Decision Tree Based Credit Card Fraud Detection By Using Non-Traditional Resampling

D. Sri Sai[1]

[1]*Department of CSE, VFSTR Deemed to be University, Guntur, India*

Abstract – Credit card fraud poses significant challenges to financial institutions and consumers worldwide. We use a decision tree-based approach for detecting fraudulent transactions in credit card data. Using a publicly available dataset, we preprocess the data by handling missing values, scaling features, and addressing class imbalance by an atypical Undersampling method where there is no loss in data and no synthetic data is used. We then train multiple decision tree classifiers for each of the data samples and evaluate their performance using metrics such as accuracy, precision, recall, and F1-score. Then we choose the best classifier as a part of our model to determine whether a transaction is fraudulent or not.

Keywords: Credit Card Fraud, Class imbalance, Decision tree, Random Forest, Gradient Boosting.

1. Introduction

The rise of electronic payments has made credit card fraud a significant concern. Fraudsters exploit vulnerabilities in credit card systems, leading to financial losses for both institutions and consumers. Traditional methods often struggle to keep pace with evolving fraud tactics. This paper explores the use of Decision Tree algorithms for credit card fraud detection. Decision Trees are machine learning models known for their interpretability and efficiency in classification tasks. This project aims to build a system that can spot the difference between real and fraudulent transactions using Decision Trees. This paper also highlights the limitations of various decision trees and how to overcome them by using pre-processing tasks such as data cleaning, feature scaling and resampling of the data.

In this paper, we will see the following classifiers CART, Random Forest Algorithm, and Gradient Boosting algorithm.

CART (Classification and Regression Trees): CART is a decision tree algorithm that splits the data into subsets based on the value of a certain feature. It recursively partitions the data into subsets until it reaches a stopping criterion, typically specified by the maximum depth of the tree or the minimum number of samples required to split a node.

Random Forest Algorithm: Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It builds each tree independently using a random subset of the features and combines their predictions to reduce overfitting and improve accuracy.

Gradient Boosting algorithm: Gradient Boosting is another ensemble learning technique that builds a strong learner by iteratively adding weak learners (usually decision trees) in a sequential manner. It fits new models to the residuals of the previous models, with each subsequent model focusing on the mistakes made by the previous ones.

These classifiers are analysed for various data(Original data, under-sampled data, Oversampled data) and the optimal classifier among these is used in the model. The performance of each classifier is listed in Table-1.

Classifier	Data	Accuracy	Precision	Recall	F1 Score	AUC ROC	Time (sec)
Decision Tree Classifier	Original Data	99.9	75.1	79.7	77.4	89.8	18.97193
	Under Sampling	89.2	87.7	91.2	89.4	89.2	0.011188
	Over Sampling	91.1	99.8	82.3	90.2	91.1	30.65578
Random Forest Classifier	Original Data	100.0	98.4	81.8	89.3	90.9	164.005
	Under Sampling	94.3	95.8	92.6	94.2	94.3	0.230949
	Over Sampling	92.8	100.0	85.5	92.2	92.8	301.7026
Gradient Boosting Classifier	Original Data	99.8	64.5	13.5	22.3	56.8	286.9221
	Under Sampling	93.9	94.5	93.2	93.9	93.9	0.654347
	Over Sampling	95.3	98.5	92.1	95.2	95.3	628.0413

TABLE - 1: CLASSIFIER PERFORMANCE

From Table – 1 we can say that Random Forest Classifier has optimal performance among the three classifiers.

A. The following contributions are made in our work

i. Investigated the resampling techniques in various existing papers to overcome the class imbalance problem and found that only traditional under-sampling and oversampling methods are used. Which results in either data loss or synthetic data being used

ii. Proposed a model which uses an under-sampling method where there is no data loss

The later sections of the paper are systematized in this way, section-2 is the literature survey in which contributions of various researchers on the same problem are mentioned in brief. The proposed system is explained in section-3. The results of the proposed system are in section-4.

