Anomaly Detection: Credit Card Fraud Detection With an Imbalance Dataset

Tony Ly

December 2023

Abstract

Contents

1 Introduction

In today's digital age, credit cards have become an indispensable tool for everyday transactions. However, this widespread adoption has also made credit cards a prime target for fraudulent activities. Credit card fraud, which encompasses unauthorized transactions using stolen or lost cards, has become a significant financial burden for both cardholders and financial institutions. According to a 2022 report by Javelin Strategy & Research, global card fraud losses reached a staggering $43 billion in 2022, highlighting the urgency of developing effective fraud detection mechanisms.

Traditional fraud detection methods, such as rule-based systems and manual reviews, have proven to be increasingly inadequate in keeping up with the evolving tactics and sophistication of fraudsters. These methods often rely on predefined rules and patterns to identify fraudulent transactions, making them susceptible to exploitation by fraudsters who constantly devise new ways to circumvent these rules. In response to these limitations, researchers have turned to machine learning and artificial intelligence (AI) techniques to develop more robust and adaptive fraud detection systems.

Neural networks, a type of machine learning algorithm inspired by the structure and function of the human brain, have emerged as a promising approach for credit card fraud detection. Neural networks have the ability to learn complex patterns and relationships from large datasets, making them well-suited for identifying the subtle anomalies that often characterize fraudulent transactions.

This thesis aims to investigate the effectiveness of neural networks in detecting credit card fraud. Specifically, we will design, implement, and evaluate a neural network model for classifying credit card transactions as either fraudulent or legitimate. We will explore various neural network architectures and hyperparameter-tuning techniques to optimize the performance of our model. Additionally, we will compare the performance of our neural network model with traditional fraud detection methods to demonstrate its superior ability to identify fraudulent transactions.

2 Related Research

Addressing Imbalanced Datasets: S. Makki et al. highlight the detrimental impact of imbalanced datasets on fraud detection. They identify LR, C5.0 decision tree, SVM, and ANN as the best algorithms for such scenarios, requiring balanced datasets for optimal training.

Multi-Stage Fraud Detection: C. Jiang et al. propose a novel multi-stage process for credit card fraud detection. This approach involves transaction collection, aggregation based on behavioral patterns, classification, training, and testing. A feedback mechanism helps identify and address anomalies.

Ensemble Learning for Fraud Detection: Ishan Sohony et al. explore ensemble learning for credit card fraud detection. They combine Random Forest and Neural Networks, leveraging the strengths of both algorithms for high accuracy

and efficient fraud detection.

Optimal Anomaly Detection Techniques: Phuong Hanh Tran et al. introduce two data-driven approaches for optimal anomaly detection in credit card transactions. These methods, kernel parameter selection and T2 control charts, offer promising results.

Comparative Analysis of Machine Learning Algorithms: Imane Sadgali et al. present a comprehensive comparison of various machine learning algorithms for credit card fraud identification. This analysis provides valuable insights into their strengths and weaknesses in this context.

Hybrid ML Model for Improved Accuracy: Debachudamani Prusti and Santhnu Kumar Rath propose a hybrid model combining Decision Tree, SVM, and k-Nearest Neighbor techniques. This hybrid approach achieves an impressive 82.58% accuracy, outperforming individual algorithms.

Unsupervised Fraud Detection with Autoencoders: Mohamad Zamini and Golamali Montazar develop an unsupervised fraud detection system using autoencoders. This method demonstrates promising results on European datasets.

Misrepresentation Location Framework with NRBE: Akila and Srinivasulu Reddy address imbalanced datasets and transaction noise with a Misrepresentation Location framework using the Non-Overlapped Risk Based Bagging Ensemble (NRBE) model. This approach offers significant improvements in BCR, BER, recall, and fraud detection cost reduction.

Random Forest for Credit Card Fraud Detection: M. S. Kumar et al. investigate the effectiveness of Random Forest for credit card fraud detection. Their proposed system achieves an accuracy of 90%, highlighting its potential in this domain.

Kernel-based Supervised Hashing for Fraud Detection: Z. Li et al. propose a fraud detection system using Kernel-based Supervised Hashing (KSH) particularly suited for large, high-dimensional datasets. This method exhibits superior performance compared to existing techniques.

Comprehensive Survey of Machine Learning Techniques for Mastercard Fraud Detection: Pawan Kumar and Fahad Iqbal conduct a comprehensive survey of machine learning techniques for MasterCard fraud detection. Their analysis emphasizes the need for more robust and adaptable systems in this constantly evolving landscape.

3 Research Methodology

The dataset we used is from Kaggle, Credit Card Fraud Detection: anonymized credit card transactions labeled as fraudulent or genuine. This dataset has a size of 284,807 transactions with a total of 31 columns (30 of them are the features and the other is the class label). 284,315 of the transaction are labeled as normal and the other 492 are labeled as fraud, which develops a severe imbalance in the dataset:

Table 3.1: Class Density Distribution

	Count	Percent
Class: 0	284315	99.83
Class: 1	492	0.17

We proposed the approach of anomaly detection methodology because of the substantial imbalance in the dataset instead of the typical approach of classification methodology. This is mainly because anomaly detection methods are less sensitive to class imbalance. The other reason is because of the density distribution of the two classes on each anonymized features.

28 of the 30 features (features V1, V2, ..., V28) are anonymized by the providers due to confidentiality concerns. The other two features are time, when during the day the transaction occurred, and amount, the amount of money in the transaction. Since features V1, V2, ..., V28 have been anonymized, the data has been converted to numbers and normalized by principle component analysis (PCA) transformation. The following plots displays the distribution of of the normal data compared to the fraud data on on each anonymized feature:

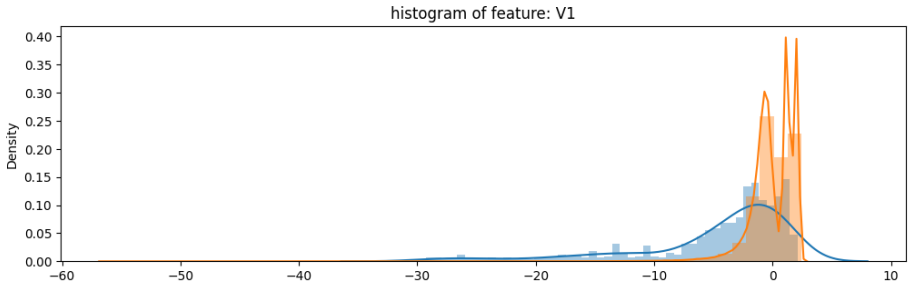

Figure 3.1: This is a graph of the class distribution density on feature V1. The orange is normal transaction and the blue is fraudulent transactions.

