International Journal of Electrical, Electronics and Computer Engineering **14**(1&2): 108-115(2025)

ISSN No. (Online): 2277-2626

An Ensemble Learning Framework for Credit Card Fraud Detection with Hyperparameter Optimization

Puneet Misra* Mohd Usman and Vishrut
Department of Computer Science
University of Lucknow, Lucknow, 226007 (Uttar Pradesh), India.

(Corresponding author: Puneet Misra)*

(Received: 16 July 2025, Accepted: 01 December 2025)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Within the financial technology sector, identifying fraudulent online transactions remains a significant challenge, primarily due to extreme class imbalance in transactional datasets. Such imbalances often degrade the predictive performance of standard machine learning models, posing a substantial risk in high-stakes financial environments. This study presents a comparative analysis of various algorithmic approaches to enhance detection reliability. To address data skewness, the Synthetic Minority Oversampling Technique (SMOTE) was applied to balance the representation of fraudulent versus legitimate instances. The core contribution of this research is the development of AXRVF, an ensemble framework that integrates the predictive strengths of Random Forest (RF), Extreme Gradient Boosting (XGB), and Artificial Neural Networks (ANN) through a soft-voting classifier. To further refine performance, hyperparameter optimization was executed across all base models. The results demonstrate that the synergy of data balancing, ensemble learning, and rigorous model tuning produces a highly robust and accurate fraud detection system suitable for real-world applications.

Keywords: Machine Learning, Ensembling Techniques, Voting Classifier, Hyperparameter Tuning, Deep Learning, SMOTE, Credit-Card Fraud.

INTRODUCTION

Internet has transformed our lives across multiple dimensions. From sectors like healthcare and education to finance, it has brought about considerable improvements. With the increase in availability of the internet, the need for online payments is on the rise. Credit card frauds are one of the most frequent and happening financial frauds. With the involvement of money, the existence of these frauds instills fear in the public. As seen in the year 2018, the global finance sector saw a loss of about 42 billion dollars because of these financial crimes (Akazue *et al.*, 2023). Moreover, in a more recent report, countries like United Kingdom faced a loss of over half a billion euros in 2020 itself which later increased to 1.2 billion as recorded by United Finance Annual Fraud Report in the 2022.

Meanwhile, according to federal trade commission (FTC), the year 2021 was the most challenging year for identity theft, showing the sensitivity of the problem. Another instance was reported by FTC, where the consumer sentinel network recorded a fraud report of 2.4 million in the year 2022 (CSN-Data-Book, 2023). With the rise in the development of Artificial Intelligence, studies have been conducted to counter these crimes. Various approaches like Statistical methods, ML models, and DL frameworks are seen to be effective methods in detecting credit card fraud.

A key challenge in card fraud detection is the imbalance nature of the dataset, due to the exceptionally high number of legitimate credit card transactions. To resolve this, data balancing techniques are used for a more reliable model.

The problem space is defined by a blend of technical, ethical, and operational limitations that push traditional machine-learning approaches to their limits. As seen in case of daily transactions that demand scalable pipelines and lightweight models for real-time inference. Additionally, the system must identify and react anomalies within milliseconds, requiring low-latency architectures. Furthermore, the cost sensitive nature of transactions makes them of high priority, False positives frustrate legitimate users and erode trust, while false negatives lead to direct financial loss, making it a risk sensitive classification task.

PROBLEM STATEMENT AND RESEARCH GAP

Various computational ML techniques can identify malicious activity; methods like Decision Trees (DT), RF and LR are much more prevalent, for involving such classification.

While LR is preferred for binary classification, decision trees are better suited for pattern recognition.

Furthermore, getting data for the fintech industry is also another challenge, because of its confidential nature. Additionally, the involvement of monetary capital, makes it rather difficult to select necessary features for improved accuracy and speed. This is because providing validation is necessary for the decisions.

Most studies focus on applying individual ML models without looking at their effectiveness after using ensembling techniques. Moreover, voting classifier ensembling is one of the few under researched methods, which unlike other techniques does not dilute the learning of models, rather is simply aggregates the predictions from pretrained models.

