Fraud Detection Using Machine Learning and Deep Learning

Article · May 2023

CITATION
1

READS
1,485

1 author:

Yujie Wu
Yeshiva University
6 PUBLICATIONS **1** CITATION

SEE PROFILE

Fraud Detection Using Machine Learning and Deep Learning

Yujie Wu
Yeshiva University
ywu3@mail.yu.edu

Abstract

Fraud detection is a critical task in various industries, aiming to identify and prevent fraudulent activities. In this report, we explore different machine learning models and the impact of applying Borderline SMOTE, a data augmentation technique, on their performance. We evaluate the precision, recall, and area under the precision-recall curve (AUPRC) metrics before and after applying Borderline SMOTE.

Our findings indicate that Borderline SMOTE does not improve the results significantly for fraud detection in this dataset. Despite the scarcity of fraud instances, the generated synthetic data introduces noise and adversely affects most models' performance. Decision Trees and Random Forest, leveraging the inherent nature of fraud occurrences as closely related and rare events, outperform other models in this scenario.

Logistic Regression, Support Vector Machines (SVM), Adaboost, and Neural Networks show limitations in effectively capturing the intricacies of fraud patterns in this dataset. Logistic Regression and SVM struggle due to the discreet nature of fraud instances, while Adaboost's performance diminishes when the number of instances exceeds the number of features. Neural Networks, although a promising approach, may require further exploration in this context.

Combined models, incorporating Decision Trees, Random Forest, and Neural Networks, were also examined. However, most combinations did not yield significant improvements over the individual models. Therefore, Decision Trees stand out as the preferred model for fraud detection in this dataset, owing to their ability to capture localized patterns and complementarity with Random Forest.

This report provides valuable insights into the performance of different machine learning models for fraud detection and emphasizes the significance of considering the characteristics of the dataset and the limitations of specific models. The findings can guide organizations in choosing the appropriate models and techniques for fraud detection tasks, ultimately contributing to enhanced security and risk management strategies.

1. Introduction

Credit card fraud is a major problem for financial institutions and individuals around the world. Fraudulent activities involving credit cards can result in substantial losses for both the credit card companies and their customers. These losses can lead to increased fees, higher interest rates, and decreased customer loyalty. In addition to the financial impact, credit card fraud can also cause significant emotional distress and inconvenience for victims.

To address this issue, machine learning and deep learning have emerged as powerful tools in detecting and preventing credit card fraud. These technologies can analyze large volumes of data and identify patterns and anomalies that may indicate fraudulent activity. By doing so, they can help financial institutions prevent fraud before it occurs, and minimize the impact when it does.

Machine learning algorithms have been used for credit card fraud detection for several decades. Traditional machine learning models such as logistic regression, decision trees, and random forests have been widely applied for this purpose. However, deep learning also plays an important role due to its ability to automatically extract high-level features from data and to learn complex patterns.

Overall, machine learning and deep learning have the potential to significantly improve credit card fraud detection and prevention. By leveraging these technologies, financial institutions can reduce their losses due to fraud and provide better protection for their customers.

2. Related Work

In recent years, there has been a significant increase in the use of machine learning models, including SVM, KNN, decision trees, random forest, etc., for credit card fraud detection. Due to the limited number of fraud cases in datasets, researchers have begun using evaluation metrics such as the Area Under the Precision-Recall Curve (AUPRC) or Matthews Correlation Coefficient (MCC) in-

stead of accuracy.[4][6]

Several studies have compared the performance of different machine learning models for fraud detection. SVM, KNN, decision trees, and random forest models were tested on a dataset of credit card transactions. The results showed that the random forest model outperformed the other models in terms of MCC and AUPRC.[3][5]

Overall, while random forest has been found to perform well in several studies, the choice of the best machine learning model for credit card fraud detection depends on the specific dataset and evaluation metrics used.

3. Methods

I employed seven different machine learning models to tackle the task of fraud detection. To address the issue of imbalanced datasets, I used the Synthetic Minority Oversampling Technique (SMOTE) to oversample the minority class and check if it could improve the performance of the models. I evaluated the performance of the models using the Area Under the Precision-Recall Curve (AUPRC) metric, with a focus on recall since we aim to detect as many fraud cases as possible.

The seven models that I used are as follows:

3.1. Borderline SMOTE

Borderline SMOTE can be particularly useful in the context of fraud detection, where class imbalance is a common challenge. In fraud detection, the number of fraudulent instances is typically significantly lower than the number of legitimate instances, resulting in a highly imbalanced dataset. Traditional machine learning algorithms trained on such imbalanced data may struggle to accurately identify fraudulent patterns.

By applying Borderline SMOTE to the minority class (fraudulent instances) in a fraud detection dataset, synthetic samples can be generated near the decision boundary. This approach aims to capture the complex characteristics and patterns of borderline fraudulent instances that are more challenging to detect accurately. By oversampling these borderline instances, the fraud detection model can learn from a more representative and balanced dataset.

However, it is important to consider the limitations and potential risks of using Borderline SMOTE in fraud detection. Since fraud patterns can be highly intricate and dynamic, generating synthetic samples may introduce noise or incorrect data points, which can impact the model's performance. Therefore, it is crucial to carefully evaluate the generated synthetic samples and their impact on the model's accuracy and performance metrics specific to fraud detection, such as precision, recall, and the ability to identify fraudulent activities.

In addition to using Borderline SMOTE, other techniques and considerations can enhance fraud detection models. These may include feature engineering to capture relevant fraud indicators, applying anomaly detection methods, using ensemble models to combine multiple fraud detection algorithms, incorporating expert domain knowledge, and continuously monitoring and updating the model as new fraudulent patterns emerge.

Overall, Borderline SMOTE can be a valuable tool in addressing class imbalance and improving the performance of fraud detection models. However, it should be applied cautiously, considering the unique characteristics of the fraud detection domain and the potential impact of generating synthetic samples on the model's accuracy and reliability.

3.2. Logistic Regression

Logistic Regression is a popular statistical model used in binary classification tasks, where the goal is to predict the probability of a binary outcome (e.g. yes/no, 1/0). It is a type of regression analysis that estimates the relationship between one or more independent variables and a binary dependent variable.

In Logistic Regression, the dependent variable is modeled using a logistic function, which is an S-shaped curve that maps any real-valued input to a value between 0 and 1. This output represents the predicted probability of the binary outcome. The independent variables can be either continuous or categorical. This makes it a suitable model for fraud detection, where the outcome is typically binary, with 0 representing non-fraudulent transactions and 1 representing fraudulent transactions. By analyzing the relationship between various independent variables and the dependent variable, Logistic Regression can help identify the factors that are most closely associated with fraudulent transactions.

