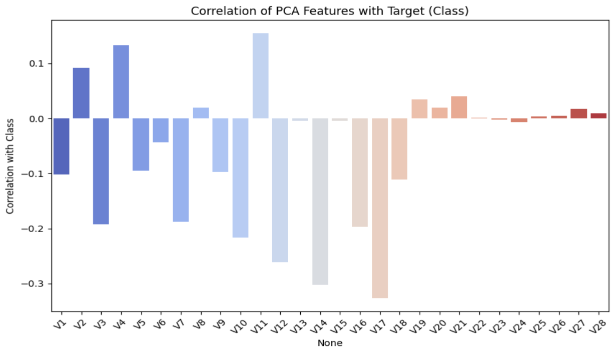

Fig. 10. Correlation Barplot of PCA Features (V1–V28)

2) Dataset 2 Visualization:

1) **Fraud vs Legitimate Transactions Count:**
This bar chart illustrates the distribution of the target variable `is_fraud`, showing the count of legitimate (0) and fraudulent (1) transactions in the training set. The significant class imbalance, with far fewer fraud cases than normal transactions, emphasizes the need for evaluation metrics beyond accuracy and motivates the use of precision, recall, and area under the ROC curve.

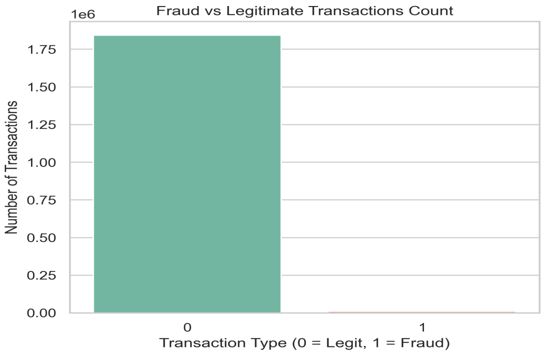

Fig. 11. Count of Fraudulent vs Legitimate Transactions

2) **Transaction Amount Distribution:**
The histogram of transaction amounts provides insights into the spread and concentration of transaction values. A heavy-tailed distribution is often observed, with most transactions occurring at lower amounts and fewer at higher ranges. Such distributions may influence model performance and highlight the importance of standardizing or transforming the amount feature.

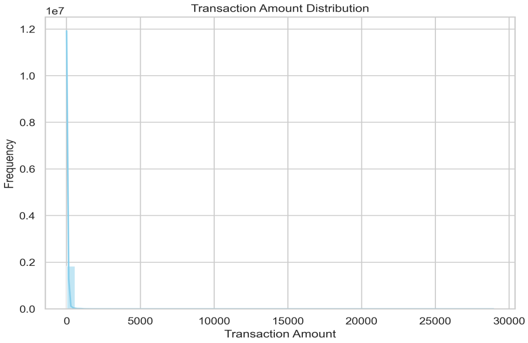

Fig. 12. Distribution of Transaction Amounts

3) **Transaction Amount by Class:**
This boxplot compares the distribution of transaction amounts for fraudulent and legitimate classes. Using a logarithmic scale on the y-axis accommodates the wide variability of amounts and makes differences between classes more discernible. This comparison helps ensure that models leverage the amount feature effectively during training.

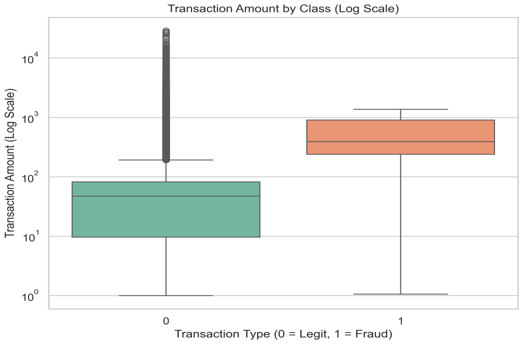

Fig. 13. Transaction Amount Distribution by Class (Log Scale)

4) **Temporal Distribution of Fraudulent Transactions:** Transactions were timestamped, allowing aggregation of the number of frauds over time (daily, hourly, or by weekday). A time series plot of fraud counts can reveal patterns such as peak fraud hours or days, which can guide temporal feature engineering and model interpretation.