Furthermore, DL models are implemented as much with these data balancing and ensembling techniques making them a considerably newer techniques to deal with the everchanging threat landscape. To detect credit card frauds more efficiently, the current study combined different algorithms along with ANN to make best use of voting classifier technique.

LITERATURE REVIEW

Aslam & Hussain (2024), assessed the performance of popularly known ML models, measuring them with metrics: accuracy, recall, and F1 score. They also showed potential of Logistic Regression (LR), Gradient Boosting Machine, and RF with high accuracy and precision. It also showcased importance of computation time in training the model in real life scenarios. Additionally, Syeda (2024), showed the effectiveness of data preprocessing and cross-validation techniques in models: LR, K nearest neighbor (KNN), RF, and ANN.

Wijaya *et al.* (2024) provided a comparative analysis and emphasized the importance of data balancing techniques like random oversampling and under sampling with algorithms like RF, LR, XGB and Decision Tree (DT) in getting accurate, robust and unbiased results.

Tanouz *et al.* (2021) studied the efficacy of different models like DT, LR, RF, Naive Bayes (NB) with a focus on imbalanced dataset. The results showed the effectiveness of RF in fraud detection, also underscoring the need for feature selection.

Sadgali *et al.* (2019) identified the most effective methods for detecting insurance fraud using techniques like Support Vector Machine (SVM), Multilayer Feed Forward Neural Network (MLFF) and NB using metric of accuracy rate for investigation.

Karthik *et al.* (2022) introduced a novel model using concept of ensembling techniques like bagging and boosting. Building on this, Khalid *et al.* (2024) proposed a novel model that integrated KNN, SVM, Bagging, RF, and Boosting classifiers while also implementing SMOTE and under-sampling techniques. This resulted in the ensembled model outperforming other models across metrics of accuracy, recall and F1 – score.

Hashemi *et al.* (2022), presented a comprehensive study utilizing class weight-tuning and hyperparameter optimization to address class imbalance and enhance performance of models while also integrating ensemble learning techniques. Furthermore, Asuai *et al.* (2024) combined ANN with XGB and GB to give a framework optimal for credit-card fraud detection and explored solutions of imbalanced dataset by utilizing oversampling methods while maintain the computational speed for a real-world approach. In accordance to this many researchers have proved the significance of information sharing and collaboration across financial institutions for strengthening fraud detection systems and mitigating risks more effectively. Furthermore, they also highlighted a clear, structured summary of key components in transaction systems, while systematically identifying the research gaps and importance for each component.

METHODOLOGY

Dataset Collection. The confidential nature of the domain makes it difficult to source real and reliable data in the fintech industry. Banks and other financial institutions do not and should not share Personally Identifiable Information (PII) of their customers. It is self-evident that these institutions would build machine-learning models in-house with their data without compromising the trust of their client base. As such, there are only a few limited options when it comes to the data-sources available at our disposal. The credit-card fraud detection dataset by “Machine-Learning Group - ULB" contains anonymized credit-card transaction data labelled as fraudulent or genuine.

The dataset used in this study is obtained from Kaggle, including 492 fraudulent transactions out of 284,807 total transactions over a period of 48 hours. The dataset is extremely unbalanced resulting in frauds only amounting to 0.172% of the total observations.

Data Analysis. Data preprocessing is a critical phase in any machine learning pipeline, serving as the foundation upon which model performance is built. In the context of credit card fraud detection, preprocessing transforms raw, often messy transaction data into a clean, structured format optimized for machine learning algorithms. The quality of preprocessing directly impacts model accuracy and training efficiency. This section details the comprehensive preprocessing pipeline applied to the credit card transaction dataset.

Inconsistencies or missing values are checked, and no such issue is seen. The percentage of the number of fraudulent transactions in the dataset is calculated to be only 0.172% showing the high disparity in the dataset as perceived in Fig. 1. To address this issue, SMOTE, a data balancing technique, is used.

SMOTE is an oversampling technique used to balance data by synthesizing the minority class to match the majority. Unlike, under sampling, no data is dropped. It helps in creating a balanced dataset, reduced the effect of overfitting, unlike random oversampling which duplicates the data without seeing the pattern.