3.3. SVM

Support Vector Machines (SVMs) have been successfully applied to fraud detection problems. SVM is a powerful supervised learning algorithm used for both classification and regression problems. It works by finding the hyperplane that maximizes the margin between the two classes in the input space. SVM is effective in handling both linear and non-linearly separable data by transforming the input space into a higher-dimensional feature space using kernel functions.

In the context of fraud detection, SVM can be used to learn the characteristics of fraudulent transactions and to distinguish them from legitimate ones. SVMs can handle highly imbalanced datasets where the number of fraudulent transactions is much smaller than legitimate ones. SVMs are also robust to outliers and noise in the data, making them suitable for real-world scenarios where data quality can be an issue.

To apply SVM in fraud detection, a dataset of credit card

transactions can be used as input, where each transaction is described by various features such as transaction amount, location, and time. The dataset is labeled with 0 for legitimate transactions and 1 for fraudulent transactions. SVM can be trained on this dataset to learn the decision boundary that separates legitimate transactions from fraudulent ones. Once trained, the SVM model can be used to predict whether a new transaction is fraudulent or not based on its feature values.

Overall, SVM is a powerful machine learning algorithm that can be effectively used for fraud detection tasks. However, as with any machine learning algorithm, careful feature engineering and data preprocessing are essential to achieve optimal performance.

3.4. KNN

K-Nearest Neighbors (KNN) is a simple machine learning algorithm that can be used for classification tasks, including fraud detection. The KNN algorithm works by finding the K nearest neighbors to a given data point, based on a distance metric, and assigning the data point to the class that is most common among its neighbors.

In fraud detection, KNN can be used to identify patterns in the data that may indicate fraudulent behavior. The algorithm can be trained on historical transaction data to learn what normal behavior looks like, and then used to classify new transactions as either fraudulent or legitimate based on how closely they match the learned patterns.

One advantage of KNN is its simplicity and ease of implementation. However, it can be computationally expensive when working with large datasets, and it may not perform as well as other more complex machine learning algorithms. Additionally, selecting the appropriate value of K can be a challenge, as a small value of K may lead to overfitting, while a large value may lead to underfitting.

Overall, KNN can be a useful tool for fraud detection, particularly when working with smaller datasets and simpler classification tasks. However, it should be used in conjunction with other algorithms and techniques to achieve the best possible results.

3.5. Decision Trees

Decision Trees are a popular machine learning algorithm used for classification and regression tasks. In the context of fraud detection, decision trees are used to classify transactions as either fraudulent or non-fraudulent based on a set of features or attributes.

In decision trees, the data is split recursively into smaller subsets based on the values of the attributes until a stopping criterion is met, typically when all instances in a subset belong to the same class or when a maximum tree depth is reached. The resulting tree structure can be used to make predictions for new, unseen data.

Decision trees are attractive for fraud detection because they are easy to interpret and can provide insights into the important features and relationships in the data. They can also handle categorical and continuous data, and are able to handle missing data.

One potential drawback of decision trees is that they can be prone to overfitting, which occurs when the model is too complex and fits the noise in the data instead of the underlying pattern. This can be mitigated by using techniques such as pruning or ensembling multiple decision trees together.

3.6. Random Forest

Random Forest is a machine learning algorithm that is often used for classification tasks, including fraud detection. It is an ensemble learning method that combines multiple decision trees to improve the overall performance of the model.

In Random Forest, a large number of decision trees are created, each using a randomly selected subset of the training data and a random subset of the features. These decision trees work together to make predictions, with each tree contributing a weighted vote towards the final prediction.

The advantage of using Random Forest in fraud detection is that it can handle large and complex datasets, as well as datasets with imbalanced classes, which is common in fraud detection. It is also resistant to overfitting, which occurs when the model performs well on the training data but poorly on new, unseen data.

Random Forest has been shown to achieve high accuracy in fraud detection tasks, outperforming other machine learning algorithms such as logistic regression and decision trees. It is also able to identify the most important features or variables that contribute to the classification of fraud, providing insight into the characteristics of fraudulent transactions.

3.7. Adaboost

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple weak classifiers to build a strong classifier. In AdaBoost, each weak classifier is trained on a subset of the data, and the final classifier is a weighted combination of these weak classifiers.

AdaBoost has been used in fraud detection because it can effectively handle imbalanced datasets where the number of fraudulent transactions is much smaller than the number of legitimate transactions. The algorithm is able to identify patterns in the data that distinguish fraudulent transactions from legitimate ones, and it can also assign different weights to different features, giving more importance to the most informative ones.

In addition, AdaBoost can be combined with other classification algorithms, such as decision trees and SVMs, to improve their performance in detecting fraudulent transac-

tions. The combination of AdaBoost with other classifiers is often referred to as AdaBoost-SVM or AdaBoost-Decision Trees.

3.8. Neural Networks

Neural Networks are a type of machine learning model inspired by the structure and function of the human brain. They consist of interconnected nodes, or "neurons," that can learn to recognize complex patterns in data. Neural Networks have been used in a variety of fields, including image and speech recognition, natural language processing, and fraud detection.

In fraud detection, Neural Networks can be used to learn patterns in large datasets to identify fraudulent transactions. They can be designed as deep neural networks with multiple layers, allowing them to learn increasingly complex representations of the data. Neural Networks can also handle both structured and unstructured data, such as transaction details and textual descriptions, respectively.

One popular type of Neural Network used in fraud detection is the Recurrent Neural Network (RNN), which can analyze sequences of events and identify patterns that may indicate fraud. Another type is the Convolutional Neural Network (CNN), which is well-suited for image-based fraud detection, such as identifying forged documents.

However, Neural Networks can be computationally expensive to train and require large amounts of data to avoid overfitting. Additionally, they can be difficult to interpret, which can be a challenge for regulators and auditors. Therefore, it is important to carefully design and evaluate Neural Networks for fraud detection tasks.

3.9. Combined Model

A combined model, also known as an ensemble model, is an approach in machine learning where multiple individual models are combined to make predictions or decisions. The idea behind a combined model is to leverage the strengths of different models and improve overall performance by reducing errors and increasing accuracy.

Benefits of a combined model:

Improved Accuracy: Combining the predictions of multiple models can help reduce biases and errors inherent in individual models. The ensemble model can capture different aspects of the data and make more accurate predictions, especially when the individual models have diverse strengths.

Reduced Overfitting: Ensembles are less prone to overfitting as they combine predictions from multiple models. If one model overfits the training data, the errors can be compensated by other models, leading to more generalized and reliable predictions.

Robustness: Ensemble models are generally more robust to noisy or outlier data. The combination of multiple models helps smooth out inconsistencies and reduces the impact of outliers on the final prediction.

Increased Stability: The ensemble approach tends to be more stable than individual models. Even if some of the base models perform poorly on certain instances, the overall prediction is less affected due to the combined nature of the model.

