Fraud Detection in Online Payments

An Analysis Using Machine Learning Techniques

Stamatios Karvounis
School of Computing
National College of Ireland
Dublin, Ireland
x18197051@student.ncirl.ie

***Abstract*— In today's digital era, securing online payment transactions is paramount. This study leverages the "Online Payments Fraud Detection Dataset" from Kaggle, containing over 6 million transactions, to evaluate the efficacy of machine learning algorithms - Logistic Regression, Random Forest, Decision Trees, and Bagging Classifier - in detecting fraud. Preliminary results highlight Random Forest's proficiency in identifying complex transaction patterns. Emphasis is also placed on the role of features such as transaction type and amount in model accuracy. The research aims to enhance online transaction security and equip financial institutions with tools to mitigate fraud risks.**

Keywords—Machine Learning, Algorithms, Logistic Regression, Random Forest, Decision Trees, Bagging

I. Introduction

The surge in online financial activities in the modern digital era has emphasized the significance of fraud detection for financial institutions. As cybercrimes escalate and pose potential monumental monetary risks, establishing robust fraud detection mechanisms becomes imperative. This study endeavors to fortify fraud detection methodologies by harnessing the power of machine learning algorithms and meticulously examining a detailed dataset of online financial exchanges. This project blueprint is geared towards concocting an efficient system for detecting fraud via machine learning. Through the dissection of a dataset pertaining to credit card transactions, the emphasis will be on pinpointing fraudulent activities with precision while curtailing false alarms. The exploration will encompass diverse classification and clustering techniques, taking into account elements like consumer habits, expenditure trends, and historical fraud patterns. The overarching goal is to bolster the integrity of online dealings, shield consumers from deceitful actions, and equip financial entities with pivotal insights to thwart fraudulent maneuvers.

A. Research Question

a. How can machine learning algorithms be utilized to detect and classify fraudulent transactions in online payments?

b. How does the performance of different machine learning algorithms compare in terms of fraud detection accuracy and efficiency?

B. Hypotheses

a. Hypothesis 1: Machine learning algorithms, such as logistic regression, decision trees, and random forests, can effectively classify fraudulent and non-fraudulent transactions in online payments.

b. Hypothesis 2: The inclusion of additional features, such as transaction amount, balance changes, and recipient details, will improve the accuracy and reliability of fraud detection models.

II. Literature Review

A. Selecting a Template (Heading 2)

Detecting fraudulent activities is a paramount challenge for financial institutions. Machine learning approaches, particularly algorithms like random forest, decision trees, and logistic regression, have emerged as potent tools for identifying suspicious transactions. In recent times, there has been a surge in research focusing on harnessing these algorithms for fraud detection.

Logistic regression, for instance, has been applied to discern fraudulent from genuine transactions using a range of input factors. An ensemble approach, incorporating logistic regression with other machine learning techniques, was explored by Dhankhad et al.[3]. This amalgamation tapped into the unique strengths of various algorithms, leading to enhanced detection capabilities.

Decision trees, renowned for their clarity and ability to unearth intricate decision-making criteria, have been instrumental in flagging fraud. They analyze various transaction attributes, such as amounts, vendor locations, and time stamps. A noteworthy study by Sadineni et al.[1] utilized decision trees to spot deceptive credit card dealings, yielding impressive precision and recall metrics.

Random forest, a methodology that synergizes multiple decision trees, has become increasingly favored in the realm of fraud detection. Its prowess lies in managing intricate datasets and discerning non-linear interdependencies between attributes. A research undertaking by Zhang et al.[5] spotlighted the efficacy of random forest in detecting credit card fraud. By evaluating transaction behaviors, customer actions, and other transaction specifics, it was observed that random forest surpassed many conventional machine learning techniques in accuracy.

Additionally, the bagging classifier has been a valuable tool in the fraud detection toolkit. By generating multiple versions of a predictor and aggregating their outputs, the bagging classifier offers enhanced stability and accuracy, especially in situations where the dataset might have noise or outliers [4].

In summation, existing studies underscore the significant potential of machine learning methods, especially random forest, decision trees, and logistic regression, in the battle against financial fraud. Their capacity to discern intricate transactional nuances translates to enhanced detection precision and fewer false alarms.

III. Methodology

A. Data Source

The dataset for this study is the "Online Payments Fraud Detection Dataset" sourced from Kaggle. This dataset encompasses 6,353,307 entries, with each entry denoting a distinct online payment transaction. It is composed of 11 features, such as transaction nature, amount, details of the sender and receiver, and fraud markers. To streamline the modeling and evaluation, the dataset will be segmented into training and testing subsets.