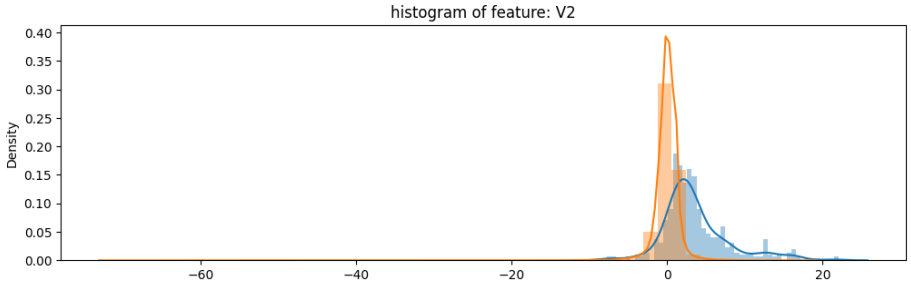

Figure 3.2: This is a graph of the class distribution density on feature V2. The orange is normal transaction and the blue is fraudulent transactions.

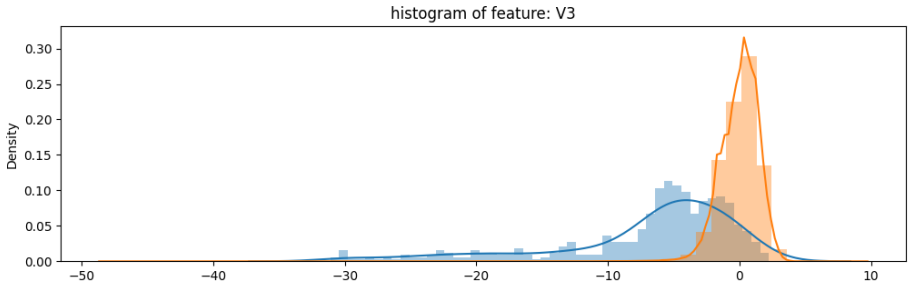

Figure 3.3: This is a graph of the class distribution density on feature V3. The orange is normal transaction and the blue is fraudulent transactions.

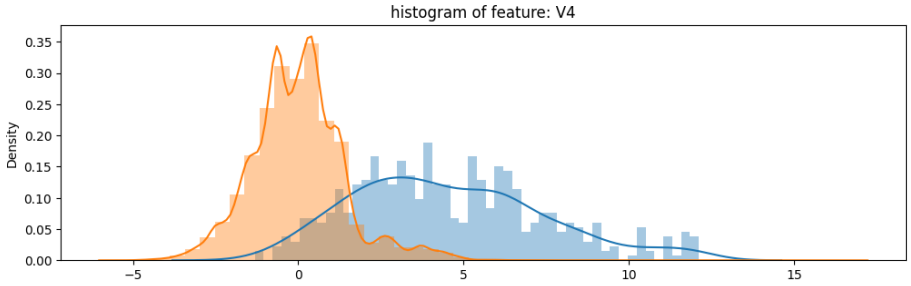

Figure 3.4: This is a graph of the class distribution density on feature V4. The orange is normal transaction and the blue is fraudulent transactions.

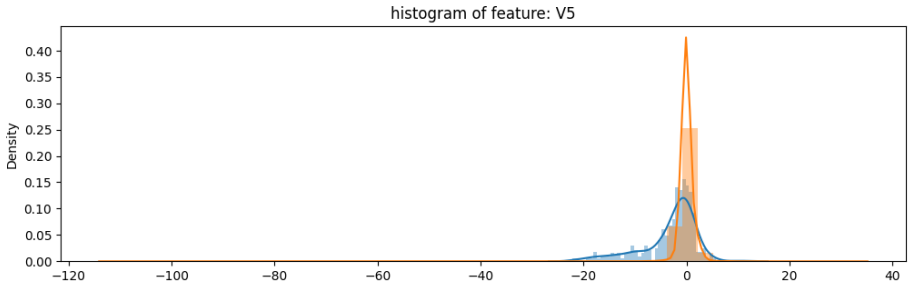

Figure 3.5: This is a graph of the class distribution density on feature V5. The orange is normal transaction and the blue is fraudulent transactions.

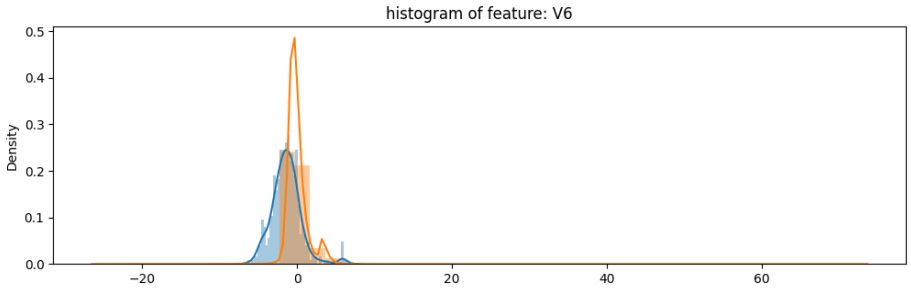

Figure 3.6: This is a graph of the class distribution density on feature V6. The orange is normal transaction and the blue is fraudulent transactions.

