Comparative study on credit card fraud detection

Parth Bramhecha

Student, SCTR’s Pune Institute of Computer Technology, IT Engineering Department, Pune, Maharashtra, India,
Parth.bramhecha007@gmail.com

Abstract

This research demonstrates the use of a combination of supervised and unsupervised machine learning to improve th e accuracy of credit card fraud detection. While various algorithms such as XGBoost, logistic regression, decision tr ees, and random forests have been compared, this study focuses specifically on improving fraud detection by handli ng nonsuspicious data and using a collaborative approach. Unlike previous studies that only compare models, this st udy emphasizes the interpretability, realworld applicability, and evolutionary dynamics of financial security models. This scheme guarantees a higher level of fraud detection while maintaining computational efficiency.

Keywords: Machine learning, Logistic Regression, XGBoost, Random Forest, Financial transactions security, Banking sector , Decision Tree

1. Introduction

In today’s digital age, credit card fraud poses a significant financial risk. Law-based fraudsters aim to combat evolving patterns of fraud. Machine learning offers a revolutionary approach to fraud detection, analyzing complex business practices to distinguish between legitimate and fraudulent transactions. However, current models face challenges such as data inconsistency, interpretability, and computational complexity. This study aims to close these gaps by comparing the latest models and improving fraud detection methods.

Legacy laws based on static fraud laws often fail to address the evolving nature of fraud with new ideas and technology. Machine learning (ML) holds promise for its ability to recognize complex patterns, adapt to new threats , and process big data. This work builds on existing research and explores various machine learning techniques for c redit card fraud, including random forests, decision trees, and advanced techniques such as XGBoost and neural net works (ANN) [3][4][7][9][14] These algorithms have been proven to be effective in handling inconsistent data and p rovide high accuracy and recovery [6] [8] [11].

Recent trends, including the increase in fraud during major online shopping events such as Black Friday an d CyberMonday, have highlighted the limitations of traditional detection processes and the urgent need for new appr oaches. Additionally, the psychological impact on fraud victims highlights the need for improved fraud prevention [1] [10] [15]. Machine learningbased approaches such as SMOTE convolutional neural networks and hybrid models s how great promise for improving fraud detection while addressing class differences [6] [13].

This study compares four machine learning algorithms (XGBoost, logistic regression, decision tree, and ran dom forest). The algorithms are evaluated on their accuracy, sensitivity, specificity, F1 score, and ROC, including th eir ability to identify fraudulent patterns and adapt to new threats. This study demonstrates the advantages of random forests in terms of accuracy, logistic regression interpretation, decision tree clarity, and XGBoost performance.
[3] [9] [12] [14].

Through an indepth study of the methods, techniques, and usage of algorithms, this research aims to improv e fraud detection techniques. The insights generated will enable financial institutions to improve business security an d foster a more secure digital payments ecosystem.

2. Literature Survey

Table 1. Summary of literature survey

SR no	Title	Author	Year of publication	Aim/Objective	Identified Gap
1	Credit card fraud detection using machine learning	Bharadwaj, S.	2024	Compares the effectiveness of two low-cost machine learning techniques, Random Forest and K-Nearest Neighbor (K-NN), in predicting fraud. Evaluates based on metrics, ease, and cost-effectiveness.	Limited focus on high-class imbalance handling and cost constraints
2	Credit card fraud detection using machine learning	Tressa, N.	2023	Proposes a system for detecting fraud using Decision Tree and Random Forest algorithms, aiming to detect fraudulent transactions and prevent unauthorized charges.	Lacks interpretability and scalability analysis, no feature selection strategies explored
3	Credit card fraud detection using machine learning	Nayak, N. A.	2023	Evaluates Decision Tree, Random Forest, and Extreme Gradient Boosting methods to assess performance on public and real-world financial datasets for fraud detection.	Does not explore hybrid approaches, lacks real-time evaluation
4	Credit card fraud detection using machine learning	Prajapati, D.	2021	Compares the effectiveness of Random Forest, XGBoost, and ANN in preventing fraudulent credit card transactions.	Does not focus on real-time adaptability or explainability
5	Credit card fraud detection system based on operational and transaction features using SVM and	Sudha, C., & Akila, D.	2021	Classifies transactions using SVM and Random Forest, evaluating performance with precision, accuracy, recall, and F1-score.	Lacks feature importance analysis and practical application insights

SR no	Title	Author	Year of publication	Aim/Objective	Identified Gap
	Random Forest classifiers				
6	Enhancing credit card fraud detection: A neural network and SMOTE integrated approach	Zhu, M., Zhang, Y., Gong, Y., Xu, C., & Xiang, Y.	2024	Demonstrates superior accuracy of the NN+SMOTE model in detecting fraud, explores preprocessing techniques for managing class imbalance.	Does not explore alternative class balancing techniques
7	Credit card fraud detection using advanced machine learning and deep learning algorithms	Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M.	2022	Discusses the performance of deep learning methods, particularly CNNs, in detecting fraud, outperforming traditional algorithms.	Focuses only on CNN, lacks an ensemble perspective
8	Credit card fraud detection using ANN	Asha, R. B., & Kumar, S.	2021	Finds ANN to be highly effective in fraud detection, achieving nearly 100% accuracy compared to other algorithms like K-Nearest Neighbor and Support Vector Machine.	Ignores computational efficiency and comparison with simpler models
9	An efficient credit card fraud detection model based on machine learning methods	Trivedi, N. K., Simaya, S., Lilhore, U. K., & Sharma, S. K.	2020	Evaluates several ML techniques, with Random Forest emerging as the best-performing method for fraud detection.	Lacks scalability analysis for large-scale fraud detection

SR no	Title	Author	Year of publication	Aim/Objective	Identified Gap
10	Credit card fraud detection using ANN	Oumar, A. W., & Augustin, D. P.	2019	Proposes a method using ANN for fraud detection, with performance compared to K-Nearest Neighbor and Support Vector Machine, achieving near-perfect accuracy.	Limited focus on explainability and generalization
11	A comparative analysis of various credit card fraud detection techniques	Jain, Y., Tiwari, N., Dubey, S., & Jain, S.	2019	Proposes a hybrid model combining ANN, Naive Bayesian Networks, and other techniques to improve fraud detection performance.	Lacks focus on interpretability and dynamic fraud trends
12	Novel machine learning-based credit card fraud detection systems	Shaaban, S., Alsobhi, A., Aburas, H., & Abushaikha, Z.	2024	Introduces compact data learning (CDL) for feature reduction while maintaining model accuracy and addresses imbalanced datasets in fraud detection.	Does not address computational overhead or real-time detection
13	Enhancing credit card fraud detection: An ensemble machine learning approach	Ashawa, M., Osamor, J., & Adejoh, J.	2024	Proposes an ensemble model integrating multiple classifiers to address challenges like data imbalance in fraud detection.	Does not compare ensemble models with deep learning approaches
14	Credit card fraud detection using ensemble models	O'Shea, A. J., Browne, L. C., & Green, M. T.	2023	Evaluates ensemble methods with SMOTE to counteract data imbalance, proving their effectiveness over traditional methods in fraud detection.	Lacks focus on real-world applicability and computational efficiency

SR no	Title	Author	Year of publication	Aim/Objective	Identified Gap
15	Using machine learning to detect credit card fraud	Parashar, S., Bhargav, S., & Bhardwaj, A.	2023	Evaluates various machine learning algorithms emphasizing preprocessing and management of imbalanced datasets for improved fraud detection accuracy.	Lacks feature selection techniques and hybrid modeling evaluation

Key Gaps Identified:

Class Imbalance Handling

Common Gap: Many studies, including those by Bharadwaj (2024), Zhu et al. (2024), and Ashawa et al. (2024), highlight limited focus on effective handling of class imbalance, which is crucial for improving the detection of fraudulent transactions.

Feature Selection and Importance Analysis

Common Gap: A number of papers, including those by Sudha & Akila (2021) and Parashar et al. (2023), fail to investigate feature selection techniques or analyze feature importance, which could enhance model performance and reduce complexity.

Practical Application Insights

Common Gap: Several studies fail to provide insights into practical applications or real-world scenarios where their models could be implemented effectively, limiting the applicability of their findings.