The basic algorithm is as follows:

i. Randomly select a minority class sample.

ii. Find the K-nearest neighbor.

iii. Select another sample and generate a synthetic example by interpolating between the two values.

iv. Repeat the process until the desired balance is achieved

The synthetic sample is created using the following formula:

$$\left[X_{\text{new}} = x + \lambda \cdot \left(X_{\text{neighbour}} - x\right)\right]$$

Where, (λ) is a random number between 0 and 1, (x)\ is the original sample, and $(X_{\text{neighbour}})$ is the selected neighbor.

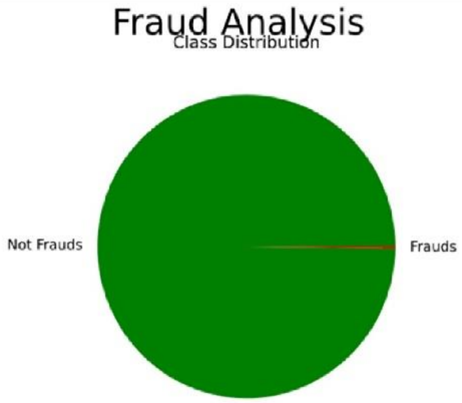

Fig. 1. Class Imbalance Visualization (Frauds vs non-frauds).

Exploratory Data Analysis (EDA). EDA is a necessary process to understand and validate a dataset using popular statistical methods and visualization tools before training the ML models.

EDA helps in detecting outliers and anomalies, while also understanding feature distributions. Moreover, missing values and noise are identified for proper mitigation.

Fig. 2 is a time-frame visualization of fraudulent activities over the 48-hour period. The feature 'Hour' is calculated using feature 'Time' because of the irrelevancy of the timestamp and group the transactions into hours for further analysis.

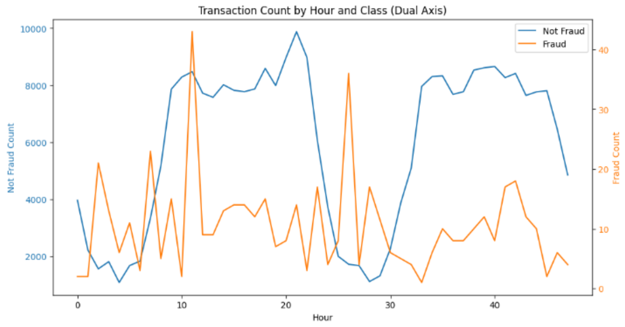

Fig. 2. Time Series Visualization (transactions over two-day period).

Regular transactions follow a natural periodic day-night cycle where the legitimate transactions go down significantly during night-time and show the most activity during the daytime. However, illegitimate transactions exhibit a pattern almost independent of the time of day.

To assess the relationships between features (V1 through V28), a correlation matrix (Fig. 3) is used, aiding in the identification of potential candidates for feature selection. Most of the parameters in the data are independent. There is some correlation between "Amount" and the features V7 and V20 and negative correlation with V2 and V5. Most other features don't show any significant correlation.

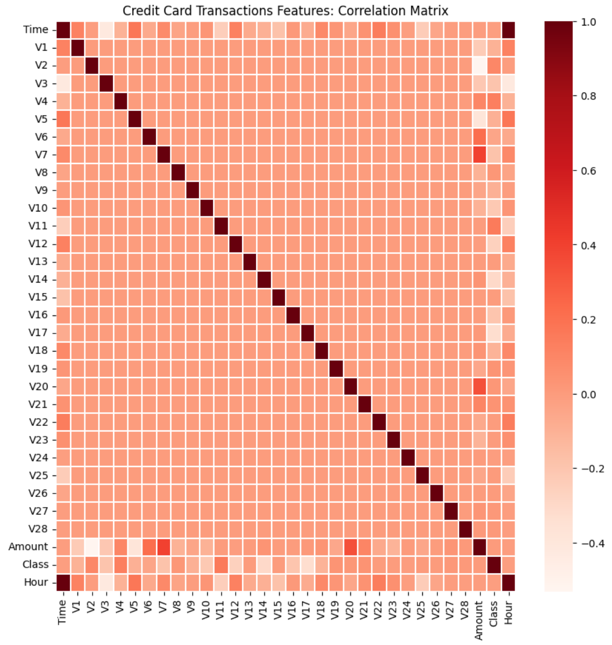

Fig. 3. Correlation Matrix of dataset features.

