Credit Card Fraud Detection

Swadha Bhatt
Department of Computer
Science
Towson, University
Sbhatt8@students.towson.ed
11

***Abstract*—**

This paper addresses the critical challenge of detecting credit card fraud, a pervasive issue threatening financial security and consumer trust. I developed a hybrid model that integrates advanced data mining techniques with a sequential neural network architecture, focusing on Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, and Decision Tree Classifier, to enhance the detection of fraudulent activities. The effectiveness of our model was validated through extensive testing with confusion matrices and performance metrics such as precision, recall, and F1-score. My approach significantly improves the handling of imbalanced datasets, a common obstacle in fraud detection, by employing SMOTE for class balancing and a multilayer perceptron for complex pattern recognition. This work not only advances the technical capabilities in fraud detection but also has profound implications for improving security measures within the financial sector.

I. Introduction

In today's fast-paced digital economy, financial fraud, especially credit card fraud, presents a significant threat to financial institutions and their clients. Effective detection of fraudulent transactions is essential not only for minimizing financial losses but also for preserving the trust and satisfaction of customers. The increasingly complex and dynamic nature of fraud requires advanced analytical solutions that can swiftly adapt to new fraudulent patterns.

This analysis serves a broad audience within the financial sector, including data scientists, fraud prevention teams, and decision-makers at financial institutions. By delving into the details of our analytical methods, these stakeholders can formulate more effective strategies and enhance their fraud detection systems. Moreover, the insights from this study can aid IT teams in building more secure systems and support customer service teams in handling fraud alerts more efficiently.

Overview of Techniques

The project uses data mining and neural network techniques to tackle the challenge of fraud detection. Key techniques include:

Data Mining: I've applied various data preprocessing steps, such as feature scaling, handling missing values, and encoding categorical variables to prepare the dataset for effective analysis. Techniques like Random UnderSampling and SMOTE (Synthetic Minority Over-sampling Technique) have been utilized to address the class imbalance problem inherent in fraud detection datasets, where fraudulent transactions are much rarer than non-fraudulent ones.

Neural Networks: To model the complex patterns that characterize fraudulent transactions, we have deployed a multilayer perceptron (MLP) neural network architecture. This model includes:

- An input layer tailored to the number of features in the dataset.
- Hidden layers with ReLU (Rectified Linear Unit) activation functions to introduce non-linearity, enabling the model to learn complex patterns.
- An output layer with a softmax activation function, designed to classify inputs into either 'fraud' or 'no fraud' categories based on the learned patterns in the training phase.

The network was compiled using the Adam optimizer for efficient learning and sparse categorical crossentropy as the loss function, optimizing for classification accuracy.

Model Evaluation:

Throughout the model training process, I used a validation split to monitor the model's performance and mitigate overfitting. The performance of the model was further evaluated using confusion matrices, which provided clear insights into the model's predictive accuracy and its ability to distinguish between fraudulent and non-fraudulent transactions.

II. Literature Review

[1] Researchers in the paper "Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach" delve into the methodology of integrating various machine learning models to enhance the detection of fraudulent credit card transactions. The paper underscores the importance of meticulous data preparation and preprocessing. These initial steps, which include addressing issues like missing values, outlier detection, and feature scaling, are critical as they ensure the data fed into the models is clean and normalized.

After the data preprocessing and preparation stages, the researchers focused on feature engineering, a crucial aspect highlighted in the methodology section. Feature engineering involves selecting and crafting the right features from the data to enhance the model's ability to distinguish between fraudulent and non-fraudulent transactions effectively.

At the heart of the methodology is the development of an ensemble machine learning model that incorporates a variety of algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest, Bagging, and Boosting. To address the data's imbalance, the researchers employed a combination of sampling techniques, including Synthetic Minority Over-sampling Technique (SMOTE), to balance the dataset.The models were evaluated using comprehensive sets of metrics such as accuracy, precision, recall, and the F1-score. These metrics provided detailed insights into each model's effectiveness in detecting fraudulent transactions.