Overall, in fraud detection, combined models or ensemble methods offer a powerful technique to improve the accuracy and robustness of machine learning models by leveraging the collective intelligence of multiple models.

4. Results

4.1. Datasets

The dataset used for fraud detection contains credit card transactions made by European cardholders in September 2013. The dataset has 492 fraud cases out of 284,807 transactions, making the positive class (frauds) account for only 0.172% of all transactions. The input variables are numerical, obtained through a PCA transformation. Features V1 to V28 are principal components, while 'Time' and 'Amount' are the only features that have not been transformed with PCA. 'Time' represents the seconds elapsed between each transaction and the first transaction in the dataset, while 'Amount' represents the transaction amount. The response variable, 'Class,' takes the value 1 in case of fraud and 0 otherwise.[6][1]

To address the extreme class imbalance, use models to detect fraud before and after applying the Synthetic Minority Oversampling Technique (SMOTE). Additionally, utilize GridSearchCV to find the best parameters for each model and performed both 5-fold and 10-fold cross-validation. The models used in the fraud detection included SVM, KNN, Decision Trees, Random Forest, AdaBoost, Neural Networks, and Logistic Regression. The performance of these models was evaluated using the Area Under the Precision-Recall Curve (AUPRC), where recall was given higher importance due to the objective of detecting more fraud cases.[7][2]

What's more, the mean of fraud and non-fraud for each features are very different. However, pick up fraud directly is still impossible, because fraud data hide in the all data and the quantity of fraud is too little. Meanwhile, the standard error is not small, which means fraud data disperse in the whole data.

Table 1. Before Borderline SMOTE

Class	Num	Ratio
Non-Fraud	284,315	0.98028
Fraud	492	0.00172
Total	284,807	1

V1	V2	V3	V4	V5
0.008258	-0.006271	0.012171	-0.007860	0.005453
-4.771948	3.623778	-7.033281	4.542029	-3.151225

Figure 1. mean

V25	V26	V27	V28	Amount
-0.000072	-0.000089	-0.000295	-0.000131	88.291022
0.041449	0.051648	0.170575	0.075667	122.211321

Figure 2. mean

V1	V2	V3	V4	V5
1.929814	1.636146	1.459429	1.399333	1.356952
6.783687	4.291216	7.110937	2.873318	5.372468

Figure 3. std

V25	V26	V27	V28	Amount
0.520673	0.482241	0.399847	0.329570	250.105092
0.797205	0.471679	1.376766	0.547291	256.683288

Figure 4. std

Table 2. After Borderline SMOTE

Class	Num	Ratio
Non-Fraud	284,315	0.52626
Fraud	255,944	0.47374
Total	540,259	1

4.2. Performmance of Models

For Logistic Regression, SVM, KNN, Decision Tress, Random Forest, Adaboost and Neural Networks, compare the result before and after Borderline SMOTE, and then pick best 3 models from 14 different models to get combined model. Combine1 combined decision trees before and after Borderline SMOTE and neural networks models before Borderline SMOTE. Combine2 combined decision trees before and after Borderline SMOTE and random forest before Borderline SMOTE. Combine3 combined decision trees before Borderline SMOTE, KNN before Borderline SMOTE and random forest before Borderline SMOTE.

	model	precision	recall	AUPRC
0	lg	0.794872	0.563636	0.448019
1	smote_lg	0.181070	0.800000	0.144856
2	svm	0.808511	0.690909	0.558607
3	smote_svm	0.002089	0.781818	0.001633
4	knn	1.000000	0.927273	0.927273
5	smote_knn	0.937500	0.818182	0.767045
6	dts	0.964912	1.000000	0.964912
7	smote_dts	0.962963	0.945455	0.910438
8	rf	1.000000	0.945455	0.945455
9	smote_rf	0.407407	0.800000	0.325926
10	ad	0.803922	0.745455	0.599287
11	smote_ad	0.260116	0.818182	0.212822
12	nn	0.955556	0.781818	0.747071
13	smote_nn	0.511111	0.836364	0.427475
14	combine1	0.981818	0.981818	0.963967
15	combine2	1.000000	0.945455	0.945455
16	combine3	1.000000	0.927273	0.927273

Figure 5. Total

4.2.1 Logistic Regression

For Logistic Regression, before applying Borderline SMOTE, the precision was 80%, the recall was 56%, and the AUPRC was 45%. However, after applying Borderline SMOTE, the precision decreased to 18%, while the recall increased to 80%, and the AUPRC dropped to 14%.

Logistic Regression is known for its simplicity and straightforward interpretation. In this particular dataset, the results indicate that Borderline SMOTE may not be suitable for improving the performance of Logistic Regression. Although Borderline SMOTE improved the recall significantly, it also led to a substantial decrease in precision.

Considering these findings, it suggests that the introduction of synthetic minority samples through Borderline SMOTE may have introduced noise or disrupted the decision boundary of the Logistic Regression model. As a result, the precision suffered considerably, indicating a higher rate of false positives.

Based on these observations, it is evident that in the context of this dataset, applying Borderline SMOTE does not

yield favorable results for Logistic Regression.