Feature Selection. Feature selection is the process of eliminating noise and overfitting risk, while also identifying the most informative variables. This results in faster models with better performance.

In this study, F1 score vs feature count peaked at 30 features implying that all features are contributing to improving the model performance. The real benefit of dropping features would be reducing the computational time (Mniai *et al.*, 2023). Fig. 4 shows the feature selection and performance curve giving the optimal feature count. It can be observed that f1-score increases sharply from 1 to 5 features, becoming moderate at 10 before fluctuating again and reaching the peak at around 30 features.

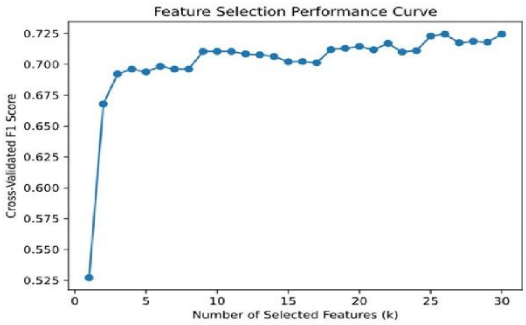

Fig. 4. Correlation Matrix of dataset features.

Metric Selection. Metric Selection is the process of choosing the evaluating measures to identify the best model on the basis of their performance. In high-risk domain like fraud detection, selecting wrong metric can result in adoption of a unreliable model.

Various metrics are considered to obtain a more unbiased and accurate result for comparison of models in this study. These evaluation metrics in the order of importance include:

(i) Recall: Recall (or Sensitivity) measure the ability of the model to correctly find the actual fraudulent cases.

$$Recall = \frac{TP}{TP + FN} \qquad (1)$$

(ii) Precision: Is a measure of quality. For all the times a model predicts fraud, how many did it get right.

$$Precison = \frac{TP}{TP + FP} \qquad (2)$$

(iii) F1-score: A harmonic Mean of Precision and Recall. It combines both metrics in a single number.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (3)$$

(iv) Area Under the Receiver-Operating Characteristic (ROC) curve: Describes the ability of the model to distinguish between the two classes, frauds and non-frauds

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \qquad (4)$$

where,

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}$$

(v) Accuracy: The most straightforward metric, it measures the correct predictions over the total number of predictions. The accuracy is not a useful metric in this case, as due to the high imbalance, it is quite easy for the model to guess non-fraud and be correct 99% of the time. The 99% accuracy is misleading as the ability to detect frauds (minority case) is of more importance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (5)$$

Where, TN is True Negative, TP is True Positive, FN is False Negative and FP is False Positive. Hence, the aforementioned metrics are used for evaluating the methods

Hyperparameter Tuning. The process of hyperparameter tuning is used to attune the model's configuration to provide optimal results. The parameters include the number of training epochs, the number of estimators used in tree algorithms, and the number of hidden layers in a deep layer network. By making use of techniques like RandomSearchCV and GridSearchCV, the model's hyperparameters were recalibrated to improve their performance characteristics even further than their base counterparts.

(i) GridSearchCV: In this, every possible combination of hyperparameters is defined to find the best combination within the grid.

(ii) RandomSearchCV: Random combinations are defined for faster and more efficient result. The result is not necessarily the best combination.

Model Algorithm Development

The performance of many ML models is taken into consideration to obtains a better result from an imbalanced dataset, including RF, DT, LR, and Gradient Boosting (GB) and DL concepts like ANN.

(i) Logistic Regression. Logistic Regression is a fundamental statistical model used for binary classification problems. Despite its name, it is a classification algorithm rather than a regression technique. It models the probability that a given transaction belongs to the fraudulent class using logistic (sigmoid) function, which maps any real-valued input between 0 and 1. Not chosen because it:

(a) Assumes linear relationship between features and log-odds

(b) May underperform with complex, non-linear patters

(c) Sensitive to feature scaling and outliers

Formulae:

$$P(y = \frac{1}{x}) = \frac{1}{1 + e^{-z}} \qquad (6)$$

where z,

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \qquad (7)$$

(ii) Decision Tree. A Decision Tree is a tree-structured classifier that makes decisions by learning simple decision rules from training data. The algorithm recursively splits the dataset based on feature values that best separate fraudulent from legitimate transactions, creating a hierarchy of if-then-else rules. Not chosen because:

(a) Prone to overfitting, especially with deep trees

(b) High variance. Small data changes can create very different trees.

