

FINANCIAL FRAUD DETECTION
USING EXPLAINABLE ARTIFICIAL INTELLIGENCE



ELİF ZÜLAL ÇAVDAR

BOĞAZİÇİ UNIVERSITY

2025

FINANCIAL FRAUD DETECTION
USING EXPLAINABLE ARTIFICIAL INTELLIGENCE

Thesis submitted to the
Institute for Graduate Studies in Social Sciences
in partial fulfillment of the requirements for the degree of

Master of Arts
in
Management Information Systems

by
Elif Zülal Çavdar

Boğaziçi University

2025

Financial Fraud Detection Using Explainable Artificial Intelligence

The thesis of Elif Zülal Çavdar

has been approved by:

Assist. Prof. Aysun Bozanta Hakyemez
(Thesis Advisor)

Prof. Bertan Yılmaz Badur

Assoc. Prof. Selçuk Kıran
(External Member)

January 2025

DECLARATION OF ORIGINALITY

I, Elif Zülal Cavdar, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature.....

Date.....

ABSTRACT

Financial Fraud Detection Using Explainable Artificial Intelligence

Fraud detection is a critical challenge in the financial industry, as the growing volume of transactions increases the risk of fraudulent activities, leading to significant financial and reputational harm. This thesis aims to address this challenge by evaluating advanced machine learning and deep learning models designed to enhance the accuracy, efficiency, and scalability of fraud detection systems, ultimately contributing to more secure financial operations. The study evaluates 14 machine and deep learning models encompassing both supervised and unsupervised learning techniques across four sampling techniques: imbalanced data, random undersampling, SMOTE, and majority voting with random undersampling on a private and a public dataset. Each model's effectiveness is compared and analyzed, highlighting their suitability for fraud detection tasks. The study also integrates explainability techniques to uncover the most influential features contributing to fraud detection by using feature importance, SHAP, LIME and DeepLIFT. The results demonstrate the efficacy of the evaluated models in detecting fraudulent activities, with Random Forests achieving the best performance among machine learning models and TabTransformers leading among deep learning models. This research contributes to the financial industry's efforts in combating fraud by presenting a comprehensive evaluation of machine learning and deep learning models, identifying optimal strategies for data sampling, feature engineering, and enhancing model interpretability to support informed decision-making.

ÖZET

Açıklanabilir Yapay Zeka ile Finansal Dolandırıcılık Tespiti

Finans sektöründe artan işlem hacmi nedeniyle dolandırıcılık faaliyetleri riski artarak ciddi finansal ve itibar kayıplarına yol açmaktadır. Bu nedenle finansal dolandırıcılık tespiti kritik bir önem arz etmektedir. Bu tez, dolandırıcılık tespit sistemlerinin doğruluk, verimlilik ve ölçeklenebilirliğini artırmayı amaçlayan gelişmiş makine öğrenimi ve derin öğrenme modellerini değerlendirerek bu konuyu ele almayı hedeflemektedir. Çalışmada hem denetimli hem de denetimsiz öğrenme tekniklerini kapsayan 14 makine ve derin öğrenme modeli, dört örnekleme tekniği (dengesiz veri, rastgele azınlık örnekleme, SMOTE ve rastgele azınlık örnekleme ile çoğunluk oylaması) üzerinden özel ve kamuya açık bir veri setinde değerlendirilmiştir. Her bir modelin etkinliği karşılaştırılmış ve analiz edilmiş, dolandırıcılık tespiti görevlerine uygunlukları vurgulanmıştır. Çalışma ayrıca, dolandırıcılık tespitine katkıda bulunan en etkili özellikleri ortaya çıkarmak için SHAP, LIME, DeepLIFT gibi açıklanabilirlik tekniklerini kullanmaktadır. Sonuçlar, değerlendirilen modellerden makine öğrenimi modelleri arasında Random Forests, derin öğrenme modelleri arasında ise TabTransformers modelinin en iyi sonucu verdiğini göstermiştir. Bu araştırma, makine öğrenimi ve derin öğrenme modellerinin kapsamlı bir değerlendirmesini sunarak, veri örnekleme, özellik mühendisliği ve model açıklanabilirliğini iyileştirme konularında optimal stratejileri belirlemek suretiyle, finans sektörünün dolandırıcılıkla mücadele çabalarına katkıda bulunmaktadır.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
CHAPTER 2: LITERATURE REVIEW	8
2.1 Financial fraud detection with AI.....	12
2.2 XAI and its applications in fraud detection	22
CHAPTER 3: METHODOLOGY	31
3.1 Datasets.....	31
3.2 Sampling methods.....	33
3.3 Algorithms.....	35
3.4 Hyperparameter tuning.....	52
3.5 Evaluation metrics.....	53
3.6 XAI techniques	55
CHAPTER 4: RESULTS	60
4.1 Exploratory data analysis (EDA)	60
4.2 Feature engineering for private data	79
4.3 EDA for engineered features	81
4.4 Model performance comparison.....	85
4.5 Explainability of best performing models.....	115
CHAPTER 5: DISCUSSION	131
CHAPTER 6: CONCLUSION AND FUTURE WORKS	137
APPENDIX A: CROSSTAB HEATMAPS FOR PRIVATE DATA CATEGORICAL VARIABLES	139
APPENDIX B: EDA RESULTS FOR PRIVATE DATA WITH ENGINEERED FEATURES	147
REFERENCES.....	152

LIST OF TABLES

Table 1. Summary of the Most Relevant Studies in Literature	29
Table 2. Chi-square Statistics and P-values for Categorical Variables	63
Table 3. T-test Statistics and P-values of Continuous Variables	71
Table 4. Performance Comparison of ML Models on Private Imbalanced Dataset..	87
Table 5. Performance Comparison of DL Models on Private Imbalanced Dataset ..	89
Table 6. Performance Comparison of USL Models on Private Imbalanced Dataset	89
Table 7. Performance Comparison of ML Models on Private Dataset with RU	90
Table 8. Performance Comparison of DL Models on Private Dataset with RU	91
Table 9. Performance Comparison of USL Models on Private Dataset with RU	93
Table 10. Performance Comparison of ML Models on Private Dataset with SMOTE	94
Table 11. Performance Comparison of DL Models on Private Dataset with SMOTE	96
Table 12. Performance Comparison of USL Models on Private Dataset with SMOTE	98
Table 13. Performance Comparison of ML Models on Private Dataset with Majority Voting	99
Table 14. Hyperparameter Values for DNN	101
Table 15. DNN Performance Before and After Hyperparameter Tuning.....	102
Table 16. Hyperparameter Values for TabTransformer	103
Table 17. TabTransformer Performance Before and After Hyperparameter Tuning	104
Table 18. Performance Comparison of ML Models on Public Imbalanced Dataset	105

Table 19. Performance Comparison of DL Models on Public Imbalanced Dataset	106
Table 20. Performance Comparison of USL Models on Public Imbalanced Dataset	
.....	108
Table 21 Performance Comparison of ML Models on Public Dataset with RU	109
Table 22. Performance Comparison of DL Models on Public Dataset with RU.....	110
Table 23. Performance Comparison of USL Models on Public Dataset with RU ..	111
Table 24. Performance Comparison of ML Models on Public Dataset with SMOTE	
.....	112
Table 25. Performance Comparison of DL Models on Public Dataset with SMOTE	
.....	113
Table 26. Performance Comparison of USL Models on Public Dataset with SMOTE	
.....	114
Table 27 Performance Comparison of ML Models on Public Dataset with Majority	
Voting	114

LIST OF FIGURES

Figure 1. ANN architecture	41
Figure 2. Cramér's V analysis of categorical variables	65
Figure 3. Histogram analysis of continuous variables	67
Figure 4. Boxplot analysis of continuous variables	68
Figure 5. Pearson correlation analysis of continuous variables	69
Figure 6. Point-biserial correlations between continuous variables and label	70
Figure 7. Spearman correlations between continuous variables and label.....	70
Figure 8. Boxplots of continuous variables by label.....	72
Figure 9. Mutual information scores of the variables	74
Figure 10. Point-biserial correlation of public dataset variables with label.....	77
Figure 11. Spearman correlation of public dataset variables with label	78
Figure 12. Mutual information scores of public dataset variables	79
Figure 13. Pearson correlations of continuous variables with feature engineering ..	83
Figure 14. Performance comparison of Random Forests with and without FE	87
Figure 15. Feature importance of Random Forests.....	116
Figure 16. SHAP summary plot of Random Forests.....	118
Figure 17. Local SHAP results for the first instance.....	119
Figure 18. Local LIME results for the first instance.....	119
Figure 19. Local SHAP results for the second instance	120
Figure 20. Local LIME results for the second instance	121
Figure 21. Local SHAP results for the third instance	121
Figure 22. Local LIME results for the third instance.....	122
Figure 23. Feature importance of TabTransformer	123

Figure 24. Global feature importance with DeepLIFT for TabTransformer	124
Figure 25. Local feature importance with DeepLIFT for the first instance	126
Figure 26. Local feature importance with LIME for the first instance	126
Figure 27. Local feature importance with DeepLIFT for the second instance	127
Figure 28. Local feature importance with LIME for the second instance	128
Figure 29. Local feature importance with DeepLIFT for the third instance	129
Figure 30. Local feature importance with LIME for the third instance	129



LIST OF APPENDIX FIGURES

Figure A1. Crosstab for V2 and label.....	139
Figure A2. Crosstab for V6 and label.....	140
Figure A3. Crosstab for V8 and label.....	141
Figure A4. Crosstab for V2 and label.....	142
Figure A5. Crosstab for V12 and label.....	143
Figure A6. Crosstab for V13 and label.....	144
Figure A7. Crosstab for V14 and label.....	145
Figure A8. Crosstab for V15 and label.....	146
Figure B1. Crosstab for ResidencyBoolean and label	147
Figure B2. Crosstab for IsTop5MCC and label	148
Figure B3. Cramer's V for categorical variables	148
Figure B4. Histograms of continuous variables.....	149
Figure B5. Boxplots of continuous variables.....	149
Figure B6. Point-biserial correlations of continuous variables with label	150
Figure B7. Spearman correlations of continuous variables with label.....	150
Figure B8. Boxplots of continuous variables by label	151
Figure B9. Mutual information scores for features.....	151

CHAPTER 1

INTRODUCTION

Fraud is a growing problem in financial systems, creating significant challenges for individuals, businesses, and governments. The financial impact of fraudulent activities is substantial, with global losses amounting to billions of dollars each year (PwC, 2022). Beyond financial costs, fraud reduces public trust in financial institutions and affects the stability of economic systems. Therefore, detecting and preventing fraud are essential priorities for financial institutions and regulatory authorities. However, identifying fraudulent behavior in today's financial systems is a difficult task due to the large number of transactions, the variety of fraud methods, and the constantly changing tactics used by fraudsters.