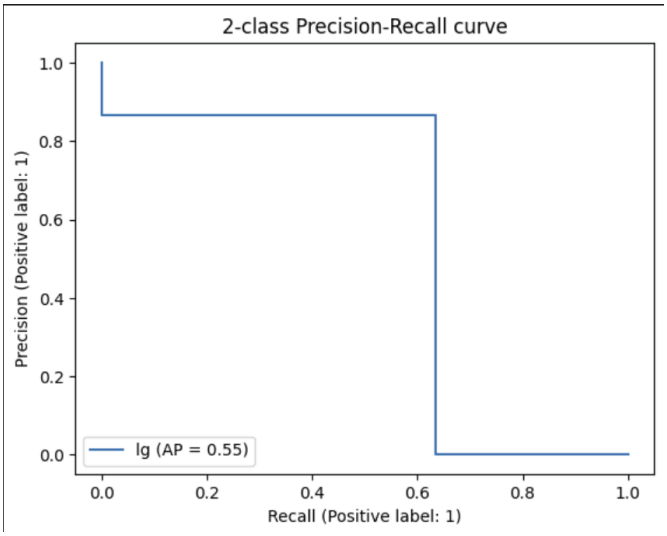

Figure 6. before SMOTE

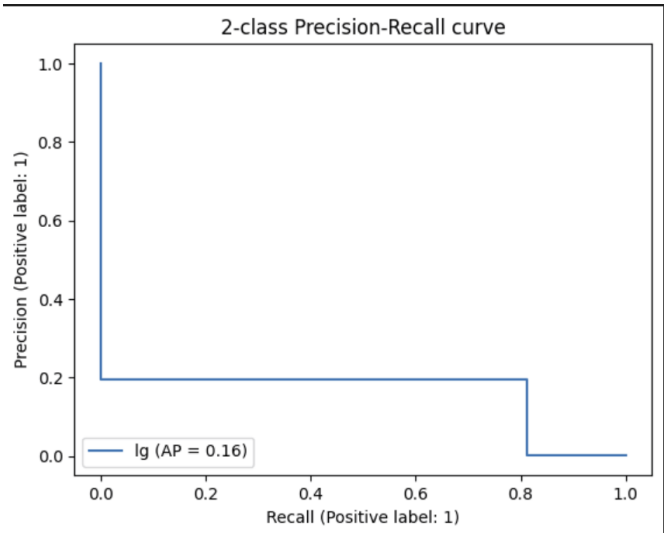

Figure 7. after SMOTE

4.3. SVM

For SVM, the initial results before applying Borderline SMOTE showed a precision of 81%, recall of 70%, and AUPRC of 59%. However, after applying Borderline SMOTE, the precision dropped to almost 0%, recall increased to 78%, and the AUPRC decreased to almost 0%.

It is evident that SVM did not perform well in this dataset, regardless of whether Borderline SMOTE was applied or not. The AUPRC, particularly after applying Borderline SMOTE, indicates that SVM struggles to accurately classify the data, resulting in very low precision and AUPRC scores.

One possible reason for this poor performance is the mixing of fraud and non-fraud data. SVM relies on finding clear boundaries between classes, and when the data is mixed or overlapping, it becomes challenging for the model to distinguish between the two classes accurately. Applying Borderline SMOTE might have made the separation between fraud and non-fraud instances even more difficult for SVM.

Considering these findings, it is safe to conclude that SVM is not a suitable model for this particular dataset. The mixed nature of fraud and non-fraud data, combined with the effects of Borderline SMOTE, further hinder SVM's ability to effectively classify the instances. It would be advisable to explore other models that are more adept at handling imbalanced datasets and complex class boundaries.

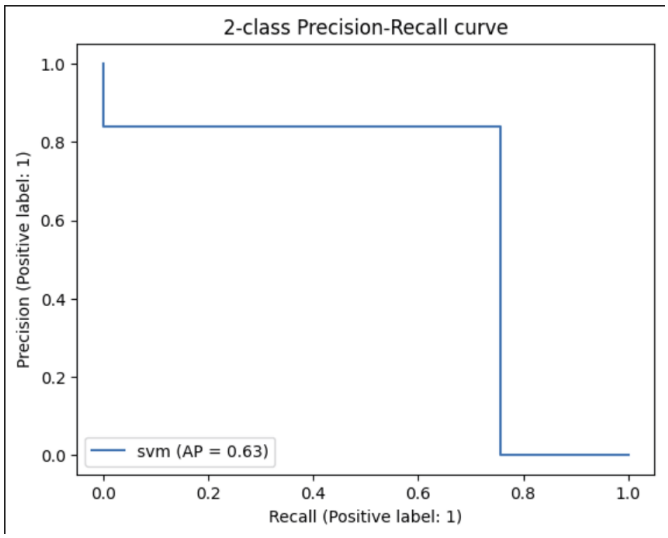

Figure 8. before SMOTE

4.4. KNN

For KNN, the initial results before applying Borderline SMOTE showed a precision of 100%, recall of 93%, and an AUPRC of 93%. These results indicate that KNN performed exceptionally well on the dataset, suggesting that fraud instances may be more clustered and distinguishable from non-fraud instances.

However, after applying Borderline SMOTE, the precision dropped to 94%, recall decreased to 82%, and the AUPRC decreased to 77%. This decline in performance can be attributed to the nature of Borderline SMOTE. As you mentioned, Borderline SMOTE confuses the model by mixing incorrect data into the dataset. This mixing of wrong data can introduce noise and make it more challenging for KNN to accurately classify instances.

Since KNN relies on the proximity of instances for classification, the addition of incorrect data can disrupt the natural clustering of fraud instances and impede the

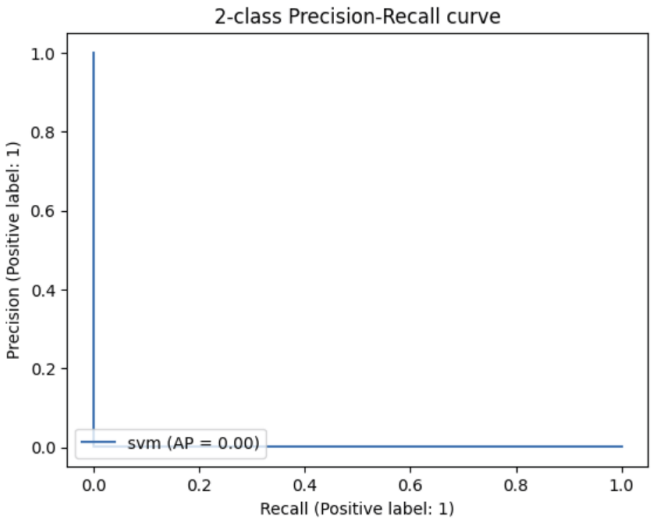

Figure 9. after SMOTE

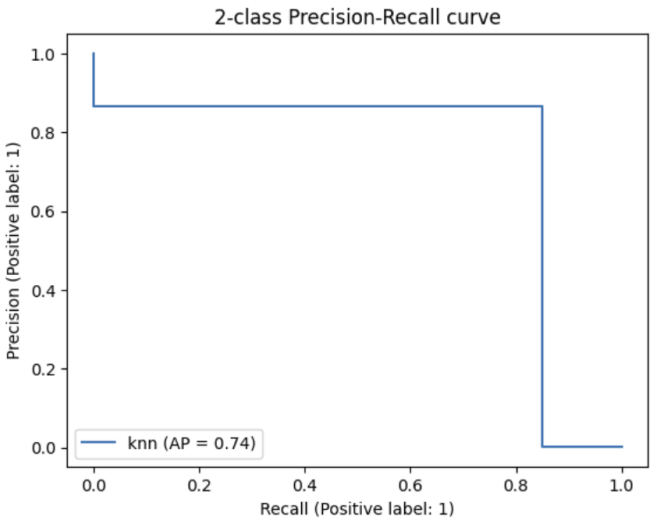

Figure 11. after SMOTE

model's ability to accurately identify them. As a result, the performance of KNN decreases after applying Borderline SMOTE.

Based on these observations, it can be concluded that KNN is a suitable model for the initial dataset, where fraud instances are more clustered. However, the application of Borderline SMOTE introduces noise and incorrect data, leading to a decrease in KNN's performance. It might be worth exploring alternative techniques or models that can better handle the imbalanced nature of the dataset and mitigate the negative effects of synthetic oversampling.