[2] In another study titled "Credit Card Fraud Detection Using Autoencoder Neural Network," the methodology involves a denoising autoencoder (DAE) paired with a deep fully connected neural network (DNN) classifier. This combination is aimed at effectively detecting fraud in credit card transactions. The data first undergoes preprocessing to ensure it is normalized and suitable for neural network application. The DAE, consisting of an encoder and a decoder, processes the data to eliminate noise, enhancing the quality for the subsequent classification by the DNN. The DNN classifier, utilizing a max activation function for binary classification, categorizes transactions into fraudulent or non-fraudulent. Metrics such as accuracy, precision, recall, and the F1-score, along with a held-out test set, are used to evaluate the model's performance.

The results section concluded with affirmations on the effectiveness of integrating denoising autoencoders with deep neural networks for fraud detection. The model not only achieved high marks in all standard performance metrics but also demonstrated a significant improvement over traditional methods, particularly in handling noisy and complex datasets typical in credit card transactions.

[3] The third paper titled "A Distributed Deep Neural Network Model for Credit Card Fraud Detection" outlines a comprehensive approach to leveraging distributed computing and deep learning to improve fraud detection across multiple financial institutions.

The paper introduces a novel Distributed Deep Neural Network (DDNN) model, designed specifically to improve the detection of credit card fraud across various financial institutions, while also prioritizing data privacy. This model uses a unique client-server setup where each bank functions as a client, independently training its own deep neural network on locally held data. These networks include an input layer, several hidden layers with ReLU activation functions, and a sigmoid output layer for binary classification. This setup allows each client to process data independently. Central to this approach is the Federated Averaging (FedAvg) algorithm used by a central server to aggregate parameters from all client models. This process mitigates issues arising from data imbalance and variations across datasets. After several iterations of local training and global updates, the collective model is tested on a separate dataset to verify its accuracy and reliability in spotting fraudulent transactions.

Distributed Deep Neural Network (DDNN) model for credit card fraud detection demonstrated its superior performance with high accuracy, precision, and recall, outperforming centralized models. The model's efficiency in processing and quick convergence is enhanced by the Federated Averaging algorithm, effectively handling data imbalances and ensuring robust fraud detection across multiple financial institutions.

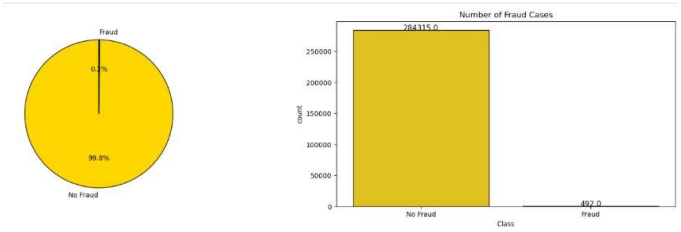
[4] Finally, the paper titled "A Convolutional Neural Network Model for Credit Card Fraud Detection" also emphasizes an extensive preprocessing phase to address class imbalance through the Adaptive Synthetic (ADASYN) sampling technique. This approach generates synthetic samples of the minority class to improve the classifier's fraud detection capability. The data is then subjected to a CNN architecture known for its pattern recognition capabilities, particularly suited for the time-series data of credit card transactions. The CNN consists of convolutional, pooling, and fully connected layers. The model's performance is assessed using accuracy, precision, recall, F1-score, and confusion matrices, showcasing high effectiveness with an accuracy of 99.82%, precision of 99.65%, and recall of 99.99%.

These papers collectively demonstrate a robust approach to tackling the challenges of credit card fraud detection through advanced machine learning and neural network techniques, significantly improving the detection capabilities and handling of complex, noisy datasets typical in credit card transactions.