The increase in digital banking and online transactions has made the problem worse, turning fraud detection into a critical area of research. Traditional rule-based systems, which were effective in the past, are becoming less capable of handling the sophisticated and adaptive methods used in modern fraud. These systems have difficulty managing the large-scale, complex data produced by financial systems and often fail to detect patterns beyond their predefined rules (KPMG & Nets, 2020). As a result, there is growing interest in using machine learning (ML) and deep learning (DL) techniques to solve these problems.

ML and DL models have shown significant potential in fraud detection by finding hidden patterns, anomalies, and complex relationships in data that traditional methods may miss (Bello & Olufemi, 2024). Common ML algorithms used in fraud detection include Decision Trees, Random Forests, Support Vector Machines (SVM), and ensemble methods such as gradient boosting machines and XGBoost (Dornadula

& Geetha, 2019). These algorithms are effective at capturing non-linear relationships and interactions between features, making them suitable for detecting complex fraud patterns. DL algorithms, on the other hand, leverage neural networks to automatically learn feature representations from raw data. Models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have gained attention for their ability to process high-dimensional data and sequential patterns, respectively (Mienye & Jere, 2024). Additionally, Autoencoders and Generative Adversarial Networks (GAN) have been applied for anomaly detection by modeling the normal transaction behavior and identifying deviations as potential fraud (Kuttiyappan, 2024; Pumsirirat & Liu, 2018). These models can adapt to new data and learn from changing patterns, making them well-suited to combat fraud.

Despite the advancements in detection methodologies, several challenges remain. There is a notable lack of studies focusing on private datasets due to confidentiality issues in fraud domain. Private datasets often contain nuanced and domain-specific information that could enhance the accuracy and applicability of fraud detection models. Unfortunately, access restrictions on such datasets limit their availability for research, which has led to a reliance on public datasets. While public datasets are valuable for reproducibility and benchmarking, they may not capture the full complexity of real-world fraud scenarios. Additionally, feature engineering remains underexplored in the context of fraud detection, due to the anonymized nature of publicly available datasets. Feature engineering involves the creation of domain-specific variables, such as transaction patterns, temporal behaviors, and aggregate metrics, that can significantly enhance model performance. Research by Bahnsen et al. (2016) highlighted the importance of engineered features like transaction velocity and cardholder behavior in improving fraud detection accuracy.

Yet, systematic studies on applying and validating feature engineering techniques across different datasets remain sparse. Without addressing these gaps, models may fail to generalize effectively across diverse contexts, limiting their practical applicability. Bridging this divide requires collaborative efforts to make private datasets more accessible for anonymized research and to advance feature engineering techniques tailored specifically for fraud detection scenarios.

Another major issue about fraud detection datasets is the class imbalance where fraudulent transactions make up only a small portion of the data. This imbalance can lead to models that favor non-fraudulent predictions while missing actual fraud cases (Haixiang et al., 2017). Addressing this imbalance often requires the use of sampling techniques, such as oversampling methods like Synthetic Minority Oversampling Technique (SMOTE) or undersampling of the majority class, to ensure the models are trained on a balanced dataset (Chawla et al., 2002). However, these methods come with their own set of challenges, such as the potential to introduce noise in the data or lose valuable information (He & Garcia, 2009). As fraudsters continue to evolve their tactics, the imbalanced nature of fraud datasets remains a critical challenge that must be tackled to develop robust detection systems.

Another important challenge is the lack of interpretability in many ML and DL models, often called the "black-box" problem (Rudin, 2019). While these models can achieve high accuracy, their complex structures make it difficult to understand how they make specific decisions. This lack of transparency is particularly concerning in financial systems, where regulatory requirements and accountability are crucial (Weber et al., 2024). Explainability methods allow stakeholders to interpret model predictions and ensure compliance with regulatory standards. Implementing these techniques effectively often require significant computational

resources and expertise. Additionally, the explainability methods should align with the specific needs of financial fraud detection. Despite these obstacles, integrating explainability into ML and DL models is increasingly recognized as essential for fostering trust, accountability, and broader adoption of these advanced technologies in high-stakes domains such as finance.

Given these challenges, this thesis examines the use of ML and DL algorithms to improve fraud detection in financial systems. By applying various supervised learning (SL) and unsupervised learning (USL) models, along with feature engineering, advanced sampling and explainability techniques, the study aims to address key issues such as imbalanced datasets and model interpretability. The research uses private bank data to develop, evaluate, and explain fraud detection models, offering practical insights for both professionals and researchers in the field.

The research seeks to answer the following questions:

- RQ1. Which ML and DL models perform best in detecting credit card fraud datasets?
- RQ2. What sampling techniques are most effective in addressing class imbalance in fraud detection datasets?
- RQ3. How can explainability techniques improve the interpretability of fraud detection models while maintaining high performance?
- RQ4. How the engineered features influence the detection of fraudulent transactions?
- RQ5. How do results differ between private bank data and a publicly available dataset?

To answer this set of research questions, this thesis leverages a private bank dataset that provides real-world relevance and valuable insights into fraud detection.

By utilizing the bank data, the study explores fraud detection within an authentic context, where transaction behaviors reflect real-world patterns. This unique dataset enables the research to delve into domain-specific nuances of fraudulent activity, offering a depth of analysis not typically available in publicly accessible datasets. The inclusion of private data underscores the need for methodologies that can cater to restricted datasets while maintaining regulatory and privacy compliance.

A key contribution of this thesis is the development of engineered features from the private dataset, emphasizing customer spending habits and behavioral patterns. Unlike traditional fraud detection methods that rely heavily on transactional data, this research introduces features that capture broader spending behaviors, by focusing on the rare and common transactions for a specific customer. These features are designed to be broadly applicable to other credit card fraud detection datasets, providing a template for enhancing model accuracy while reducing reliance on raw transactional features. By shifting the focus towards behavioral insights, this approach adds an interpretability layer to fraud detection systems.

The study also addresses one of the most critical challenges in fraud detection: class imbalance. Fraudulent transactions typically represent a small fraction of the overall dataset, leading to models that are biased towards non-fraudulent predictions. This thesis evaluates and compares multiple techniques to manage class imbalance, including random undersampling (RU), SMOTE, and majority voting with RU. Through systematic experimentation, this study identifies effective strategies to optimize model performance while minimizing information loss and mitigating overfitting.

Another prominent feature of this research is the broad spectrum of ML and DL models that are evaluated. By including both SL and USL approaches, the study

provides a holistic view of fraud detection capabilities across different algorithms. The models range from traditional algorithms, such as Logistic Regression and SVM, to advanced DL architectures like TabTransformer and Autoencoders. This comprehensive evaluation not only highlights the relative strengths and weaknesses of these approaches but also offers practical guidance for their implementation in real-world systems.

Explainable Artificial Intelligence (XAI) methods are integral to this thesis, addressing the "black-box" nature of complex models. By implementing impurity based and permutation-based feature importance, SHapley Additive ExPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME) and Deep Learning Important Features (DeepLIFT), the research ensures that the decision-making processes behind the models are transparent and interpretable. This focus on explainability aligns with the needs of financial institutions, where regulatory compliance and stakeholder trust are paramount. The integration of these techniques into the study highlights the importance of accountability in deploying fraud detection systems.

To enhance reproducibility and facilitate benchmarking, the thesis compares findings from the private dataset with results derived from the publicly available Kaggle Credit Card Fraud Detection dataset. This comparative analysis sheds light on the consistency and transferability of the proposed methods. By identifying similarities and differences between private and public datasets, the research offers valuable insights into how methodologies can be adapted to various contexts, ensuring broader applicability of the findings.

This thesis is organized into several key chapters that provide a structured exploration of the research questions and contributions starting with the Literature

Review chapter which provides an in-depth examination of fraud detection, focusing on the role of artificial intelligence (AI) in this domain. It discusses the datasets commonly used in fraud detection research, the challenge of class imbalance and its mitigation techniques, and the ML and DL algorithms employed. The chapter concludes with a discussion on XAI and its implications in the context of fraud detection.

Following the literature review, Methodology details the research design, starting with the datasets used in the study, including both private and publicly available datasets. It then elaborates on the sampling techniques applied to address class imbalance, the algorithms evaluated, and the explainability techniques integrated to interpret model predictions. Afterwards, Results chapter presents the findings of the study, beginning with exploratory data analysis (EDA) and feature engineering outcomes. It then examines model performance across different sampling techniques for both private and public datasets. The chapter includes hyperparameter tuning results for the best-performing DL model on the private dataset and concludes with the explainability results of the top-performing ML and DL models.

The discussion chapter interprets the results, comparing them with findings from previous research. This chapter highlights the implications of the study's contributions, including the effectiveness of feature engineering, the impact of class imbalance techniques, and the role of explainability methods in enhancing trust and usability. The thesis concludes with the main findings and contributions of the research. It also identifies limitations and proposes directions for future work to advance fraud detection research further.

CHAPTER 2

LITERATURE REVIEW

This chapter explores what fraud is, discussing its various forms, underlying motivations, and the increasing reliance on AI-driven solutions to combat fraudulent activities by focusing on financial fraud. The key challenges associated with the application of AI in fraud detection are explored by evaluating current solutions, assessing their effectiveness, and highlighting existing gaps in the literature that require further research and innovation. Additionally, the chapter introduces XAI and its significance in enhancing transparency and trust in AI-driven fraud detection systems. By reviewing these key areas, this chapter provides a comprehensive understanding of how AI is shaping the future of fraud detection.

Fraud is defined in the Fraud Risk Management Guide by Association of Certified Fraud Examiners (2023, p.8) as “any intentional act or omission designed to deceive others, resulting in the victim suffering a loss and/or the perpetrator achieving a gain”. The complexity of fraud is often illustrated through the Fraud Triangle, which consists of three key elements -pressure, opportunity, and rationalization (Huang et al., 2016). It serves as a foundational framework for understanding the motivations behind fraudulent behavior and helps organizations enhance their fraud detection capabilities by identifying and analyzing these risk factors (Huang et al., 2016).