2. Literature Survey

[1] Prasad, Yogendra, Sreni Chowdary Cherapalli Bavitha, Earagaraju Mounisha, and Chatna Reethika proposed a methodology combining machine learning algorithms including XGBoost, Support Vector Machine, Decision Tree, Random Forest, and Logistic Regression with Convolutional Neural Networks (CNN) for credit card fraud detection. Deep learning techniques and data-matching trials were employed to improve detection accuracy. The study evaluated the proposed model based on key metrics such as F1 score, accuracy, precision, recall, and Area Under Curve (AUC). Results showed promising performance, with notable improvements in fraud detection accuracy achieved through the integration of deep learning

techniques. Despite the promising results, the proposed model faces limitations such as computational complexity, especially with the integration of CNNs. Additionally, the effectiveness of the model may be impacted by imbalanced datasets, although techniques like the Synthetic Minority Oversampling Technique (SMOTE) were employed to address this challenge.

[2] Muhammad Liman Gambo, Anazida Zainal, and Mohamad Nizam Kassim proposed a Convolutional Neural Network (CNN) model for credit card fraud detection, employing the Adaptive Synthetic (ADASYN) sampling technique to address dataset imbalance. While achieving high precision and recall rates, the model's scalability may be hindered by computational demands, especially with larger datasets. Additionally, as fraud tactics evolve, the model's effectiveness may diminish over time, requiring regular updates and retraining to maintain performance. Despite its promising results, the model's reliance on labelled data and the potential for false positives or negatives remain noteworthy considerations for real-world deployment. Future research could focus on enhancing the model's scalability and robustness to adapt to dynamic fraud patterns while minimizing false alarms.

[3] Ghosh and Reilly's neural network model for credit card fraud detection outperformed traditional methods, offering improved accuracy and reduced false positives. Trained on extensive labelled data, it achieved higher detection rates. However, the model's effectiveness depends on substantial computational resources for processing large datasets. Additionally, its accuracy may fluctuate due to variations in sample sizes. Implementation costs are another concern, potentially limiting accessibility. Despite these limitations, the model represents a promising advancement in fraud detection technology.

[4] The paper titled "A Survey on Credit Card Fraud Detection using Machine Learning" by Rimpal R. Popat and Jayesh Chaudhary provides a comprehensive overview of credit card fraud detection techniques. It discusses various methods including Artificial Immune System, Bayesian Belief Network, Logistic Regression, Decision Tree, Neural Network, Support Vector Machine, Genetic Algorithm, and Hybrid Methods. The authors analyze the strengths, limitations, and challenges associated with each technique, offering valuable insights for researchers and practitioners in the field. This survey serves as a valuable resource for understanding and implementing credit card fraud detection systems.

[5] Nityanand Sharma and Vivek Ranjan introduced a novel approach to credit card fraud detection, employing a hybrid of PSO and K-Means Clustering. Their method outperforms traditional supervised techniques, offering advancements in unsupervised fraud detection. However, while promising, the model's efficacy may be sensitive to parameter settings, and it could face scalability challenges with larger datasets. Additionally, the heuristic optimization process introduces a risk of overfitting, necessitating careful tuning and

validation. Despite these limitations, the proposed methodology represents a significant step forward in leveraging unsupervised learning for fraud detection, paving the way for further research and refinement in this critical domain.

[6] The paper "Credit Card Fraud Detection Using Logistic Regression" by M. Devika, S. Ravi Kishan, L. Sai Manohar, and N. Vijaya proposes a fraud detection methodology employing logistic regression. They address increasing online transaction fraud with a web application utilizing machine learning models. However, advanced preprocessing methods and real-time fraud detection challenges are not explored. The study emphasizes logistic regression's role in efficient fraud detection, but further investigation is needed to enhance accuracy and address real-time transaction monitoring.