(c) Biased towards features with many categories

(d) Can create overly complex trees that don't generalize well.

(iii) K Nearest Neighbor. KNN presents significant limitations when applied to fraud detection systems, particularly in production environments handling high-volume transaction data. While the algorithm's simplicity is appealing, its practical disadvantages make it less suitable compared to more robust ml approaches. KNN suffers from a high computational cost, as it requires calculating distances to all training examples for each prediction. This becomes prohibitively slow with large fraud datasets containing millions of transactions, making real-time fraud detection challenging and expensive to deploy at scale. Beyond computation time, KNN is also memory intensive because the model is essentially the data itself—the entire training dataset must be stored in memory. This is impractical for enterprise scale fraud systems with billions of historical transactions, leading to significant infrastructure costs

Distance metric (Euclidean):

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2} \qquad (8)$$

The top performing models were evaluated and considered for ensembling technique (voting classifier). The following algorithms with the given weights were used:

(iv) Random Forest (weight: 0.411). An ensemble of DTs, created to reduce overfitting while improving accuracy and stability. It excels in handling non-linear interactions and relationships in the data. It was selected due to its ability to improve the stability of the model and make precise predictions.

Random Forest excels at handling the imbalanced nature of fraud datasets and the complex interactions between transaction features. Its feature importance metrics help identify which factors (transaction amount, time, merchant category) most reliable indicate fraud. Random forest significantly reduces overfitting compared to single decision trees. Furthermore, they perform well with high-dimensional data while also handling missing values effectively.

Ensemble prediction (classification):

$$\hat{y} = \arg\max_{c} \sum_{t=1}^{T} \mathbb{I}\left(h_t(x) = c\right) \tag{9}$$

(v) XGBoost (weight: 0.331). The extreme gradient boosting model is designed for high accuracy and fast computation. It improves previous mistakes iteratively and handles structured data exceptionally well. It was selected due to its well-rounded performance in providing good results.

XGB consistently achieves superior performance in credit-card fraud detection and real-world applications. Its ability to learn complex patterns, handles class imbalance, and provide nuanced probability scores makes it particularly effective for distinguishing subtle differences between legitimate and fraudulent transactions. Moreover, it performs well with tabulated data.

(vi) Artificial Neural Network (weight: 0.257). ANNs: make use of the concepts of DL to provide a more flexible, personalized approach of building a model. A normal ANN model is made of various layers containing a certain number of neurons which work on data. ANNs are largely classified into:

Feed Forward Neural Network: FNNs transfer data in a unidirectional manner, from input to output layer while going through the given hidden layers. This is used in this study because of its advantages in pattern recognition and classification.

Feed Back Neural Networks: Unlike FNNs, direction of information is bidirectional, i.e., after initial computations, the error is calculated along with the contribution of different weights. It accordingly adjusts the weights for better results.

A high-level overview of ANN architecture components:

a) Input Layer: Receives transaction features (amount, location, time, merchant type)

b) Hidden Layers or Deep Layers: One or more layers where neurons apply activation functions to weighted sums of inputs, extracting progressively abstract representations.

c) Output Layer: Produces the final classification (fraud probability).

The learning process during training, the network makes predictions, calculates the error using a loss function (e.g. “binary cross-entropy"), and adjusts weights backwards through the network using gradient descent to minimize this error.

Furthermore, ensemble approaches are utilized because of the improvement they provide after combining the strengths of different models, hence reducing errors, noise, and bias. Additionally, voting classifiers work on the concept of weights to provide a more flexible framework.

(vii) Ensemble Techniques. Ensemble learning is a powerful ml paradigm based on the principle that combining multiple models often produces better predictive performance than any one single model alone.