Fraud can be categorized into several types, each with unique characteristics and implications. Common categories include financial fraud, identity theft, healthcare fraud, and cyber-related fraud. Financial fraud often involves deceptive practices that result in financial loss to individuals or organizations, while identity

theft typically refers to the unauthorized use of someone else's personal information to commit fraud (Kahn & Roberds, 2005). In the healthcare sector, fraud can be revealed as unnecessary medical procedures or billing for services not performed, highlighting the diverse nature of fraudulent activities (Chen et al., 2020; Sharma et al., 2023). Furthermore, the rise of digital transactions has led to an increase in cyber-related fraud, which includes scams like phishing and business email compromise (BEC) (Cross & Gillett, 2020; Zhu et al., 2021).

Credit card fraud is a widespread form of financial fraud that can be categorized into two main types: existing account fraud and new account fraud. Existing account fraud typically involves the theft of credit card information to make unauthorized purchases, while new account fraud occurs when an individual uses another person's information to open a new credit account (Kahn & Roberds, 2005). The increasing reliance on digital payment systems has heightened these issues, making it essential for consumers and financial institutions to adopt robust security measures to mitigate risks associated with credit card fraud (Zhu et al., 2021). The financial losses attributed to credit card fraud are substantial, with estimates indicating that losses can reach upwards of \$1 billion annually due to the rapid growth of online payments and electronic banking (Peng & Wang, 2022).

The detection of fraud is essential for mitigating financial losses and maintaining the integrity of various systems, including financial institutions, healthcare, and corporate environments. Various techniques and methodologies have been developed to enhance the effectiveness of fraud detection, each with its strengths and limitations. One of the foundational approaches to fraud detection is the use of data mining techniques. Data mining involves the extraction of useful information from large datasets, enabling the identification of hidden patterns and

anomalies that may indicate fraudulent activity (Phua et al., 2010). ML algorithms which can be categorized into SL and USL techniques are used to detect fraudulent transactions as a part of the data mining process. SL involves training models on labeled datasets, allowing them to learn patterns associated with fraudulent transactions (Gyamfi & Abdulai, 2018). Techniques such as decision trees, SVMs, and neural networks have been widely adopted in this context (Pumsirirat & Liu, 2018). On the other hand, USL techniques, such as k-Nearest Neighbors (kNN) and local outlier factor are employed to detect anomalies in transaction data without prior labeling, making them particularly useful for identifying fraud patterns (Bauder & Khoshgof, 2017). The flexibility of these methods allows for the adaptation of fraud detection systems to evolving fraudulent strategies, which is crucial in today's rapidly changing digital landscape (Aras, 2023). Furthermore, Bhardwaj and Gupta (2022) discuss the impact of feature selection on the effectiveness of ML algorithms in identifying fraudulent transactions. They note that the choice of features significantly influences the model's performance, which is echoed by Ileberi et al. (2022), who emphasize the importance of selecting relevant features for effective anomaly detection.

In addition to data mining, the integration of DL models has further enhanced the capabilities of fraud detection systems (Alarfaj et al., 2022). These models can adapt to new fraud patterns and improve detection accuracy over time. For instance, studies have explored the use of advanced algorithms such as Long Short-Term Memory (LSTM) networks and Deep Belief Networks, which have shown promise in detecting anomalies in credit card transactions (Kalid et al., 2020). Furthermore, Autoencoders and RBM have been utilized to develop sophisticated models that can learn complex representations of transaction data, thereby improving the accuracy of

fraud detection (Pumsirirat & Liu, 2018). Another study has proposed GAN and Graph Neural Networks as innovative solutions for improving fraud detection capabilities (Kuttiyappan, 2024; Tran et al., 2019). Alharbi et al. propose a DL approach that leverages a text-to-image mechanism to enhance detection capabilities, illustrating the innovative directions in which research is heading (Alharbi et al., 2022). Dash et al. (2020) discusses the development of a Legendre Neural Network for credit card fraud detection, highlighting its superior performance compared to traditional methods like multilayer perceptron and decision trees. This study illustrates the challenges of nonstationary data and unbalanced class distributions in fraud detection, emphasizing the need for advanced algorithms in this domain. These models are particularly effective in identifying subtle patterns that may indicate fraudulent behavior, which traditional rule-based systems might overlook (Aras, 2023; Raj & Portia, 2011).

Additionally, the use of ensemble methods, which combine multiple algorithms to improve detection performance, has gained attraction in recent years (Ghrib et al, 2024). Hussein et al. (2021) discusses a novel approach to credit card fraud detection by employing a stacking ensemble technique that combines fuzzy-rough nearest neighbor and sequential minimal optimization with Logistic Regression. Muter & Molood (2020) introduces a method that combines SVM, Modified Particle Swarm Optimization, and Modified Multi-Objective Particle Swarm Optimization, showcasing the effectiveness of ensemble algorithms in enhancing detection accuracy and speed for fraudulent transactions.

In conclusion, fraud is a complex issue requiring comprehensive strategies to detect and mitigate its impacts across various domains. Rooted in motivations described by the Fraud Triangle, fraud appears in diverse forms such as financial

fraud, identity theft, and cyber-related scams, each requiring tailored detection techniques. The integration of advanced methodologies, such as AI, has significantly improved fraud detection systems by uncovering complex patterns and anomalies that traditional approaches might overlook. Innovations like Autoencoders, GAN, and ensemble methods have demonstrated exceptional adaptability to evolving fraudulent behaviors, underscoring the importance of continuous technological advancement. By combining foundational frameworks with cutting-edge tools, organizations can better safeguard systems, reduce financial losses, and maintain public trust in an increasingly digital landscape.

2.1 Financial fraud detection with AI

This section focuses on the application of AI in financial fraud detection, exploring the key components that contribute to its effectiveness. It begins by examining the datasets commonly used in fraud detection research, highlighting their features and challenges. Next, it addresses the issue of class imbalance, a prevalent challenge in fraud datasets, and discusses various sampling techniques designed to address this imbalance. The review then delves into SL models, categorizing them into ML, DL and transformer models. Finally, USL models are discussed, emphasizing their utility in identifying novel fraud patterns without labeled data. Together, these topics provide a comprehensive overview of the current advancements and methodologies in AI-driven fraud detection.

2.1.1 Financial fraud datasets

Fraud detection is a critical area of research that employs various types of datasets to identify and mitigate fraudulent activities across different sectors. The datasets

employed in fraud detection research can be broadly categorized into three main types: public datasets, which are openly accessible to the research community; private datasets, which are typically proprietary and restricted in access; and synthetic datasets, which are artificially generated to simulate real-world data while preserving certain statistical properties. Each type of data offers unique advantages and challenges, and the choice of data significantly influences the effectiveness of fraud detection methodologies.

Public datasets are frequently utilized in fraud detection research due to their accessibility and relevance. The Credit Card Fraud Detection Dataset¹ also known as the European Card Holders Dataset is one of the most recognized public datasets, containing a substantial amount of real transaction data characterized by significant class imbalance, making it a benchmark for evaluating fraud detection algorithms (Ghrib et al., 2024; Leevy, 2023; Pumsirirat & Yan, 2018; Raghavan & El Gayar, 2019). The 2019 IEEE-CIS dataset² from Vesta Corporation includes around 590,000 transactions and is also one of the highly used credit card fraud datasets. It contains identity and transaction details, featuring attributes like device type, card info, and timestamps, and is highly imbalanced, with few fraudulent cases (Kanika & Singla, 2022; Mienye & Jere, 2024).

Due to privacy and security concerns, private or authentic fraud datasets are not readily accessible to researchers and practitioners. As a result, these datasets are relatively uncommon in academic literature. An example of fraud detection research utilizing private datasets is presented by Thennakoon et al. (2019), who employed a credit card fraud dataset obtained under a confidential disclosure agreement with a banking institution to develop a real-time fraud detection system. Gomez et al.

¹ <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

² <https://www.kaggle.com/competitions/ieee-fraud-detection/data>

(2018) utilized a real-world dataset and applied feature engineering to create aggregated features. Another study on fraud detection using private dataset involves a financial institution providing anonymized data on 80 million credit card transactions over eight months. The dataset, including transaction and account details, was labeled for fraud detection and exhibited a significant class imbalance, with only 0.14% of transactions marked as fraudulent (Roy et al., 2018).

Synthetic datasets play a significant role in fraud detection research as well. The PaySim³ financial simulator generates synthetic datasets that replicate real transaction behaviors while incorporating fraudulent activities, allowing researchers to explore various detection techniques without ethical concerns associated with real data usage (Lopez-Rojas et al., 2018). Another study by Erik Altman (2019) from IBM Research Center employs various techniques to synthesize credit card data⁴ within a 40,000-line code base. Focusing on modeling individual consumers, it creates a population reflecting the U.S. distribution in terms of age, occupation, income, credit scores, and geographic factors. This approach enables the testing of algorithms in a risk-free environment, facilitating innovation in fraud detection strategies.

2.1.2 Class imbalance and sampling techniques for financial fraud detection

Class imbalance is an extensive issue in ML, particularly in the context of fraud detection, where the number of legitimate transactions vastly outnumbers fraudulent ones. This imbalance can lead to models that are biased towards the majority class, resulting in poor detection rates for the minority class, which is often the class of interest in fraud detection scenarios. Peng & Wang (2022) emphasize the need for

³ <https://github.com/EdgarLopezPhD/PaySim>

⁴ <https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions>

effective data processing techniques to overcome class imbalance, such as oversampling and undersampling, to manage the skewed distribution of credit card transactions.

One of the most widely used techniques for addressing class imbalance is the SMOTE. SMOTE generates synthetic examples of the minority class to create a more balanced dataset (Chawla et al., 2002). Research has shown that applying SMOTE can significantly enhance the performance of classifiers in fraud detection tasks. For instance, one research demonstrated that using SMOTE in conjunction with feature selection methods improved the detection of financial accounting fraud in Turkish SMEs, where the dataset exhibited a class imbalance with 1384 non-fraudulent cases compared to 321 fraudulent ones (Hamal & Şenvar, 2021). Similarly, another study found that employing SMOTE with Random Forest classifiers improved the detection of fraudulent financial statements, highlighting the effectiveness of this technique in financial contexts (Ye et al., 2019). In addition to SMOTE, other oversampling techniques have been explored. Wibowo and Fatichah (2021) conducted an in-depth analysis of various oversampling methods, concluding that techniques like random oversampling and SMOTE yield better performance metrics in high-class imbalanced datasets, particularly in fraud detection scenarios. Esenogho et al. (2022) introduces a hybrid data resampling technique achieved by using the SMOTE and edited nearest neighbor (SMOTE-ENN) method that performs both oversampling and undersampling of the data. The use of these techniques is crucial for improving the performance of ML models in detecting fraud.