[7] Anish Mahajan, Vivek Singh Baghel, and Ramkumar Jayaraman introduced logistic regression coupled with under-sampling of the majority class and oversampling of the minority class for credit card fraud detection. While exhibiting promising fraud detection capabilities, the model lacks real-time functionality and may suffer reduced performance over time. Despite these limitations, the methodology provides valuable insights into enhancing fraud detection in financial systems. It underscores the importance of further research and development to address these limitations and improve the model's applicability in real-world scenarios, ultimately contributing to a more secure financial environment for all stakeholders involved.

[8] The paper "Credit Card Fraud Detection Using Machine Learning Techniques" by Indrani Vejalla, Sai Preethi Battula, Kartheek Kalluri, and Hemantha Kumar Kalluri from VFSTR University, India, proposes credit card fraud detection using supervised machine learning techniques. They address the imbalanced dataset issue through resampling methods. However, while achieving high accuracy, the study lacks a detailed discussion on model interpretability and scalability to large datasets, which are crucial for real-world applications.

[9] The paper explores credit card fraud detection using machine learning on real transaction data. It evaluates Random Forest, Logistic Regression, and AdaBoost algorithms based on the MCC score. Employing Streamlit, it develops a web app for fraud detection. Notably, Random Forest outperforms with a high MCC score of 86%. Future work aims to reduce misclassifications and implement a majority voting system. This research underscores the effectiveness of machine learning in combating fraudulent financial activities, offering insights for future enhancements in fraud detection systems.

[10] Aditi Singh, Anamika Chauhan, Anoushka Singh, and Anshul Aggarwal proposed a methodology for credit card fraud detection using logistic regression, decision trees, random forest, and CatBoost algorithms. CatBoost achieved the highest accuracy at 99.87%, outperforming other methods. However, the study does not address real-time implementation challenges or scalability issues for large datasets. Additionally, it lacks discussion on the

interpretability of the models and the computational resources required for training and inference. Including these aspects could enhance the applicability and practicality of the proposed methodology in real-world scenarios.

[11] The paper titled "Evaluation of Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison" by Qazaleh Sadat Mirhashemi, Negar Nasiri, and Mohammad Reza Keyvanpour compares three machine learning algorithms for credit card fraud detection: Decision Tree, Regression Logistic, and Random Forest. It proposes a hybrid approach for fraud detection and concludes that the proposed model outperforms previous studies in identifying fraudulent transactions. The study emphasizes the importance of choosing effective fraud detection methods due to the increasing prevalence of credit card fraud in digital transactions.

[12] The paper by Anusha et al. introduces an Intelligent Learning Scheme for Digital Fraud Detection (ILSDFD) using deep learning principles to combat credit card fraud. It emphasizes the dynamic nature of fraudulent activities and the need for adaptive detection systems. Employing autoencoders, ILSDFD distinguishes between legitimate and fraudulent transactions in real time. Comparative analysis with conventional methods demonstrates superior performance in accuracy, precision, and recall rates. The study contributes valuable insights to the ongoing development of robust fraud detection systems.

[13] Deepak Singh Nijwala, Rohan Verma, Sudhanshu Maurya, and M. P. Thapliyal proposed a credit card fraud detection model employing Extreme Gradient Boost (XGBoost) with SMOTE oversampling. The model achieved high accuracy, demonstrating effective detection of fraudulent transactions. However, challenges persist with dynamic fraudster behaviour, data constraints, and dataset imbalance, which may limit the model's effectiveness in real-world scenarios, necessitating further research to address these complexities and enhance fraud detection capabilities.

[14] The paper by Yathartha Singh, Kiran Singh, and Vivek Singh Chauhan presents an advanced approach to detecting credit card fraud using machine learning techniques. By applying anomaly detection algorithms like "neighbour outliers" and "forest isolation" on PCA-transformed transaction data, the study aims for comprehensive fraud detection while minimizing misclassifications. However, specific accuracy metrics are not provided, and the summary lacks explicit mention of limitations in the proposed model.