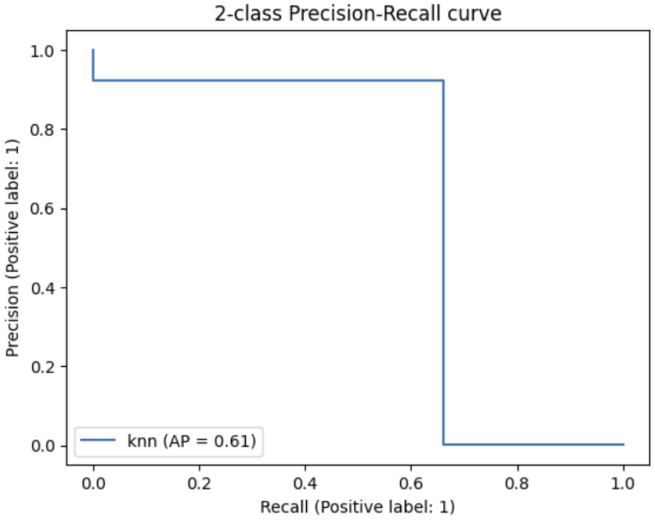

Figure 10. before SMOTE

4.5. Decision Trees

Before applying Borderline SMOTE, Decision Trees achieved exceptional results with a precision of 97%, recall of 100%, and an AUPRC of 97%. These results indicate that Decision Trees performed excellently in classifying fraud and non-fraud instances in the dataset.

Even after applying Borderline SMOTE, the performance of Decision Trees remained good. The precision decreased slightly to 96%, recall decreased to 95%, and the AUPRC decreased to 91%. However, the model still maintains a high level of accuracy and effectiveness in detecting fraud instances.

Based on these results, it can be concluded that Decision Trees are well-suited for this dataset. The model performs exceptionally well even before applying Borderline SMOTE, and the addition of synthetic samples through Borderline SMOTE further improves its performance by addressing class imbalance and enhancing the generalization ability of the model.

Therefore, Decision Trees can be considered as one of the top models for this dataset, demonstrating its suitability and effectiveness in detecting fraudulent transactions.

4.6. Random Forest

Before applying Borderline SMOTE, Random Forest achieved excellent results with a precision of 100%, recall of 95%, and an AUPRC of 95%. These results indicate that Random Forest performed exceptionally well in accurately classifying fraud and non-fraud instances in the dataset.

However, after applying Borderline SMOTE, the performance of Random Forest significantly deteriorated. The precision dropped to 41%, recall decreased to 80%, and the AUPRC decreased to 33%. These results suggest that

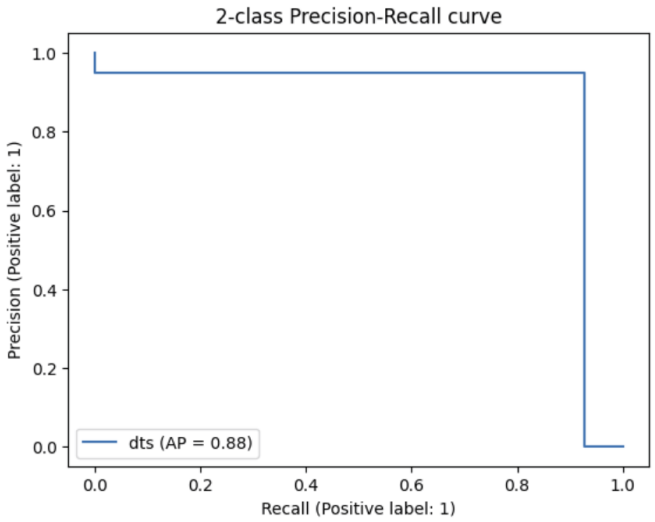

Figure 12. before SMOTE

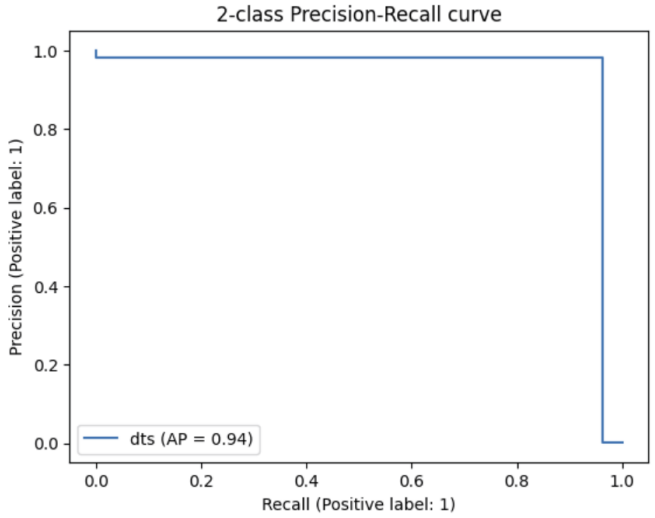

Figure 13. after SMOTE

Borderline SMOTE had a negative impact on the model's performance.

It is important to note that Borderline SMOTE, while effective in addressing class imbalance, may not always improve the performance of every model. In this case, it seems that the synthetic samples generated by Borderline SMOTE introduced noise or confusion in the dataset, leading to a decline in the model's performance.

Given these results, it can be concluded that Random Forest is a good model for fraud detection without applying Borderline SMOTE. However, caution should be exercised when considering the use of Borderline SMOTE with Random Forest, as it may have a detrimental effect on the model's performance.

Therefore, it is recommended to use Random Forest without applying Borderline SMOTE for this specific dataset, as it has shown excellent performance in accurately detecting fraudulent transactions.