Undersampling, on the other hand, involves reducing the number of instances in the majority class to balance the dataset. While this method can help overcome class imbalance, it may also lead to the loss of potentially valuable information.

Verma and Tyagi (2022) proposed a selective class sampling technique that strategically reduces the majority class while retaining critical data points, thereby enhancing the performance of Random Forest classifiers in credit card fraud detection. This approach underscores the necessity of careful consideration when implementing undersampling techniques to avoid detrimental effects on model performance. Hybrid methods that integrate both oversampling and undersampling techniques have also gained attraction in literature. For instance, a study explored various sampling techniques, including SMOTE and SMOTE-Tomek, in conjunction with different classification models. Their findings indicated that hybrid approaches could effectively improve the detection of fraudulent credit card transactions by balancing the dataset while maintaining the integrity of the information (Mahesh et al., 2022). Furthermore, Makki et al. (2019) emphasized the importance of combining oversampling with undersampling methods to mitigate the effects of class imbalance, demonstrating that such hybrid approaches can lead to improved detection rates for credit card fraud.

2.1.3 Usage of SL models in financial fraud detection

SL models have become the cornerstone of fraud detection research by leveraging labeled data to learn discriminative patterns that distinguish legitimate from fraudulent transactions. They typically rely on historical examples of confirmed fraud to train predictive algorithms, which then generalize to new, unseen instances (Awoyemi et al., 2017). The following section delves into the ML and DL models which are commonly used in fraud detection studies.

2.1.3.1 ML models used in financial fraud detection

Logistic Regression is one of the most widely used statistical methods for binary classification problems, including fraud detection. It serves as a baseline model due to its interpretability and efficiency in handling large datasets. Studies have shown that while Logistic Regression can effectively identify fraudulent transactions, its performance may be limited in complex scenarios where relationships between features are non-linear (Ileberi et al., 2022; Zhang et al., 2018). However, it remains a valuable tool, particularly when combined with other methods to enhance predictive accuracy.

SVM is another powerful technique employed in fraud detection. SVM excels in high-dimensional spaces and are effective in cases where the number of dimensions exceeds the number of samples. They work by finding the optimal hyperplane that separates different classes, making them suitable for detecting fraudulent activities that may not be linearly separable (Doshi et al., 2023; Jan, 2018).

Random Forests, an ensemble learning method, utilize multiple decision trees to improve classification accuracy. This method is particularly effective in handling large datasets with numerous features, as it reduces the risk of overfitting and enhances generalization (Herland et al., 2018). Studies have demonstrated that Random Forests often yield high precision and recall rates in fraud detection tasks, making them a preferred choice for many practitioners (Nabrawi, 2023). Their ability to provide feature importance scores also aids in understanding the underlying factors contributing to fraudulent behavior (Iranzad & Liu, 2024).

XGBoost, or Extreme Gradient Boosting, has gained popularity due to its efficiency and performance in various ML competitions. It is particularly noted for

its ability to handle missing data and its robustness against overfitting (Noviandy, 2023; Hájek et al., 2022). Research indicates that XGBoost outperforms many traditional methods, achieving higher true positive rates while minimizing false positives in fraud detection applications (Hájek et al., 2022; Noviandy, 2023). Its integration with techniques like SMOTE has further enhanced its effectiveness in dealing with imbalanced datasets, a common challenge in fraud detection (Noviandy, 2023).

2.1.3.2 DL models used in financial fraud detection

Multilayer Perceptron (MLP), as a foundational architecture in DL, has been effectively utilized for credit card fraud detection. For instance, a study demonstrated that MLP could achieve high precision in identifying fraudulent transactions, emphasizing their effectiveness in handling imbalanced datasets which are typical in fraud detection scenarios (Kasasbeh & Ahmad, 2022). Furthermore, the integration of MLPs within the ensemble methods has been shown to improve performance metrics significantly, as evidenced by the Optimized Stacking Ensemble (OSE) approach, which combines MLP with other classifiers to enhance predictive accuracy (Veigas et al., 2021).

Recurrent Neural Networks (RNN), particularly LSTM and Gated Recurrent Unit (GRU), are well-suited for sequential data, making them ideal for fraud detection in time-series transactions. Research indicates that LSTMs outperform traditional RNNs in various performance metrics, achieving accuracy rates as high as 94.88% in financial statement fraud detection (Jan, 2021). Additionally, GRU has been employed in hybrid models that utilize optimization techniques to enhance detection capabilities in banking transactions (Kolli & Tatavarthi, 2020). The

sequential nature of RNN allows it to capture temporal dependencies in transaction data, which is crucial for identifying fraudulent patterns that may evolve over time (Branco et al., 2020). Bidirectional LSTM (BiLSTM) extends the capabilities of standard LSTM by processing data in both forward and backward directions, thus capturing context from both past and future states. This dual processing can significantly enhance the model's understanding of transaction sequences, leading to improved detection rates in complex fraud scenarios (Ghrib et al., 2024). The ability of BiLSTM to leverage comprehensive context makes them particularly effective in environments where transaction patterns are not static.

While Convolutional Neural Networks (CNN) are traditionally associated with image processing, their application in fraud detection has gained traction. CNN can effectively extract features from transaction data by treating sequences of transactions as spatial data, allowing for the identification of patterns that may indicate fraudulent behavior (Bello & Olufemi, 2024). For instance, Alarfaj et al. (2022) demonstrated that CNN architectures, when optimized with additional layers, significantly enhance detection accuracy in credit card fraud scenarios. Similarly, Chen and Lai (2021) proposed a deep CNN model specifically designed for credit card fraud detection, highlighting its ability to identify unknown attack patterns through big data analytics. These findings are corroborated by a study, which categorized fraud detection into anomaly and misuse detection, emphasizing the role of DL in recognizing novel fraudulent behaviors (Pumsirirat & Liu, 2018).

Transformer models, introduced by Vaswani et al. in 2017, has revolutionized the field of natural language processing (NLP) and has been widely adopted across various applications due to its unique architecture based on self-attention mechanisms. This architecture allows for the parallel processing of input data,

significantly enhancing the efficiency and scalability of model training compared to traditional RNN (Geneva, 2020; He et al., 2020). The self-attention mechanism enables the model to weigh the importance of different words in a sentence, facilitating a deeper understanding of context and relationships within the text (He et al., 2020; Wolf et al., 2020).

Transformer models have recently begun to be employed for the purposes of fraud detection, demonstrating a significant expansion beyond their initial applications in NLP and other domains. In a recent study by Yu et al. (2024), a comprehensive comparison is conducted between a transformer-based model known as TabNet and traditional ML algorithms for credit card fraud detection purposes. The study highlights the evolving capabilities of transformer architectures in handling complex, tabular data and their potential advantages over more conventional methods within the domain of fraud analytics. Hu et al. (2023) states the potential of pre-trained transformer models, specifically BERT4ETH, in detecting fraud within Ethereum transactions. They argue that the versatility of transformers allows them to be fine-tuned for multiple fraud detection tasks with minimal adjustments, thereby enhancing their applicability across different fraud scenarios. This adaptability is crucial, as fraud patterns are often dynamic and require models that can evolve with changing data landscapes.

2.1.4 Usage of USL models in financial fraud detection

USL techniques are particularly advantageous in fraud detection due to their ability to identify anomalies without the need for labeled datasets. This characteristic is crucial in scenarios where fraudulent activities are rare and historical data may not adequately represent new fraud patterns. For instance, Debener et al. (2023)

emphasize the potential of Isolation Forests as an USL method for insurance fraud detection, noting that modern USL techniques have been underutilized in this context. Similarly, another researcher points out a significant research gap in applying USL techniques, particularly clustering algorithms, for financial fraud detection, indicating that this area remains largely unexplored (Husnaningtyas, 2023). These findings underscore the necessity for further investigation into USL methods to enhance fraud detection capabilities.

Among the various USL techniques, Autoencoders and RBM have gained prominence due to their effectiveness in reconstructing normal transaction patterns and identifying deviations that may signify fraud. For example, one research explored the use of unsupervised DL techniques, including Autoencoders, for insurance fraud detection, revealing their capacity to uncover hidden patterns in data that may indicate fraudulent claims (Gomes et al., 2021). Additionally, a comparative study on credit card fraud detection emphasizes the advantages of USL methods, including Autoencoders, in identifying fraudulent transactions without prior labeling (Niu, 2019).

RBM also plays a significant role in unsupervised fraud detection. Their probabilistic nature allows them to model the distribution of normal transactions effectively, making it easier to identify outliers. The work by Chen et al. (2021) discusses the integration of RBMs in fraud detection systems, highlighting their ability to learn from unlabeled data and adapt to new fraud patterns. This adaptability is essential, as fraud patterns can change quickly, and traditional SL models may struggle to keep pace with these changes.

Pumsirirat and Liu (2018) developed a model that combines Autoencoders and RBMs to detect anomalies in credit card transactions, demonstrating that these

models can effectively reconstruct normal behavior and highlight fraudulent activities. This approach is particularly relevant as fraudsters often mimic legitimate user behavior, making it challenging for traditional methods to detect anomalies based solely on historical data. Furthermore, Sharma et al. (2022) reinforce this notion by illustrating how DL-based Autoencoders can adapt to evolving fraud patterns, thus improving detection accuracy.

In conclusion, this review highlights the diverse and innovative applications of AI in fraud detection, addressing key challenges and methodologies across various domains. By exploring the role of datasets—public, private, and synthetic—this review emphasizes the foundational importance of data in building effective detection systems. It examines strategies to mitigate class imbalance, a critical issue in fraud detection, and evaluates the effectiveness of both traditional and advanced sampling techniques. The exploration of ML models showcases the progression from ML algorithms to DL architectures, including transformers, underlining their ability to capture complex fraud patterns. Furthermore, the discussion of USL methods, such as Autoencoders and RBMs, underscores their potential in detecting anomalies without reliance on labeled data. Together, these insights provide a comprehensive understanding of how AI is revolutionizing fraud detection, paving the way for more adaptive and efficient solutions in combating evolving fraudulent activities.