[15] Lakshmi S V S S and Selvani Deepthi Kavila employed logistic regression, decision trees, and random forest for credit card fraud detection, addressing dataset imbalance through oversampling. While their study elucidates multiple machine learning methodologies, it lacks comprehensive accuracy metrics, limiting a detailed evaluation of model efficacy. Moreover, the investigation's scope might benefit from exploring additional techniques beyond the selected algorithms. Despite these constraints, their work contributes

valuable insights into fraud detection systems, albeit with room for further enhancement and exploration of alternative approaches for improved performance evaluation and detection accuracy.

A. *Brief on the existing system:*

In existing credit card fraud detection systems, resampling methods are frequently utilized to tackle the challenge of imbalanced datasets, aiming to alleviate the bias towards the majority class (non-fraudulent transactions) and enhance the performance of machine learning models in identifying fraudulent activities. These methods encompass a range of techniques, including the Synthetic Minority Oversampling Technique (SMOTE), which generates synthetic instances of the minority class by interpolating between existing samples, and Adaptive Synthetic (ADASYN), an extension of SMOTE that adaptively generates synthetic samples based on local density. Additionally, under-sampling of the majority class involves randomly removing instances to balance the class distribution, while over-sampling of the minority class replicates instances to increase its representation. Hybrid methods often combine these approaches or integrate them with ensemble learning or feature engineering to further improve model performance. The utilization of resampling methods underscores the importance of mitigating class imbalance to enhance the accuracy and reliability of fraud detection models, although careful selection and evaluation of techniques based on dataset characteristics and task requirements are crucial. Ongoing research is essential to explore innovative resampling approaches and their impact on real-world model effectiveness.

3. Dataset

The dataset contains transactions made by credit cards in September 2013 by European cardholders.

This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

4. Proposed Model

To overcome the problem of class imbalance in the existing systems, Oversampling Techniques and Under-sampling Techniques are used. Such as SMOTE an oversampling technique where the synthetic samples are generated for the minority class.

Another technique is Random Undersampling an under-sampling technique where the majority class gets reduced to the size of the minority class by removing random records from the majority class. The problem with these techniques is either valid data is lost, or synthetic(not real) data is used. However, in this paper, we used a resampling technique where the large dataset with high-class imbalance is divided into multiple small balanced data samples, where each data sample's size is equal to twice the size of the minority class. We obtained these data samples by taking the minority class and appending a random sample of the majority class of size equal to the size of the minority class. This process is repeated until all the records in the majority class are covered.

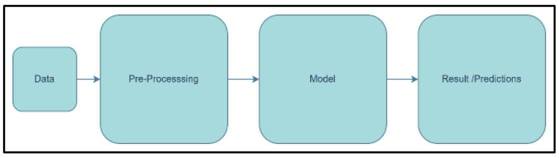

Figure-1: OVERVIEW OF THE PROPOSED MODEL

A. *Methodology:*

i. Data Collection: The first step involves importing the credit card fraud dataset. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

S. No.	Attribute	Description
1	Time	Time elapsed from the first transaction
2	V1 – V29	Personal information about credit card holder
3	Amount	Amount used in the transaction
4	Class	Valid(0)/Fraud(1)

TABLE -2: DATASET DESCRIPTION

ii. Pre-Processing: We employed a common technique where missing entries are replaced with the mean value of each feature. Following this, data scaling is performed. This ensures all features are on a similar scale, potentially improving the model's performance by giving each feature an equal influence during training. It's important to note that scaling is applied to all features except for the target variable (typically labelled as "Class"). After scaling the features we resample the data with class imbalance into multiple small balanced data samples.