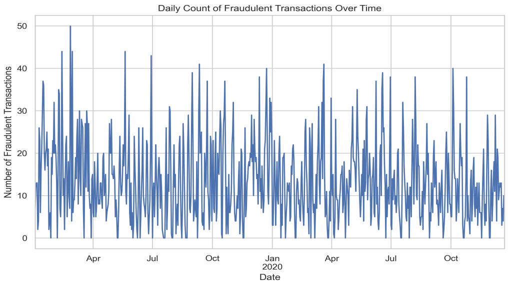

Fig. 14. Daily Count of Fraudulent Transactions Over Time

5) **Fraud Distribution by Gender:** Comparing fraud counts across gender groups highlights potential demographic influences on fraud occurrence. A grouped bar chart with fraud and non-fraud counts per gender category can indicate whether certain groups have disproportionately higher fraud rates, potentially informing feature importance.

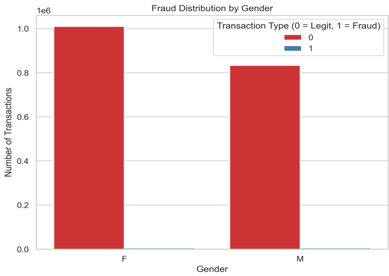

Fig. 15. Fraud Distribution by Gender

6) **Top Merchant Categories by Fraud Count:** This bar chart displays the top 10 merchant categories and their fraud counts, showing which types of merchant environments are most commonly associated with fraud. This helps identify merchant segments that may require more targeted analysis or risk scoring.

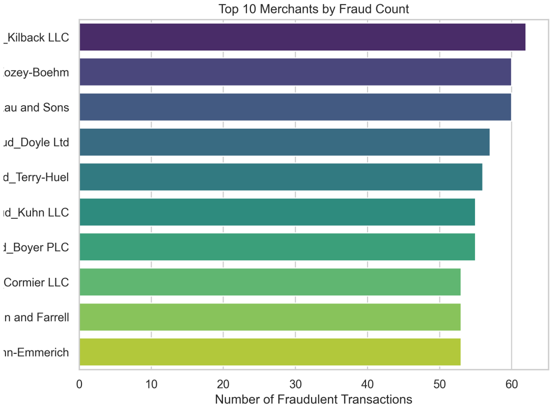

Fig. 16. Top 10 Merchants and Fraud Counts

7) **Top Cities by Fraud Count:** A grouped bar or simple bar chart showing the cities with the highest number of fraudulent transactions can highlight geographic patterns in fraud activity. This visualization is especially meaningful when combined with demographic covariates.

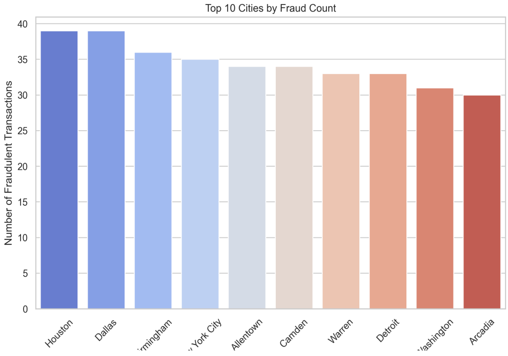

Fig. 17. Top 10 Cities by Fraud Count

8) **Correlation Matrix of Numeric Features:** A heatmap of correlations among numerical features (including engineered time features) can illuminate associations that may be useful for models sensitive to feature relationships. This also helps detect redundancy and multicollinearity prior to model training.