2.2 XAI and its applications in fraud detection

This section examines commonly used XAI techniques and their taxonomies, followed by an analysis of their application in fraud detection. The rapid advancement of AI technologies has led to their widespread application across various sectors. However, the inherent complexity and opacity of many AI models

pose significant challenges regarding their interpretability and explainability. In most of the implication areas of AI, it is not possible to accept risks of relying on systems that cannot justify their decisions. XAI seeks to tackle this challenge by promoting greater transparency in AI. Its goal is to develop methods that enhance model explainability without compromising performance. (Adadi & Berrada, 2018)

XAI is fundamentally concerned with making AI systems more understandable to users, thereby fostering trust and facilitating informed decision-making. Increasing reliance on AI for critical decisions necessitates a deeper understanding of these systems' workings to build trust among users (Tiwari, 2023). This sentiment is cited by Lozano-Murcia et al. (2023), who argue that transparency is essential for ML models, particularly in high-stakes domains like finance and insurance, where auditability and accountability are paramount. The challenge lies in developing frameworks that simplify the explainability of complex models, which often operate as "black boxes".

Arrieta et al. (2020) provide a taxonomy of XAI methods, identifying opportunities and challenges in the field. They emphasize the intersection of data fusion and explainability, advocating for a responsible AI framework that prioritizes fairness and accountability. One of the core contributions of the article is its comprehensive taxonomy of XAI methods, which categorizes approaches based on various criteria, including the nature of the explanations provided and the underlying methodologies employed. The authors delineate two primary categories: transparent models, which are inherently interpretable (e.g., linear regression, decision trees), and opaque models, which require additional techniques to elucidate their decision-making processes (e.g., DL models) (Arrieta et al., 2020; Das, 2020). This classification is crucial for understanding the landscape of XAI techniques and their

applicability in different contexts. Post-hoc explainability techniques also discussed in the article for the opaque models. Model-agnostic techniques for post-hoc explainability are designed so that they can be applied to any ML model, regardless of the model’s internal structure or learning mechanism. According to the article, these techniques often rely on three main strategies: (1) model simplification, where a complex model is approximated by a simpler, more interpretable surrogate (e.g., rule extraction techniques such as LIME and Genetic Rule Extraction); (2) feature relevance estimation, which assigns importance scores to input features (e.g., SHAP, sensitivity analysis, and influence functions) to show how each feature contributes to a prediction; and (3) visual explanations, which present feature relevance or other interpretable aspects through plots or heatmaps to better communicate model behavior. By separating the explanatory method from the trained model, these approaches accommodate diverse architectures and data types, offering flexibility for practitioners seeking insight into opaque models.

LIME is one of the most common post-hoc techniques designed to approximate the decision boundary of any complex, black-box ML model with a simpler, interpretable surrogate in the local region around a target instance (Ribeiro et al., 2016). LIME perturbs the input features to generate synthetic samples, obtains the black-box model’s predictions on these samples, and then fits a simpler model (often linear) to understand which features are most important for a specific prediction (Ribeiro et al., 2016). Another commonly practiced method, SHAP, employs cooperative game theory, particularly Shapley values, to quantify how each feature contributes to a prediction (Lundberg & Lee, 2017). SHAP provides explanations with strong theoretical guarantees, such as local accuracy, consistency, and missingness, making it relatively robust for both global and local interpretability

(Lundberg & Lee, 2017). While LIME often requires fewer computational resources and is intuitive to implement for generating local explanations, SHAP can give more nuanced attributions at the expense of potentially higher computational cost (Psychoula et al., 2021). Both techniques have gained broad adoption in practice, underscoring the increasing need for transparency and accountability in modern AI systems.

Furthermore, the authors address the challenges associated with implementing XAI in real-world scenarios. One significant challenge is the trade-off between model accuracy and interpretability; often, more complex models yield better performance but at the cost of transparency (Arrieta et al., 2020; Dwivedi et al., 2023). The article emphasizes the need for robust evaluation metrics to assess the effectiveness of XAI methods, as current metrics may not adequately capture the nuances of explainability (Gerlach et al., 2022). Moreover, the integration of XAI with other critical areas, such as fairness and bias mitigation in AI, presents additional complexities that require interdisciplinary collaboration and innovative solutions. This perspective aligns with the findings of a research which argues that the lack of understanding of AI systems hinders their practical implementation, reinforcing the need for XAI methodologies that enhance model transparency (Clement et al., 2023).

Moreover, it is crucial to acknowledge that XAI does not exist in isolation; its influence reaches into the societal, legal, and policy realms where transparency and accountability have profound implications. Therefore, a comprehensive understanding of XAI must address not only how models make decisions but also the broader impact of these decisions on stakeholders, institutions, and regulatory frameworks. The legal domain presents unique challenges for explainability. As

Górski et al. (2020) note that legal decision-making requires not only data transparency but also a clear understanding of the decision-making processes involved. This dual requirement underscores the need for a comprehensive approach to explainability that encompasses both ontological (data) and epistemic (model) transparency. Coyle and Weller highlight the policy challenges associated with explainable ML systems, particularly in contexts requiring political accountability (Coyle & Weller, 2020). This concern is echoed by McKay, who discusses the real-world factors that complicate the understanding of algorithmic decisions, emphasizing that transparency alone may not suffice (McKay, 2022).

The integration of XAI into fraud detection systems has become increasingly vital as organizations seek to enhance transparency and trust in automated decision-making processes. Creating automated decision-making systems involves ensuring that the rationale behind decisions is accessible and understandable, emphasizing accuracy and reliability, identifying potential biases in underlying data and features, and considering the societal and individual impacts within the system's deployment context (Psychoula et al., 2021). Mill et al. (2023) highlight the importance of XAI considering recent regulatory changes that necessitate the adoption of advanced ML techniques. Their study explores the operational environment of credit card transactions, which is essential for the successful implementation of XAI solutions. By understanding the complexity of transaction data, researchers can develop more effective and explainable models that enhance the detection of fraudulent activities. Cirqueira et al. (2021) emphasize the necessity of aligning fraud experts' requirements with XAI capabilities to build trust in AI predictions. They propose a user-centric framework for integrating expert knowledge into XAI decision support

systems, emphasizing its role in enhancing trust and effectiveness in real-world fraud detection.

Psychoula et al. (2021) provide a comprehensive overview of various explainability techniques such as LIME and SHAP which are applicable to fraud detection. They found that both LIME and SHAP are effective tools for explaining transaction fraud models, but they involve trade-offs. While LIME provides faster single-instance explanations, SHAP offers greater reliability, making the choice dependent on balancing model complexity, explanation reliability, and runtime. Their analysis also reveals that explainability is not merely a technical requirement but a critical factor in the acceptance and effectiveness of AI solutions in combating fraud. In a more specialized approach, Qin et al. (2022) introduce a graph-based method for fraud detection that leverages neural meta-graph search techniques. This innovative framework allows for the identification of complex relationships within data that traditional methods may overlook. The explainability of the model is enhanced through the visualization of these relationships, providing stakeholders with insights into the reasoning behind fraud detection outcomes.

Li et al. (2024) created SEFraud for Industrial and Commercial Bank of China (ICBC), which is a novel self-explainable graph-based fraud detection framework designed to address the limitations of post-hoc methods like GNNExplainer, which are computationally expensive and impractical for real-world applications. SEFraud integrates fraud detection with interpretability, achieving significant improvements in AUC (8.6%) and Recall (8.5%) compared to the second-best method, along with a 10 times faster inference time. It employs customized heterogeneous graph transformer networks with learnable feature and edge masks, enhanced by a triplet loss to improve mask learning. It delivers accurate,

interpretable results that align with expert business knowledge, demonstrating its efficiency and scalability in large-scale online services.

Zhu (2022) explores cross-domain fraud detection by modeling user behavior sequences with a hierarchical explainable network. This approach not only captures the nuances of user interactions across different domains but also provides explanations that are crucial for understanding the model's predictions. The hierarchical structure of the network facilitates a clearer interpretation of how various factors contribute to the likelihood of fraud. Farbmacher et al. (2022) present an explainable attention network specifically designed for claims management fraud detection. Their model utilizes attention mechanisms to highlight the most relevant features influencing fraud predictions, thereby enhancing interpretability. This focus on explainability is essential for claims managers who must justify decisions based on AI outputs.

In conclusion, by making complex models understandable, XAI fosters accountability and aligns automated decisions with human oversight. In fraud detection, XAI techniques such as SHAP and SEFraud have proven instrumental in enhancing transparency while improving detection accuracy and scalability in large-scale applications. As regulatory and societal demands for fairness and transparency increase, XAI continues to evolve, offering scalable solutions that balance complexity with usability, ultimately enabling more reliable and ethical AI adoption.

Table 1 provides a comparative overview of prior studies and their methodological approaches in the domain of credit card fraud detection. The studies included in the table are chosen due to their focus on credit card fraud detection and comprehensive work across different models and techniques. Table 1 categorizes these studies based on their dataset type (public or private), learning types (SL or

USL), models used (ML models, DL models, or transformer models), and additional analytical techniques such as feature engineering, sampling techniques, and XAI. The table highlights the contributions of various authors and contrasts their approaches with our study, which comprehensively addresses all these aspects, showcasing a more holistic approach.

