CREDIT CARD FRAUD DETECTION

Boya Zeng

2024 May

https://github.com/Boya-Z/Fraud-Detectoion

Credit card fraud represents a significant challenge to financial security and customer trust. The project aims to develop a deep learning model that can effectively identify and classify transactions as fraudulent or legitimate, helping to mitigate losses due to fraud.

— Problem Statement

Data Overview

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

Dataset Attributes

V1 - V28

Numerical features that are a result of PCA transformation.

Time

Seconds elapsed between each transaction and the 1st transaction.

Amount

Transaction amount.

Class

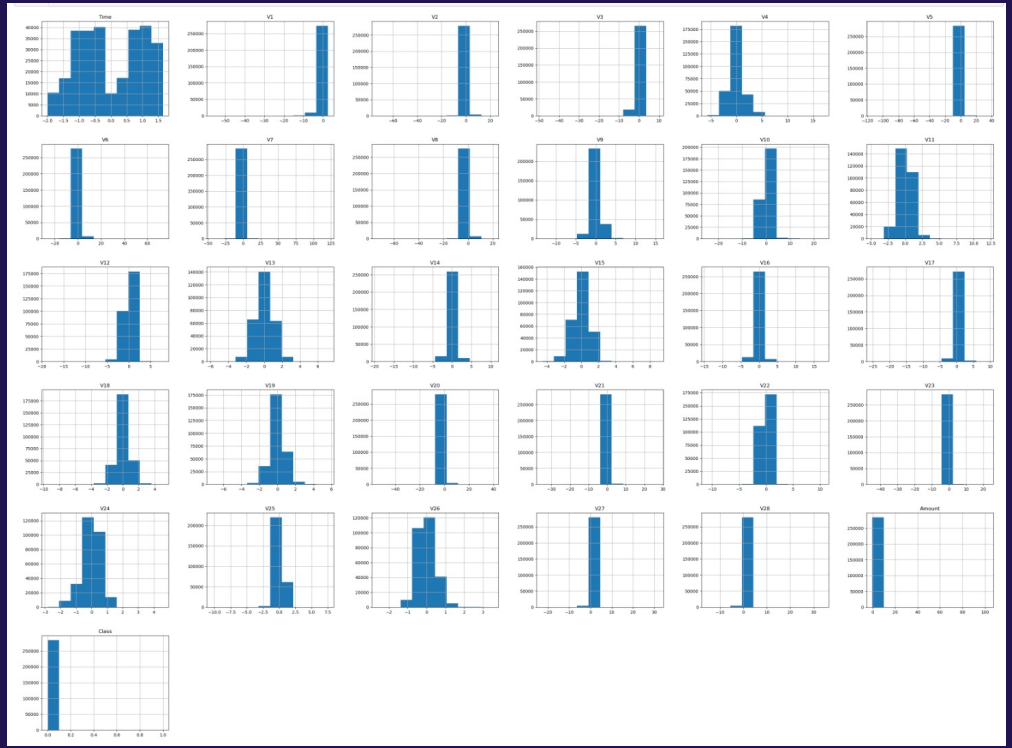
Fraud or otherwise (1 or 0)

Assumptions about Data

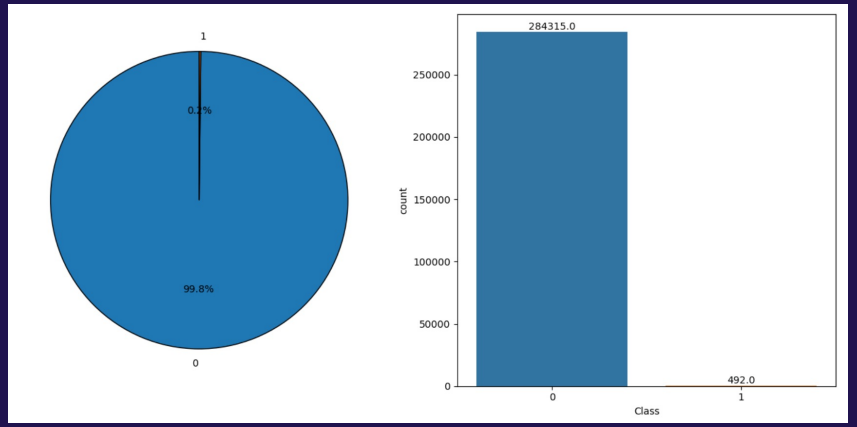
1.The principal components included in the dataset (after PCA transformation) contain sufficient information to identify patterns associated with fraud.
2.The labels in the dataset are accurate and there are no misclassifications.
3.The features even after scaling, retain their predictive value.