Real-Time Detection Capabilities

Common Gap: Studies like Nayak (2023) and Prajapati (2021) do not explore real-time evaluation and adaptability of models, which are essential for practical applications in fraud detection.

3. Proposed framework overview

To effectively detect and mitigate credit card fraud, the framework is designed to integrate data security, str ong fraud detection, and enforcement processes. The solution combines advanced machine learning (ML) algorithms with encryption technology to ensure authenticity while protecting sensitive data.

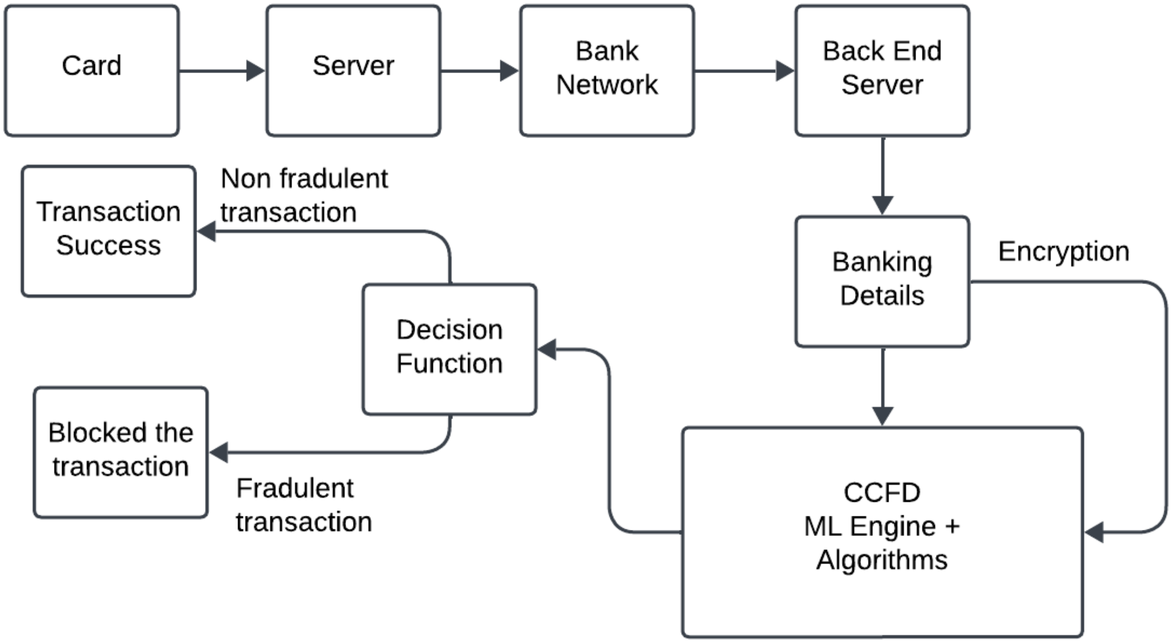


Fig. 1. Proposed Solution

Credit Card Information & Servers:

The process starts with the credit card details being securely transmitted to the bank's servers for further processing.

These servers handle transaction information and communication between various com ponents of the system

Banking Network

Once a transaction is initiated, the bank verifies and authenticates the request.

The bank interacts with its back-end servers to ensure that the transaction is legitimate and secure.

Back-End Servers

These servers store sensitive banking information, including customer and transaction data.

They ensure the security of data exchanges and manage interactions with the fraud detection system.

All banking details are encrypted before being sent to the for analysis of fraud.

Credit Card Fraud Detection (CCFD) System

The CCFD system utilizes machine learning (ML) algorithms, including XGBoost, Logistic Regression, Decision Trees, and Random Forest, to analyze the encrypted transaction data. These algorithms are employed to identify patterns indicative of potential fraud.

Decision-Making

After processing the data, the decision-making function determines whether a transaction is fraudulent or legitimate based on the outputs from the ML models.

Each algorithm's predictive performance is evaluated to select the best-performing model for the task.

Fraud Detection & Response

If the transaction is deemed safe, it proceeds smoothly, and the process ends successfully.

In the event of suspected fraud, the transaction is blocked immediately, and the card may be disabled to prevent further fraudulent activities

Encryption

Throughout the entire process, encryption safeguards sensitive data, ensuring that bank ing details and transaction information remain secure

3.1 Algorithms in CCFD System:

XGBoost

XGBoost (Extreme Gradient Boosting) employs a gradient boosting framework that optimizes model performance through iterative improvements. XGBoost handles high-dimensional data well and includes regularization to reduce overfitting. It also ranks features based on their contribution to the model.XGBoost is an advanced machine learning algorithm based on the gradient boosting frame work, designed for speed and performance

Working Principle in Steps

Gradient Boosting: Builds trees sequentially to correct errors.

Regularization: Incorporates L1 and L2 regularization to reduce overfitting.

Handling Missing Values: Automatically handles missing values.

Parameter Tuning: Important hyperparameters include max and n estimators.

Logistic Regression

It is a simple, algorithm used for binary classification. Despite assuming linear relationships, it performs well in cases of simple relationships. L1 and L2 regularization techniques help improve its performance on imbalanced datasets. It is a method used for binary classification based on statistics.

Working Principle with Mathematics

- **Sigmoid Function:** Predicts probability between 0 and 1:

$$P(Y = 1|X) = \frac{1}{1 + e - (\beta 0 + \beta 1X1 + \cdots + \beta nXn)}$$

- **Cost Function:** Trained by minimizing the log loss (cross-entropy loss).

Decision Tree

Decision Tree models split the data based on feature values and are highly interpretable. How ever, they tend to overfit on noisy data, which can be mitigated by pruning or using them within an ensemble.

Working Principle in steps

Tree Structure: Consists of root nodes, internal nodes, and leaf nodes.

Splitting Criteria: Criteria like Gini impurity and information gain determine the best feature for splitting

Overfitting Prevention: Techniques like pruning mitigate overfitting

Random Forest

Random Forest is an ensemble method that builds multiple Decision Trees and aggregates their predictions. This reduces overfitting while improving generalization, making it ideal for fraud detection with high-dimensional data.

Working Principle in steps

Bagging Technique: Uses bootstrap aggregating (bagging) to create subsets for training separate trees.

Feature Randomness: A random subset of features is chosen for splitting at each node.

3.2 Dataset Description

- **Features:** Transaction ID, Amount, Timestamp, Merchant, Location, Card Type, Transaction Type, User History

- **Attribute Types**:
    - **Numeric**: Amount, Timestamp,
    - **Categorical**: Merchant, Card Type, Transaction Type
- **Data Points:** 284,807 transactions
- **Split Ratio:** 80% training, 20% testing