III. METHODOLOGY

Exploratory Data Analysis (EDA)

The initial phase of my project involved conducting an Exploratory Data Analysis (EDA) to understand the underlying patterns and characteristics of the data. Key aspects of the EDA included:



- Statistical Summaries
- Correlation Analysis
- Visualization

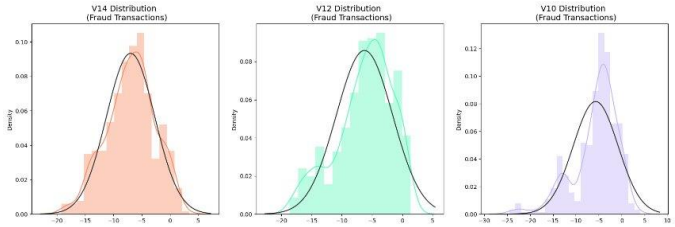
With the help of EDA, I was able to get a deeper understanding of the data characteristics. The data includes 284,807 transactions over period of two days with 492 of those being fraud transactions. The data is highly unbalanced with the positive class of frauds during a minor fraction of all the transactions. Additionally, the dataset Features are predominantly numerical and result from a Principal Component Analysis (PCA) transformation to protect user confidentiality. Only two features, 'Time' and 'Amount', are not transformed by PCA:

- Time: Seconds elapsed between each transaction and the first transaction in the dataset.
- Amount: Transaction amount, useful for cost-sensitive learning models.

Data Preprocessing

To prepare my dataset for effective modeling, several data preprocessing steps were undertaken:

- Handling Missing Values: I did not have to really handle any missing data because my dataset did not have have any missing data.
- Stratified Sampling: To handle the significant class imbalance shown in the dataset (only 0.172% fraud cases), I employed stratified sampling. This technique ensures that each split of the data contains approximately the same percentage of samples of each target class as the original set. Here's how I implemented it:

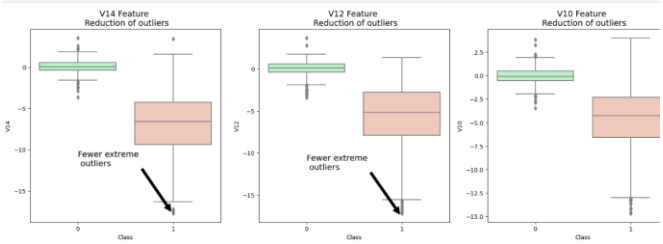


Used histograms to visually inspect the distribution of the scaled features. This helps in understanding how the scaling transformations have adjusted the distributions and in verifying the presence of any remaining outliers.

  - Utilizing `StratifiedKFold` for splitting the dataset ensures that the ratio of fraud to non-fraud transactions is consistent across both training and testing datasets. This method helps in maintaining a uniform distribution of classes, which is crucial for training models on imbalanced data.

After splitting the data using Stratified Sampling. I detected outliers and removed them.

- Outlier Detection: Using the IQR (Interquartile Range) method, I identified outliers in the dataset. This method is effective in defining thresholds (using 1.5x the IQR below Q1 and above Q3) to find extreme values which can skew the model's performance. Outlier Removal: After identifying these outliers, I removed them from the dataset to ensure a more robust and generalizable model training process.



- Dimensionality Reduction: Techniques such as Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) were used to reduce the number of variables under consideration. PCA was utilized to reduce the dimensionality of the data by transforming features into a set of linearly uncorrelated components, while t-SNE was applied to visualize high-dimensional data in two or three dimensions, providing insights into the data clustering.

These steps collectively enhance the quality of my data, addressing issues of scale, outliers, and class imbalance, which are critical for effective model training in fraud detection scenarios.

Data Mining Techniques

After the preprocessing techniques were applied to the dataset, I proceeded to implement various data mining techniques. The data was split into 80% for training and 20% for validation. The first technique applied was Logistic Regression, chosen primarily for two reasons:

1. Probability Assessment: Logistic Regression provides probabilities regarding class membership—fraud or non-fraud—which allows for an assessment of the risk level associated with each transaction.
2. Feature Impact Understanding: This technique facilitates a clearer understanding of how each feature, such as the amount of the transaction or the time of the transaction, influences the likelihood of a fraudulent occurrence. This depth of insight is invaluable for identifying critical predictors of fraud.