Exploratory Data Analysis

Distribution of Each Variable



Fraud vs Genuine transactions



Dataset Dimension (284807, 31)

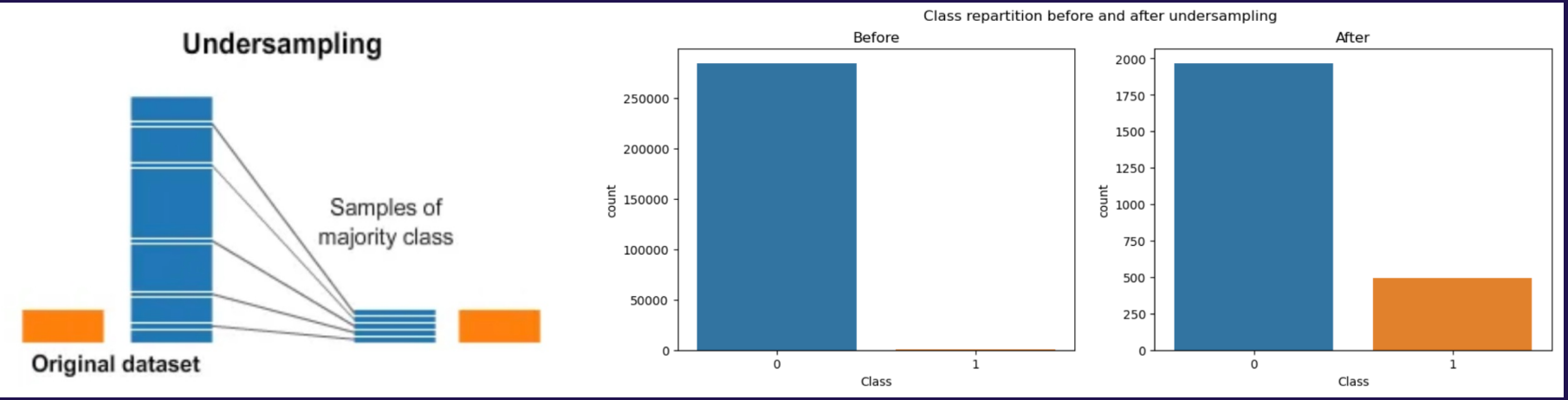
No Missing Value Present

The visual analysis of 'Amount' shows a right-skewed distribution, typical for transactional data, where most transactions involve smaller amounts.

The dataset contains 284,807 transactions, of which only 492 (0.17%) are fraudulent, indicating a highly imbalanced dataset.

Feature Engineering & Transformations

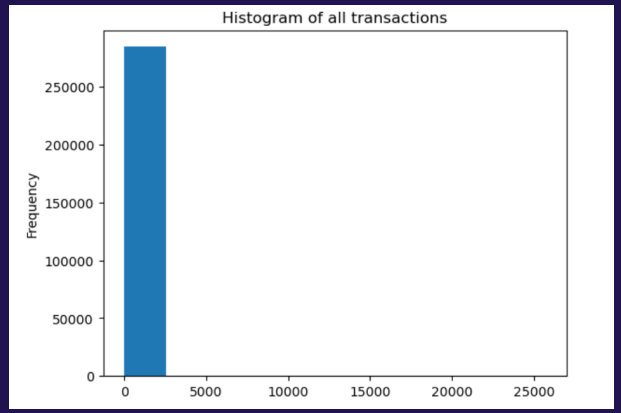
Step1: Balancing the Dataset--Under-sampling the Majority Class