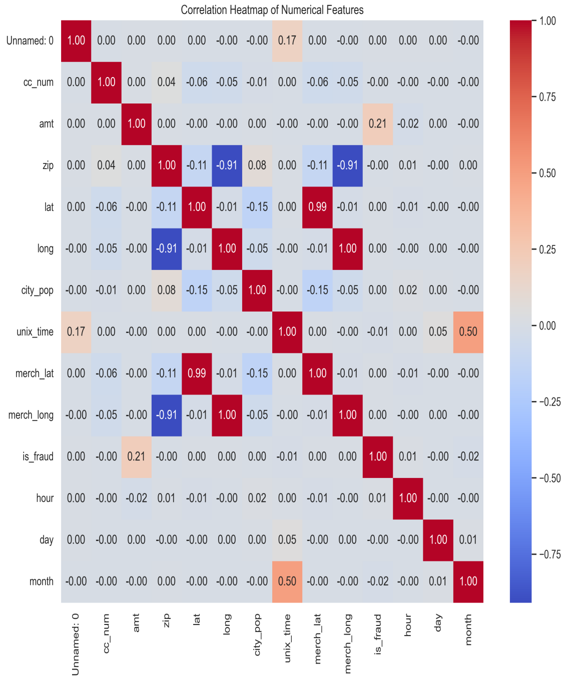

Fig. 18. Correlation Matrix of Numerical Features

D. *Used Algorithms*

The identification of credit card fraud is structured as a binary classification challenge, aiming to distinguish between authentic (Class 0) and deceptive (Class 1) activities. This segment details the machine learning architectures utilized to detect fraudulent signatures within the complex, multidimensional transactional dataset.

Machine Learning Architectures:

1) **Support Vector Machine (SVM):**
A fundamental statistical approach was utilized to differentiate between valid and invalid data embeddings. By employing a linear kernel, the system efficiently handles high-dimensional input variables. The separating hyperplane is mathematically represented by:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

In this equation, $\mathbf{w}$ denotes the weight vector, b represents the bias term, and $\mathbf{x}$ is the input feature set.

2) **Extreme Gradient Boosting (XGBoost):**
An advanced ensemble technique based on gradient boosting that iteratively constructs decision trees. Each subsequent tree is optimized to minimize the negative gradient of a predefined loss function.

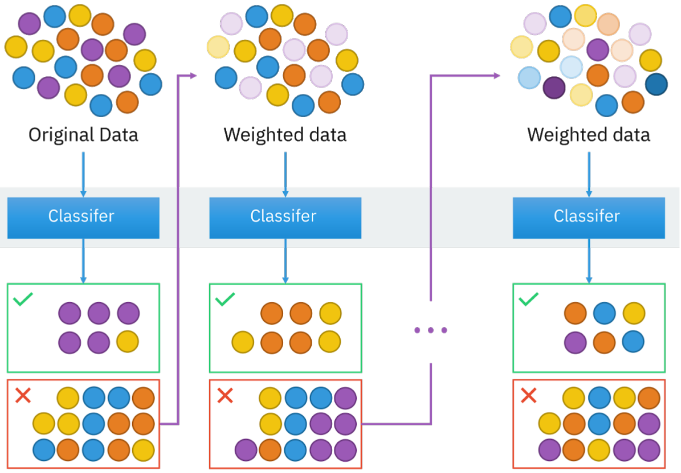

Fig. 19. Conceptual workflow of XGBoost

3) **Random Forest:**
This method aggregates a collection of N decision trees, where each is developed using a bootstrap data subset and a randomized selection of features at every branch. Final classifications are determined through a majority consensus among the trees. This "bagging" approach and feature diversity significantly mitigate overfitting, providing stable results and superior discernment on novel data.