Table 1. Summary of the Most Relevant Studies in Literature

References	Datasets		Learning Types		Models			Other Analysis		
	Public	Private	SL	USL	MLM	DLM	TM	FE	ST	XAI
Ghrib et al., 2024	✓		✓		✓	✓			✓	
Yu et al., 2024	✓		✓		✓		✓		✓	
Alharbi et al., 2022	✓		✓		✓	✓				
Alarfaj et al., 2022	✓		✓		✓	✓			✓	
Psychoula et al.,2021	✓		✓	✓	✓	✓				✓
Veigas et al., 2021	✓		✓	✓	✓	✓			✓	
Niu et al., 2019	✓		✓	✓	✓	✓			✓	
Pumsirirat&Liu,2018	✓			✓		✓				
Roy et al., 2018		✓	✓			✓		✓	✓	
Gomez et al., 2018		✓	✓			✓		✓	✓	
Our Study	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: SL: Supervised Learning, USL: Unsupervised Learning, MLM: ML Models, DLM: DL Models, TM: Transformer Models, FE: Feature Engineering, ST: Sampling Techniques, XAI: Explainable AI

Our study makes significant contributions to the field of credit card fraud detection by addressing several limitations present in the existing literature. Unlike most studies that rely exclusively on public datasets, this research utilizes both private and public datasets, enabling a more comprehensive evaluation of detection methods across diverse data types. Furthermore, additional features were generated through feature engineering, which improved the accuracy of the models—a step often neglected in previous studies where datasets are typically used without generating new features. While many studies employ sampling techniques to address class imbalance, this work integrates these methods alongside a novel combination of ML models, DL models, and transformer models, representing a holistic approach not previously documented. This study employs both SL and USL techniques to compare their performance, addressing a gap in the literature where most studies focus exclusively on one approach. Additionally, it incorporates XAI methods, including LIME, SHAP, and DeepLift, to provide interpretability for the models, addressing the limited application of such techniques in fraud detection research. These contributions collectively advance the understanding and development of robust, interpretable, and effective fraud detection systems.

CHAPTER 3

METHODOLOGY

This section outlines the methodological approach undertaken in this study to develop and validate the credit card fraud detection models. It begins with an overview of the datasets utilized, including both private and publicly available data, followed by a description of the sampling techniques applied to address computational and data balance challenges. Next, the algorithms employed for fraud detection are discussed, detailing the rationale behind their selection and implementation. The process of hyperparameter tuning is also described, emphasizing how optimal model configurations were identified to enhance predictive performance. Evaluation metrics are presented to assess the models comprehensively, ensuring their effectiveness in detecting fraudulent transactions. Finally, the explainability methods are introduced, highlighting how model predictions are interpreted and insights into fraud patterns are derived. For all implementations and evaluations carried by this study, Python was employed as the primary programming language, utilizing a range of libraries and frameworks.

3.1 Datasets

The private dataset utilized for this study comprises credit and debit card transaction data obtained from a private bank in Türkiye. Fraudulent transactions are identified and labeled by a rule-based fraud detection system, supplemented by customer-reported complaints. There are 14,146 fraudulent transactions recorded in 2023, alongside hundreds of millions of non-fraudulent transactions in the database. Since it is not feasible to process millions of transactions due to computational limitations,

random sampling was performed on the non-fraudulent transactions, resulting in a subset of 141460 non-fraudulent transactions being included in the dataset used for the study. This count was selected to facilitate the application of majority voting with RU, where 10 different subsets of 14,146 non-fraudulent transactions were used to train the models. The dataset includes 11 features, of which 8 are transactional attributes and 3 are related to customer demographics, excluding customer and card IDs.

Certain features within the dataset were directly extracted from the credit card transactions database. These features represent standard attributes of credit card transactions and are routinely maintained in the organization's fraud database. Unfortunately, due to confidentiality restrictions, the exact features cannot be disclosed. However, some examples include transaction amount, merchant category, security type of transaction, and payment system provider, which are commonly found in credit card transaction datasets.

The dataset was further enriched by merging transaction-level data with customer demographic information from a separate customer database. This integration added features such as customer age, residency, and other basic attributes, which were retrieved using SQL queries. These features provide additional context for fraud detection, complementing the transactional data. Additional features were obtained through feature engineering, including customer profiles, credit card profiles, and the purchasing patterns of customers.

To validate the results, a publicly available dataset, the Kaggle Credit Card Fraud Detection dataset, was employed. This dataset, introduced in the academic study "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy" by Dal Pozzolo et al. (2015), contains anonymized transaction records

collected from European cardholders over two days in September 2013. It includes 284,807 transactions, of which 492 are labeled as fraudulent, resulting in a highly imbalanced class distribution. The features consist of 28 principal components derived from a Principal Component Analysis (PCA) transformation, alongside the original transaction amount and timestamp features. The PCA-transformed features are labeled V1 through V28 to ensure privacy. This dataset serves as a benchmark for evaluating and comparing the performance of fraud detection models in a reproducible manner. It complements the analysis of the private dataset by providing an external validation of the findings.

3.2 Sampling methods

In the context of ML and data analysis, effective sampling methods are crucial for addressing class imbalance, particularly when the dataset exhibits a significant disparity between the majority and minority classes. This section describes the sampling methods employed in this study, specifically focusing on RU, oversampling using SMOTE, and the implementation of majority voting through repeated RU. All sampling techniques were applied to the training set, while the imbalanced test set remained consistent across all evaluations.

3.2.1 Random undersampling (RU)

RU is a widely recognized technique utilized to mitigate class imbalance by reducing the number of instances in the majority class. This method involves randomly selecting a subset of instances from the majority class while retaining all instances from the minority class, thereby achieving a more balanced dataset. It can prevent the model from becoming biased toward the majority class but carries the risk of losing

valuable information from the discarded samples (Kubat & Matwin, 1997). Despite its limitations, RU is computationally efficient and often serves as a baseline technique in handling class imbalance.

3.2.2 Oversampling with SMOTE

Random oversampling is a technique for addressing class imbalance by increasing the representation of the minority class in the dataset. This is achieved by randomly duplicating instances of the minority class until its size matches that of the majority class. While this method effectively balances the class distribution and ensures that the minority class is adequately represented, it carries certain risks. The duplication of minority class instances can lead to overfitting, where the model becomes overly specialized to the repeated examples, reducing its ability to generalize to unseen data (He & Garcia, 2009).

SMOTE (Chawla et al., 2002) was introduced to address the challenges posed by random oversampling, which can lead to overfitting by duplicating minority class instances. SMOTE generates synthetic examples of the minority class by interpolating between existing minority class samples and their nearest neighbors. This process results in a more generalized representation of the minority class in the feature space. SMOTE has been widely adopted due to its ability to enhance classifier performance on imbalanced datasets without duplicating existing samples. SMOTE operates by identifying k -nearest neighbors in the feature space for each instance of the minority class. A synthetic sample is then generated by selecting one of these neighbors and interpolating a feature vector between the original instance and the chosen neighbor. This process is iteratively repeated until the minority class achieves the desired size (Chawla et al., 2002).

3.2.3 Majority voting using RU

To further enhance the dataset balancing process, this study implemented a hybrid ensemble approach involving majority voting with RU. This technique performs RU multiple times, creating diverse subsets of the majority class. In this study, ten iterations of RU were conducted. Each iteration resulted in a unique balanced dataset, and ML models were independently trained on these datasets.

During the prediction phase, the outputs from the models trained on the undersampled datasets were aggregated, and the majority vote determined the final classification. By generating multiple balanced datasets through RU, the method introduces diversity into the training process, which helps mitigate the risk of overfitting or bias associated with a single dataset. The ensemble nature of the method enhances the stability and robustness of predictions, particularly for noisy or complex datasets. Furthermore, the process reduces sensitivity to the randomness inherent in a single instance of RU, ensuring more reliable model performance (Sadhukhan et al., 2021).

In summary, by employing these sampling techniques, the study ensured that the dataset was sufficiently balanced, allowing ML algorithms to focus on distinguishing fraudulent transactions without being influenced by class imbalance. The selection of these methods aligns with best practices in handling imbalanced datasets in fraud detection, emphasizing the importance of methodological rigor and robustness.

3.3 Algorithms

This section provides a detailed explanation of the ML algorithms employed in this study, including their mathematical foundations, architecture, and key characteristics.

Each algorithm's strengths and limitations are discussed in the context of fraud detection. For clarity and comprehensiveness, formulas, taxonomies, and, where applicable, visual representations of architectures are provided.

3.3.1 Logistic Regression

Logistic Regression (Cox, 1958) is a statistical model designed for binary classification tasks, where the goal is to predict the probability of an outcome belonging to one of two classes. It models this probability using a sigmoid (logistic) function. The Logistic Regression model expresses the probability $P(y = 1|X)$, where y is the binary outcome variable, and X represents the independent variables. This probability is modeled using the sigmoid function which is shown in Equation (1):

$$P(y = 1|X) = \frac{1}{1+e^{-z}} \quad (1)$$

where z is a linear combination of the independent variables:

$$z = \beta_0 + \sum_{i=1}^n \beta_i X_i \quad (2)$$

where β_0 is the intercept, β_i represents the coefficients of the independent variables X_i , and n is the number of features in the dataset.

The sigmoid function transforms the linear input z into a value between 0 and 1, which represents the predicted probability. This probabilistic output allows Logistic Regression to be used for classification by applying a threshold. Typically, a threshold of 0.5 is used to classify a sample into one of the two classes. The decision rule can be expressed as in the Equation (3):

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|X) \geq 0.5, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Logistic Regression optimizes its parameters ($\beta_0, \beta_1, \dots, \beta_n$) by minimizing the binary cross-entropy loss function, also known as the log-loss. The loss function is given by:

$$\mathcal{L}(\beta) = -\frac{1}{m} \sum_{i=1}^m [y_i \log(P(y_i|X_i)) + (1 - y_i) \log(1 - P(y_i|X_i))] \quad (4)$$

where m is the number of samples, y_i is the actual class label for the i -th sample, and $P(y_i|X_i)$ is the predicted probability.

This model assumes a linear relationship between features and the log-odds of the outcome. While simple and computationally efficient, Logistic Regression is limited to problems with linearly separable data. Despite this, it serves as a strong baseline for fraud detection due to its probabilistic output and interpretability.

3.3.2 Support Vector Machine (SVM)

SVM (Cortes & Vapnik, 1995), is a SL algorithm used for classification and regression. It seeks to find the optimal hyperplane that separates data points from different classes while maximizing the margin, defined as the distance between the hyperplane and the closest points, known as support vectors. These support vectors are critical in determining the decision boundary. The SVM optimization problem is expressed by Equation (5):

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (5)$$

subject to Equation (6):

$$y_i(w \cdot x_i + b) \geq 1, \forall i \quad (6)$$

where w is the weight vector, b is the bias term, x_i are input features, and y_i are binary class labels (± 1). The norm $\|w\|$ is minimized to maximize the margin.

For non-linearly separable data, SVM employs slack variables ξ_i to allow some misclassifications, leading to the soft-margin formulation shown in Equation (7):

$$\min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^m \xi_i \quad (7)$$

where C is a regularization parameter balancing margin maximization and classification errors.