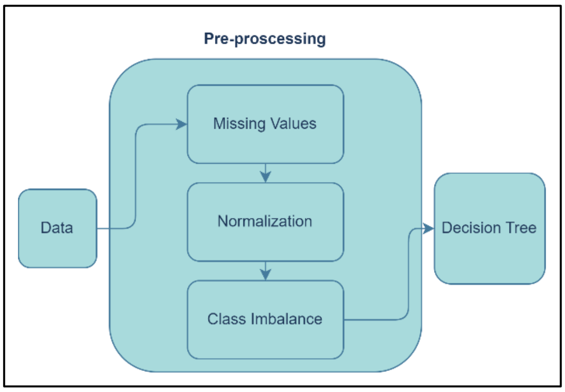

Figure-2: DATA PRE-PROCESSING

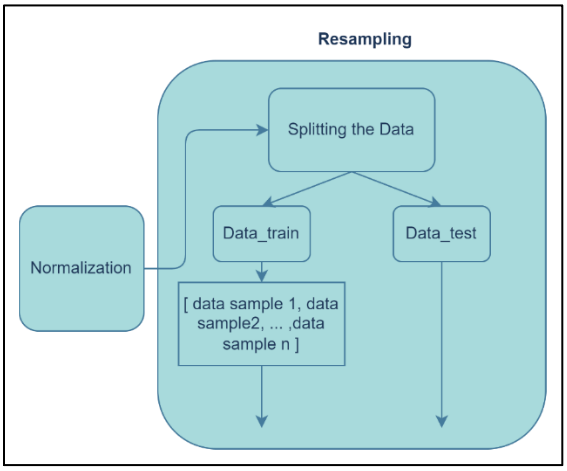

Figure-3: DATA RESAMPLING

iii. Training and Testing: The dataset is divided into two main parts known as training and testing sets. The training set (typically around 70% of the data) is used to train the model. The testing set (usually around 30% of the data) is used to evaluate the model's performance on unseen data. This helps ensure the model doesn't simply memorize the training data and can generalize to identify fraud in new transactions

B. Classifiers used

i. Bagging: A bagging classifier generates its predictions based on multiple classifiers which are parallelly trained on multiple instances/samples of the dataset. The predictions of these multiple classifiers are combined to form the predictions of the bagging classifier.

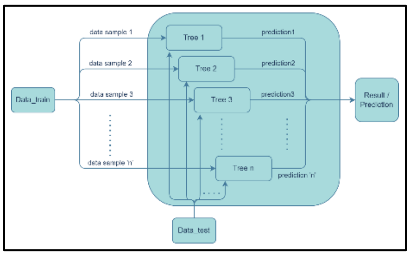

Figure-4: BAGGING OF MULTIPLE DECISION TREES

ii. Decision Tree: A decision tree is a supervised learning algorithm used in machine learning. It partitions the dataset recursively based on selected features, creating a tree-like structure with branches and nodes. Each node represents a feature, and each branch represents a decision based on that feature. The leaf nodes represent the final output or prediction.

iii. Random Forest: Random Forest is an ensemble learning technique that utilizes multiple decision trees to improve predictive accuracy and reduce overfitting. It operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or the mean prediction (regression) of individual trees. Each tree in the forest is built using a random subset of the features and a random subset of the training data, ensuring diversity among the trees. When making predictions, each tree "votes" on the outcome, and the mode or mean of these predictions is taken as the final result. Random Forest is robust against overfitting and noisy data, making it a popular choice for classification and regression tasks.

iv. Gradient Boosting: Gradient Boosting is another ensemble learning technique that builds a strong predictive model by combining multiple weak learners, typically decision trees. Unlike Random Forest, which builds trees independently, Gradient Boosting builds trees sequentially, with each new tree correcting errors made by the previous ones. In each iteration, the model fits a new tree to the residuals (the difference between the actual and predicted values) of the previous predictions, gradually reducing the overall error. By optimizing a loss function, such as mean squared error for regression or cross-entropy for classification, Gradient Boosting iteratively improves model performance. It is particularly effective in handling heterogeneous features and capturing complex relationships in data, often outperforming single decision trees and other machine learning algorithms in predictive accuracy.

5. Results

Our proposed model is implemented on Google Collab using the Windows 11 operating system, with an Intel core i5 processor of 8 GB RAM. The proposed model is implemented on over 2,84,807 records collected from Kaggle, one of the largest repositories which consists of a wide range of datasets. The empirical outcomes manifest that our model yields finer results than the existing one.