The second technique employed was K-Nearest Neighbors (KNN). Despite its simplicity, KNN is highly effective at localizing and clustering patterns of fraudulent transactions, making it a powerful tool for detecting anomalies.

The third technique utilized was Support Vector Classification (SVC). Chosen for its proficiency in high-dimensional spaces, SVC is ideal for handling the complex feature sets typical of credit card transaction data, which often include numerous preprocessing derivatives like PCA components.

Lastly, the Decision Tree Classifier was applied. This technique mirrors human decision-making logic and excels at illustrating non-linear relationships between features. Its ability to model complex interactions makes it particularly useful in scenarios where simpler models, such as Logistic Regression, might falter in accurately detecting fraud.

Each of these techniques was selected to leverage their unique strengths in constructing a robust model capable of identifying and predicting fraudulent activities effectively.

Evaluation of the Data Mining Models

The evaluation metrics used to assess the four data mining techniques—precision, recall, F1-score, and accuracy—are critical for understanding the effectiveness of models, especially in applications like fraud detection.

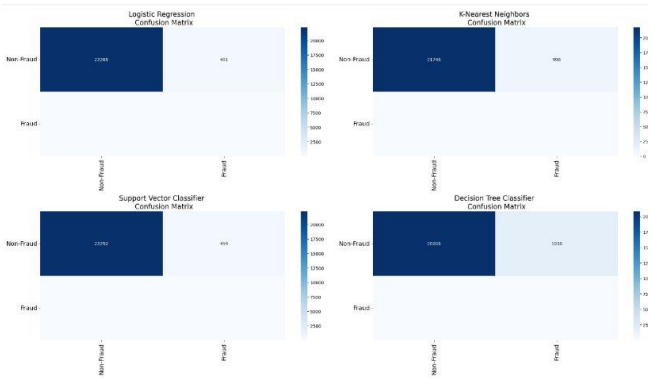
- Precision Metrics is the ratio of correctly predicted positive observations to the total predicted positives. Using the precision metric with our credit card fraud data is crucial because high precision indicates the reliability of the model in detecting actual frauds. Conversely, low precision can result in many false positives, which can lead to customer dissatisfaction.

- Recall is the ratio of correctly predicted positive observations to all observations in the actual class. This is vital in our credit card fraud analysis because high recall reflects the model's ability to capture as many actual fraud cases as possible, whereas low recall indicates that a significant number of fraudulent transactions are missed.
- F1-Score is the weighted average of precision and recall. The F1-score is particularly useful for our credit card fraud data because it provides a more balanced perspective on the model's performance than accuracy alone. Additionally, it helps to understand that both precision and recall are crucial in scenarios where both false positives and false negatives can have serious implications.
- Accuracy is the ratio of correctly predicted observations to the total observations. Accuracy is a helpful metric because it provides a quick overview of how often the model is correct in terms of both fraud and non-fraud predictions.

The results section of this paper will provide an in-depth discussion of how each model performed. However, the results indicate a common pattern across all models: high precision and recall for non-fraud transactions but significantly poorer performance for fraud transactions. The disparity in performance metrics between the two classes can be attributed to the inherent class imbalance in the dataset. To address this imbalance, the SMOTE technique was employed.

Tackling Imbalances in the Dataset

SMOTE, or Synthetic Minority Over-sampling Technique, is used to address one of the common challenges in building classification models: class imbalance. In datasets where one class significantly outnumbers another—like in credit card fraud detection, where fraudulent transactions are much rarer than non-fraudulent ones—models tend to develop a bias towards the majority class. This can lead to poor predictive performance, especially in accurately identifying the minority class, which is often the class of greater interest.