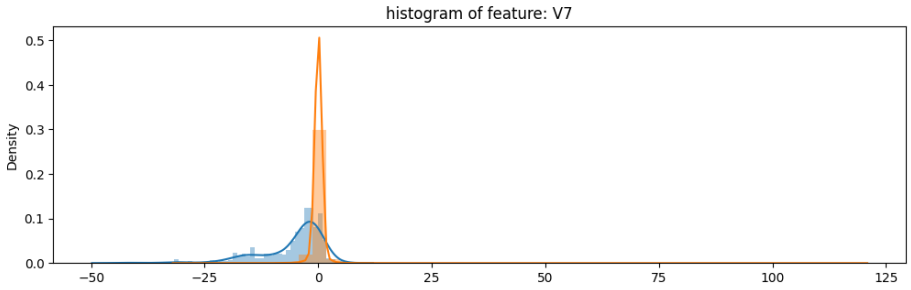

Figure 3.7: This is a graph of the class distribution density on feature V7. The orange is normal transaction and the blue is fraudulent transactions.

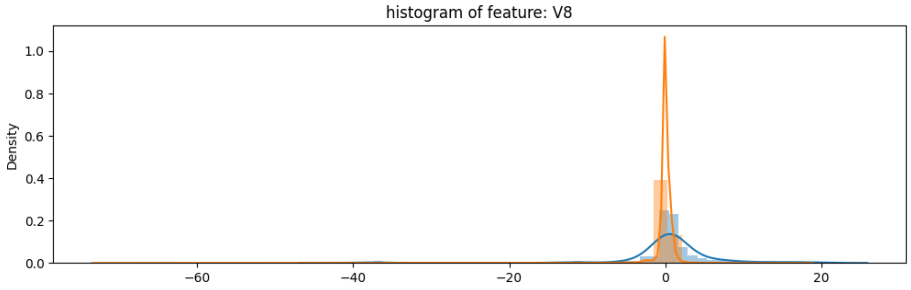

Figure 3.8: This is a graph of the class distribution density on feature V8. The orange is normal transaction and the blue is fraudulent transactions.

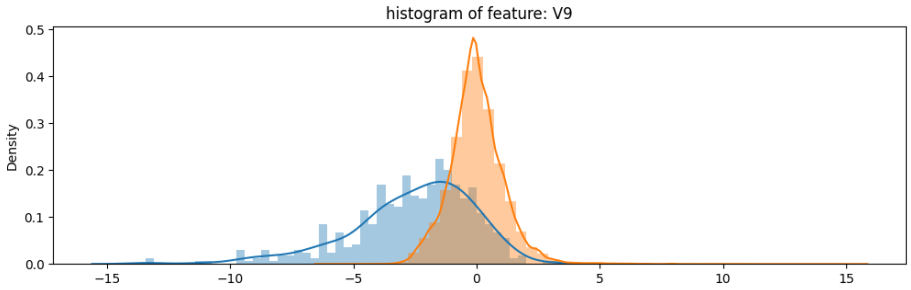

Figure 3.9: This is a graph of the class distribution density on feature V9. The orange is normal transaction and the blue is fraudulent transactions.

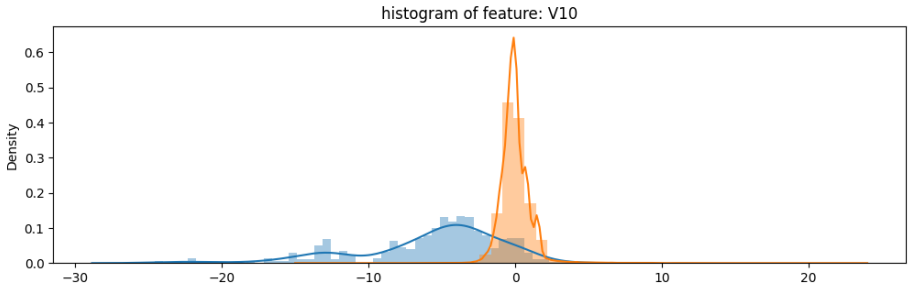

Figure 3.10: This is a graph of the class distribution density on feature V10. The orange is normal transaction and the blue is fraudulent transactions.

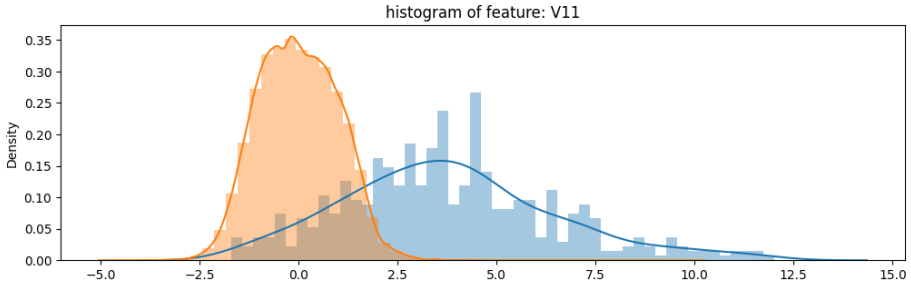

Figure 3.11: This is a graph of the class distribution density on feature V11. The orange is normal transaction and the blue is fraudulent transactions.

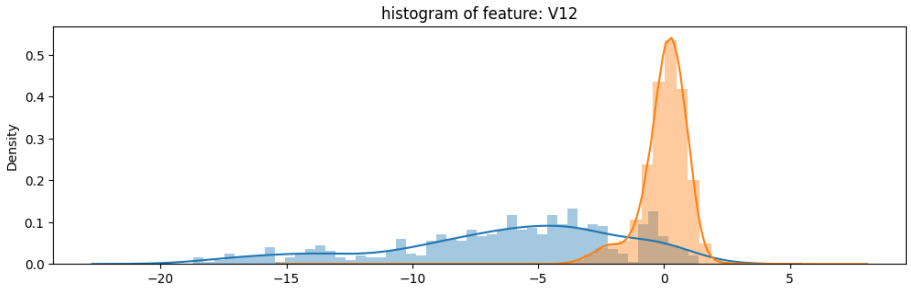

Figure 3.12: This is a graph of the class distribution density on feature V12. The orange is normal transaction and the blue is fraudulent transactions.