A. Performance evaluation metrics

a. Accuracy: It is one of the metrics used to evaluate which strategy is the most effective at discovering patterns and correlations among data samples utilizing input or even training data. The accuracy of the proposed model is computed using the following formula.

$$\text{Accuracy} = \frac{\text{TRP} + \text{TRN}}{\text{TRP} + \text{FLN} + \text{TRN} + \text{FLP}}$$

Where TLP=true positives, TLN =true negatives, FLP=false positives, FLN =false negatives.

b. Precision: It is a metric that measures the consistency of the model's positive predictions, and it is computed using the following formula.

$$\text{Precision} = \frac{\text{TRP}}{\text{TRP} + \text{FLP}}$$

Where TRP=true positives, FLP=false positives.

c. Recall: The ability of a system to locate all similar instances in a dataset is referred to as recall and it is determined by using the following formula.

$$\text{Recall} = \frac{\text{TRP}}{\text{TRP} + \text{FLN}}$$

Where TRP=true positives, FLN =false negatives.

d. F-score: It is among the most significant ML evaluation measures. It concisely summarises a model's prediction performance by merging two previously opposing metrics, precision, and recall. The F-score of the proposed model is computed by using the following formula.

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

Table – 3 shows the performance of each of the classifiers when trained on the original dataset

Classifier	Accuracy	Precision	Recall	F1 Score	AUC ROC
Decision Tree Classifier	99.9	75.1	79.7	77.4	89.8
Random Forest Classifier	100	98.4	81.8	89.3	90.9
Gradient BoostingClassifier	99.8	64.5	13.5	22.3	56.8

TABLE - 3: PERFORMANCE OF CLASSIFIERS BEFORE IMPLEMENTING THE MODEL

Table 4 and Figure 5 shows the evaluation metrics for the model with a decision tree classifier for different test sizes.

DecisionTreeClassifier					
Test Size(%)	Accuracy	Precision	Recall	F1 Score	AUC-ROC
10	93.9	92.2	95.9	94.0	93.9
20	94.9	94.9	94.9	94.9	94.9
30	94.3	95.8	92.6	94.2	94.3
40	94.7	96.3	92.9	94.6	94.7
50	93.9	95.8	91.9	93.8	93.9

TABLE - 4: PERFORMANCE OF DECISION TREE CLASSIFIER WITH DIFFERENT TEST SIZES

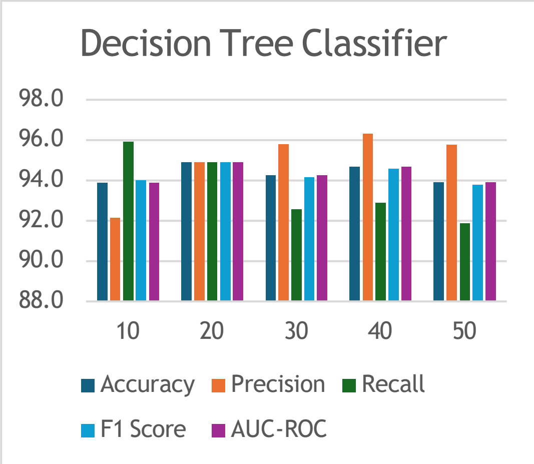

Figure - 5: PERFORMANCE OF DECISION TREE CLASSIFIER WITH DIFFERENT TEST SIZES

Table 5 and Figure 6 shows the evaluation metrics for the model with a Random Forest classifier for different test sizes.

RandomForestClasifier					
Test Size(%)	Accuracy	Precision	Recall	F1 Score	AUC-ROC
10	96.9	97.9	95.9	96.9	96.9
20	95.9	98.9	92.9	95.8	95.9
30	94.6	97.1	91.9	94.4	94.6
40	94.9	98.4	91.4	94.7	94.9
50	94.3	97.8	90.7	94.1	94.3

TABLE - 5: PERFORMANCE OF RANDOM FOREST CLASSIFIER WITH DIFFERENT TEST SIZES.