The dataset's features are detailed as follows:

- Step: This feature captures time units, with each unit equating to an hour, facilitating the chronological tracking of transactions.

- Type: This feature categorizes the transaction, shedding light on the various transaction methods or types.

- Amount: Representing the transaction's monetary value, this feature gives a numerical perspective on the transaction's scale.

- NameOrig: This feature identifies the transaction's initiator, enabling the study of particular user patterns.

- OldbalanceOrg: Representing the initiator's account balance pre-transaction, it sets a baseline for balance alterations.

- NewbalanceOrig: Post-transaction, this feature denotes the updated balance of the initiator's account, allowing for balance change assessments.

- NameDest: This feature designates the transaction's beneficiary, providing a lens into the money's direction and potential fraudulent activities centered on certain beneficiaries.

- OldbalanceDest: Reflecting the beneficiary's account balance before the transaction, it offers a backdrop for beneficiary balance shifts.

- NewbalanceDest: This feature indicates the beneficiary's updated account balance post-transaction, aiding in analyzing balance fluctuations.

- IsFraud: As a binary marker, this feature indicates the legitimacy of a transaction, forming the foundation for model training and evaluation.

- IsFlaggedFraud: Highlighting transactions deemed suspicious by the system or set criteria, this feature adds an extra layer of scrutiny to transactions that might be high-risk.

B. Descriptive Statistics

The dataset under examination comprises various transactional attributes, each offering insights into the dynamics of online payments. A comprehensive descriptive analysis of these attributes is presented below:

- Transaction Type(type) : The dataset catalogues five distinct transaction modalities. Predominantly, the transaction mode 'CASH_OUT' emerges as the most recurrent, registering a total of 670,801 instances.

- Transaction Magnitude (amount): Transactions, on average, hover around a value of approximately 179,972. The spectrum of transactional amounts spans from a minimum of 0 to an upper limit of 69,337,200. The central tendency, represented by the median, situates at 74,835.82. Furthermore, three-quarters of the transactions have amounts that do not exceed 208,812.8.

- Origination Account Details (nameOrig): A total of 1,907,926 unique origination account identifiers are discerned from the dataset. Notably, certain accounts, exemplified by 'C2077948692', have engaged in more than a singular transaction, pointing towards repetitive transactional behavior.

- Origination Account Balance Dynamics (oldbalanceOrg and newbalanceOrig): Prior to transactional activity, accounts hold an average balance of approximately 834,615.7. Subsequent to the transaction, this metric marginally escalates to around 856,019.5. The dataset captures an apex origination balance of 57,316,260 pre-transaction, which experiences a decrement to 47,316,260 post-transaction. Intriguingly, the median balance prior to the transaction is 14,339.35, which predominantly diminishes to zero following the transaction.

- Destination Account Dynamics (nameDest): The dataset encompasses 1,055,280 unique destination account monikers. Among these, 'C1899073220' emerges as a frequent transactional endpoint, having been the recipient in 38 separate instances.

- Destination Account Balance Trajectory (oldbalanceDest and newbalanceDest): Prior to the influx from a transaction, destination accounts, on an average, maintain a balance close to 1,103,267. This figure experiences an increment, averaging at about 1,227,516, post the transaction. The highest balance recorded at a destination account touches 355,381,400 before the transaction, and sees a slight surge to 356,015,900 post the transaction.

- Fraudulence Indicator (isFraud): A critical observation from the dataset is the pronounced imbalance in the representation of fraudulent transactions. A mere 0.1275% of the transactions are tagged as fraudulent. This variable is binary in nature, with '0' denoting legitimate transactions and '1' signifying fraudulent activities.

C. Visualizing the Data

Data visualization offers a clear lens to discern patterns, and relationships amongst variables. In the following segment, the dataset will be visualized across several key attributes to get more clear insights.

- Distribution by Transaction Type:

As indicated in Figure 1, data landscape is dominated by a substantial volume of CASH_OUT and PAYMENT transactions, reflecting these as the predominant financial activities for users. CASH_IN, although frequent, trails behind in its occurrence, underlining that deposit transactions aren't as recurrent as withdrawals and payments. The other transaction categories, notably TRANSFER, are sporadic, paling in comparison to the top three.