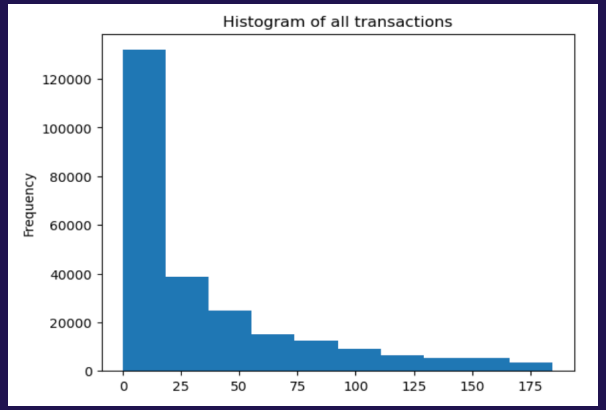


Step2: Features were scaled using StandardScaler to standardize their range and distribution, which helps improve the learning process of neural networks and CNN.

Feature Engineering & Transformations

Step3: Applied the IQR method to remove outliers from the 'Amount' feature.





Step4: Split the data into training and testing sets.

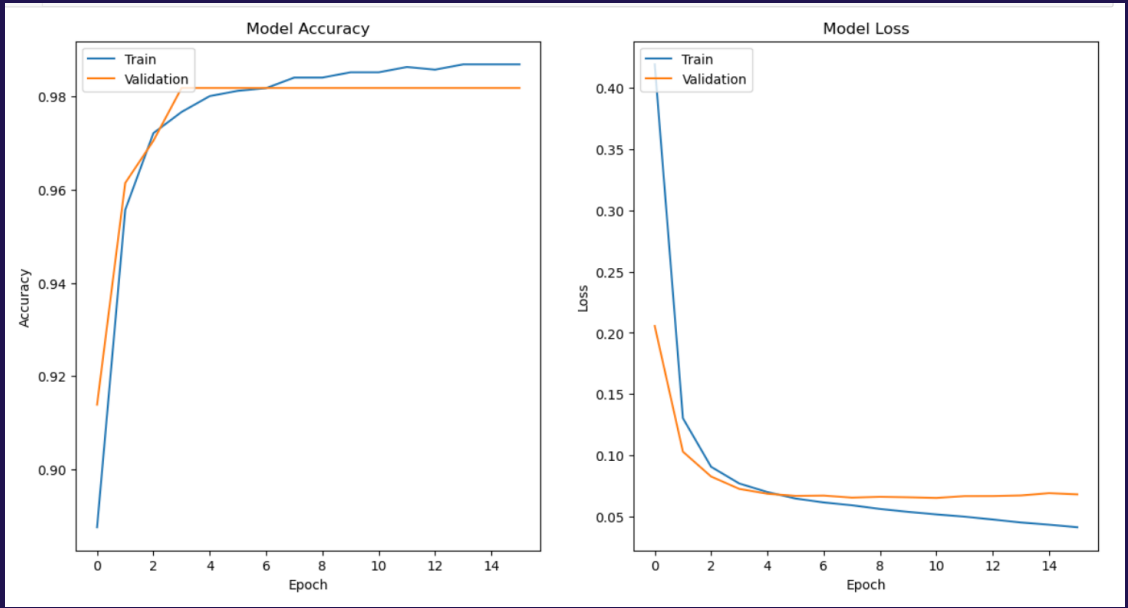
X_train	y_train	X_test	y_test
(1760, 30)	(1760,)	(441, 30)	(441,)

Step5: Reshape the data for compatibility with CNN input requirements.