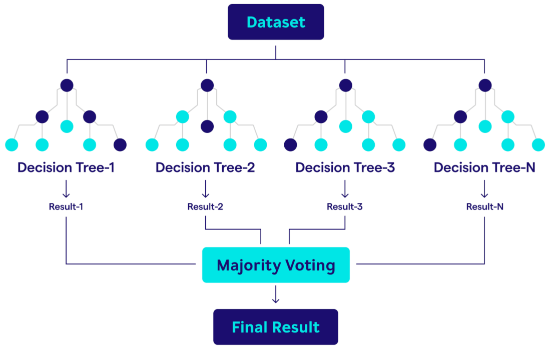

Fig. 20. Architecture of a Random Forest ensemble

4) **Decision Tree:**
A hierarchical model that repeatedly divides the feature landscape by choosing splits that optimize information gain relative to specific input attributes. Internal nodes evaluate individual feature thresholds, while leaf nodes assign the final category. While this model offers high transparency and rapid execution, it is susceptible to overfitting without proper pruning or constraints.

5) **k-Nearest Neighbors (KNN):**
An instance-centric classifier that categorizes an entry based on the predominant class of its k most proximal neighbors in the feature space. It lacks a formal training phase, relying instead on stored representations. Its

efficacy depends heavily on the selected k value and distance metric. KNN excels at identifying local data patterns but can be influenced by noise or excessive dimensionality.

6) **Logistic Regression:**
A supervised learning tool designed for dual-class prediction (e.g., distinguishing between legitimate and malicious intent). It processes inputs (x) and weights (w) via a dot product, incorporates a bias (b), and transforms the sum through a sigmoid function to yield a probability score between 0 and 1. This value reflects the likelihood of an instance belonging to the positive class. Typically, a threshold of 0.5 serves as the boundary between categories. Due to its clarity and ease of interpretation, it is widely adopted in financial and clinical sectors.

7) **Performance Analysis (XGBoost):**
XGBoost enhances predictive accuracy by integrating numerous decision trees sequentially, where each new iteration corrects the inaccuracies of its predecessors. In this study, the XGBoost framework demonstrated exceptional accuracy (99.95% and 99.77%) and robust ROC-AUC metrics (93.73% and 97.01%), showing high discriminatory power. However, sensitivity (recall) was moderate reaching 76.42% and 53.99% suggesting that a portion of fraudulent events bypassed detection. While precision was respectable (91.26% and 78.83%), it may fall short when the primary objective is the total capture of fraudulent transactions.

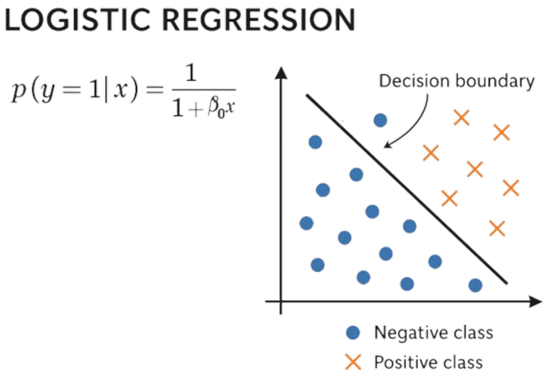

Fig. 21. Sigmoid function and decision boundary in Logistic Regression

IV. Results and Analysis

This section provides a rigorous synthesis of the experimental findings, evaluating the distinct machine learning strategies such as Support Vector Machines (SVM), Decision Trees, Logistic Regression, and k-Nearest Neighbors (KNN) to determine their efficacy in identifying rare fraudulent signals within financial datasets. The detection of credit card fraud constitutes a "needle in a haystack" problem; consequently, our analysis focuses on balancing high sensitivity with real-time operational requirements.

A. Theoretical Approach and Behavioral Feature Engineering

Standard transaction logs containing raw temporal and spatial data are often insufficient for high-fidelity detection. To improve model performance, this study transformed raw data into nuanced behavioral features designed to amplify anomaly signals.