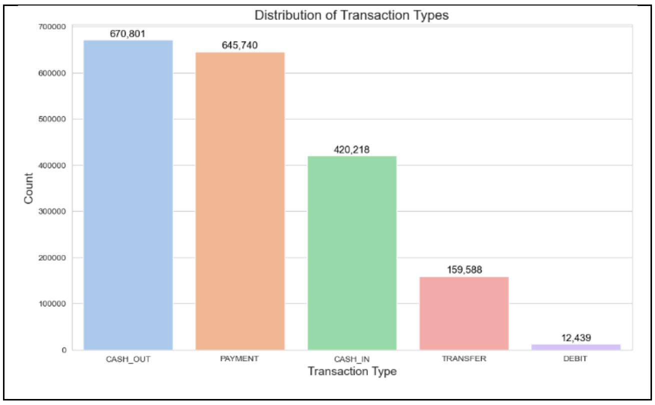


Figure 1 Distribution by transaction type

- Distribution of Transaction Amounts:

Figure 2 shows that smaller transaction amounts constitute the bulk of the data. A trend of diminishing frequency is observed as transaction amounts augment. Really high-valued transactions are sparse. Certain amplified transaction amounts might recur, hinting at standard financial practices or inherent data patterns.

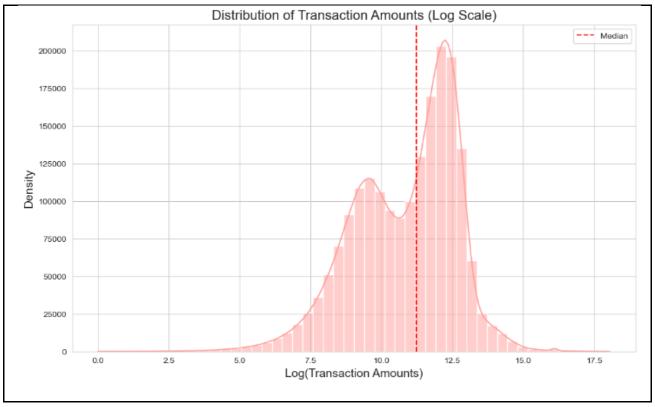


Figure 2 Distribution of Transaction Amounts

- Distribution of Origination Account Balances (oldbalanceOrg and newbalanceOrig):

Figure 3 shows a significant chunk of accounts commenced with paltry balances, and post-transaction, the majority sustained these modest amounts. Accounts flush with funds initially are a rarity, with their prevalence seemingly dwindling post-transactions. Despite the pervasive low balances, there's evidence of sizable transactions, spotlighted by accounts with substantial balances both pre and post transactions.

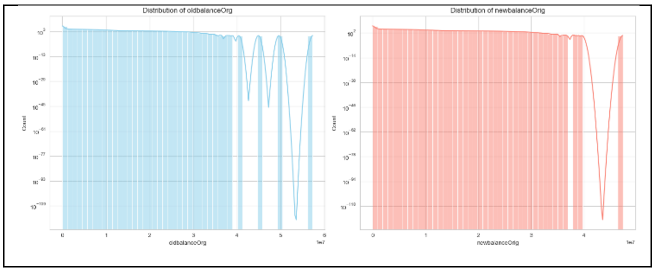
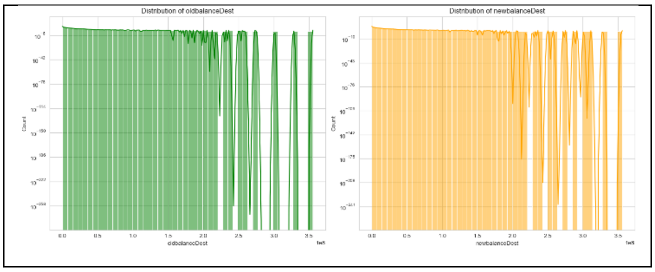


Figure 3 Distribution of oldbalanceOrg and newbalanceOrig

- Distribution of Destination Account Balances (oldbalanceDest and newbalanceDest):

Figure 4 shows a large portion of destination accounts inaugurated with scanty balances. After the inflow, most of these accounts either clung to or reverted to meager amounts. Affluent destination accounts at the outset are an exception, with their frequency marginally tapering post-transactions. The data underscores the coexistence of both modest and hefty transactions, corroborated by accounts with extensive balances both pre and post transactions.



D. Handling Categorical Data

Within the dataset, the column labeled 'type' defines the nature of transactions and is categorical. For effective assimilation by machine learning models and to ensure a seamless numeric representation, one-hot encoding was employed to modify this specific column.