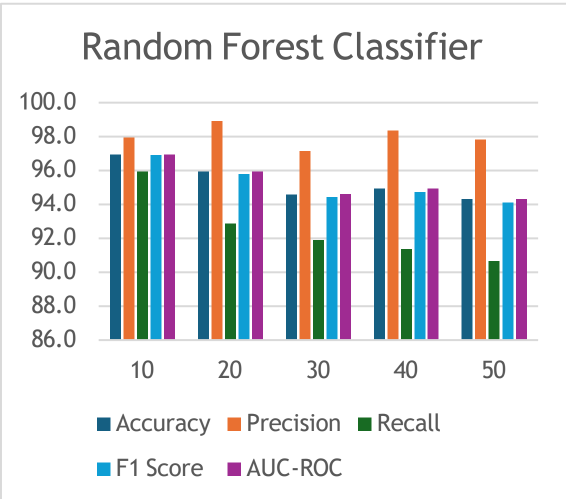

Figure - 6: PERFORMANCE OF RANDOM FOREST CLASSIFIER WITH DIFFERENT TEST SIZES.

Table 6 and Figure 7 shows the evaluation metrics for the model with Gradient Boosting classifier for different test sizes.

GradientBoostingClassifier					
Test Size(%)	Accuracy	Precision	Recall	F1 Score	AUC-ROC
10	94.9	92.3	98.0	95.1	94.9
20	95.4	95.9	94.9	95.4	95.4
30	94.3	95.8	92.6	94.2	94.3
40	94.4	96.3	92.4	94.3	94.4
50	94.5	96.6	92.3	94.4	94.5

TABLE - 6: PERFORMANCE OF GRADIENT BOOSTING CLASSIFIER WITH DIFFERENT TEST SIZES.

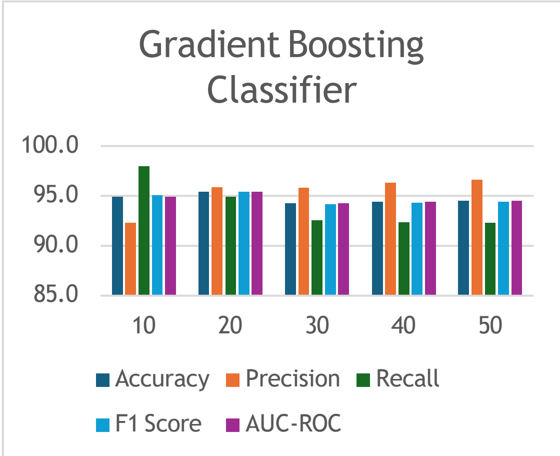

Figure - 7: PERFORMANCE OF GRADIENT BOOSTING CLASSIFIER WITH DIFFERENT TEST SIZES

B. Results Discussion

In Table 3 we can see that the accuracy of the classifier is very high this is because the classification is applied to imbalanced data, and this is the reason we used multiple evaluation metrics to evaluate the performance of our model. Our proposed model has higher precision, recall, and F1 scores than the classifiers. The average precision of all the classifiers is 79.3 and the precision of our model with 20% test size and with Gradient Boosting Classifier is 95.4, the average recall of all the classifiers is 58.3 and the recall of our model with 20% test size and with Gradient Boosting Classifier is 94.9, average F1 Score of all the classifiers is 63 and the recall of our model with 20% test size and with Gradient Boosting Classifier is 95.4, average AUC-ROC of all the classifiers is 79.1 and the AUC-ROC of our model with 20% test size and with Gradient Boosting Classifier is 95.4. The prediction results of our proposed model are returned in a CSV file along with printing in the console.

6. Conclusion

In this study, we used Kaggle's credit card dataset to validate the efficiency of various supervised machine learning models in predicting the likelihood of fraudulent transactions. We used Accuracy, Precision, Recall, F1 Score, and AUC-ROC as determinants to reach a specific outcome. Accuracy alone is not considered an attribute since it is not sensitive to class imbalance and does not provide a clear answer. We have implemented our model with a Decision Tree Classifier, Random Forest Classifier, and Gradient Boosting Classifier predicting that the best-suited model is the Gradient Boosting model. As a future scope, the model's performance could be increased if we consider bagging higher models instead of bagging base classifiers.