To handle non-linear boundaries, the kernel trick maps data to a higher-dimensional space where a linear hyperplane can be applied. Common kernels include linear, polynomial, and radial basis function (RBF) kernels.

SVM's decision rule is given by Equation (8):

$$f(x) = \text{sign}(w \cdot x + b) \quad (8)$$

SVM is effective in high-dimensional spaces and provides robust performance for complex decision boundaries. However, it is computationally intensive for large datasets and sensitive to kernel and parameter selection.

3.3.3 Random Forests

Random Forests (Breiman, 2001) is an ensemble learning method designed to improve the performance and stability of individual decision trees. It operates by constructing multiple decision trees during training and aggregating their predictions to achieve better generalization and reduce overfitting. The approach leverages two key concepts: bootstrap aggregating (bagging) and random feature selection.

The algorithm works as follows: For a dataset with m samples and n features, multiple decision trees are trained independently. Each tree is built on a bootstrap sample (a random sample of size m , drawn with replacement from the original dataset). At each split in the tree, instead of considering all n features, only a random subset of k features is selected. This randomness reduces correlation between individual trees, leading to improved model performance.

For a classification problem, the Random Forest predicts the class label $\hat{y}$ for a given input X by aggregating the predictions from all T trees as shown in Equation (9):

$$\hat{y} = \text{mode}\{h_t(X), t = 1, 2, \dots, T\} \quad (9)$$

where $h_t(X)$ is the prediction of the t -th tree, and the mode function selects the most frequent class among the trees.

For regression equation, the calculation of prediction is the average of the outputs of all trees is shown in Equation (10):

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(X) \quad (10)$$

Each decision tree is constructed to minimize the impurity at each node. Common impurity measures include Gini impurity and entropy. For Gini impurity, used in classification tasks, the impurity at a node is shown in Equation (11):

$$G = 1 - \sum_{i=1}^C p_i^2 \quad (11)$$

where p_i is the proportion of samples belonging to class i , and C is the number of classes.

Random Forests mitigate overfitting by averaging diverse trees, ensuring robustness to noise and capturing non-linear relationships. While computationally intensive and less interpretable than simpler models, they handle imbalanced data and complex interactions effectively, making them ideal for fraud detection.

3.3.4 XGBoost

XGBoost, or eXtreme Gradient Boosting (Chen & Guestrin, 2016), is a scalable and efficient implementation of gradient-boosted decision trees. It is designed to optimize both speed and performance through innovations such as regularization, parallelization, and a novel tree-building approach. XGBoost builds trees

sequentially, where each tree attempts to minimize the errors of the preceding ones, creating a strong ensemble model.

The core objective of XGBoost is to minimize a regularized loss function, combining a differentiable loss function for prediction errors and a complexity term to prevent overfitting. For a dataset with n samples, the objective function for tree t can be written as Equation (12):

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (12)$$

where $l(y_i, \hat{y}_i)$ is the loss function, often squared error for regression or log-loss for classification, $f_t(x)$ represents the t -th decision tree, $\Omega(f_t) = \gamma T + \frac{1}{2} \lambda |w|^2$ is the regularization term, penalizing the number of leaves T and leaf weights w using hyperparameters γ and λ .

XGBoost uses second-order Taylor expansion to approximate the loss function, enabling efficient optimization. The approximate objective becomes:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \quad (13)$$

where g_i and h_i are the first and second derivatives of the loss function with respect to the predictions which is shown in Equation (13).

XGBoost builds trees to minimize residual errors, using feature splits based on gain and metrics like gain, cover, or frequency to evaluate feature importance. Innovations like weighted quantile sketch for missing data, sparsity handling, and parallelized construction ensure efficiency, while regularization prevents overfitting. Its scalability, accuracy, and ability to handle complex interactions make it ideal for fraud detection and large-scale problems.

3.3.5 Multilayer Perceptron (MLP)

MLP is a specific type of ANN characterized by their fully connected layers and feedforward architecture. ANNs are a foundational concept in ML, inspired by the structure and function of biological neural networks. They were first introduced by McCulloch and Pitts in 1943, who proposed a simplified model of a neuron, capable of binary decision-making. ANNs consist of layers of interconnected nodes (neurons), each of which performs a mathematical operation on its inputs to produce an output.

MLPs are first conceptualized by Rosenblatt in 1958 with the introduction of the perceptron, MLPs expanded on this concept by incorporating multiple hidden layers and nonlinear activation functions, enabling them to approximate complex, nonlinear functions. An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is fully connected, meaning every neuron in a layer is connected to every neuron in the subsequent layer. This connectivity ensures that the network can capture intricate patterns in the data. Figure 1 visualizes the architecture of an ANN with a single hidden layer.

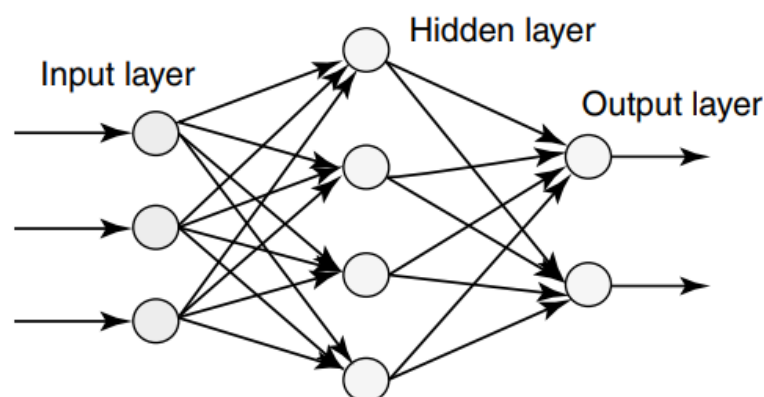


Figure 1. ANN architecture

Source: [Handbook of Measuring System Design, 2005]

The forward pass of an MLP involves computing the output of each layer sequentially, displayed in Equation (14):

$$a^{[l]} = f(W^{[l]}a^{[l-1]} + b^{[l]}) \quad (14)$$

where $W^{[l]}$ and $b^{[l]}$ are the weights and biases for layer l , $a^{[l-1]}$ is the activation from the previous layer, and f is the activation function. The final output layer typically uses a softmax function for classification tasks or a linear activation for regression tasks.

MLPs are trained using stochastic gradient descent (SGD) or its variants. This is achieved using backpropagation, an algorithm introduced by Rumelhart, Hinton, and Williams in 1986. Backpropagation computes the gradient of the loss function with respect to each weight in the network using the chain rule and updates the weights through gradient descent shown in Equation (15):

$$w_i \leftarrow w_i - \eta \frac{\partial L}{\partial w_i} \quad (15)$$

where η is the learning rate. Over multiple iterations, the network learns to adjust its parameters to better fit the data.

Activation functions play a critical role in MLPs, introducing nonlinearity to the network. The sigmoid function maps inputs to the range (0, 1), making it useful for binary classification tasks. The ReLU function addresses the vanishing gradient problem by providing non-saturating gradients. For multi-class classification, softmax activation normalizes the outputs to represent probabilities.

MLPs are foundational to many ML applications and serve as a basis for more complex architectures. Despite their simplicity compared to modern DL models, they remain effective for structured data and smaller-scale problems, demonstrating the power and versatility of neural networks.

3.3.6 Deep Neural Network (DNN)

DNNs are an extension of ANNs that include multiple hidden layers between the input and output layers. While traditional ANNs typically have one or two hidden layers, DNNs incorporate several layers, enabling the network to learn hierarchical representations of data. This hierarchical learning allows DNNs to extract simple features in the lower layers and progressively combine them to form more complex representations in the higher layers. The foundation for modern DNNs was established by Hinton et al. (2006), who introduced an efficient layer-wise pretraining algorithm, making it feasible to train deep architectures effectively.

The mathematical formulation of a DNN builds on the ANN framework. For each layer l in the network, the computation can be expressed as shown in Equation (16):

$$z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}, \quad a^{[l]} = f(z^{[l]}) \quad (16)$$

where $W^{[l]}$ represents the weight matrix, $b^{[l]}$ is the bias vector, $a^{[l-1]}$ is the activation from the previous layer, and f is the activation function. The network's depth—measured by the number of layers—enables it to model complex, high-dimensional functions. However, increased depth introduces challenges such as vanishing gradients, overfitting, and computational complexity.

To address these challenges, various techniques are employed. For example, batch normalization standardizes the inputs to each layer, improving convergence and stability. Dropout, a regularization technique, prevents overfitting by randomly deactivating a fraction of neurons during training. Additionally, advanced optimizers such as Adam, which adapt learning rates based on the first and second moments of the gradient, enhance the training process.

DNNs have revolutionized fields such as computer vision, NLP, and speech recognition. Their ability to process and analyze large-scale data has been a key factor in the success of DL applications.

3.3.7 Recurrent Neural Networks (RNN)

RNN is a class of neural networks designed to process sequential data, such as time-series data, text, or audio. Unlike feedforward neural networks, RNNs have connections that form cycles, allowing information to persist across time steps. This capability was first conceptualized by Hopfield in 1982 and later formalized into RNNs by Rumelhart, Hinton, and Williams in 1986, who introduced the concept of backpropagation through time (BPTT) for training recurrent structures.

The core idea of an RNN is to use a hidden state, h_t , which captures information from the previous time step as shown in Equation (17). At each time step t , the network computes:

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \quad (17)$$

where h_t is the hidden state at time t , h_{t-1} is the hidden state from the previous time step, x_t is the input at time t , W_h , W_x are weight matrices for the hidden state and input, b is bias vector and f is activation function (commonly tanh or ReLU).

The output at each time step is computed as shown in Equation (18):

$$y_t = g(W_y h_t + b_y) \quad (18)$$

where g is often a softmax or sigmoid function, depending on the task.

While RNNs excel at capturing dependencies in sequential data, they suffer from the vanishing and exploding gradient problems, which hinder learning long-term dependencies. These limitations led to the development of advanced architecture like GRUs and LSTMs.

3.3.8 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU), introduced by Cho et al. in 2014, is a variant of the RNN designed to address the vanishing gradient problem and improve the modeling of long-term dependencies. GRUs use gating mechanisms to control the flow of information, enabling the network to retain relevant information over longer sequences while discarding irrelevant details.