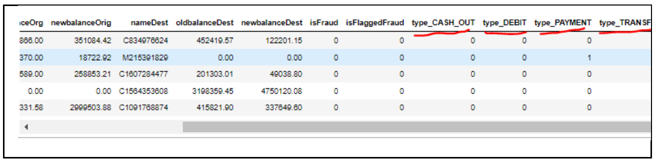
One-hot encoding is a prevalent strategy to transmute categorical variables into a structured format that's readily digestible for machine learning processes. Instead of allowing the 'type' column to exist as a singular column filled with textual categories, one-hot encoding broadens its structure by inaugurating a distinct binary column for every unique category present in the original data.

To illustrate, should our 'type' column host categories like 'CASH_OUT' or 'TRANSFER', post-encoding, we'd be presented with individual columns named 'type_CASH_OUT' and 'type_TRANSFER' respectively. These columns would be populated with binary values:

A '1' signifying that a specific transaction aligns with that category.

A '0' indicating the contrary.

By adopting this methodology for the 'type' column, potential misinterpretations were eliminated perceived ordinal relationships among categories, bestowing our machine learning models with a clarified numeric perspective of each transaction variety. This enhancement ensures our models can distinguish between different transaction types with heightened precision, leading to improved predictive outcomes. The dataset looks like this at this point:



iceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	type_CASH_OUT	type_DEBIT	type_PAYMENT	type_TRANSF
866.00	351084.42	C834976624	452419.57	122201.15	0	0	0	0	0	
370.00	18722.92	M215391829	0.00	0.00	0	0	0	0	1	
589.00	258853.21	C1607284477	201303.01	49038.80	0	0	0	0	0	
0.00	0.00	C1564353608	3198359.45	4750120.08	0	0	0	0	0	
331.58	2999503.88	C1091768874	415821.90	337649.60	0	0	0	0	0	

E. Dropping Identifiers Columns

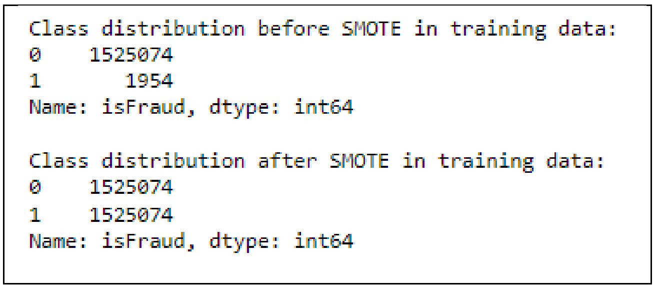
In the process of preparing the data for machine learning modeling, the nameOrig and nameDest columns were dropped from the dataset. There are several compelling reasons for this decision:

High Cardinality: Both columns contain unique identifiers for customers or accounts. Such columns tend to have a large number of unique values, known as high cardinality. Encoding such columns would result in a vast number of new features, making the dataset much larger and potentially harder to model. Identifiers, by their nature, are used to label or distinguish entities and typically do not carry intrinsic predictive information for the target variable. Including them in the model might lead to overfitting, where the model starts to memorize these identifiers rather than learning the actual patterns in the data. In addition, Machine learning algorithms require numeric input features. The nameOrig and nameDest columns contain string values, which would need to be converted to a numerical format. Given the high cardinality of these columns, traditional encoding methods might not be efficient or meaningful. Lastly, removing columns that don't provide significant predictive value helps in simplifying the model. A simpler model is often more interpretable and easier to validate, maintain, and deploy.

F. Handling Imbalanced Data

Given that the "isFraud" variable shows imbalance in distributions, with class 0 accounting for approximately 99.87% and class 1 representing a mere 0.13%, the target variable exhibits a clear imbalance. To rectify this and ensure a balanced representation, the SMOTE technique is applied.

Dealing with imbalanced datasets is crucial, especially in tasks like fraud detection, where the minority class (fraudulent transactions) is typically of more interest. Imbalanced datasets can lead to models that have high accuracy but poor recall for the minority class, as the model may simply predict the majority class for most instances. The SMOTE technique is a popular method to address this imbalance. It works by creating synthetic samples in the feature space. After applying SMOTE, we observe that the number of samples in the class isFraud=1 has increased to match the number of samples in class isFraud=0. This means the training data is now balanced in terms of the target variable.



```
Class distribution before SMOTE in training data:
0    1525074
1       1954
Name: isFraud, dtype: int64

Class distribution after SMOTE in training data:
0    1525074
1    1525074
Name: isFraud, dtype: int64
```

Figure 4 Before and After SMOTE

IV. EVALUATION

In this section, a comprehensive exploration of four distinct machine learning algorithms is conducted to decipher the patterns and intricacies within our dataset. These algorithms have been chosen based on their applicability to classification problems, their performance in similar contexts, and their ability to handle the nuances of our specific dataset. The four algorithms that will be employed are:

Logistic Regression: A foundational algorithm renowned for its simplicity and effectiveness, particularly when the relationship between the independent and dependent variable is somewhat linear. It serves as a benchmark for many classification tasks.