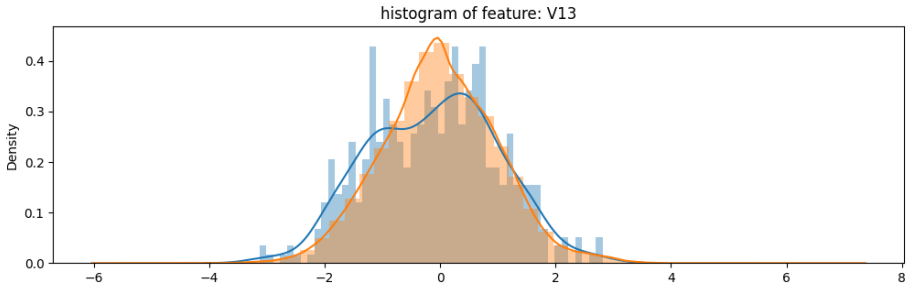

Figure 3.13: This is a graph of the class distribution density on feature V13. The orange is normal transaction and the blue is fraudulent transactions.

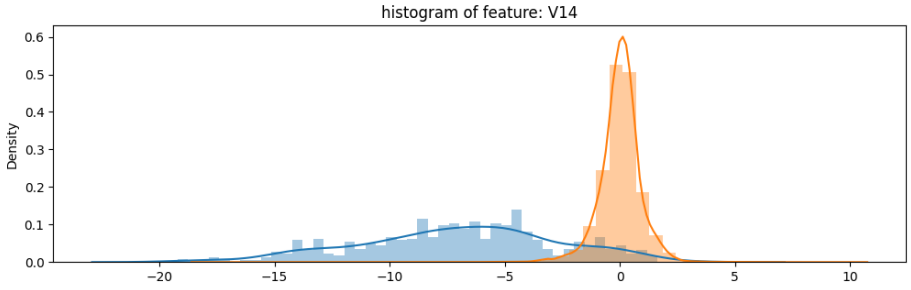

Figure 3.14: This is a graph of the class distribution density on feature V14. The orange is normal transaction and the blue is fraudulent transactions.

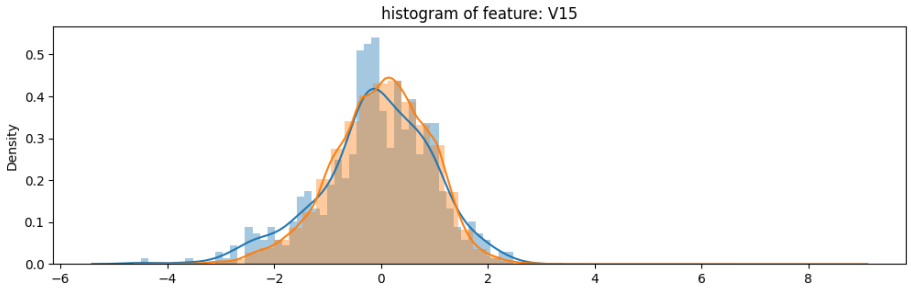

Figure 3.15: This is a graph of the class distribution density on feature V15. The orange is normal transaction and the blue is fraudulent transactions.

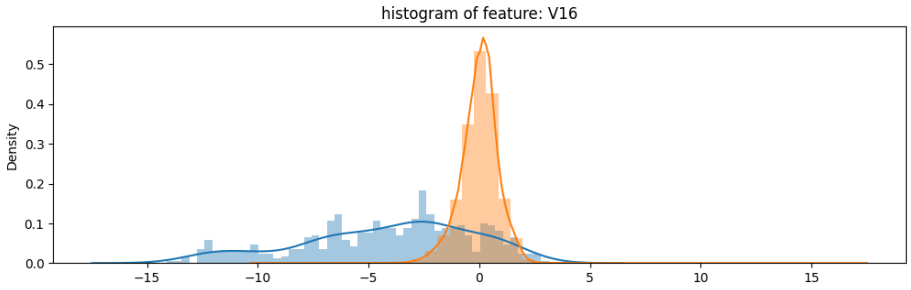

Figure 3.16: This is a graph of the class distribution density on feature V16. The orange is normal transaction and the blue is fraudulent transactions.

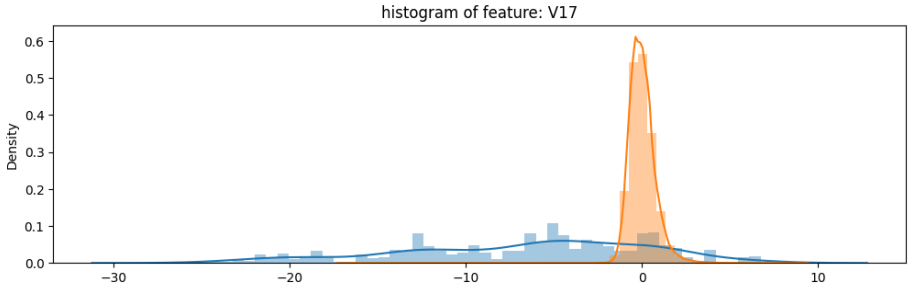

Figure 3.17: This is a graph of the class distribution density on feature V17. The orange is normal transaction and the blue is fraudulent transactions.

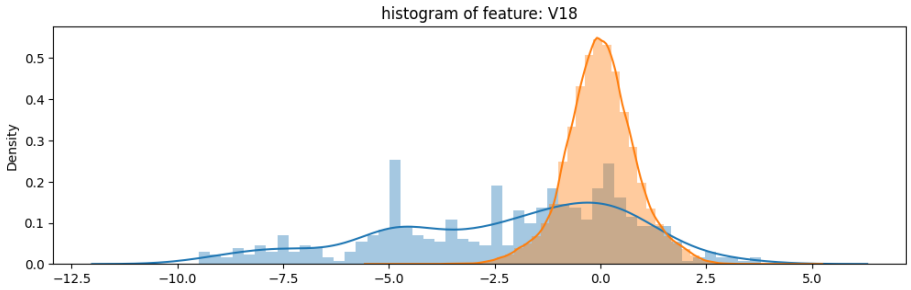

Figure 3.18: This is a graph of the class distribution density on feature V18. The orange is normal transaction and the blue is fraudulent transactions.