Different models make different types of errors. By combining them strategically, we can cancel out individual weaknesses while amplifying collective strengths. A model that excels at catching one type of fraud pattern might miss another, but an ensemble can capture both.

Types of Ensemble Methods:

a) Bagging (bootstrap aggregation): Trains multiple instances of the same algorithm on different random subsets of data. Example: RF (ensemble of decision trees)

b) Boosting: Trains models sequentially, with each model correcting errors of the previous ones. Example: XGB, AdaBoost, GB.

c) Stacking: Trains diverse models and uses another model (meta-learner) to combine their predictions

d) Voting: Combines predictions from multiple different algorithms.

e) Voting Classifier

The Voting Classifier is an ensemble meta-algorithm that combines conceptually different ML models and uses a majority vote (hard voting) or average predicted probabilities (soft voting) to make final predictions. In the context of this credit-card fraud detection project, it integrates the strengths of Random Forest, XGB and ANNs.

This study uses a soft-voting strategy, in which the average weight of each class is calculated and then the class with the maximum average probability wins. It is preferable in the case of unbalanced datasets as soft voting can help resolve the partiality towards the majority class.

Model Evaluation. The main criterion for assessing the models is based on their ability to detect fraudulent transactions. The performance characteristics are evaluated and compared across various ML techniques and algorithms like DT, XGB, RF, LR, etc. Metrics like Precision, Recall, F1-Score, and Area Under Curve-Receiver Operating-Characteristics (AUC-ROC) are used to evaluate model performance characteristics.

RESULTS

The outcome of AXRVF, as depicted in Table 1, clearly indicates that suggested model outperforms the previously mentioned popular models. This is possible after hyperparameter tuning resulting in best f1 score and recall of 0.8621 and 0.8696 respectively along with a consistent score in precision and auc-roc of 0.8547 and 0.9860. Additionally, the model reduced least number of false negatives and false positives to 17 and 15 which were less than the values obtained by Clive Asuai *et al.* (2024).

Table 1: Statistical analysis of the models.

Model	Recall	Precision	F1-Score	AUC-ROC
LR	0.5918	0.8405	0.6946	0.7958
DT	0.7346	0.7578	0.7461	0.8671
RF	0.6836	0.9178	0.7836	0.8417
XGB	0.7040	0.7931	0.7459	0.8518
kNN	0.6938	0.8947	0.7816	0.8468
LR (SMOTE)	0.8533	0.0573	0.1076	0.9213
DT (SMOTE)	0.7448	0.2491	0.3734	0.8705
RF (SMOTE)	0.7244	0.8452	0.7802	0.8621
XGB (SMOTE)	0.8061	0.7596	0.7821	0.9028
kNN (SMOTE)	0.8061	0.4730	0.5962	0.9022
ANN	0.8088	0.8661	0.8364	0.8936
RF (Hypertuned)	0.8197	0.8695	0.8439	0.9841
XGB (Hypertuned)	0.7803	0.8957	0.8340	0.9927
AXRVF	0.8547	0.8696	0.8621	0.9860

To provide a better understanding of the performance, the graphs comparing the precision, recall, f1-score and AUC-ROC of models are provided in Fig. 5-8 respectively. The result clearly depicts that AXRVF shows best f1-scores and recall in class by combining strengths of the previously stated models, while also showing a good performance under precision and auc-roc metrics.

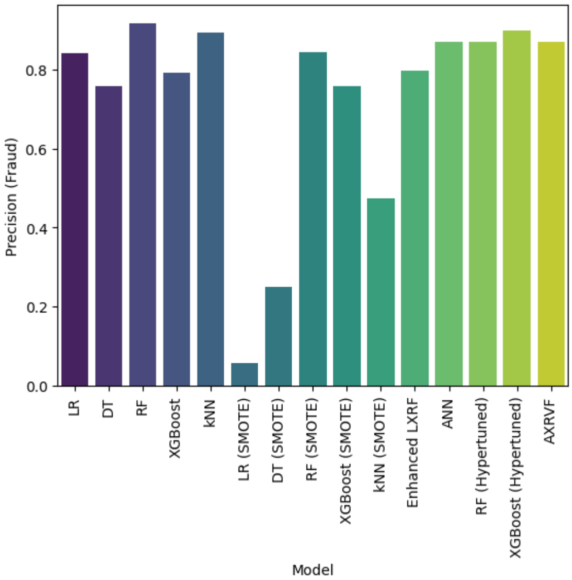

Fig. 5. Comparison of Precisionwith different models.