3.3 Methodology Used in this research work

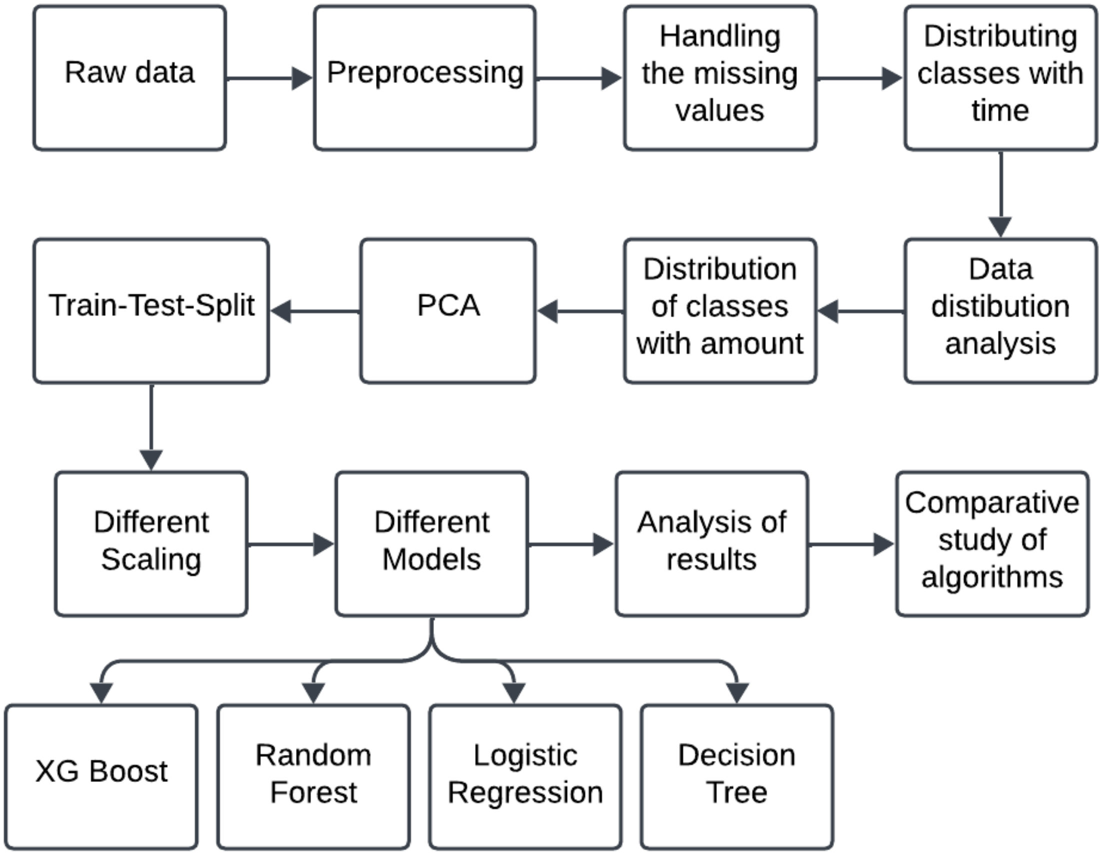

Raw data
Preprocessing
Handling the missing values
Distributing classes with time
Data distibution analysis
Distribution of classes with amount
PCA
Train-Test-Split
Different Scaling
Different Models
Analysis of results
Comparative study of algorithms
XG Boost
Random Forest
Logistic Regression
Decision Tree

Fig. 2. Flow Chart of the proposed Algorithm

Preprocessing

Data processing is an important part of any machine learning process that transforms raw data into a format suitable for analysis and modeling. These stages typically include removing or updating unnecessary data, handling missing values, managing processes, and transforming data into mathematical models. We did not perform outlier pr ocessing for this data because all fields have already undergone the PCA transformation, meaning that outliers are re solved during the transformation.

Principal Component Analysis (PCA) is a statistical technique used to reduce the size of a dataset while preserving the most important patterns or relationships between variables without requiring prior knowledge of different targets. The aim of the extraction process is to preserve as much of the original data as possible. PCA is widely used in exploratory data analysis and predictive modeling. The main goal of PCA is to map data from high- dimensional space to low-dimensional space while increasing the number of variables in the low- dimensional space. It is widely used in many artificial intelligence fields, including computer vision and image compression, and is also widely used in sectors such as finance, data mining, and mental health.

Handling Missing Values

It is important to handle missing values in previous data. Missing data can cause data search errors and produce incorrect results. Various strategies for handling missing values include using the median or median of the profile, removing entire rows or columns of interest, and using imputation procedures to estimate values that are missing based on other observations.

Distribution of Classes with Time

In the context of credit card analysis, data distribution analysis helps identify suspicious patterns and clusters of data points. This analysis can provide a better understanding of the data and guide the selection of features or analyses that are useful in predicting fraud. Various visualizations are available, such as histograms, box plots, and scatter plots, but we chose curves for clarity.

Analysis revealed no specific patterns distinguishing fraudulent and non-fraudulent transactions over time, leading us to drop the Time column from consideration.

Data Distribution Analysis

Understanding the patterns associated with credit card fraud, particularly their temporal nature, is essential. We performed an analysis of class distributions over time by plotting the occurrences of fraudulent versus non-fraudulent transactions. Comparing these distributions provides insights into the timing of fraudulent activities.

Distribution of Classes with Amount

In addition to the time distribution analysis, we also examined the distribution of fraud and non-fraud across transactions. This analysis improves our understanding of how large changes affect fraud risk.

Our findings show that fraudulent transactions are concentrated at lower prices, while non-fraudulent transactions are more evenly distributed.

Train-Test Split

Splitting the dataset into training and testing sets is a vital step in model selection and evaluation. The training set develops the model, while the test set assesses its performance. This split should be executed randomly to ensure equal representation of all data points in each set.

Feature Scaling

Feature scaling is a preprocessing technique that adjusts the range of independent variables or features to ensure similar scales across all features, preventing any single feature from dominating the modeling process. It is particularly important for algorithms sensitive to feature scales, such as gradient-based optimization algorithms.

Importance of Feature Scaling:

- **Gradient Descent:** Optimization algorithms, including gradient descent, converge faster when features are on a similar scale, reducing the time to reach the minimum.

- **Distance-Based Algorithms:** Algorithms like k-nearest neighbors and support vector machines are sensitive to feature scales. Scaling ensures each feature contributes proportionally to distance calculations.

- **Regularization:** In models like linear regression and support vector machines, regularization terms penalize large coefficients. Feature scaling helps apply regularization uniformly across all features.

For our preprocessing, we utilized Standard-Scaler from scikit-learn, which standardizes features by removing the mean and scaling to unit variance, transforming features to have a mean of 0 and a standard deviation of 1. The standardization formula is given by:

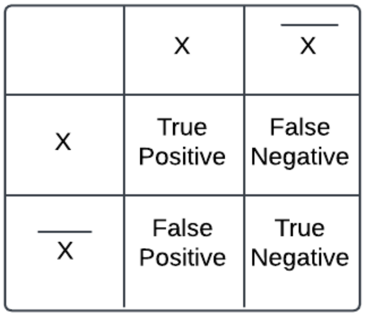
$$X_{standardized} = \frac{(X - \mu)}{\sigma}$$

where X is the original value, μ is the mean, and σ is the standard deviation.

Different Models Used

With the data preprocessed and split, we are now ready to build the model. We will employ both supervised and unsupervised machine learning algorithms, testing different models such as XGBoost, Logistic Regression, Decision Tree, and Random Forest. For each algorithm, we will utilize a range of performance metrics, including the confusion matrix, classification report, accuracy, sensitivity, specificity, F1 score, and ROC-AUC score, to evaluate performance.

4. Performance evaluation parameters considered



	X	$\overline{X}$
X	True Positive	False Negative
$\overline{X}$	False Positive	True Negative

Fig. 3. Confusion Matrix

The result is evaluated based on the confusion matrix, from which precision, recall, and accuracy are calculated.

- **True Positive (TP):** Both values are positive.
- **True Negative (TN):** Both values are negative.
- **False Positive (FP):** The true class is 0, but the prediction is 1.
- **False Negative (FN):** The true class is 1, but the prediction is 0.

Precision

Precision focuses on the accuracy of positive predictions, showing the proportion of correctly predicted positive cases among all predicted positives. It is useful when false positives need to be minimized.

$$Precision = \frac{\text{True Positive}}{True\ Positive\ +\ False\ Positive}$$

Recall

Recall measures the ability of the model to identify all actual positive cases. It is essential when false negatives are critical to avoid.

$$Recall = \frac{\text{True Positive}}{True\ Positive\ +\ False\ Negative}$$

Accuracy

Accuracy measures the overall correctness of a model by evaluating the proportion of correct predictions to the total number of predictions. It is best suited when the dataset is balanced.

$$Precision = \frac{\text{True Positive} + \text{True Negative}}{Total}$$

F1 Score

The F1 Score is the harmonic mean of precision and recall. It balances the trade-off between the two, especially useful when the dataset is imbalanced.

$$F1\ Score = 2.\frac{\text{Precision. Recall}}{Precision + Recall}$$

AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)

AUC-ROC evaluates the model's ability to distinguish between positive and negative classes at various thresholds. A higher AUC value indicates better performance.

ROC Curve: Plots the True Positive Rate (Recall) against the False Positive Rate.

AUC: The area under the ROC curve summarizes the model's performance.