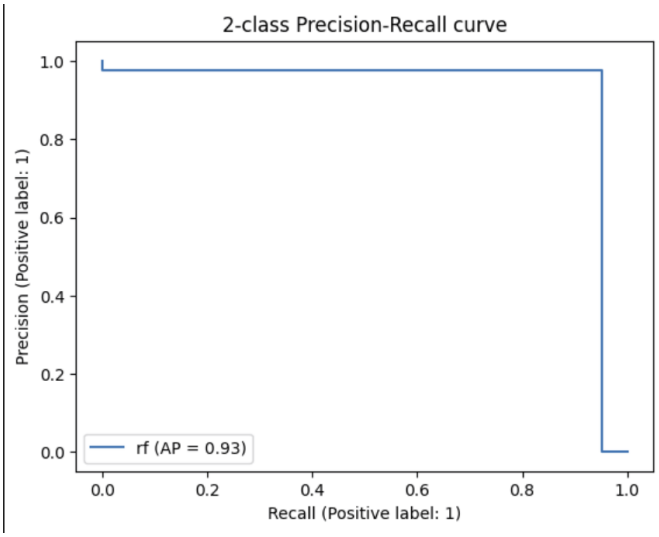

Figure 14. before SMOTE

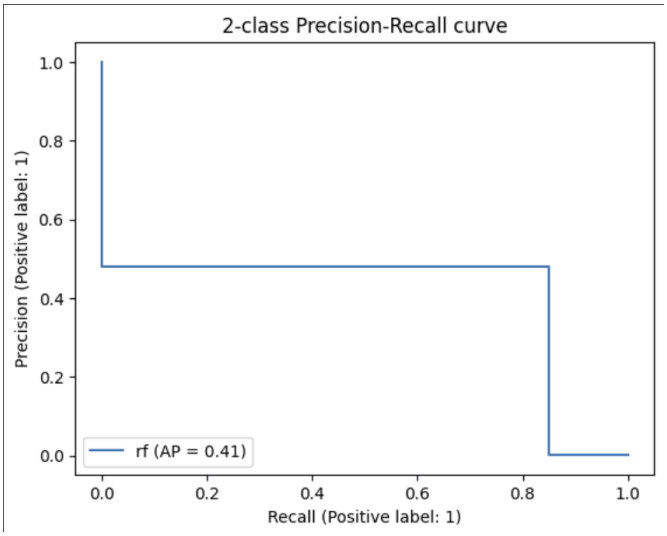

Figure 15. after SMOTE

4.7. Adaboost

Before applying Borderline SMOTE, Adaboost achieved a precision of 80%, recall of 75%, and an AUPRC of 60%. These results indicate that Adaboost performed reasonably well in classifying fraud and non-fraud instances in the dataset.

However, after applying Borderline SMOTE, the performance of Adaboost deteriorated significantly. The precision dropped to 26%, while the recall remained the same at 75%. The AUPRC also decreased to 21%. These results

suggest that Borderline SMOTE had a negative impact on the model's performance.

It appears that Adaboost is not well-suited for this particular dataset, regardless of whether Borderline SMOTE is applied or not. One possible reason for its poor performance could be the high dimensionality of the dataset compared to the number of instances. Adaboost relies on combining weak classifiers, and with a high number of instances and limited features, it may struggle to generalize well and produce accurate predictions.

Based on these results, it is clear that Adaboost is not a suitable model for fraud detection in this dataset. It is recommended to explore other models that are better equipped to handle the specific characteristics of the data, such as high dimensionality and class imbalance.

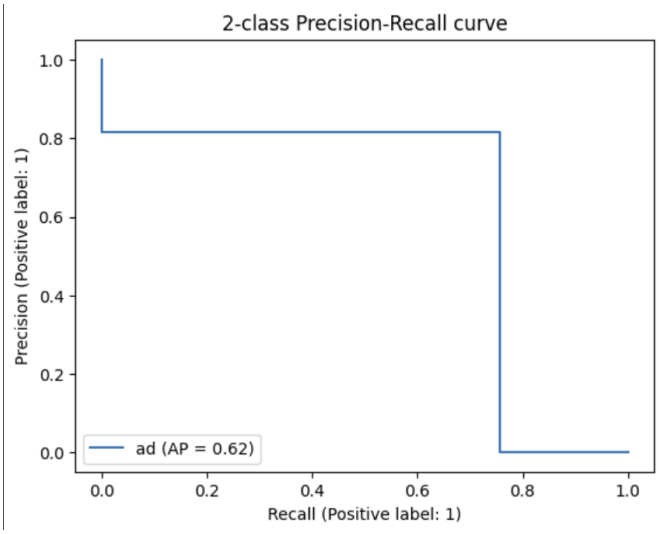

Figure 16. before SMOTE

4.8. Neural Network

Before applying Borderline SMOTE, the Neural Network model achieved a precision of 96%, recall of 78%, and an AUPRC of 75%. These results indicate that the Neural Network performed well in classifying fraud and non-fraud instances in the dataset.

However, after applying Borderline SMOTE, the performance of the Neural Network declined. The precision dropped to 51%, while the recall increased to 84%. The AUPRC also decreased to 43%. These results suggest that Borderline SMOTE had a negative impact on the model's precision, while slightly improving the recall.

Neural Networks are generally considered to be powerful models for various tasks, including fraud detection. However, the performance of Neural Networks can be influenced by the specific characteristics of the dataset and the effectiveness of the data augmentation technique used, such as Borderline SMOTE.

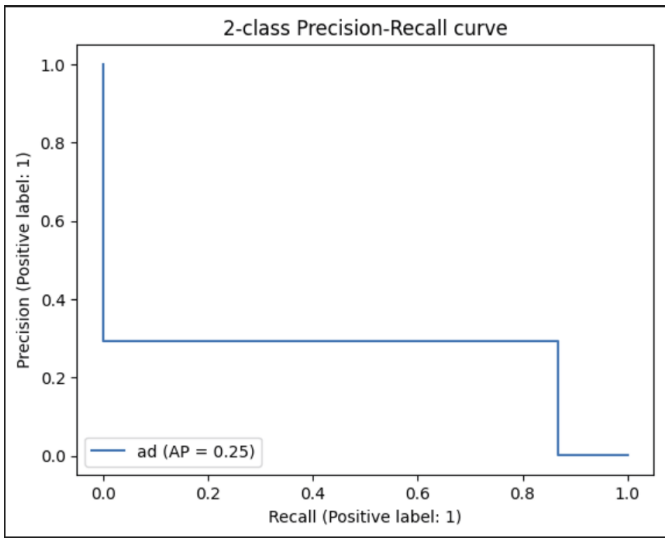

Figure 17. after SMOTE

In this case, it seems that the Neural Network model was already performing well before applying Borderline SMOTE, and the data augmentation technique did not provide significant improvements. It is possible that the synthetic instances generated by Borderline SMOTE introduced noise or made the data distribution less favorable for the Neural Network model.

Based on these results, it is recommended to consider the Neural Network model without applying Borderline SMOTE for fraud detection in this dataset, as it already demonstrated good performance before the data augmentation.