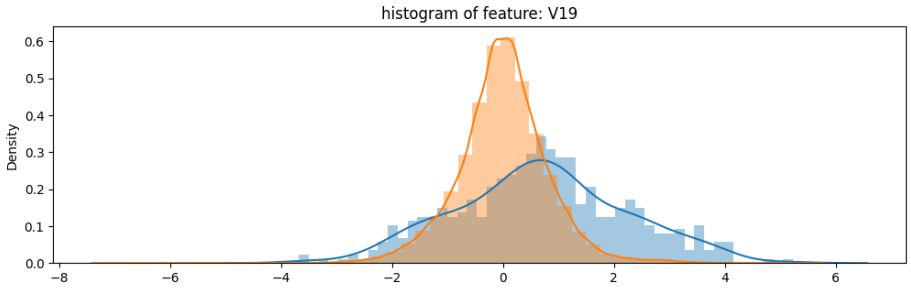

Figure 3.19: This is a graph of the class distribution density on feature V19. The orange is normal transaction and the blue is fraudulent transactions.

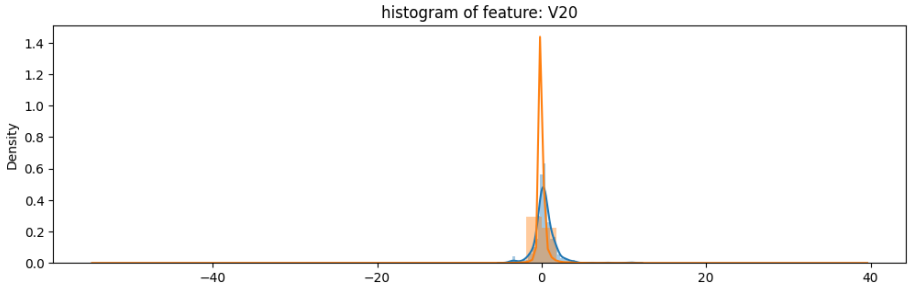

Figure 3.20: This is a graph of the class distribution density on feature V20. The orange is normal transaction and the blue is fraudulent transactions.

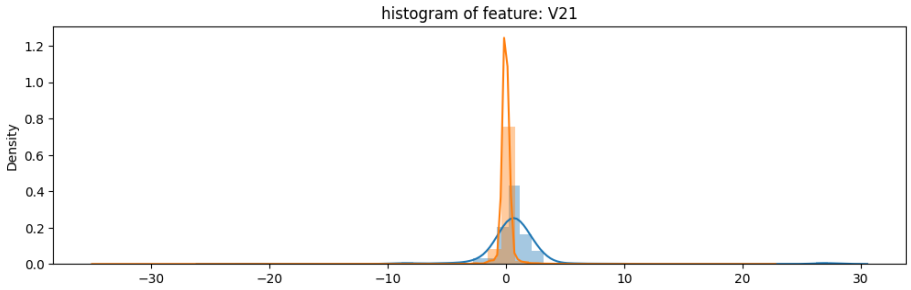

Figure 3.21: This is a graph of the class distribution density on feature V21. The orange is normal transaction and the blue is fraudulent transactions.

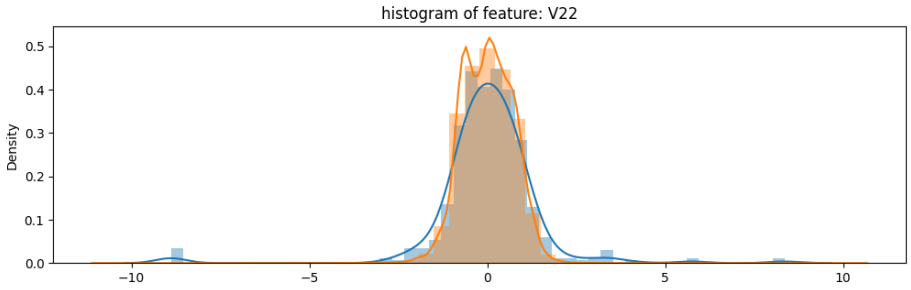

Figure 3.22: This is a graph of the class distribution density on feature V22. The orange is normal transaction and the blue is fraudulent transactions.

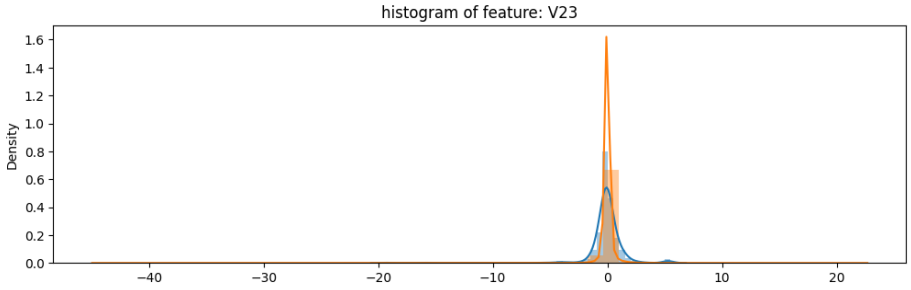

Figure 3.23: This is a graph of the class distribution density on feature V23. The orange is normal transaction and the blue is fraudulent transactions.