- True Positives (TP): Transactions correctly classified as fraudulent.
- True Negatives (TN):Transactions correctly classified as non-fraudulent.
- False Positives (FP):Non-fraudulent transactions incorrectly classified as fraudulent.
- False Negatives (FN):Fraudulent transactions incorrectly classified as non-fraudulent.

After the SMOTE technique was applied to the dataset, confusion matrices were computed for each of the data mining classifiers to gain better insight into each model's performance in terms of true positives, true negatives, false positives, and false negatives. From the confusion matrices, it is evident that Logistic Regression and Support Vector Classifiers perform best in terms of maintaining a high number of true negatives and low false positives. However, the Decision Tree Classifier shows potential in detecting actual fraud transactions, but this comes at the high cost of a large number of false negatives.

Additionally, cross-validation was applied to the four different classifiers to determine which one performs best. From the cross-validation results, it is clear that Logistic Regression is the most effective classifier for this particular credit card fraud dataset.

Neural Network Model

Following the data preprocessing and application of data mining techniques, a neural network model was designed and implemented to classify transactions as fraudulent or non-fraudulent:

Model Architecture

The neural network is constructed using a sequential model, which is typical for straightforward processes where a layer's output is the input to the next layer. The architecture consists of:

1. Input Layer: The first layer is a dense layer designed to receive input features directly from the dataset. The number of neurons in this layer matches the number of input features in the dataset. This setup ensures that each feature can independently contribute to the model's learning process.
2. Hidden Layers: After the input layer, the model includes at least one hidden layer, which is typical for capturing complex patterns in the data. Each neuron in these layers uses a ReLU (Rectified Linear Unit) activation function. ReLU is favored in hidden layers because it introduces non-linearity to the model, allowing it to learn more complex functions and interactions between features.
3. Output Layer: The final layer of the model is another dense layer with a softmax activation function. In a binary classification task like fraud detection, this layer would typically have two neurons, corresponding to the two class labels: fraud and no fraud. The softmax function is used to output a probability distribution over the two classes, making it easy to determine which class is more likely according to the model's prediction.

Model Complication

The compilation of the model includes several key components:

- Optimizer: I am using the Adam optimizer. It's a popular choice because it manages how quickly the model learns, adjusting the speed (or learning rate) as needed. This means the model can learn fast without missing out on important details, which helps it get better at making predictions quicker and more efficiently.
- Loss Function: For determining how well the model is doing during training, something called sparse categorical cross-entropy was used. This is perfect for the data's needs because the data involves categories (like whether a transaction is fraudulent or not) that are clearly defined and exclusive. This function helps by measuring how often the model makes the right call on which category each example belongs to.
- Metrics: To gauge how well the model is performing, accuracy of the model was looked at.This is a straightforward measure that tells me what percentage of predictions the model got right.

Model Training

The model was trained over several epochs with batch processing, using a validation split to monitor performance and mitigate overfitting. Data shuffling was incorporated to ensure that each batch and epoch would not be biased by the order of data.

Model Evaluation

After training, the model's performance was evaluated on a held-out test set using accuracy metrics and confusion matrices, providing insights into the model's ability to generalize to unseen data.

IV. Results

Data Mining Classifiers' Evaluation

The analysis employed four different classification models to detect fraud in credit card transactions: Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and Decision Tree Classifier. Each model's performance was evaluated based on precision, recall, F1-score, and support for each class (fraud and no fraud), as well as overall accuracy, macro average, and weighted average scores. The findings are summarized below:

Logistic Regression:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	22746
1	0.07	0.95	0.13	39
accuracy			0.98	22785
macro avg	0.54	0.96	0.56	22785
weighted avg	1.00	0.98	0.99	22785

KNears Neighbors:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	22746
1	0.05	0.95	0.10	39
accuracy			0.97	22785
macro avg	0.53	0.96	0.54	22785
weighted avg	1.00	0.97	0.98	22785