Random Forest: An ensemble learning method that creates multiple decision trees during training and outputs the mode of the classes for classification. It is known for its high accuracy, ability to handle large datasets with higher dimensionality, and its ability to handle missing values.

Decision Trees: A flowchart-like structure where each internal node represents a feature(or attribute), each branch represents a decision rule, and each leaf node represents an outcome. It's intuitive and easy to interpret, making it valuable for insights.

Bagging Classifier: An ensemble meta-estimator that fits base classifiers on random subsets of the original dataset and then aggregates their individual predictions to form a final prediction. It aims to reduce overfitting and improve the accuracy of models.

By leveraging these diverse algorithms, the aim is to achieve a robust and accurate fraud detection system..

A. Logistic Regression

Precision for class 0 (Not Fraud): The model correctly predicted 99.87% of the non-fraudulent transactions out of all the transactions it predicted as non-fraudulent.

Recall for class 0 (Not Fraud): The model was able to capture 91.72% of the actual non-fraudulent transactions.

Precision for class 1 (Fraud): The model correctly predicted only 1.35% of the fraudulent transactions out of all the transactions it predicted as fraudulent. This is a low precision rate.

Recall for class 1 (Fraud): Impressively, the model was able to capture 89.81% of the actual fraudulent transactions, which is quite high.

F1-score: Gives a balance between precision and recall. The F1-score for fraudulent transactions is low, primarily because of the low precision.

Confusion Matrix: True Negatives (349,760): These are the actual non-fraudulent transactions that were correctly predicted as non-fraudulent. False Positives (31,517): These are the actual non-fraudulent transactions that were incorrectly predicted as fraudulent. False Negatives (49): These are the actual fraudulent transactions that were incorrectly predicted as non-fraudulent. True Positives (432): These are the actual fraudulent transactions that were correctly predicted as fraudulent.

ROC-AUC Score:The ROC-AUC score is 0.9077, which is reasonably good. An ROC-AUC score of 1 represents a perfect classifier, and 0.5 represents a worthless classifier. So, 0.9077 indicates a good capability to distinguish between the positive and negative classes.

The model does an impressive job in terms of recall for the fraudulent class, capturing almost 90% of the fraudulent transactions. This is crucial in fraud detection, as failing to identify a fraudulent transaction (False Negative) can have severe consequences.

However, the precision for the fraudulent class is quite low, which means that while the model is capturing most frauds, it's also misclassifying a significant number of non-fraudulent transactions as fraudulent (False Positives). This could lead to a lot of false alarms, which in a real-world scenario might lead to unnecessary checks or actions on legitimate transactions. The overall accuracy is high (92%), but in the context of imbalanced datasets, accuracy can be misleading. The more relevant metrics here are precision, recall, and the F1-score, especially for the minority class (fraudulent transactions).

B. Random Forest

Precision for class 0 (Not Fraud): The model correctly predicted virtually 100%100% of the non-fraudulent transactions out of all the transactions it labeled as non-fraudulent. This is excellent.

Recall for class 0 (Not Fraud): The model captured 99.92%99.92% of the actual non-fraudulent transactions. This is also outstanding.

Precision for class 1 (Fraud): Out of the transactions that the model predicted as fraudulent, 61%61% were actually fraudulent. This precision is a significant improvement over the Logistic Regression model.

Recall for class 1 (Fraud): The model was able to identify 92.72%92.72% of the actual fraudulent transactions. This is slightly higher than the recall of the Logistic Regression model.

F1-score: The F1-score for fraudulent transactions is 0.730.73, which is much higher than that of the Logistic Regression model. It provides a balanced metric between precision and recall.

Confusion Matrix: True Negatives (380,987): These are the actual non-fraudulent transactions that were correctly predicted as non-fraudulent. False Positives (290): These are the actual non-fraudulent transactions that were incorrectly flagged as fraudulent. False Negatives (35): These are the actual fraudulent transactions that were missed and labeled as non-fraudulent. True Positives (446): These are the actual fraudulent transactions that were correctly identified as fraudulent.

ROC-AUC Score: The ROC-AUC score is 0.96320.9632, which is outstanding. It indicates a strong ability of the model to distinguish between the positive (fraudulent) and negative (non-fraudulent) classes.