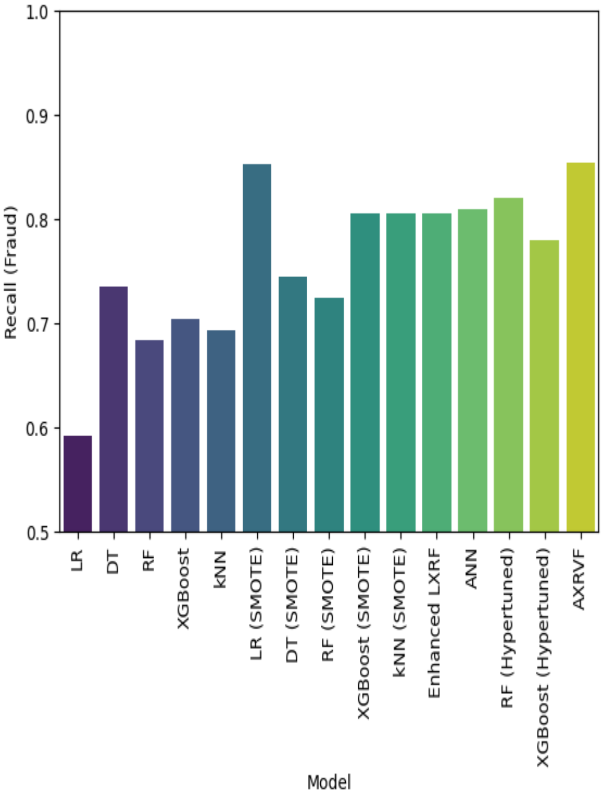

Fig. 6. Comparison of Recall of models.

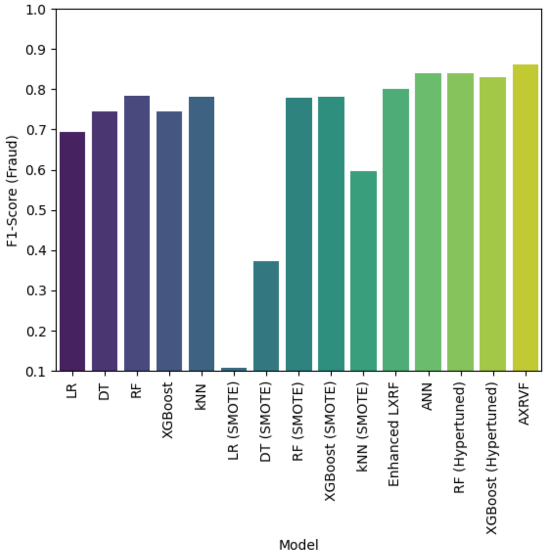

Fig. 7. Comparison of F1-Scorewith different models.

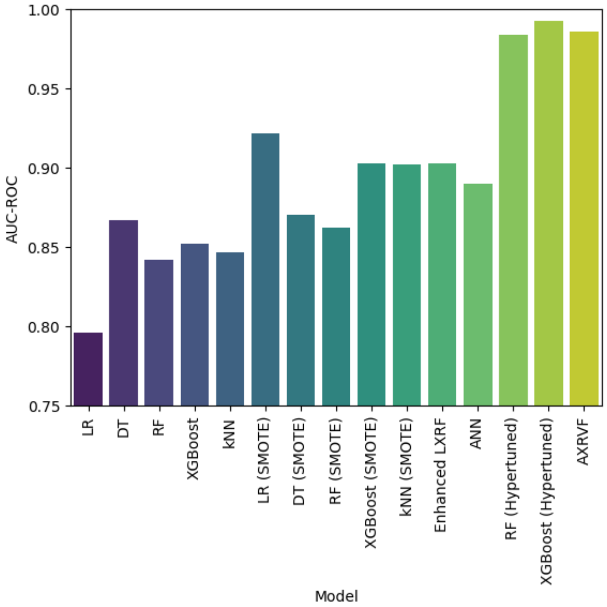

Fig. 8. Comparison of AUC-ROC with different models.