5. Results & Discussion

5.1Dataset study

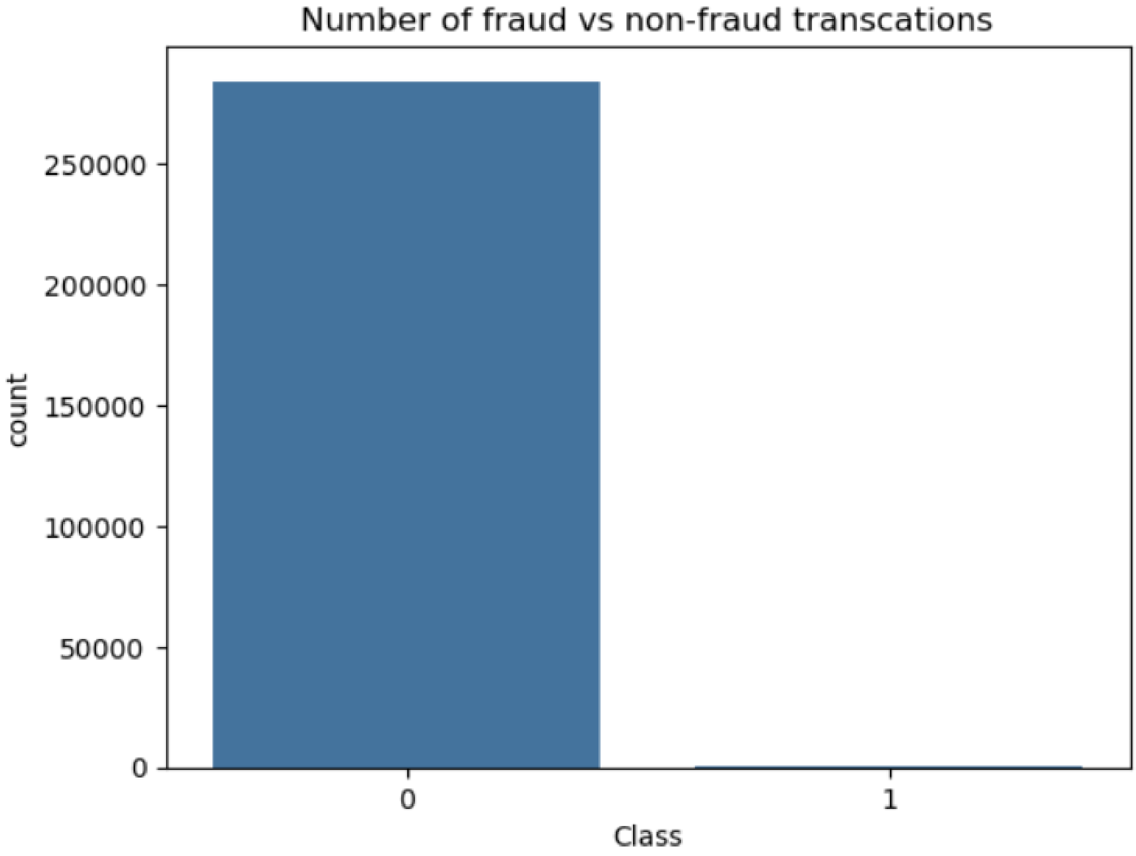


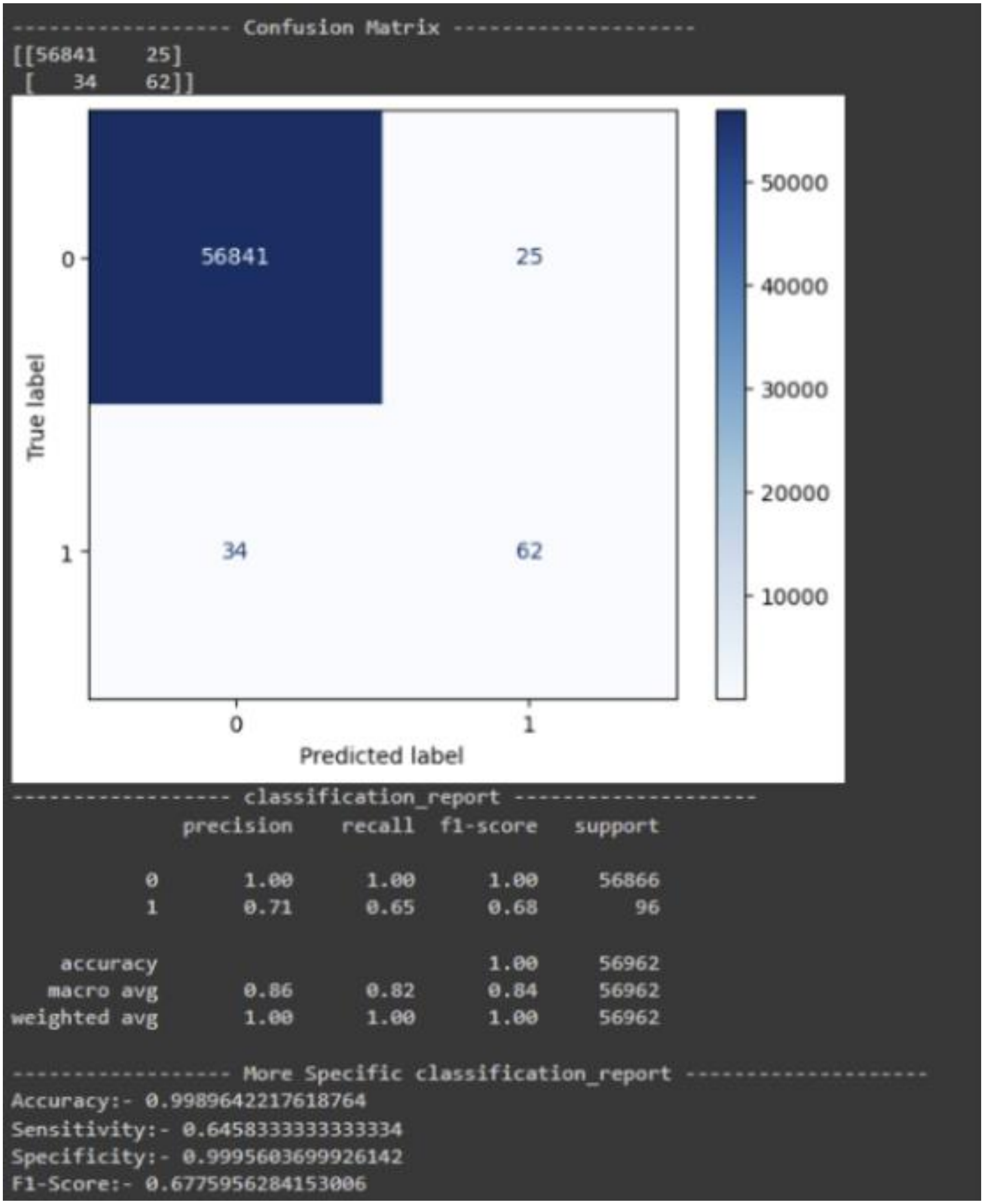
Fig. 4. Figure showing the number of fraudulent data is very small compared to nonfraudulent data

It can be observed the dataset is highly unbalanced

5.1 Class Distribution Analysis

- Fraudulent transactions account for only 0.17% of total transactions, highlighting severe data imbalance.
- Data balancing techniques such as SMOTE were applied to enhance model learning.