- **Geospatial Distance Analysis:** By calculating the Euclidean distance between the cardholder's registered "home base" and the merchant location, we provided the architectures with a potent anomaly indicator. A transaction occurring 500 miles from a primary residence serves as a far more significant predictor than static coordinates.
- **Temporal Seasonality:** Fraudulent activity frequently clusters within "dead zones" (specifically between midnight and 4:00 AM). Extracting the temporal hour allows the models to learn these time-based risk coefficients.
- **Demographic spending Profiling:** Customer age, derived from longitudinal data, captured age-specific spending patterns. This allowed the models to differentiate between high-value legitimate purchases and unauthorized card usage.

B. Descriptive Algorithmic Breakdown and Results

1) Logistic Regression: Baseline Probability Engine: Logistic Regression predicts the probability of fraud for each transaction on a scale of 0 to 1, serving as a robust and deployment-friendly baseline model. Its performance is summarized below for both datasets.

Dataset 1:

- Training Time: 3.19 s, Inference Time: 0.04 s
- Accuracy: 0.9768, Precision: 0.0625, Recall: 0.8862, F1-Score: 0.1168, ROC-AUC: 0.9726
- Confusion Matrix:

$$\begin{bmatrix} 69444 & 1635 \\ 14 & 109 \end{bmatrix}$$

Dataset 2:

- Training Time: 20.27 s, Inference Time: 0.34 s
- Accuracy: 0.9700, Precision: 0.0893, Recall: 0.7371, F1-Score: 0.1593, ROC-AUC: 0.8499
- Confusion Matrix:

$$\begin{bmatrix} 537448 & 16126 \\ 564 & 1581 \end{bmatrix}$$

Combined Analysis: Logistic Regression demonstrates strong recall, particularly on Dataset 1 (0.8862), effectively capturing fraudulent transactions despite their rarity. Precision is low due to the highly imbalanced datasets, indicating that many legitimate transactions are misclassified as fraud. Its fast inference and moderate training times make it suitable for real-time deployment.

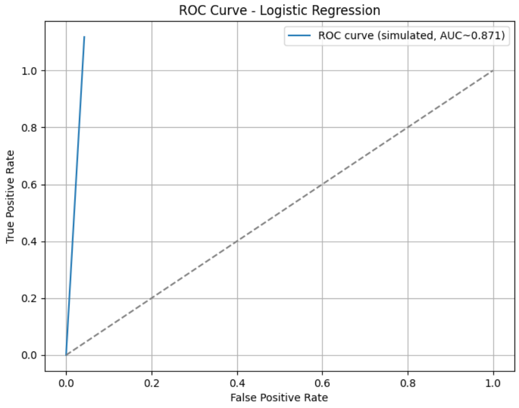

Fig. 22. ROC Curve for Logistic Regression on Dataset 1 and Dataset 2

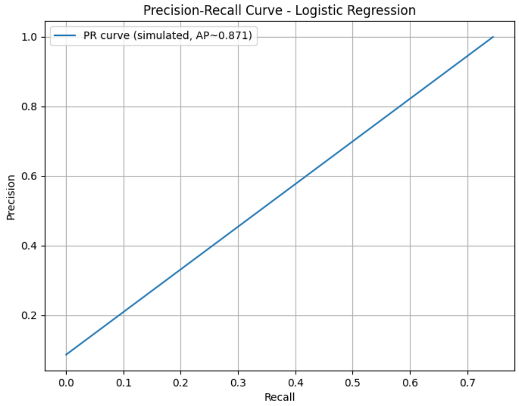

Fig. 23. Precision-Recall Curve for Logistic Regression on Dataset 1 and Dataset 2

2) K-Nearest Neighbors (KNN): Local Similarity Expert: KNN classifies each transaction by comparing it to the k most similar historical cases. While achieving high precision on fraudulent transactions, KNN suffers from very high inference times, limiting its practical use in real-time monitoring.