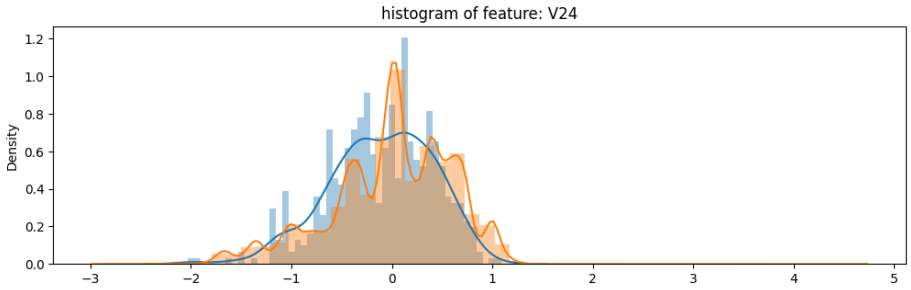

Figure 3.24: This is a graph of the class distribution density on feature V24. The orange is normal transaction and the blue is fraudulent transactions.

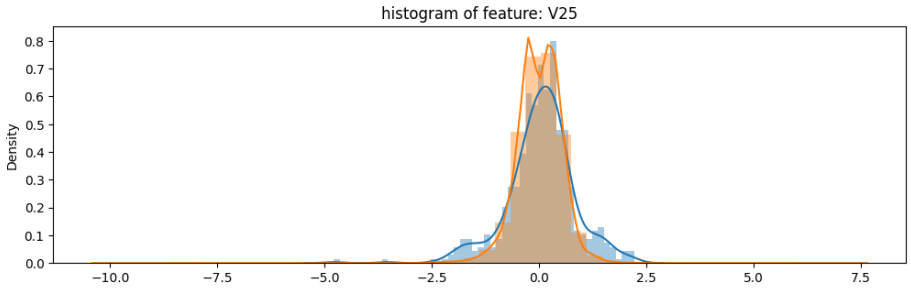

Figure 3.25: This is a graph of the class distribution density on feature V25. The orange is normal transaction and the blue is fraudulent transactions.

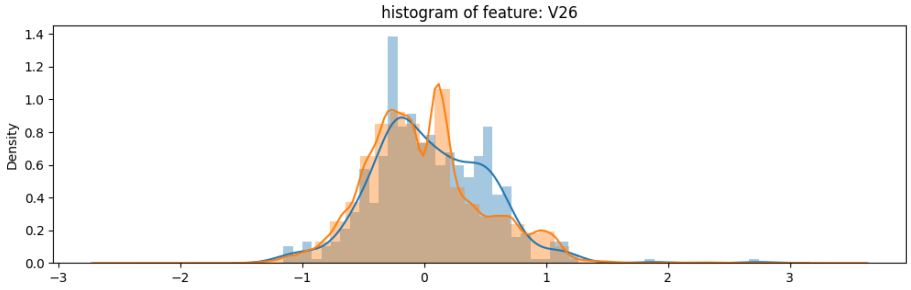

Figure 3.26: This is a graph of the class distribution density on feature V26. The orange is normal transaction and the blue is fraudulent transactions.

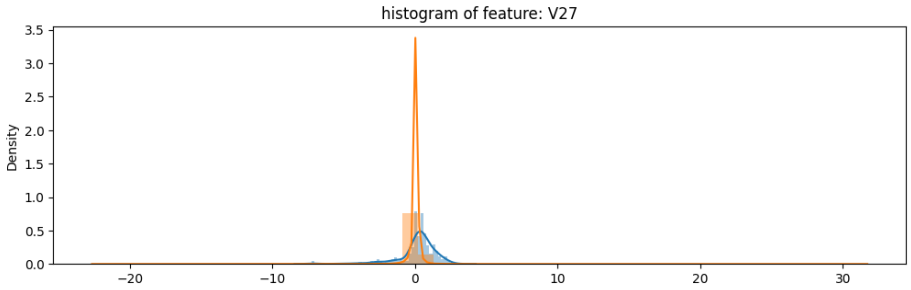

Figure 3.27: This is a graph of the class distribution density on feature V27. The orange is normal transaction and the blue is fraudulent transactions.

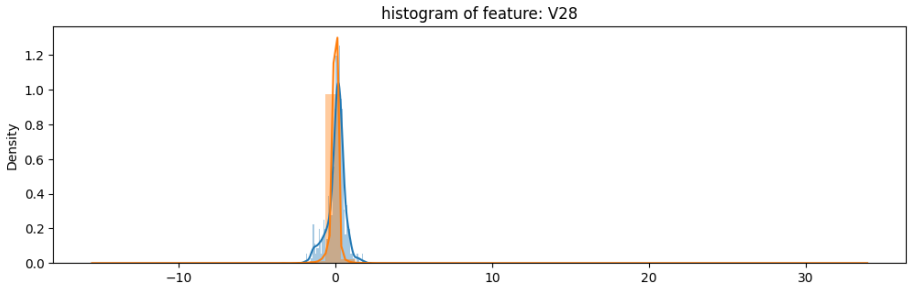

Figure 3.28: This is a graph of the class distribution density on feature V28. The orange is normal transaction and the blue is fraudulent transactions.

11 of the 28 plots reveals that the two types of transactions have similar distributions on features: V8, V13, V15, V20, V22, V23, V24, V25, V26, V27, and V28. We made the decision to approach this dataset with anomaly detection methods instead of the typical approach of classification methods. More than half of the anonymized features reveals difference in the density which proves to preferable in anomaly detection.

We propose the autoencoder model for the task of anomaly detection. The autoencoder is a type of artificial neural network that learns by compressing the dimensionality of the input and try to recreate the input from the encoded representation through the two functions: an encoding function and decoding function (refer to Figure 3.29). The encoder transforms the input into a smaller dimensionality, known as a bottleneck. The decoder recreates the input based on the reduced dimensional representation. The autoencoder is trained to minimize the difference, called reconstruction error, between the input data and the reconstructed data. This process forces the autoencoder to learn efficient representations of the data that capture the most important information while discarding irrelevant details. In our particular situation, the autoencoder is trained only on the normal transaction to learn the representation of normal

transaction, which is then tested on a mix of normal and fraudulent transaction. The anomaly detection is based on the reconstruction error with respect to the threshold to classify if a transaction is normal or fraudulent.