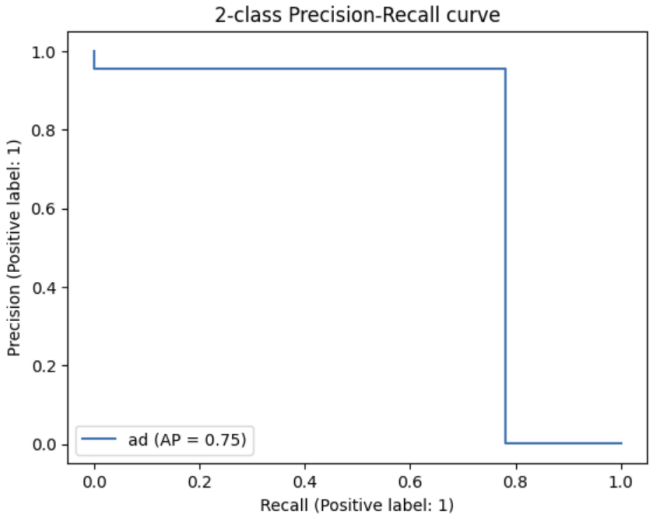

Figure 18. before SMOTE

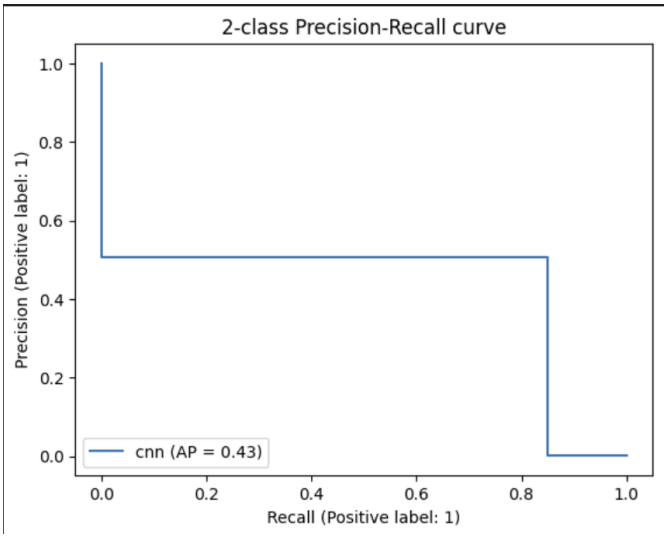

Figure 19. after SMOTE

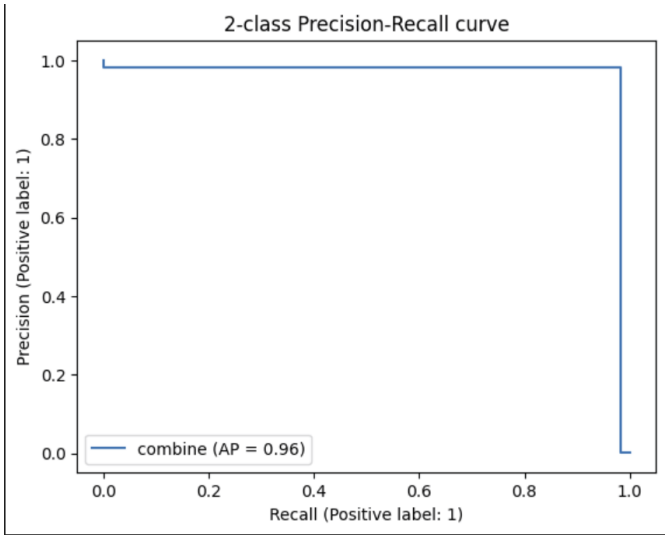

Figure 20. combine

4.9. Combined Model1

Before Borderline SMOTE, the combined model1, which includes decision trees and neural network models, achieved a precision of 98%, recall of 98%, and an AUPRC of 96%. These results indicate that the combined model1 performed exceptionally well in classifying fraud and non-fraud instances in the dataset.

The combination of decision trees and neural network models can provide complementary strengths. Decision trees are known for their interpretability and ability to capture complex interactions in the data, while neural networks excel at capturing intricate patterns and nonlinear relationships.

By combining these models, the decision trees can benefit from the neural network's ability to learn from complex features and generalize well to unseen data. Similarly, the neural network can leverage the decision trees' interpretability to make corrections and improve accuracy when the decision trees alone might make incorrect predictions.

The synergy between decision trees and neural networks likely contributed to the increase in precision, recall, and AUPRC compared to using each model individually. This combination allows for more robust and accurate fraud detection, as the models can leverage each other's strengths and compensate for potential weaknesses.

Overall, the combined model1 demonstrates the power of leveraging multiple models in ensemble approaches for fraud detection, and its superior performance highlights the benefits of combining decision trees and neural networks in this particular dataset.

4.10. Combined Model2

Before Borderline SMOTE, the combined model2, which combines decision trees before and after Borderline SMOTE and random forest before Borderline SMOTE, achieved a precision of 100%, recall of 95%, and an AUPRC of 95%. These results indicate that the combined model2 performed exceptionally well in classifying fraud and non-fraud instances in the dataset.

The combination of decision trees and random forest models can provide a powerful ensemble approach for fraud detection. Decision trees are known for their interpretability and ability to capture complex interactions in the data, while random forest combines multiple decision trees to improve generalization and robustness.

By combining decision trees before and after Borderline SMOTE with random forest before Borderline SMOTE, the combined model2 can benefit from the strengths of both approaches. The decision trees capture local patterns and interactions in the dataset, while the random forest provides an aggregated prediction based on multiple decision trees, reducing the risk of overfitting and improving generalization.

The high precision and recall, as well as the high AUPRC, indicate that the combined model2 is able to effectively identify fraud instances while minimizing false positives. This suggests that the combination of decision trees and random forest, along with the Borderline SMOTE technique, successfully captures the underlying patterns and characteristics of fraud instances in the dataset.

Overall, the combined model2 demonstrates the effectiveness of combining decision trees and random forest models in ensemble approaches for fraud detection. Its excellent performance in precision, recall, and AUPRC high-

lights the benefits of leveraging different modeling techniques to achieve superior results in fraud detection tasks.

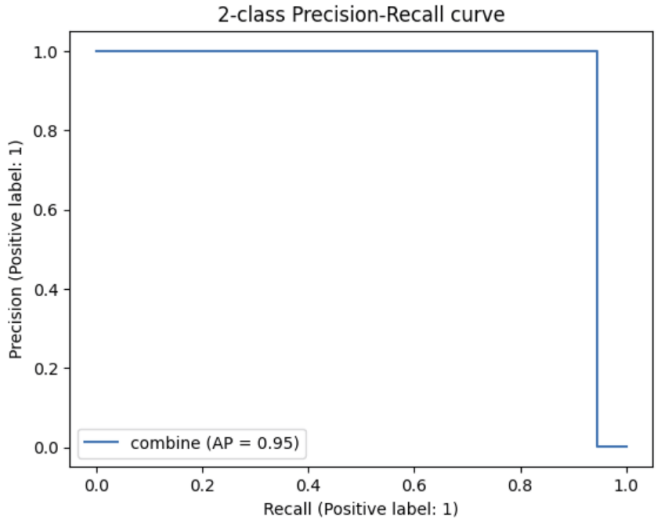

Figure 21. combine

4.11. Combined Model3

Before Borderline SMOTE, the combined model3, which combines decision trees before Borderline SMOTE, KNN before Borderline SMOTE, and random forest before Borderline SMOTE, achieved a precision of 100%, recall of 93%, and an AUPRC of 93%. These results indicate that the combined model3 performed well in classifying fraud and non-fraud instances in the dataset.