Support Vector Classifier:

	precision	recall	f1-score	support
0	1.00	0.98	0.99	22746
1	0.06	0.92	0.11	39
accuracy			0.98	22785
macro avg	0.53	0.95	0.55	22785
weighted avg	1.00	0.98	0.99	22785

Decision Tree Classifier:

	precision	recall	f1-score	support
0	1.00	0.97	0.99	22746
1	0.05	0.87	0.10	39
accuracy			0.97	22785
macro avg	0.53	0.92	0.54	22785
weighted avg	1.00	0.97	0.98	22785

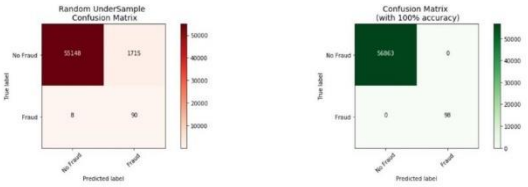
1. Logistic Regression:
   a. Precision: High precision for non-fraudulent transactions (1.00) but significantly lower for fraudulent transactions (0.07).
   b. Recall: Excellent recall for non-fraudulent (0.98) and fraudulent transactions (0.95)
   c. F1-Score: Very high for non-fraudulent transactions (0.99) but low for fraudulent ones (0.13).
   d. Accuracy: Overall accuracy of the model stood at 0.98 with a weighted average F1-score of 0.99.
2. K-Nearest Neighbors (KNN):
   a. Precision: Similar to Logistic Regression, KNN achieved high precision for non-fraudulent transactions (1.00) but low for fraudulent ones (0.05).
   b. Recall: Slightly lower recall for non-fraudulent (0.97) and similar for fraudulent transactions (0.95).
   c. F1-Score: Consistent with Logistic Regression, high for non-fraudulent transactions (0.98) but notably lower for fraudulent transactions (0.10).
   d. Accuracy: Overall accuracy stood at 0.97 with a weighted average F1-score of 0.98.
3. Support Vector Classifier (SVC):
   a. Precision: Maintained high precision for non-fraudulent transactions (1.00) but the lowest for fraudulent transactions among all models (0.06).
   b. Recall: High recall for fraudulent transactions (0.92), indicating effective detection capabilities.
   c. F1-Score: High for non-fraudulent transactions (0.99) but very low for fraudulent ones (0.11).
   d. Accuracy: Overall model accuracy was 0.98 with a weighted average F1-score of 0.99.
4. Decision Tree Classifier:
   a. Precision: Similar precision scores as other models for non-fraudulent transactions (1.00) but low for fraudulent transactions (0.05).
   b. Recall: Lower recall for non-fraudulent transactions (0.97) and the lowest for fraudulent ones (0.87) among the models.

c. F1-Score: Consistent with other models for non-fraudulent transactions (0.99) but low for fraudulent transactions (0.10).
d. Accuracy: Exhibited a total accuracy of 0.97 with a weighted average F1-score of 0.98.

Discussion:

The models are good at flagging potential frauds, as shown by the high recall rates. However, the downside is that they also mark too many safe transactions as fraudulent, leading to a lot of false alarms. The models need to get more refined in order to get better at distinguishing real frauds from false ones. This might involve creating more sophisticated features, exploring different methods for spotting outliers, or using a combination of models to improve accuracy. In summary, while our models are great at catching non-fraudulent transactions, they still struggle to identify actual fraud without also catching a lot of false positives.

Neural Network Model's Evaluation



Confusion Matrix for Random UnderSample Model

- True Positives (TP): The model correctly identified 90 fraudulent transactions, demonstrating its effectiveness in detecting fraud.
- True Negatives (TN): A total of 55,148 transactions were accurately classified as non-fraudulent, indicating a strong ability to recognize legitimate activities.
- False Positives (FP): There were 1,715 instances where legitimate transactions were incorrectly flagged as fraudulent. This high number of false positives could lead to unnecessary verification processes and potential customer dissatisfaction.
- False Negatives (FN): The model missed 8 fraudulent transactions, slightly compromising the system's reliability in catching fraud.