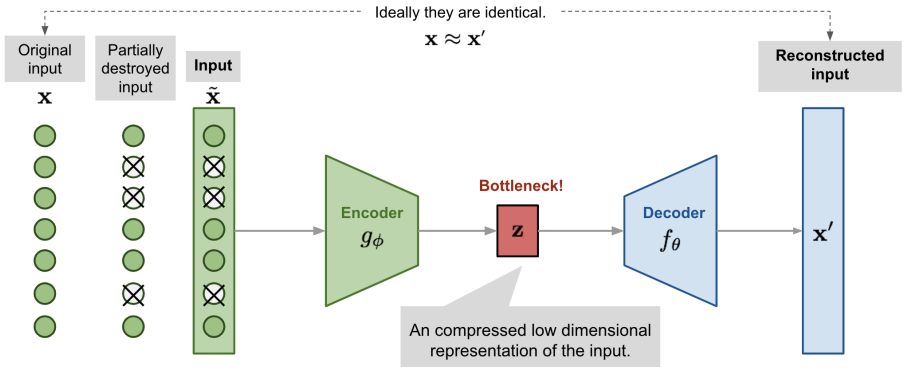

Figure 3.29: This is a diagram of the proposed autoencoder model.

The autoencoder architecture we proposed consists of an encoder with 4 layers and a decoder with 4 layers (refer to Figure 3.30). We decided that the bottleneck should have the dimensionality of 6 because the plots of features: V4, V10, V11, V12, V14, V16, V17 displayed the greatest disparity between the distribution density. For the training hyperparameters, we used the Adam optimizer, the accuracy metric, the mean squared error (MSE) loss function, and an epoch of 50 with a batch size of 512 at a learning rate of 1e-7.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
dense_40 (Dense)	(None, 25)	775
dense_41 (Dense)	(None, 19)	494
dense_42 (Dense)	(None, 13)	260
dense_43 (Dense)	(None, 6)	84
dense_44 (Dense)	(None, 6)	42
dense_45 (Dense)	(None, 13)	91
dense_46 (Dense)	(None, 19)	266
dense_47 (Dense)	(None, 30)	600

Total params: 2612 (10.20 KB)
Trainable params: 2612 (10.20 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 3.30: This is a diagram of the proposed autoencoder architecture.

Since the dataset we are using is heavily imbalanced, the performance of the model is not evaluated with the common accuracy metric. Instead, we use the confusion matrix (Table 3.2), precision (Equation 1), recall (Equation 2), and F1-score (Equation 3). We focused primarily on the precision metric when fine-tuning the model because of the context of credit card fraud detection, we do not want to an abundant of false positives.

Table 3.2: Confusion Matrix.

	Predicted: 0	Predicted: 1
Actual: 0	True Negative (TN)	False Positive (FP)
Actual: 1	False Negative (FN)	True Positive (TF)

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2}{precision^{-1} + recall^{-1}} \tag{3}$$

4 Empirical Evidence

The training and validation loss curve of our model exhibits an optimal training scenario (Figure 4.1). Both training and validation loss steadily decrease over the training epochs, indicating effective learning and generalization. The gap between the two lines remains small, signifying that the model is not memorizing the training data but instead learning generalizable patterns. This suggests that the chosen model architecture, hyperparameters, and training process are well-suited for the task at hand, and the model is likely to perform well on unseen data.

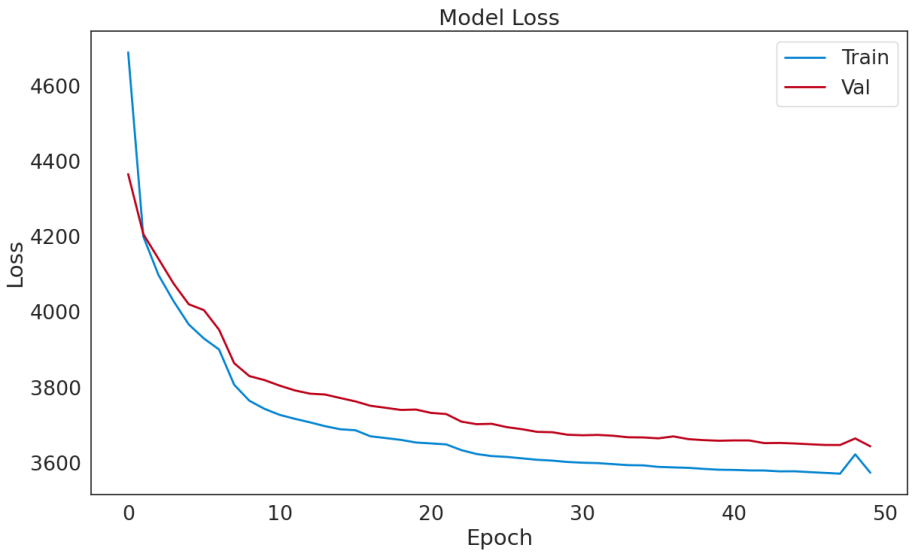

Figure 4.1: This is a graph of the training loss vs validation loss.

The precision and recall trade off of our model allows us to generalize the range of our threshold. The threshold serves as a decision boundary that separates data points belonging to different classes: normal and fraudulent transaction. Based on Figure 4.2 and our focus of keeping precision score as high as possible without lowering the recall score too low, we decided our threshold to be in the range of 45 to 55.