X_train	X_test
(441, 30, 1)	(441, 30, 1)

Model with Checks for Overfitting/Underfitting

Multilayer Neural Network



Model Accuracy Graph:

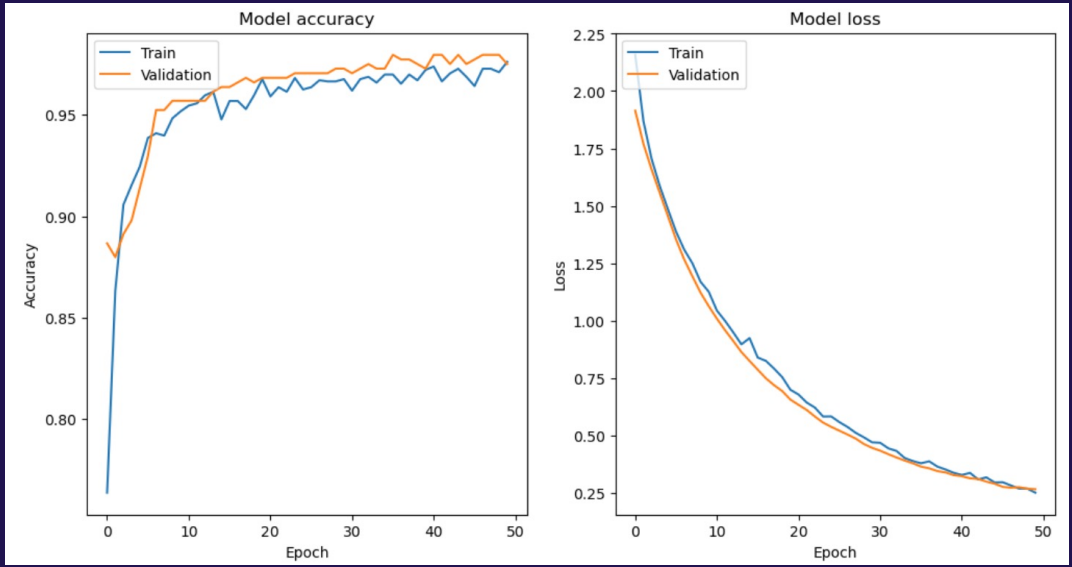
- The training accuracy (blue) and validation accuracy (orange) both increase and converge, indicating that the model is learning effectively from the training data and generalizing well to the validation data.
- Both curves plateau around the same accuracy level, showing no significant gap, which **suggests minimal overfitting**.

Model Loss Graph :

- loss (orange) decrease initially, indicating that the model is learning to reduce error.
- The validation loss starts to plateau and even slightly increase after a certain number of epochs, which is a sign of potential overfitting. However, early stopping helps to mitigate this by stopping training once the validation loss ceases to improve.

Model with Checks for Overfitting/Underfitting

Convolutional Neural Network (CNN)



Model Accuracy Graph:

- The training accuracy (blue) and validation accuracy (orange) both increase and converge, indicating that the model is learning effectively from the training data and generalizing well to the validation data.
- Both curves plateau around the same accuracy level, showing no significant gap, which suggests minimal overfitting.

Model Loss Graph :

- The training loss (blue) and validation loss (orange) decrease initially, indicating that the model is learning to reduce error.
- The validation loss starts to plateau and even slightly increase after a certain number of epochs, which is a sign of potential overfitting. However, early stopping helps to mitigate this by stopping training once the validation loss ceases to improve.

Model Selection with Regularization

Neural Network

Keras Tuner is used to perform Random Search on the hyperparameters. The best model is selected based on validation accuracy.

Early stopping is used during the search to prevent overfitting.

CNN

CNN model selected based on Bayesian Optimization results.

Model trained with early stopping and the best hyperparameters found.

Dropout layers to prevent overfitting.

L2 regularization to penalize large weights.

Batch normalization to stabilize learning.