The Random Forest classifier demonstrates a robust performance in the fraud detection task. The model not only captures a large majority of the actual fraudulent transactions but also maintains a good precision, resulting in fewer false alarms compared to the Logistic Regression model.

While the overall accuracy is near perfect, the more critical metrics for this problem are the precision and recall for the minority class (fraudulent transactions). The Random Forest model showcases a commendable balance between these metrics. The false positives (290) are substantially reduced compared to the Logistic Regression model, leading to fewer legitimate transactions being flagged as suspicious.

A recall of 92.72%92.72% for the fraudulent class is impressive, capturing most of the fraudulent activities. However, 35 fraudulent transactions were still missed, indicating there's room for further improvement.

In conclusion, the Random Forest classifier shows strong potential as a solution for fraud detection in this dataset. The significant improvement in precision for the fraudulent class, combined with high recall, suggests that this model would be valuable in a real-world fraud detection system.

C. Decision Trees

Precision for class 0 (Not Fraud): The model correctly predicted virtually 100%100% of the non-fraudulent transactions out of all the transactions it labeled as non-fraudulent. This is excellent.

Recall for class 0 (Not Fraud): The model captured 99.91%99.91% of the actual non-fraudulent transactions. This is exceptional.

Precision for class 1 (Fraud): Out of the transactions that the model predicted as fraudulent, 56%56% were actually fraudulent. This precision is lower than the Random Forest model but higher than the Logistic Regression model.

Recall for class 1 (Fraud): The model was able to identify 94.38%94.38% of the actual fraudulent transactions. This recall is slightly higher than both the Logistic Regression and Random Forest models.

F1-score: The F1-score for fraudulent transactions is 0.700.70, indicating a good balance between precision and recall, though slightly lower than the Random Forest model.

Confusion Matrix: True Negatives (380,921): These are the actual non-fraudulent transactions that were correctly predicted as non-fraudulent. False Positives (356): These are the actual non-fraudulent transactions that were incorrectly flagged as fraudulent. False Negatives (27): These are the actual fraudulent transactions that were missed and labeled as non-fraudulent. True Positives (454): These are the actual fraudulent transactions that were correctly identified as fraudulent.

ROC-AUC Score: The ROC-AUC score is 0.97150.9715, which is outstanding. It suggests that the model has an excellent capability to differentiate between the positive (fraudulent) and negative (non-fraudulent) classes.

The Decision Tree classifier demonstrates strong performance in fraud detection, especially when considering its interpretability advantage over more complex models like Random Forest. While its precision for the fraudulent class is slightly lower than Random Forest, the Decision Tree model achieves a high recall, capturing a significant portion of the actual fraudulent transactions.

The total number of false positives (356) is slightly higher than the Random Forest model, which means there might be a few more legitimate transactions flagged as suspicious. However, the false negatives (27) are fewer than the Logistic Regression model, indicating fewer missed fraudulent transactions. Given its simplicity and interpretability, the Decision Tree's performance is commendable. However, it might benefit from some hyperparameter tuning or pruning to mitigate overfitting and further optimize its performance.

In summary, the Decision Tree classifier offers a solid performance for fraud detection in this dataset, making it a viable option, especially when model interpretability is a priority.

D. *Bagging Classifier*

Precision for class 0 (Not Fraud): This model correctly predicted 99.87%99.87% of the non-fraudulent transactions out of all the transactions it labeled as non-fraudulent.

Recall for class 0 (Not Fraud): The model captured 96.50%96.50% of the actual non-fraudulent transactions.

Precision for class 1 (Fraud): Out of the transactions the model predicted as fraudulent, only 3.37%3.37% were actually fraudulent. This precision is lower than all the models we've discussed so far.

Recall for class 1 (Fraud): The model impressively identified 96.67%96.67% of the actual fraudulent transactions.

F1-score: The F1-score for fraudulent transactions is 0.070.07, significantly lower than the other models because of the low precision, even though recall is high.

Confusion Matrix: True Negatives (367,927): These are the actual non-fraudulent transactions that were correctly predicted as non-fraudulent. False Positives (13,350): These are the actual non-fraudulent transactions that were incorrectly flagged as fraudulent. This number is higher than in the other models. False Negatives (16): These are the actual fraudulent transactions that were missed and labeled as non-fraudulent. This number is impressively low. True Positives (465): These are the actual fraudulent transactions that were correctly identified.

ROC-AUC Score: The ROC-AUC score is 0.96590.9659, which is outstanding. It suggests the model's capability to differentiate between the positive (fraudulent) and negative (non-fraudulent) classes is excellent.