7. References

[1] P. Y. Prasad, A. S. Chowdary, C. Bavitha, E. Mounisha and C. Reethika, "A Comparison Study of Fraud Detection in Usage of Credit Cards using Machine Learning," *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2023, pp. 1204-1209, doi: 10.1109/ICOEI56765.2023.10125838.

[2] M. L. Gambo, A. Zainal and M. N. Kassim, "A Convolutional Neural Network Model for Credit Card Fraud Detection," *2022 International Conference on Data Science and Its Applications (ICoDSA)*, Bandung, Indonesia, 2022, pp. 198-202, doi: 10.1109/ICoDSA55874.2022.9862930.

[3] B. Al Smadi and M. Min, "A Critical review of Credit Card Fraud Detection Techniques," *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2020, pp. 0732-0736, doi: 10.1109/UEMCON51285.2020.9298075

[4] R. R. Popat and J. Chaudhary, "A Survey on Credit Card Fraud Detection Using Machine Learning," *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2018, pp. 1120-1125, doi: 10.1109/ICOEI.2018.8553963.

[5] N. Sharma and V. Ranjan, "Credit Card Fraud Detection: A Hybrid of PSO and K-Means Clustering Unsupervised Approach," *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2023, pp. 445-450, doi: 10.1109/Confluence56041.2023.10048876.

[6] M. Devika, S. R. Kishan, L. S. Manohar and N. Vijaya, "Credit Card Fraud Detection Using Logistic Regression," *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment,*

Computing & Communication Engineering (ICATIECE), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/ICATIECE56365.2022.10046976.

[7] Mahajan, V. S. Baghel and R. Jayaraman, "Credit Card Fraud Detection using Logistic Regression with Imbalanced Dataset," *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2023, pp. 339-342.

[8] Vejalla, S. P. Battula, K. Kalluri and H. K. Kalluri, "Credit Card Fraud Detection Using Machine Learning Techniques," *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, Nagpur, India, 2023, pp. 1-4, doi: 10.1109/PCEMS58491.2023.10136040.

[9] V. Jain, H. Kavitha and S. Mohana Kumar, "Credit Card Fraud Detection Web Application using Streamlit and Machine Learning," *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, Hassan, India, 2022, pp. 1-5, doi: 10.1109/ICDSIS55133.2022.9915901.

[10] Singh, A. Singh, A. Aggarwal and A. Chauhan, "Design and Implementation of Different Machine Learning Algorithms for Credit Card Fraud Detection," *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Maldives, Maldives, 2022, pp. 1-6, doi: 10.1109/ICECCME55909.2022.9988588.

[11] Q. S. Mirhashemi, N. Nasiri and M. R. Keyvanpour, "Evaluation of Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," *2023 9th International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, 2023, pp. 247-252, doi: 10.1109/ICWR57742.2023.10139098.

[12] P, S. Bharath, N. Rajendran, S. D. Devi and S. Saravanakumar, "Experimental Evaluation of Smart Credit Card Fraud Detection System using Intelligent Learning Scheme," *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICSES60034.2023.10465367.

[13] D. S. Nijwala, S. Maurya, M. P. Thapliyal and R. Verma, "Extreme Gradient Boost Classifier based Credit Card Fraud Detection Model," *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)*, Dehradun, India, 2023, pp. 500-504, doi: 10.1109/DICCT56244.2023.10110188.

[14] Y. Singh, K. Singh and V. Singh Chauhan, "Fraud Detection Techniques for Credit Card Transactions," *2022 3rd*

International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2022, pp. 821-824, doi: 10.1109/ICIEM54221.2022.9853183.

[15] Vejalla, S. P. Battula, K. Kalluri and H. K. Kalluri, "Credit Card Fraud Detection Using Machine Learning Techniques," *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, Nagpur, India, 2023, pp. 1-4, doi: 10.1109/PCEMS58491.2023.10136040.