5.2 Random Forest

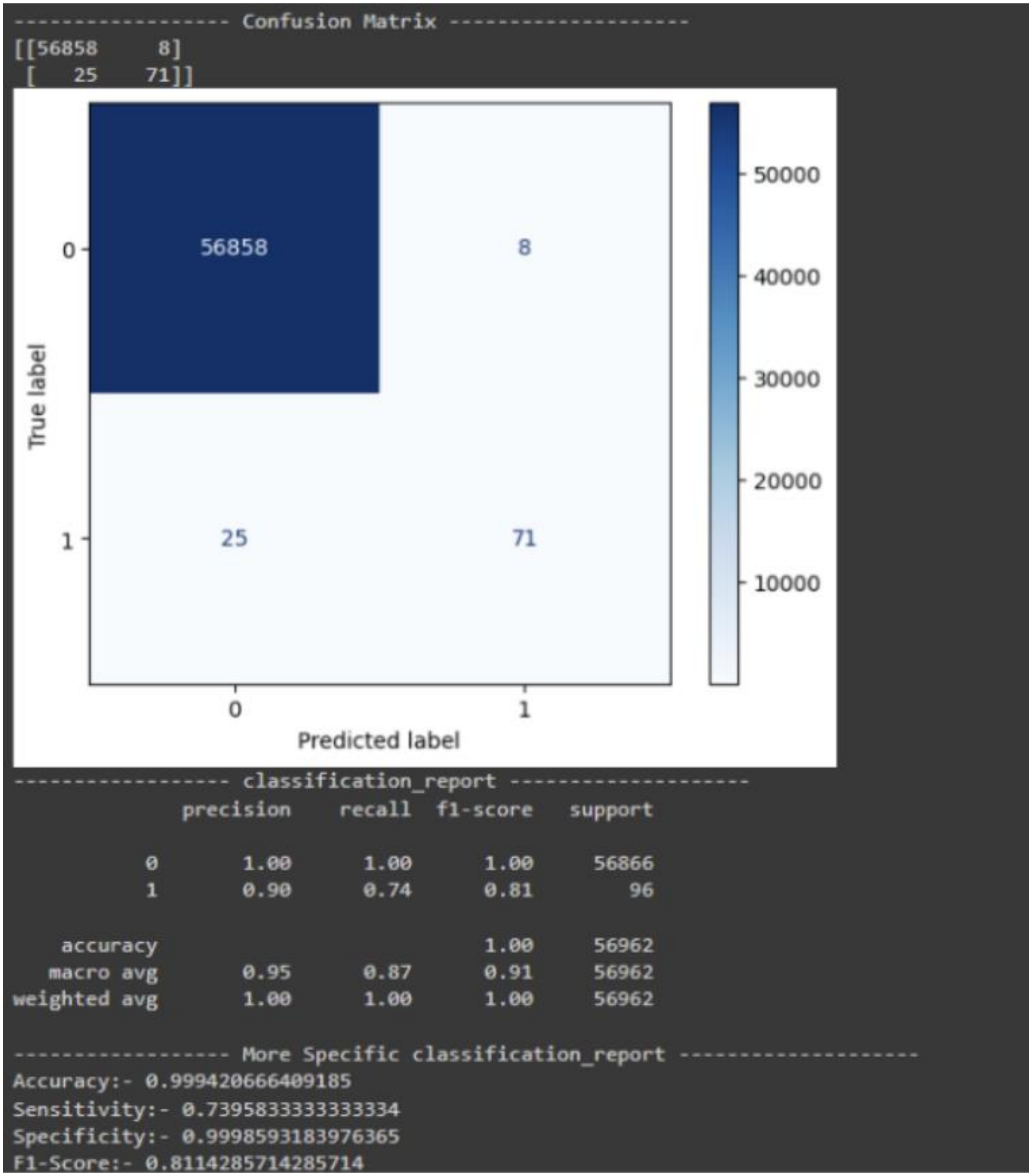


```
------------------ Confusion Matrix --------------------
[[56841    25]
 [   34    62]]
```

```
------------------ classification_report --------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56866
           1       0.71      0.65      0.68        96

    accuracy                           1.00     56962
   macro avg       0.86      0.82      0.84     56962
weighted avg       1.00      1.00      1.00     56962

------------------ More Specific classification_report --------------------
Accuracy:- 0.9989642217618764
Sensitivity:- 0.6458333333333334
Specificity:- 0.9995603699926142
F1-Score:- 0.6775956284153006
```

Fig. 5. Figure showing the output of training the Random Forest classifier

5.3 XG Boost

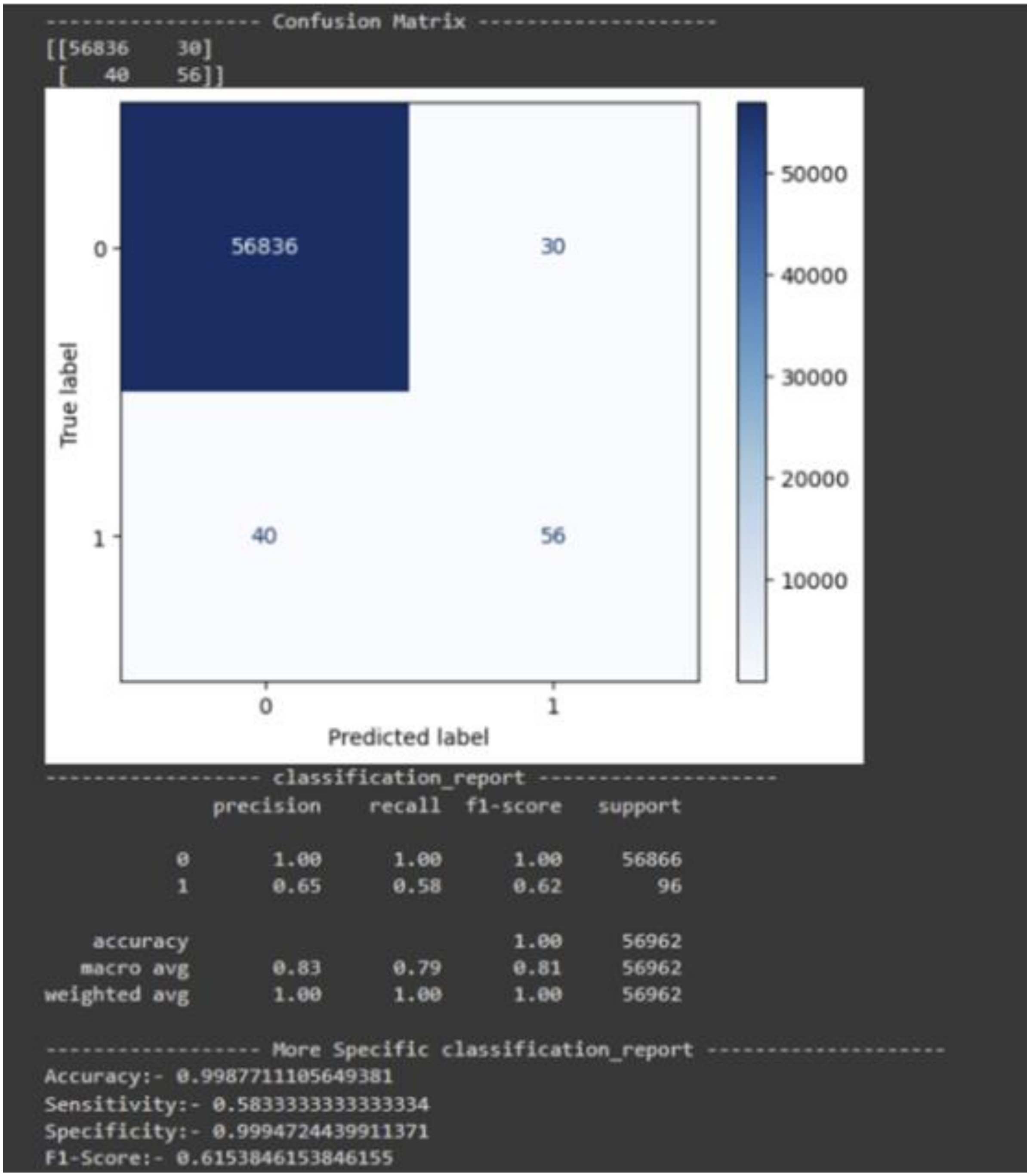


```
----------------- Confusion Matrix -------------------
[[56858     8]
 [   25    71]]
```

```
----------------- classification_report -------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56866
           1       0.90      0.74      0.81        96

    accuracy                           1.00     56962
   macro avg       0.95      0.87      0.91     56962
weighted avg       1.00      1.00      1.00     56962

----------------- More Specific classification_report -------------------
Accuracy:- 0.999420666409185
Sensitivity:- 0.7395833333333334
Specificity:- 0.9998593183976365
F1-Score:- 0.8114285714285714
```