The Bagging Classifier demonstrates a high capability to detect fraudulent transactions, as indicated by the high recall and ROC-AUC score. However, its precision for detecting fraud is notably low, which means while it catches most of the actual fraud cases, it also mislabels a significant number of legitimate transactions as fraudulent (False Positives). This might lead to unnecessary alerts and checks on genuine transactions, which can be inconvenient for customers and administrators. The model's overall accuracy is still very high, but the cost of false positives in a fraud detection scenario can be significant. Hence, the trade-off between precision and recall needs careful consideration.

While Bagging reduces the variance in the predictions, in this case, it seems to have increased the number of false positives. This might be due to the high variability in the majority class, which gets sampled multiple times across different bags. In summary, while the Bagging Classifier's ability to detect actual fraud is commendable, its high number of false positives might make it less desirable than models like Random Forest for practical deployment.

E. *Evaluation and Comparison*

The table presented below offers a comprehensive summary of various performance measures associated with the four machine learning models under consideration: Logistic Regression, Random Forest, Decision Trees, and Bagging Classifier. In that way, ee aim to facilitate an objective evaluation of each model's strengths and potential areas of improvement, thereby guiding the selection of the most suitable approach for fraud detection in our dataset.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	0.92	0.01	0.90	0.03	0.9077
Random Forest	1.00	0.61	0.93	0.73	0.9632
Decision Trees	1.00	0.56	0.94	0.70	0.9715
Bagging Classifier	0.96	0.03	0.97	0.07	0.9659

Logistic Regression

Recall: At 90%, this model correctly identifies a very high proportion of actual frauds. This means out of all the real fraud cases, the model catches 90% of them.

Precision: The low precision of 1% indicates that out of all the transactions that the model predicts as fraudulent, only 1% of them are actual frauds. This will lead to a large number of false alarms, potentially causing inconvenience to customers and increasing the workload for fraud detection teams.

While the model is excellent at flagging potential frauds (high recall), it has a significant drawback of raising too many false alerts (low precision). This model might be too cautious, flagging many non-fraudulent transactions as suspicious.

Random Forest

Recall & Precision: Achieving a precision of 61% and a recall of 93% for the fraudulent class means that the Random Forest model is both precise and sensitive. It can detect a high percentage of actual fraud cases while keeping the number of false alarms relatively low.

ROC-AUC Score: A high score of 0.9632 indicates the model's capability to distinguish between the positive and negative classes effectively.

Random Forest has a balanced performance, making it a strong candidate for deployment. It will catch most frauds while minimizing customer disturbance due to false alerts.

Decision Trees

Recall & Precision: While the recall is high (94%), the precision is at 56%. It means the model correctly identifies most of the frauds but at the expense of a higher number of false positives compared to Random Forest.

Decision Trees provide a transparent model which can be easily interpreted and explained. However, its performance, while still good, is slightly overshadowed by the Random Forest.

Bagging Classifier

Recall & Precision: The recall is impressively high at 97%, but the precision is very low at 3%. This means the model is overzealous in flagging transactions as fraudulent, leading to a lot of false positives.

The Bagging Classifier is aggressive in its approach to detect fraud, which might be beneficial if the cost of missing a fraud is very high. However, the trade-off is a higher number of regular transactions getting flagged, which can lead to customer dissatisfaction.

The graph below provides a visual comparison of the different models we've employed. By illustrating the performance metrics of each model side by side, this visualization aids in discerning the relative efficacy of Logistic Regression, Random Forest, Decision Trees, and the Bagging Classifier in the context of our dataset. Through this visual representation.

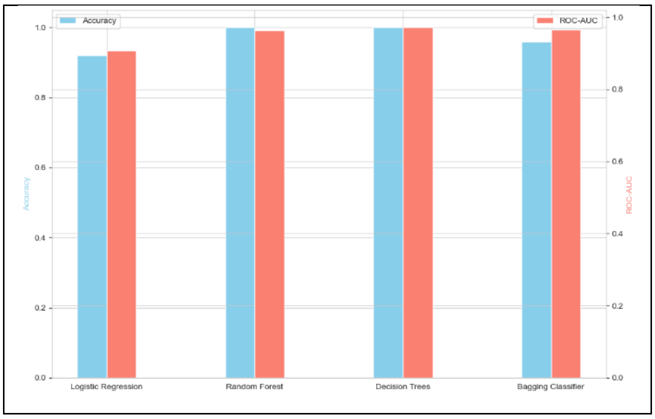
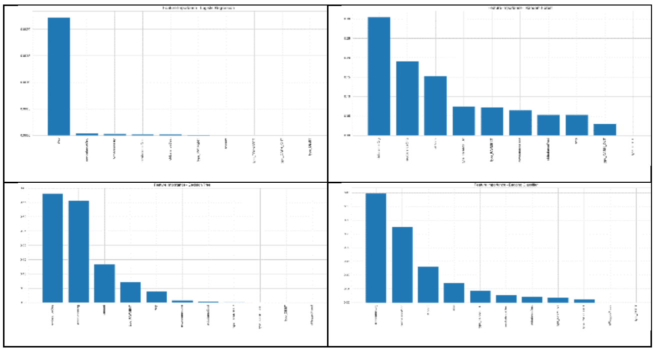