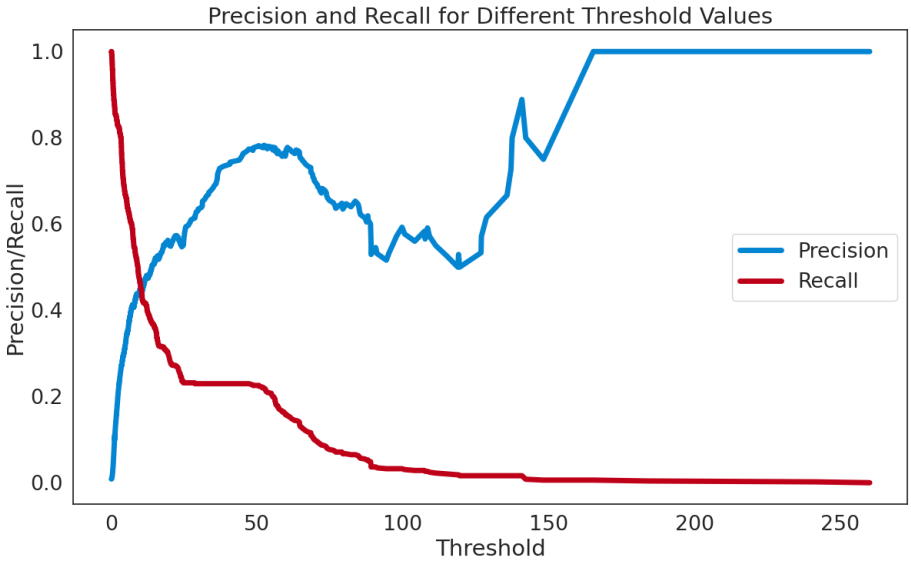

Figure 4.2: This is a graph depicts the Precision Recall Trade off.

We were able to finalize our threshold to the value of 47. The reconstruction error plot (Figure 4.3) depicts the reconstruction error calculated of the reconstructed inputs to the actual inputs. The threshold at 47 gives us the best results in terms of precision in account of recall.

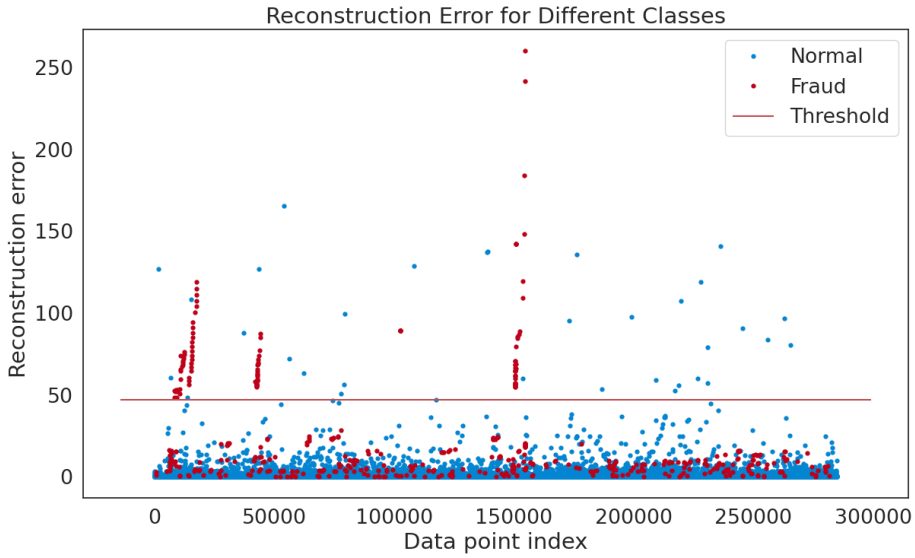

Figure 4.3: This is a graph displays the Reconstruction Error with a Threshold of 47.

Our model achieved a precision score of 80.71%, a recall score of 22.97%, and a F1-score 0f 35.67%. The confusion matrix (Table 4.1) visualize the performance of the model.

Table 4.1: Confusion Matrix.

	Predicted: 0	Predicted: 1
Actual: 0	56820	27
Actual: 1	379	113

5 Conclusion

While the proposed model achieved some encouraging results in specific areas, its overall performance fell short of our initial expectations. Despite extensive experimentation with various architectures and hyperparameters, the model struggled to achieve a consistently high level of accuracy across all tasks. This suggests potential limitations in the chosen model architecture, training process, or data preparation techniques.

Further investigation is needed to identify the underlying causes of the model's shortcomings. Analyzing the training and validation loss curves, exploring different model architectures, and experimenting with alternative data augmentation techniques are some potential avenues for improvement. Additionally, investigating interpretability techniques could offer valuable insights into the model's decision-making process and guide further refinement.

While the current results do not meet our initial goals, they provide a valuable foundation for future research. The insights gained from this study can be leveraged to develop more robust and accurate models in the future. By addressing the identified limitations and exploring alternative approaches, we can continue to advance the field and contribute to the development of effective and reliable models for the specific task at hand.Unfortunately, our model's performance did not

6 References

References

[1] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[2] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, 2018.

[3] I. Sohony, R. Pratap, and U. Nambiar, "Ensemble learning for credit card fraud detection," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, ser. CODS-COMAD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 289–294.

[4] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, "Real time data-driven approaches for credit card fraud detection," in *Proceedings of the 2018 International Conference on E-Business and Applications*, ser. ICEBA 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 6–9.

[5] I. Sadgali, N. Sael, and F. Benabbou, "Fraud detection in credit card transaction using neural networks," in *Proceedings of the 4th International Conference on Smart City Applications*, ser. SCA '19. New York, NY, USA: Association for Computing Machinery, 2019.

[6] D. Prusti and S. K. Rath, "Web service based credit card fraud detection by applying machine learning techniques," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 492–497.

[7] M. Zamini and G. Montazer, "Credit card fraud detection using autoencoder based clustering," in *2018 9th International Symposium on Telecommunications (IST)*, 2018, pp. 486–491.

[8] S. Akila and U. S. Reddy, "Credit card fraud detection using non-overlapped risk based bagging ensemble (nrbe)," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2017, pp. 1–4.

[9] M. S. Kumar, V. Soundarya, S. Kavitha, E. Keerthika, and E. Aswini, "Credit card fraud detection using random forest algorithm," in *2019 3rd International Conference on Computing and Communications Technologies (ICCCT)*, 2019, pp. 149–153.

[10] Z. Li, G. Liu, S. Wang, S. Xuan, and C. Jiang, "Credit card fraud detection via kernel-based supervised hashing," in *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2018, pp. 1249–1254.

[11] P. Kumar and F. Iqbal, "Credit card fraud identification using machine learning approaches," in *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019, pp. 1–4.