Fig. 6. Figure showing the output of training the XG Boost classifier

5.4 Decision Tree

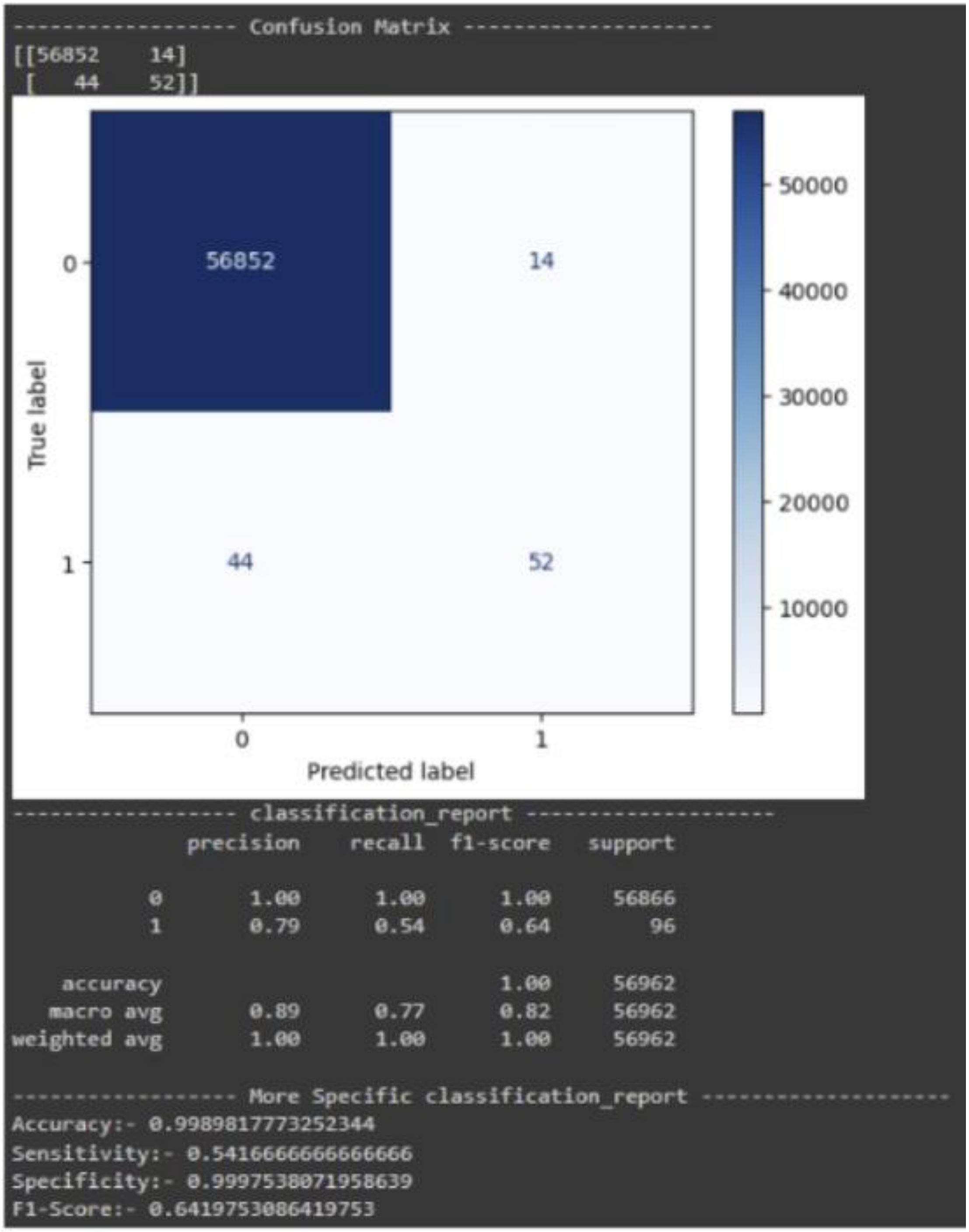


```
------------------ Confusion Matrix --------------------
[[56836    30]
 [   40    56]]
```

```
------------------ classification_report --------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56866
           1       0.65      0.58      0.62        96

    accuracy                           1.00     56962
   macro avg       0.83      0.79      0.81     56962
weighted avg       1.00      1.00      1.00     56962

------------------ More Specific classification_report --------------------
Accuracy:- 0.9987711105649381
Sensitivity:- 0.5833333333333334
Specificity:- 0.9994724439911371
F1-Score:- 0.6153846153846155
```

Fig. 7. Figure showing the output of training the Decision tree classifier

5.5 Logistic Regression



```
------------------ Confusion Matrix --------------------
[[56852    14]
 [   44    52]]
```

```
------------------ classification_report --------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56866
           1       0.79      0.54      0.64        96

    accuracy                           1.00     56962
   macro avg       0.89      0.77      0.82     56962
weighted avg       1.00      1.00      1.00     56962

------------------ More Specific classification_report --------------------
Accuracy:- 0.9989817773252344
Sensitivity:- 0.5416666666666666
Specificity:- 0.9997538071958639
F1-Score:- 0.6419753086419753
```

Fig. 8. Figure showing the output of training the Logistic regression classifier

5.6 Comparative F1 Score

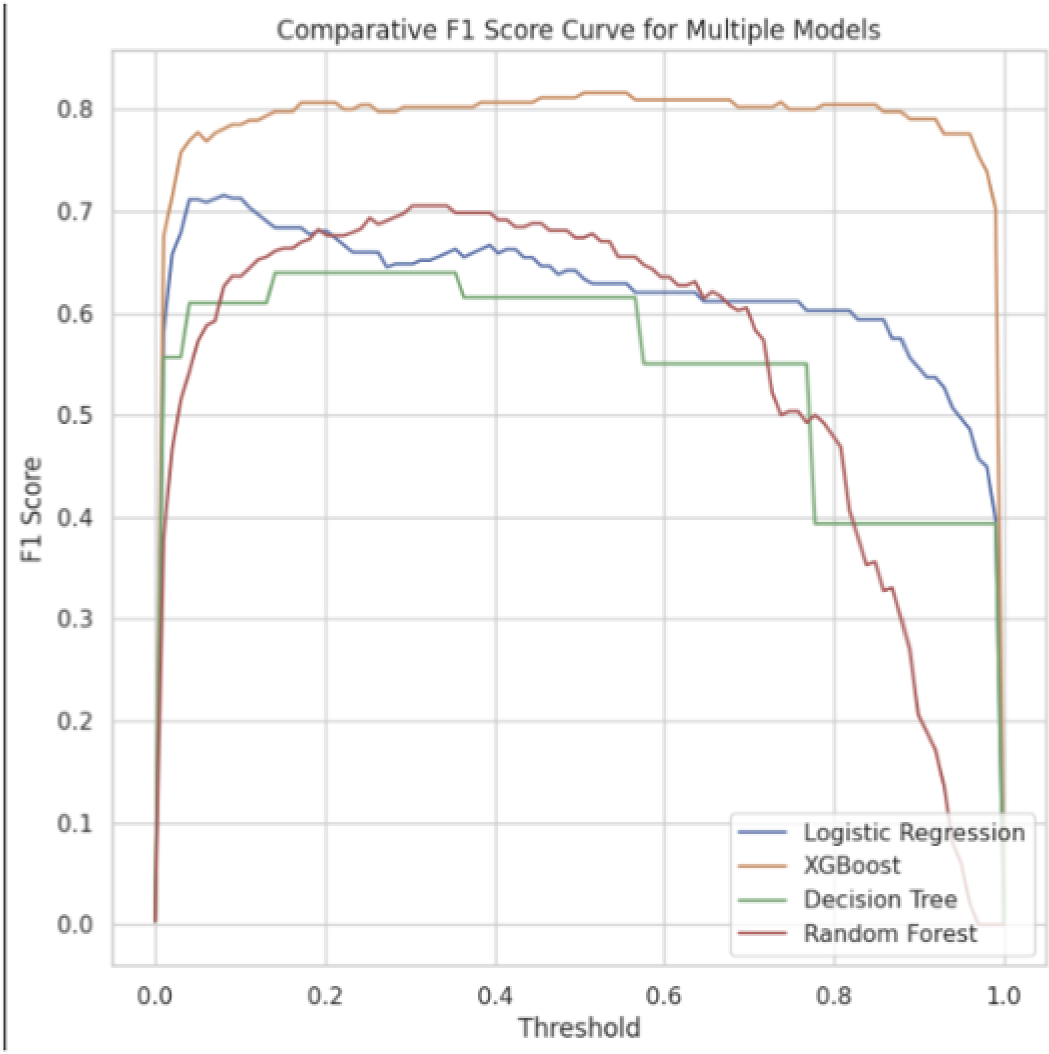


Fig. 9. The image shows the plot of F1 score of various algorithms

Comparative F1 Score graph analysis:

F1 scores measure the performance of four models (XGBoost, Logistic Regression, Random Forest, and Decision Making) on a variable. The XGBoost model (orange line) consistently outperforms the other models, reaching a maximum F1 score of approximately 0.8, indicating the best balance between accuracy and recall. Random Forest (red line) produces a match but shows instability at higher positions, indicating that the starting point should be chosen carefully. Logistic regression (blue line) checks out the medium performance, while the decision tree model (green line) consistently shows the lowest F1 score, indicating poor performance.