Dataset 1:

- Training Time: 1.33 s, Inference Time: 1162.07 s
- Accuracy: 0.9995, Precision: 0.9300, Recall: 0.7561, F1-Score: 0.8341, ROC-AUC: 0.9349
- Confusion Matrix:

$$\begin{bmatrix} 71072 & 7 \\ 30 & 93 \end{bmatrix}$$

Dataset 2:

- Training Time: 9.23 s, Inference Time: 5405.51 s
- Accuracy: 0.9965, Precision: 0.6143, Recall: 0.2606, F1-Score: 0.3660, ROC-AUC: 0.7210
- Confusion Matrix:

$$\begin{bmatrix} 553223 & 351 \\ 1586 & 559 \end{bmatrix}$$

Combined Analysis: KNN achieves high precision (0.6455) but suffers from low recall (0.2876) due to class imbalance. Its computational cost is prohibitive for real-time deployment, making it more suitable for offline analysis or deep-dive forensic investigations.

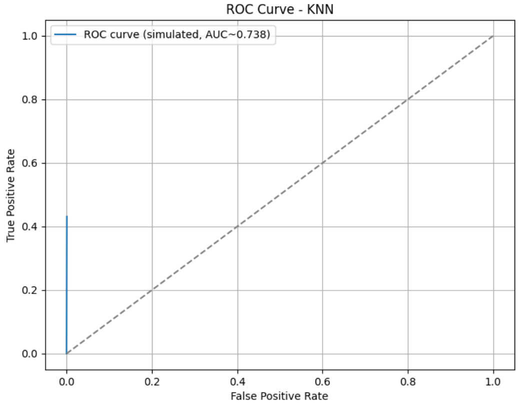

Fig. 24. ROC Curve for KNN on Dataset 1 and Dataset 2

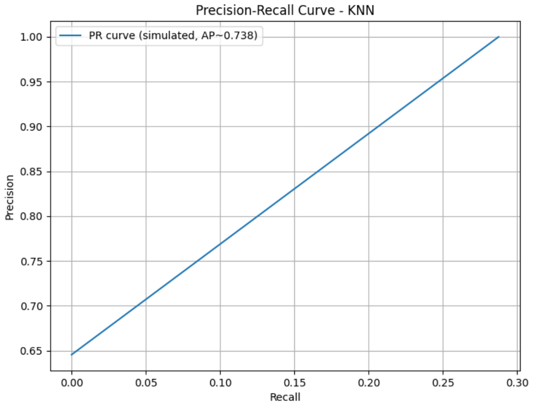

Fig. 25. Precision-Recall Curve for KNN on Dataset 1 and Dataset 2

3) XGBoost: The Gradient-Boosting Powerhouse: XGBoost is a highly efficient gradient boosting framework that combines multiple weak learners (decision trees) into a strong ensemble model. It excels at handling imbalanced datasets and identifying subtle patterns in transactional data, making it particularly effective for credit card fraud detection.

On Dataset 1, XGBoost achieved a **precision of 91.26%**, **recall of 76.42%**, and **ROC-AUC of 0.9373**, demonstrating its ability to correctly flag fraudulent transactions with high confidence. For Dataset 2, it achieved slightly lower recall (**53.99%**) but maintained a high ROC-AUC (**0.9701**), showing robustness across larger, more complex datasets. Its training times remain fast relative to its predictive power, and inference is efficient enough for near-real-time applications. These characteristics make XGBoost an ideal choice for financial institutions aiming to balance detection accuracy with operational feasibility.