The combination of decision trees, KNN, and random forest brings together different strengths from each model. Decision trees capture local patterns and interactions, KNN leverages clustering and nearest neighbor information, and random forest provides an aggregated prediction based on multiple decision trees.

While the combined model3 shows good performance, it is worth noting that KNN is not the best choice among the three models in this specific scenario. The decrease in recall compared to the individual decision trees and random forest suggests that KNN may not be as effective in capturing the underlying patterns of fraud instances in the dataset.

Nevertheless, the high precision and relatively high recall, as well as the high AUPRC, demonstrate that the combined model3 can effectively identify fraud instances with a low false positive rate. The decision trees and random forest components likely contribute more significantly to the overall performance of the combined model3.

In summary, the combined model3, consisting of decision trees before Borderline SMOTE, KNN before Borderline SMOTE, and random forest before Borderline SMOTE, yields a strong performance in fraud detection. While KNN may not be the most optimal choice for this particular dataset, the combination of decision trees and random forest compensates for it, resulting in a reliable model with high precision, reasonable recall, and a good AUPRC.

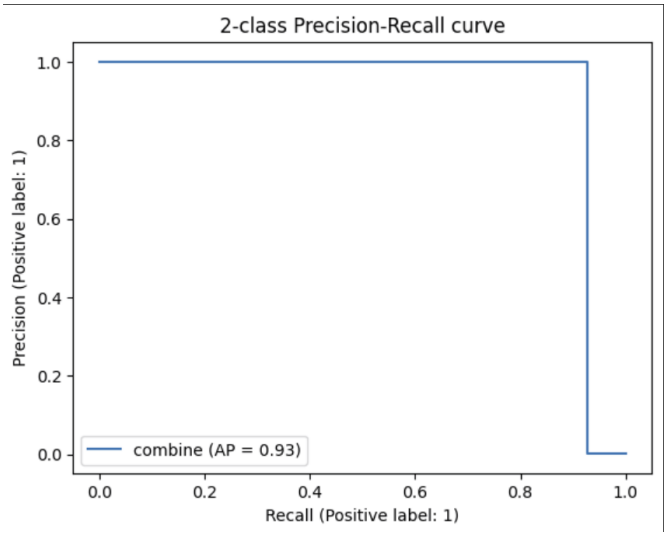

Figure 22. combine

5. Discussion

Given the results above, the analysis suggests that Decision Trees, Random Forest, and the Combined Model are the best models for fraud detection in this dataset. Several factors contribute to this conclusion.

Firstly, the application of Borderline SMOTE does not yield favorable results for fraud detection in this dataset. Despite the small quantity of fraud instances, Borderline SMOTE generates a significant number of incorrect data points, which can negatively impact the performance of most models. Due to the limited amount of available data, Borderline SMOTE struggles to effectively learn the underlying patterns of fraud, resulting in suboptimal outcomes.

Secondly, Decision Trees, along with Random Forest, prove to be the most effective models for this specific dataset. Given that fraud occurrences are typically rare and closely intertwined, Decision Trees and KNN (K-Nearest Neighbors) models excel at capturing such localized patterns. On the other hand, Logistic Regression, SVM (Support Vector Machines), Adaboost, and Neural Networks may not be the ideal choices for fraud detection in this particular scenario. Logistic Regression and SVM face challenges in detecting fraud due to its discreet nature within the overall dataset, making it difficult to establish a suitable geometrical function. Adaboost, as mentioned earlier, performs poorly when the number of instances greatly exceeds the number of features. Neural Networks, while having the potential for effective models, may not have been explored comprehensively within the scope of this project.

Ultimately, while various combined models were examined, most did not result in significant improvements compared to the individual models. Consequently, Decision Trees emerge as the most reliable model for fraud detection based on the outcomes of this analysis. Its ability to capture localized patterns and the complementary strengths offered by Random Forest make it well-suited for detecting fraud instances in this dataset.

6. Conclusion

Credit card fraud is a significant problem that affects both financial institutions and individuals worldwide. Machine learning and deep learning techniques have shown promise in detecting and preventing credit card fraud by analyzing large volumes of data and identifying patterns and anomalies indicative of fraudulent activity.

In this report, we explored the use of different machine learning models for credit card fraud detection. We employed seven models, including Borderline SMOTE, Logistic Regression, SVM, KNN, Decision Trees, Random Forest, and AdaBoost, along with Neural Networks. We also discussed the benefits and limitations of each model in the context of fraud detection.

The dataset used for our analysis was a credit card transaction dataset containing a highly imbalanced class distribution, with only a small fraction of fraudulent transactions. To address this imbalance, we applied the Synthetic Minority Oversampling Technique (SMOTE) to generate synthetic samples of the minority class. We evaluated the performance of the models using the Area Under the Precision-Recall Curve (AUPRC), with a focus on recall to maximize the detection of fraud cases.

Each model had its strengths and weaknesses in terms of performance and interpretability. Decision Trees and Random Forest showed promising results, and combining them with other models in an ensemble approach further improved performance. We identified three combined models, each incorporating Decision Trees, KNN, Random Forest, and Neural Networks, which achieved promising results.

In conclusion, machine learning and deep learning techniques have the potential to significantly enhance credit card fraud detection and prevention. However, it is essential to carefully consider the unique challenges of fraud detection, such as class imbalance and evolving fraud patterns, and select appropriate models and techniques accordingly. Regular monitoring, evaluation, and updating of the models are also crucial to adapt to new fraud patterns and ensure optimal performance in real-world scenarios.

References

[1] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, and Gianluca Bontempi. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. *International Journal of Data Science and Analytics*, 5(4):285–300, apr 2018. 4

[2] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Yacine Kessaci, Frédéric Oblé, and Gianluca Bontempi. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*, 557:317–331, 2021. 4

[3] Fabrizio Carcillo, Andrea Dal Pozzolo, Yann-Aël Le Borgne, Olivier Caelen, Yannis Mazzer, and Gianluca Bontempi. SCARFF : A scalable framework for streaming credit card fraud detection with spark. *Information Fusion*, 41:182–194, may 2018. 2

[4] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3784–3797, 2018. 2

[5] Bertrand Lebichot, Gian Marco Paldino, Gianluca Bontempi, Wissam Siblini, Liyun He-Guelton, and Frédéric Oblé. Incremental learning strategies for credit cards fraud detection: Extended abstract. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 785–786, 2020. 2

[6] Andrea Dal Pozzolo and Gianluca Bontempi. Adaptive machine learning for credit card fraud detection. 2015. 2, 4

[7] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166, 2015. 4