Figure 5 Comparison of the models

While each model has its strengths and weaknesses, the Random Forest stands out as the most balanced in terms of precision and recall. In a real-world scenario, the choice of the model would also consider factors like computational efficiency, interpretability, and the specific costs associated with false positives and false negatives. The high number of false positives from models like Logistic Regression and Bagging Classifier could lead to potential customer dissatisfaction, increased operational costs in verifying each alert, and potential loss of trust in the system's alerts. On the other hand, missing a fraudulent transaction (false negative) could result in financial losses and damage to reputation. Hence, it's a delicate balance that needs to be struck.

F. *Evaluation and Comparison*

In the realm of machine learning, understanding the significance of individual features can provide insights into the model's decision-making process. A feature's importance indicates how often a particular feature played a pivotal role in splitting the data, and by what magnitude it influenced the model's predictions.



From the visualizations, specific conclusions can be drawn about which features play a dominant role in determining the legitimacy of a transaction. These insights not only affirm the relevance of certain transaction attributes but also guide future data collection, preprocessing, and feature engineering efforts. For financial institutions, understanding these key features can also inform preventive measures and policy changes to thwart fraudulent activities more effectively.

V. Conclusions and Future Work

Throughout this study, the application of machine learning techniques has been explored, specifically Logistic Regression, Random Forest, Decision Trees, and the Bagging Classifier, to detect fraudulent transactions in online payments. The findings of this project are summarized as follows.

Random Forest emerged as a particularly robust model, demonstrating high precision, recall, and an outstanding ROC-AUC score. This suggests its adeptness at capturing intricate transactional patterns and its potential as a primary tool for fraud detection.

Logistic Regression, while offering a decent recall, indicated a lower precision for detecting fraudulent transactions. This implies a higher number of false positives, which might not be ideal in real-world scenarios where falsely flagging legitimate transactions can lead to customer dissatisfaction.

Both Decision Trees and the Bagging Classifier showcased reasonable performance, with Decision Trees slightly edging out in terms of precision.

The importance of feature engineering and preprocessing, including handling imbalanced datasets, was evident in improving model performance.

Given more time, deeper investigation into creating and refining features based on domain knowledge would be applied, potentially uncovering more nuanced patterns indicative of fraud. Advanced algorithms, such as Neural Networks or Gradient Boosting Machines, weren't explored in this study and might provide better insights or performance. The study was retrospective. In a real-world scenario, implementing these models in a real-time fraud detection system would be the ultimate test of their effectiveness.

In addressing our research questions, we've ascertained that machine learning algorithms can indeed be potent tools in detecting fraudulent transactions. The patterns and features extracted from the dataset, especially when combined with these algorithms, can significantly enhance fraud detection capabilities. The key implication of our findings underscores the potential of machine learning, and particularly ensemble methods like Random Forest, in revolutionizing the domain of online transaction security. However, as with all models, continuous refinement and validation are essential for maintaining their effectiveness in ever-evolving fraud landscapes.

REFERENCES

[1] Sadineni, P. K. (2020, October). Detection of fraudulent transactions in credit card using machine learning algorithms. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 659-660). IEEE.

[2] Ba, H. (2019). Improving detection of credit card fraudulent transactions using generative adversarial networks. arXiv preprint arXiv:1907.03355

[3] Dhankhad, S., Mohammed, E., & Far, B. (2018, July). Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. In 2018 IEEE international conference on information reuse and integration (IRI) (pp. 122-125). IEEE.

[4] Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using machine learning. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 488-493). IEEE.

[5] Zhang, K., Zhang, Y., & Yang, Y. (2016). Credit Card Fraud Detection Based on Random Forest and SMOTE. Journal of Physics: Conference Series, 716(1), 012105