CONCLUSIONS

The study emphasizes the role of data balancing, ensemble learning and hyperparameter tuning techniques in efficient detection of credit-card fraud. The impact of feature selection on the training of models is also discussed by providing a feature selection curve. Several ML algorithms are systematically compared with various augmentation techniques. The effectiveness of SMOTE is also shown in obtaining higher metric evaluation in tree-based models. A deep layer neural network was also trained with a feed forward approach for pattern recognition. Ultimately, an ensemble model is proposed using soft-voting classifier technique, which is coined as AXRVF. The model showed exemplary performance in recall and f1-score while maintaining excellent scores across other metrics. Moreover, this study acts as a foundation for future frameworks inthe field of fintech to keep up with the evolving threatlandscape. Additionally, this can be further implemented for sectors like insurance claims, telecom fraud, and identity theft.

FUTURE SCOPE

While this study demonstrates the effectiveness of a voting-based ensemble framework combined with optimization techniques and SMOTE for handling class imbalance in credit card fraud detection, several directions remain for future exploration. Advanced ensembling strategies can be investigated to further enhance predictive performance and speed. Incorporating deep learning architectures like Transformer or LSTM may enable the detection of complex patterns in transaction data. Real time deployment of the framework could facilitate fraud prevention in online banking applications, while integration with explainable AI techniques would improve interpretability. Additionally, extending the methodology to domains like insurance claims or digital payments can assess the generalizability across different datasets.

REFERENCES

Akazue, M. I., Debekeme, I. A., Edje, A. E., Asuai, C., & Osame, U. J. (2023). Unmasking fraudsters: ensemble features selection to enhance random forest fraud detection. *Journal of Computing Theories and Applications, 1*(2), 201-211.

Aslam, A., & Hussain, A. (2024). A performance analysis of machine learning techniques for credit card fraud detection. *Journal of Artificial Intelligence, 6*, 1.

Asuai, C., Nana, O. K., & Destiny, I. E. (2024). Optimizing credit card fraud detection: A multialgorithm approach with artificial neural networks and gradient boosting model. *International Research Journal of Modernization in Engineering Technology and Science, 6*(12), 2582-5208.

Federal Trade Commission. CSN-Data-Book-2022. no. February 2023. Available online: https://www.ftc.gov/system/files/ftc_ gov/pdf/CSN-Data-Book-2022.pdf (accessed on 11 March 2023).

Hashemi, S. K., Mirtaheri, S. L., & Greco, S. (2022). Fraud detection in banking data by machine learning techniques. *IEEE Access, 11*, 3034-3043.

Karthik, V. S. S., Mishra, A., & Reddy, U. S. (2022). Credit card fraud detection by modelling behaviour pattern using hybrid ensemble model. *Arabian Journal for Science and Engineering, 47*(2), 1987-1997.

Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J., & Adejoh, J. (2024). Enhancing credit card fraud detection: an ensemble machine learning approach. *Big Data and Cognitive Computing, 8*(1), 6.

Mniai, A., Tarik, M., & Jebari, K. (2023). A novel framework for credit card fraud detection. *IEEE Access*, *11*, 112776-112786.

Sadgali, I., Sael, N., & Benabbou, F. (2019). Performance of machine learning techniques in the detection of financial frauds. *Procedia Computer Science, 148*, 45-54.

Syeda Farjana Farabi, M. P. (2024). Enhancing Credit Card Fraud Detection: A Comprehensive Study of Machine Learning Algorithms and Performance Evaluation. *Journal of Business and Management Studies.*

Tanouz, D., Subramanian, R. R., Eswar, D., Reddy, G. P., Kumar, A. R., & Praneeth, C. V. (2021). Credit card fraud detection using machine learning. In 2021 5th international conference on intelligent computing and control systems (ICICCS) (pp. 967-972). IEEE.

Wijaya, M. G., Pinaringgi, M. F., & Zakiyyah, A. Y. (2024). Comparative Analysis of Machine Learning Algorithms and Data Balancing Techniques for Credit Card Fraud Detection. *Procedia Computer Science, 245,* 677-688.