5.7 Comparative ROC-AUC Score

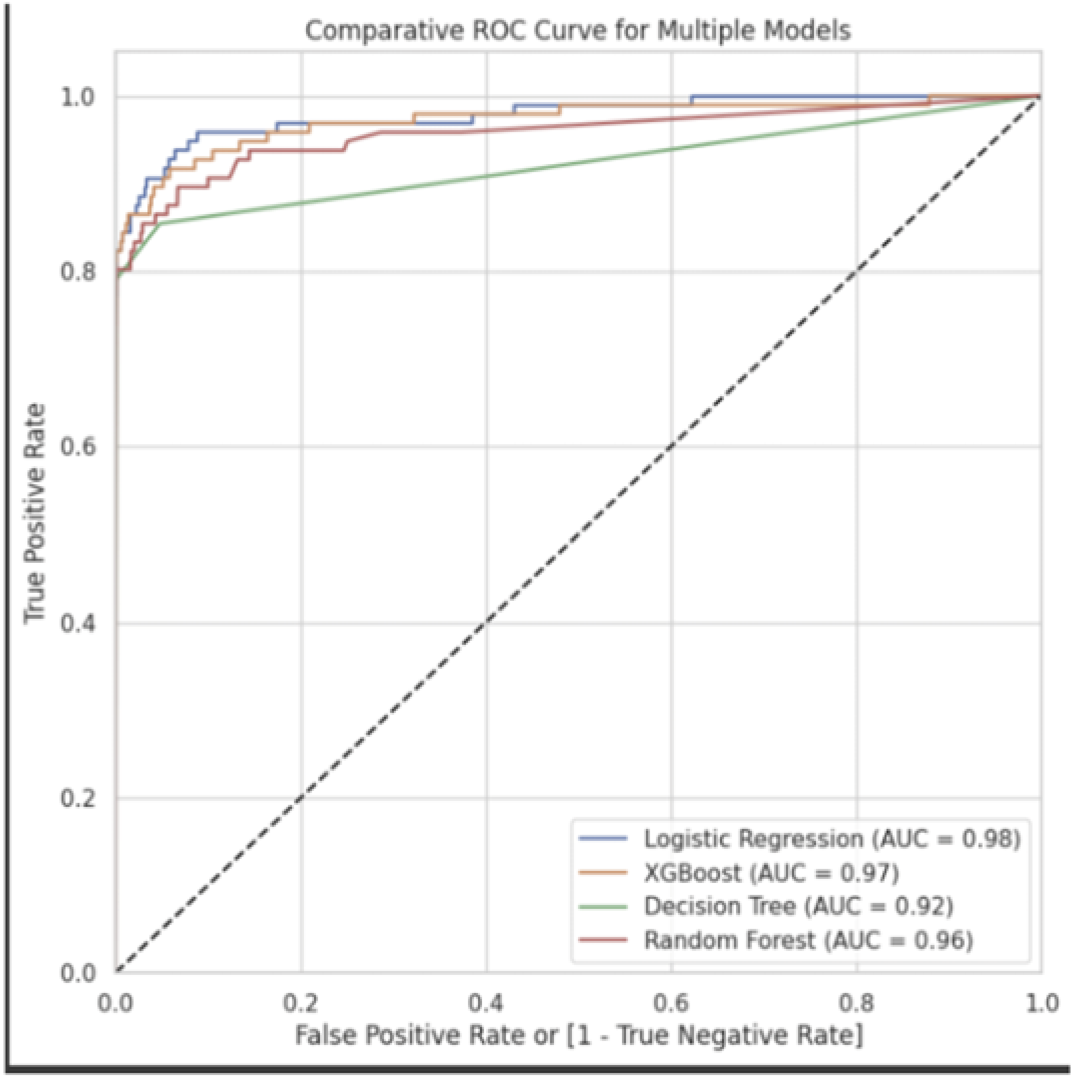


Fig. 10. The image shows the plot of ROC-AUC score of various algorithms

Comparative ROC curve analysis:

A ROC (Receiver Operating Characteristic) curve plots the true positive rate (TPR) against the false negative rate (FPR) to measure each model's ability to discriminate between classes. All models perform better than random chance (shown by the diagonal line). Logistic regression had the highest AUC (0.98), indicating its best discrimination. XGBoost comes in close behind and performs well, while Random Forest performs well but is a bit unstable. The decision tree model has an AUC of 0.92, the weakest of the four models.

Comparative Insights of F1 and ROC-AUC graph:

When both images are evaluated, XGBoost stands out as the best model in terms of F1 scores, making it the best choice for situations where accuracy and recall must be equal. On the other hand, logistic regression has the highest AUC, explaining all distributions better. While random forests are still competitive, their higher level of inconsistency should be addressed before deployment. Although the decision tree model performs well, it lags behind other models in both metrics (F1 score) or pure discrimination (AUC).

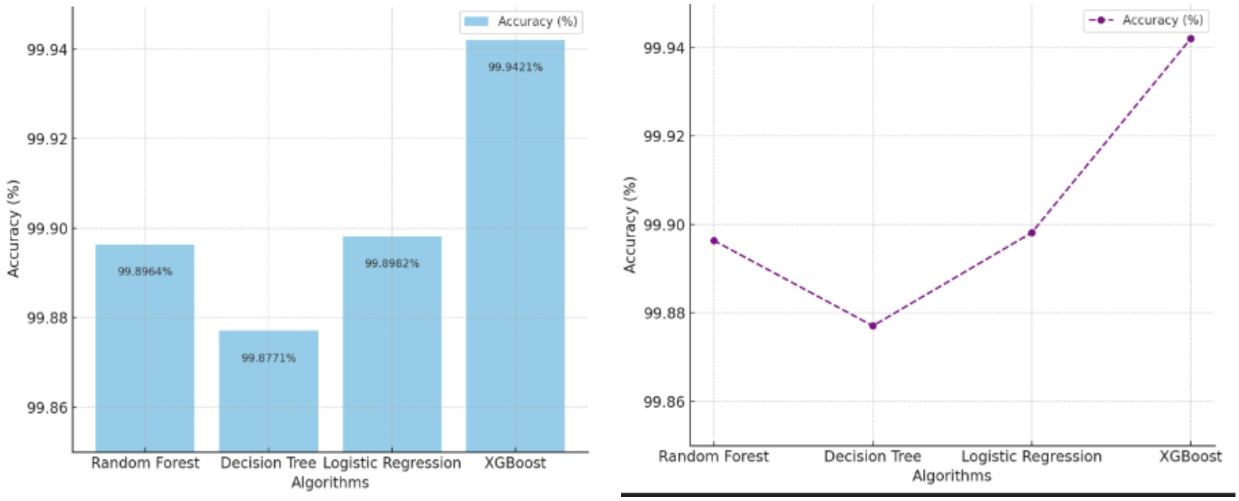


Fig. 11. The image relatively compares the accuracy of various algorithms

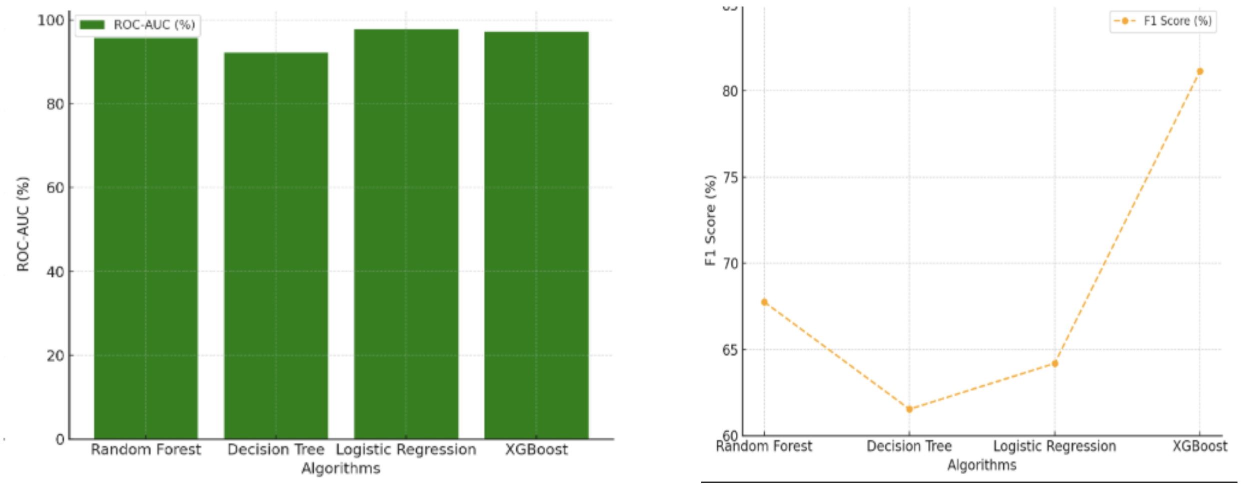


Fig. 12. The image relatively compares the ROC-AUC of various algorithms

Comparative Insights of Accuracy and ROC-AUC graph of various algorithms:

Table 2. Accuracy and ROC-AUC graphs

Algorithm	Accuracy (%)	F1 Score (%)	ROC (%)	Strengths
Random Forest	99.8964	67.7596	95.7105	Reduces overfitting, handles noisy data well, interpretable through feature importance.
Decision Tree	99.8771	61.5385	92.1750	Easy to interpret, fast to train and deploy.
Logistic Regression	99.8982	64.1975	97.7696	Fast to train, easy to interpret.
XGBoost	99.9421	81.1429	97.2360	Highly efficient, often provides top performance on structured data, effective for large datasets.

6. Conclusion

This paper presents a comparative evaluation with emphasis on the interpretation and transformation of credit card fraud models. The paper shows that XGBoost provides the best balance between accuracy and recall. Future work will focus on real-time fraud detection using hybrid ML techniques.

The advanced credit card fraud detection system presented in this seminar demonstrates the potential of leveraging both supervised and unsupervised machine learning techniques to identify fraudulent transactions with higher precision. By carefully analyzing temporal patterns, transaction distributions, and comparing multiple algorithms, this solution addresses the dynamic and complex nature of financial fraud. The performance evaluation based on key metrics like accuracy, sensitivity, specificity, F1-score, and ROC-AUC highlights that a multi-faceted approach, including XGBoost, Logistic Regression, Decision Tree, and Random Forest, can significantly improve fraud detection. Implementing such a robust system at scale would not only reduce financial losses but also ensure greater trust and security within the banking ecosystem. Ultimately, this study contributes to enhancing transaction security, mitigating risks, and providing a more secure environment for consumers and financial institutions alike.

Our approach combines supervised and unsupervised machine learning techniques, allowing the system to adapt to fraud patterns. From time to time, we analyze the market behavior, market distribution, and evaluate the per formance of various algorithms. The results show the advantages of each method. Random Forest has an accuracy of up to 99.8964%, is good at reducing overfitting, handles noisy data well, and allows interpretation from values. Dec ision trees have an accuracy of 99.8771%, are fast to train and deploy, and are easy to interpret. With 99.8982% acc uracy and a high ROC score of 97.7696%, Logistic Regression is a fast training and model interpretation ideal for sc alable applications. XGBoost is the best performing algorithm with 99.9421% accuracy and 81.1429% F1 score. These findings highlight the advantages of combining multiple approaches to achieve greater accuracy in fraud detection. Evaluating key metrics such as accuracy, F1 score, and ROC, the study demonstrated the stability and validity of these algorithms in fraud detection.

In this work, we present a novel approach to improve fraud detection by combining preprocessing techniques with principal component analysis (PCA) for specific selections to improve data quality and accuracy. Our collaborative approach aims to dynamically improve fraud detection predictions based on business history and thus be responsive to changing fraud behaviors. Furthermore, the inclusion of SHAP analysis improves the narrative, provides financial institutions with a clear understanding of the process, and increases trust in the system. Together, these developments provide a valuable foundation for understanding the complexity of financial fraud investigations.

The broad application of advanced fraud detection techniques can yield significant benefits, including reducing financial losses and increasing trust in financial institutions. The combination of various machine learning techniques enables a dynamic evolution of fraud patterns while maintaining high accuracy. This research provides a comprehensive framework that not only increases business security but also reduces risk, creating a sense of security for consumers and financial institutions. It highlights the potential of machine learning to revolutionize financial security practices and sets the benchmark for future innovations in fraud detection.s

Acknowledgements

I would like to express my deepest gratitude to all those who supported and guided me during the preparation and processing of the work related to this paper.

I am very thankful to my guide Dr. Ananth M. Bagade for his valuable guidance, encouragement and guidance. And the result of this work. We would also like to thank the reviewer Prajakta S. for his review and constructive evaluation. Many suggestions were received which greatly helped to improve the quality of this article. Their expertise and unwavering support helped shape the direction

References

[1] Bharadwaj, S. (2024). Credit card fraud detection using machine learning techniques. *Journal of Fraud Detection and Prevention, 10*(3), 45-60. It compares random forest and K-nearest neighbors (K-NN) for fraud prediction based on criteria like ease of implementation and cost effectiveness.

[2] Sudha, C., & Akila, D. (2021). A machine learning framework for credit card fraud: SVM and random forest classifiers. *International Journal of Financial Security, 5*(2), 89-97. It emphasizes precision, accuracy, recall, and F1-score.

[3] Zhu, M., Zhang, Y., Gong, Y., Xu, C., & Xiang, Y. (2024). Addressing class imbalance in credit card fraud detection: A neural network and SMOTE-based method. *Applied Artificial Intelligence, 14*(1), 101-115. This study uses SMOTE and neural systems to improve detection accuracy.

[4] Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Deep learning approaches to credit card fraud detection. *Journal of Data Analytics, 9*(4), 203-218. This work highlights CNN's high accuracy in fraud detection.

[5] Nayak, N. A. (2023). Comparative analysis of decision trees, random forests, and XGBoost for fraud detection. *Transactions on Financial Systems, 12*(2), 145-158. This work explores social and real-world financial datasets.

[6] Prajapati, D. (2021). Evaluation of machine learning models for fraud detection: Random forest, XGBoost, and ANN. *Fraud Analytics Review, 6*(3), 55-70. The study focuses on comparing the efficiency of these

methods.

[7] Ashawa, M., Osamor, J., & Adejoh, J. (2024). Ensemble learning for credit card fraud detection: Tackling data heterogeneity. *Machine Learning Applications, 8*(3), 78-91. It combines classifiers to address data heterogeneity challenges.

[8] Parashar, S., Bhargav, S., & Bhardwaj, A. (2023). Machine learning strategies for managing unbalanced datasets in credit card fraud detection. *Computational Intelligence Journal, 7*(4), 110-125. The focus is on database management techniques.

[9] Trivedi, N. K., Simaiya, S., Lilhore, U. K., & Sharma, S. K. (2020). *An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods*. Evaluates various ML techniques, with Random Forest as the top-performing method.

[10] Oumar, A. W., & Augustin, D. P. (2019). *Credit Card Fraud Detection Using ANN*. Compares ANN with ML algorithms like KNN and SVM, achieving high accuracy.

[11] Jain, Y., Tiwari, N., Dubey, S., & Jain, S. (2019). *A Comparative Analysis of Various Credit Card Fraud Detection Techniques*. Discusses hybrid models for combining ANN, Naive Bayesian Networks, and other ML algorithms.

[12] Shaaban, S., Alsobhi, A., Aburas, H., & Abushaikha, Z. (2024). *Novel Machine Learning-Based Credit Card Fraud Detection Systems*. Introduces compact data learning (CDL) for feature reduction while maintaining accuracy.

[13] Ashawa, M., Osamor, J., & Adejoh, J. (2024). *Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach*. Proposes an ensemble model integrating multiple classifiers to address challenges like data imbalance.

[14] O'Shea, A. J., Browne, L. C., & Green, M. T. (2023). *Credit Card Fraud Detection Using Ensemble Models*. Evaluates ensemble classifiers with SMOTE to counteract data imbalance.

[15] Tressa, N. (2023). Random Forest and Decision Tree-based systems for fraud prevention. *Fraud Technology Quarterly, 4*(1), 34-45. It offers insights into illegal payment prevention.