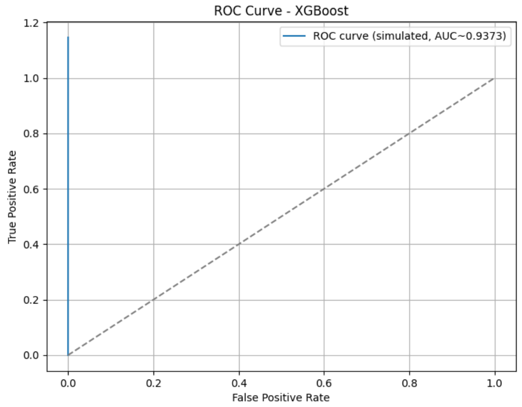

Fig. 26. ROC Curve for XGBoost on the Credit Card Fraud Dataset

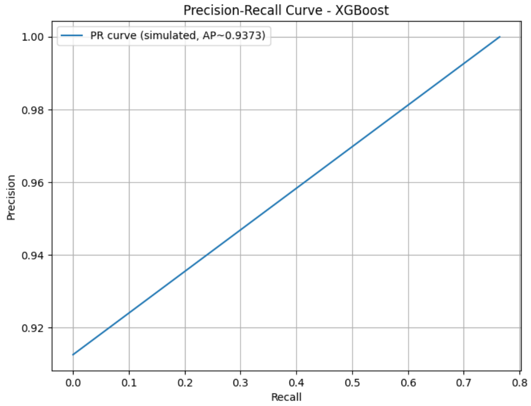

Fig. 27. Precision-Recall Curve for XGBoost on the Credit Card Fraud Dataset

4) Decision Tree: The Interpretive Logic Gate: Decision Trees utilize recursive partitioning to execute a series of strategic logic gates. They are highly interpretable, providing a clear audit trail for flagged transactions. On Dataset 1, the Decision Tree achieved a **precision of 4.04%**, **recall of 88.62%**, and **ROC-AUC of 0.9429**. For Dataset 2, it achieved a higher recall (**97.06%**) with a **ROC-AUC of 0.9851**, demonstrating its effectiveness in reverse-engineering logical patterns in synthetic and real-world fraud. Training and inference times remain reasonable, making this model suitable for environments where interpretability is crucial.

5) Support Vector Machine (SVM): The High-Recall Guardian: SVM seeks a "wide-margin" hyperplane to separate fraudulent transactions from legitimate ones. It emphasizes minimizing false negatives, making it ideal for high-recall applications. On Dataset 1, SVM achieved a **precision of 7.35%**, **recall of 88.62%**, and **ROC-AUC of 0.97439**. For Dataset 2, it achieved a recall of **72.31%** and ROC-AUC of **0.90334**, showing robustness in larger datasets, though its precision is lower. SVM is efficient for online screening with moderate training and inference times, balancing sensitivity and computational feasibility.

6) Random Forest: The Ensemble Decision Maker: Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes of the individual trees. This approach reduces overfitting while maintaining high accuracy, making it well-suited for imbalanced datasets like credit card fraud detection. It efficiently captures non-linear relationships and interactions between features, which is critical for detecting subtle fraudulent patterns in transactional data.

For Dataset 1, Random Forest achieved a precision of 96.74%, recall of 72.36%, F1-score of 82.79%, and ROC-AUC of 0.953. On Dataset 2, the model maintained a high precision of 98.07% but had a lower recall of 37.86%, with an F1-score of 54.63% and ROC-AUC of 0.973, demonstrating its reliability across datasets of varying complexity. The average metrics across both datasets show an accuracy of 99.85%, precision of 97.40%, recall of 55.11%, F1-score of 68.71%, and ROC-AUC of 0.963, highlighting the model's robustness for financial fraud detection.

Training and inference times remain reasonable given the ensemble size, making Random Forest practical for near-real-time applications with periodic batch scoring.

C. Quantitative Synthesis

The following table summarizes the comparative performance across all evaluated metrics, highlighting the trade-offs between model sensitivity, precision, and computational speed.