Regularization Techniques

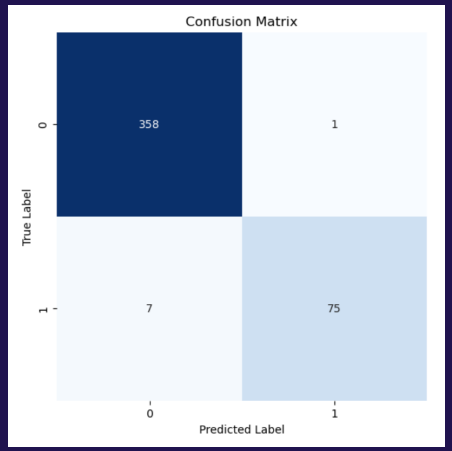
Results and Learnings from the Methodology

Model Selection:

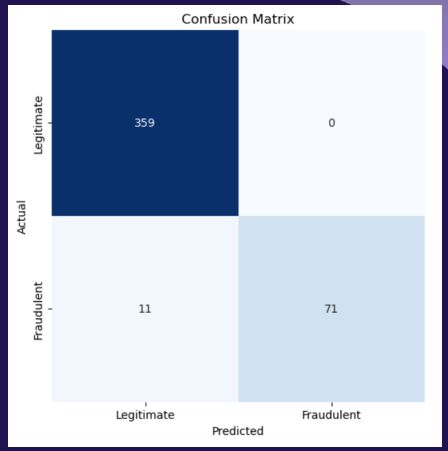
- The NN model may be preferred in scenarios where a balance between false positives and false negatives is crucial.
- The CNN model may be preferred in scenarios where minimizing false positives is more critical, despite a slight increase in false negatives.

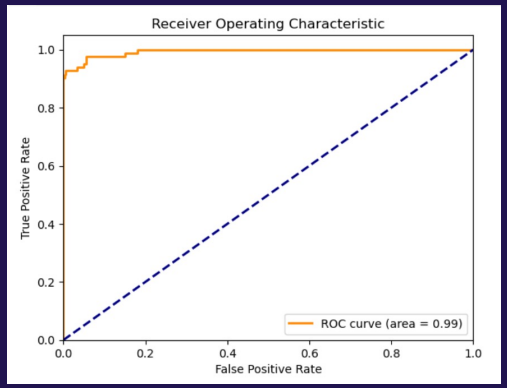
The ROC curves for both models indicate excellent performance, with AUC values close to 1.

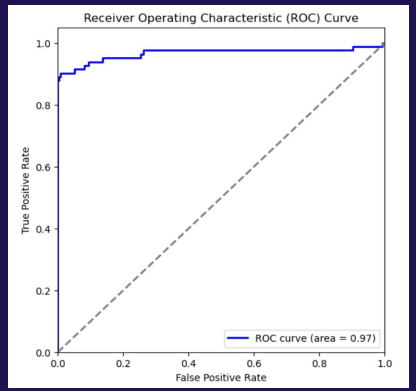
NN



CNN







Results and Learnings from the Methodology

	NN	CNN
Accuracy	0.9818	0.9750
Precision	0.9868	1.0
Recall	0.9146	0.8659
F1-Score	0.9493	0.9281

Conclusion

- The **NN model** may be preferred when a balance between false positives and false negatives is crucial, as it provides high recall and F1-score.
- The **CNN model** may be preferred in scenarios where minimizing false positives is more critical, despite a slight increase in false negatives, due to its perfect precision.

Learning

• The accuracy and F1-score were high for both models, demonstrating their effectiveness in classifying transactions as fraudulent or legitimate.

• The NN model had a slightly higher F1-score, indicating a better balance between precision and recall.

• The CNN model achieved perfect precision, meaning it did not classify any legitimate transactions as fraudulent, which is critical in financial fraud detection.

• Early stopping and regularization were crucial in ensuring the models did not overfit the training data.

Future Work

01 Options

- *Explore other deep learning architectures, such as Recurrent Neural Networks (RNNs) or Transformer models, to potentially improve performance.*

02 Options

- *Explore other feature engineering techniques, such as creating interaction terms or more sophisticated time-related features.*

03 Options

- *Implement more advanced techniques for handling imbalanced data, such as SMOTE (Synthetic Minority Over-sampling Technique).*

04 Options

- *Continuous learning and model updating with new transaction data to adapt to emerging fraud patterns.*

Thanks