This matrix suggests a balanced performance by the model, albeit with a concern over the number of false positives, which could strain resources and affect customer relations.

Confusion Matrix for Model with 100% Accuracy

- True Positives (TP): Perfectly identified all 98 fraudulent cases without errors.
- True Negatives (TN): Correctly recognized 58,683 non-fraudulent transactions, reflecting ideal detection capabilities.
- False Positives (FP) and False Negatives (FN): No false positives or false negatives were recorded, indicating a flawless classification process.

While this matrix displays an exemplary performance, such perfect accuracy is uncommon in practical scenarios and may suggest overfitting or a lack of complexity in the test data.

Discussion:

Two main things that need to be discussed, first of which is realism of the results: The perfect accuracy matrix, while impressive, is atypical and may not represent the model's performance in a real-world setting. On the other hand, the Random UnderSample model's results might be more indicative of actual operational conditions, where managing the trade-offs between detecting frauds and minimizing false alarms is crucial.

Second thing that need to be discussed is Model's Needs: The presence of false positives and minimal false negatives in the Random UnderSample model underscores the need for ongoing model tuning to enhance precision and recall. This could involve refining detection algorithms, implementing more robust feature engineering practices, or exploring hybrid models that combine various analytical strengths.

V. CONCLUSION

The aim of this study was to enhance the detection of credit card fraud using sophisticated data mining techniques and neural network models. Through a detailed analysis involving methods such as Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, and Decision Tree Classifier, we've gained deeper insights into the challenges and complexities of fraud detection in scenarios characterized by highly imbalanced data.

Key Findings:

The data mining approaches adopted in this project showed a significant ability to differentiate between fraudulent and legitimate transactions. However, the models tended to produce a higher number of false positives. This means that while they were effective in identifying fraud, they sometimes incorrectly flagged genuine transactions as fraudulent.

The neural network model, optimized with preprocessing techniques like SMOTE to counter class imbalance, demonstrated strong performance. It was particularly effective in identifying fraudulent transactions, as evidenced by the results of the confusion matrices used for performance evaluation.

Limitations and Directions for Future Research:

A notable limitation encountered in this study is the trade-off between sensitivity (recall) and specificity (precision). The high rate of false positives, although effective for fraud detection, could lead to customer dissatisfaction due to unwarranted checks.

Future efforts could look into combining different ensemble techniques to lower the rate of false positives while maintaining high recall. Experimenting with hybrid models that leverage the strengths of multiple analytical techniques could further refine predictive accuracy.

This study sets a robust foundation for tackling credit card fraud, providing a scalable and adaptable framework that can be enhanced in future research. Moving forward, the focus will be on refining these models to achieve higher accuracy and developing real-time detection systems to further secure consumer transactions.

References

[1] A. R. Khalid, N. Owoh, O. Uthmani, M. Ashawa, J. Osamor, and J. Adejoh, "Enhancing credit card fraud detection: an ensemble machine learning approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 6, 2024.

[2] J. Zou, J. Zhang, and P. Jiang, "Credit card fraud detection using autoencoder neural network," *arXiv preprint arXiv:1908.11553*, 2019.

[3] Y. T. Lei, C. Q. Ma, Y. S. Ren, X. Q. Chen, S. Narayan, and A. N. Q. Huynh, "A distributed deep neural network model for credit card fraud detection," *Finance Research Letters*, vol. 58, p. 104547, 2023.

[4] M. L. Gambo, A. Zainal, and M. N. Kassim, "A convolutional neural network model for credit card fraud detection," in *2022 International Conference on Data Science and Its Applications (ICoDSA)*, IEEE, Jul. 2022, pp. 198-202.