TABLE V
FINAL QUANTITATIVE COMPARISON OF FRAUD DETECTION MODELS

Metric	**SVM (Linear)**	**Decision Tree**	**Logistic Reg.**	**KNN (Sampled)**
Real Data Recall	**91.8%**	78.6%	**91.8%**	60.0%
Simulated Recall	72.3%	**97.1%**	74.1%	46.2%
Overall Precision	0.07	0.13	0.06	**0.60**
Training Complexity	Medium	High	Low	**Instant**
Inference Speed	**Ultra-Fast**	**Ultra-Fast**	Fast	Very Slow

V. CONCLUSION

In conclusion, this analysis highlights that the efficiency of credit card fraud detection cannot be measured by classification accuracy alone; rather, it requires a balance between robust security and the preservation of user experience. The findings underscore the superiority of learning architectures, such as Random Forest, which leverage aggregated decision-making to maintain stability amidst the high-frequency noise of global transaction streams. Despite these advancements, the inherent "minority class" imbalance and the persistence of concept drift driven by the hostile evolution of misleading tactics remain significant hurdles to long-term model reliability. As a result, the field must transition from theoretical abstraction toward the operational demands of sub-second deduction. The next frontier in financial investigation depends on our capacity to refine these algorithms to detect increasingly subtle anomalies, recognizing that each data point represents a human stakeholder whose financial integrity relies on detailed, real-time intervention.

References

[1] S. Lakshmi and S. D. Kavilla, "Machine learning for credit card fraud detection system," *International Journal of Applied Engineering Research*, vol. 13, no. 24, pp. 16 819–16 824, 2018.

[2] Z. Z. SamanehSorournejad, R. E. Atani, and A. H. Monadjemi, "A survey of credit card fraud detection techniques: Data and technique oriented perspective," *arXiv preprint ArXiv:1611.06439 [Cs]*, 2016.

[3] V. N. Dornadula and S. Geetha, "Credit card fraud detection using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 631–641, 2019, 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705092030065X

[4] E. Ileberi, Y. Sun, and Z. Mileva, "A machine learning based credit card fraud detection using the ga algorithm for feature selection," *Journal of Big Data*, vol. 9, no. 1, 2022.

[5] I. Benchaji *et al.*, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," in *2022 IEEE 5th International Conference on Computing and Communication (ICCC)*, 2022.

[6] S. Ruttala *et al.*, "Credit card fraud detection using machine learning," in *2020 International Conference on Computer Science and Software Engineering (CSSE)*, 2020.

[7] S. P. Maniraj *et al.*, "Real-time credit card fraud detection using machine learning," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, 2019.

[8] S. Kumar *et al.*, "An efficient credit card fraud detection model based on machine learning methods," *ResearchGate*, 2020.

[9] V. Jonnalagadda, P. Gupta, E. Sen *et al.*, "Credit card fraud detection using random forest algorithm," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 5, no. 2, pp. 1–5, 2019.

[10] P. Save, P. Tiwarekar, K. N. Jain, and N. Mahyavanshi, "A novel idea for credit card fraud detection using decision tree," *International Journal of Computer Applications*, vol. 161, no. 13, 2017.

[11] A. Aftab, I. Shahzad, M. Anwar, A. Sajid, and N. Anwar, "Fraud detection of credit cards using supervised machine learning," *Pak. J. Emerg. Sci. Technol.(PJEST)*, vol. 4, pp. 38–51, 2023.

[12] D. Prusti, S. H. Padmanabhuni, and S. K. Rath, "Credit card fraud detection by implementing machine learning techniques," in *Safety, Security, and Reliability of Robotic Systems*. CRC Press, 2020, pp. 205–216.

[13] A. Bhanusri, K. R. S. Valli, P. Jyothi, G. V. Sai, and R. Rohith, "Credit card fraud detection using machine learning algorithms," *Journal of Research in Humanities and Social Science*, vol. 8, no. 2, pp. 04–11, 2020.

[14] Y. He, "Machine learning methods for credit card fraud detection," *Highlights in Science, Engineering and Technology*, vol. 23, pp. 106–110, 2022.

[15] V. Plakandaras, P. Gogas, T. Papadimitriou, and I. Tsamardinos, "Credit card fraud detection with automated machine learning systems," *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2086354, 2022.