The GRU simplifies the architecture of LSTM networks by combining the forget and input gates into a single update gate. At each time step, the GRU computes the following:

$$\begin{aligned} f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ \tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{19}$$

where z_t is update gate, determines how much of the past information to retain, r_t is reset gate, determines how much of the past state to forget, $\tilde{h}_t$ is candidate hidden state, h_t is current hidden state, and $\odot$ is element-wise multiplication.

GRUs are efficient and suitable for fraud detection tasks that require analyzing large-scale transactional data for anomalies. Their gating mechanism allows them to retain relevant patterns over longer sequences without incurring excessive computational costs. This makes GRUs particularly effective for fraud detection systems that prioritize both speed and accuracy.

3.3.9 Long Short-Term Memory (LSTM)

LSTM networks (Hochreiter & Schmidhuber, 1997), are another variant of RNNs designed to overcome the limitations of traditional RNNs. LSTMs use memory cells and gating mechanisms to selectively retain, update, and discard information, enabling them to capture long-term dependencies effectively. An LSTM cell has three gates: forget gate, input gate, and output gate. At each time step t , the LSTM computes as shown in Equation (20) above.

LSTMs are particularly well-suited for tasks requiring the modeling of long-term dependencies, such as language modeling, time-series forecasting and fraud detection. Their memory cells allow them to retain and process complex sequential patterns, making them ideal for identifying subtle and long-term fraudulent behaviors. However, their computational complexity may require careful optimization in real-time systems.

3.3.10 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM networks (Schuster & Paliwal, 1997) extend the LSTM by incorporating two separate LSTMs: one processes the sequence in the forward direction, and the other processes it in the backward direction. This allows the network to capture both past and future context for each time step, making it particularly effective for tasks such as text classification, named entity recognition, and machine translation.

The forward and backward hidden states are computed as shown in Equations (21) and (22):

$$\vec{h}_t = \text{LSTM}_{\text{forward}}(x_t, \vec{h}_{t-1}) \quad (21)$$

$$\overleftarrow{h}_t = \text{LSTM}_{\text{backward}}(x_t, \overleftarrow{h}_{t+1}) \quad (22)$$

The final hidden state is obtained by concatenating the forward and backward states as shown in Equation (23):

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (23)$$

BiLSTMs excel in tasks where understanding both past and future context is critical. Their ability to capture bidirectional dependencies makes them a powerful tool for various NLP applications. They are particularly effective for detecting fraud in scenarios where surrounding transactions influence the likelihood of an anomaly. Despite their benefits, their high computational cost may make them more suitable for offline analysis or high-stakes investigations.

3.3.11 Convolutional Neural Networks (CNN)

CNNs are a class of DL models specifically designed for processing structured data such as images, though they are also applicable to other domains with grid-like data structures. CNNs were first introduced by LeCun et al. in 1989, with their success popularized by the LeNet architecture. Unlike traditional fully connected networks, CNNs use convolutional layers to extract spatial hierarchies of features from input data.

The core operation in CNNs is the convolution, where a filter (or kernel) slides over the input data to compute feature maps. This is mathematically represented in Equation (24):

$$y_{i,j,k} = \sum_{m,n,c} x_{i+m,j+n,c} \cdot w_{m,n,c,k} + b_k \quad (24)$$

where x is the input, w is the filter, b_k is the bias, and y is the resulting feature map. Additional layers like pooling (e.g., max pooling) are used to downsample the data, reducing computational complexity and capturing dominant features. Fully

connected layers at the end aggregate the extracted features for final classification or regression.

CNNs are widely used in tasks such as image classification, object detection, and facial recognition, and can also be applied to non-image data transformed into grid-like formats. In fraud detection, CNNs are effective when transaction data is represented as visual patterns, such as heatmaps or matrices, enabling them to detect anomalies. While not traditionally suited for sequential data, CNNs can complement models like RNNs or LSTMs in hybrid approaches, leveraging their ability to extract spatial features from structured representations

3.3.12 TabTransformer

TabTransformer (Huang et al., 2020) is a DL model tailored for tabular data, which employs a transformer-based architecture, adapted from NLP, to capture intricate interactions among categorical and numerical features. The model replaces traditional embedding techniques for categorical data with a self-attention mechanism, enabling it to learn contextual embeddings that consider the relationships between different features. Its architecture consists of a feature transformer that uses multi-head self-attention to process categorical embeddings and a decision layer that aggregates these embeddings with numerical features for final predictions.

The self-attention mechanism is the core of TabTransformer's architecture. It projects input embeddings into query, key, and value matrices, computes attention scores as the scaled dot-product between queries and keys, and applies these scores to the value matrices. The attention is computed as shown in Equation (25):

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (25)$$

where Q , K , and V are the query, key, and value matrices derived from the input embeddings, and d_k is the dimensionality of the keys. Multi-head attention enhances this process by computing attention outputs across multiple subspaces, concatenating them, and applying a final linear transformation. A position-wise feedforward network follows each attention layer, transforming the embeddings through nonlinearity and projection.

TabTransformer is effective for modeling high-cardinality categorical features and capturing complex feature interactions. It has been successfully applied in classification, regression, and ranking tasks in domains such as finance, healthcare, and e-commerce. In fraud detection, it is particularly suitable because tabular datasets often include categorical and numerical features, such as transaction histories. Its ability to capture intricate patterns makes it effective in identifying fraudulent behavior. However, the model requires substantial computational resources and may underperform on small datasets, necessitating careful optimization for best results.

3.3.13 Restricted Boltzmann Machine (RBM)

RBMs are a type of stochastic neural network introduced by Smolensky in 1986 and popularized by Hinton in 2006. They are energy-based models consisting of two layers: a visible layer, representing the input data, and a hidden layer, capturing latent features. Unlike traditional Boltzmann Machines, RBMs do not have connections between neurons within the same layer, ensuring a 'restricted' structure that facilitates efficient learning.

The RBM learns a probabilistic representation of the input data by minimizing an energy function defined as in Equation (26):

$$E(v, h) = -\sum_i v_i b_i - \sum_j h_j c_j - \sum_{i,j} v_i h_j w_{ij} \quad (26)$$

where v_i is the states of the visible units, h_j is the states of the hidden units, w_{ij} are weights between visible and hidden units, b_i, c_j are the bias terms for the visible and hidden units.

The RBM models the joint probability distribution of the visible and hidden layers as shown in Equation (27):

$$P(v, h) = \frac{\exp(-E(v, h))}{Z} \quad (27)$$

where Z is the partition function, ensuring the distribution sums to 1. During training, RBMs use contrastive divergence to approximate the gradient of the log-likelihood and update weights.

RBMs are widely used for dimensionality reduction, feature extraction, and pretraining DNNs, particularly in Deep Belief Networks. They are suitable for applications like recommendation systems and collaborative filtering, where they uncover latent features in user-item interaction matrices. For fraud detection, RBMs can learn hidden patterns in transaction data, helping to identify anomalous behaviors. However, their reliance on approximate inference methods can make them computationally challenging, and their performance is often surpassed by more modern DL approaches.

3.3.14 Autoencoder

Autoencoders are unsupervised neural networks introduced in 1993 by Hinton and Zemel, designed to learn compressed representations of data by reconstructing the input at the output layer. They consist of two main parts: an encoder that compresses the input into a latent space representation and a decoder that reconstructs the input from this representation. The goal of the Autoencoder is to minimize the

reconstruction loss, often measured as the mean squared error or binary cross-entropy between the input and output shown in Equation (28):

$$L = ||x - \hat{x}||^2 \quad (28)$$

where x is the input data and $\hat{x}$ is the reconstructed data.

The encoder applies a transformation to compress the data as shown in Equation (29):

$$z = f(Wx + b) \quad (29)$$

where z is the latent representation, W and b are the weights and biases, and f is a nonlinear activation function such as ReLU or sigmoid. The decoder performs the inverse operation, reconstructing the data as shown in Equation (30):

$$\hat{x} = g(W'z + b') \quad (30)$$

where g is another activation function, and W' and b' are the weights and biases of the decoder.

Autoencoders are commonly used for dimensionality reduction, anomaly detection, and denoising. Variants like Variational Autoencoders extend the basic architecture to probabilistic models, while Sparse Autoencoders and Denoising Autoencoders introduce constraints to learn more meaningful representations. For fraud detection, Autoencoders are particularly effective in identifying anomalies, as they excel at reconstructing normal patterns in data while poorly reconstructing unusual or fraudulent activities. They are versatile and computationally efficient but require sufficient data to avoid overfitting and are limited by their reliance on the quality of the reconstruction loss as a proxy for learning meaningful representations.

3.4 Hyperparameter tuning

Hyperparameter tuning is a critical step in ML and DL model development, ensuring that the models achieve optimal performance while avoiding overfitting or underfitting. In this study, Optuna (Akiba et al., 2019), an advanced hyperparameter optimization framework, was used to perform efficient and automated hyperparameter tuning for all models.

The tuning process began by defining an objective function, which evaluates the model's performance for a given set of hyperparameters. This function accepts a trial, representing a single combination of hyperparameters, and performs the following steps. First, it samples a set of hyperparameters from the search space using Optuna's dynamic sampling strategy. Next, the model is trained on the training dataset with these hyperparameters. Finally, the model's performance is validated on a separate validation dataset by computing the f1-score for class 1 (fraudulent transactions) as the primary evaluation metric.

Mathematically, the optimization problem is shown in Equation (31):

$$\text{maximize}_{\lambda \in \mathcal{H}} \text{F1}_{\text{class 1}}(\lambda) \quad (31)$$

where λ represents a set of hyperparameters within the defined hyperparameter search space $\mathcal{H}$, $\text{F1}_{\text{class 1}}$ is the f1-score for the fraudulent class, emphasizing the critical need to minimize false negatives and false positives for class 1.

Optuna optimizes this objective function using a Tree-structured Parzen Estimator (TPE) (Bergstra et al., 2013). This approach models the relationship between hyperparameter configurations and their corresponding performance, allowing the algorithm to explore promising regions of the search space while avoiding unproductive areas.

During the optimization process, trials are pruned based on intermediate results. If a trial's performance is unlikely to surpass the current best result, Optuna terminates it early, thereby conserving computational resources. The iterative process continues until a predefined number of trials is reached or computational limits are met. The optimal hyperparameter configuration is the one that maximizes $F1_{\text{class 1}}$ on the validation set. This configuration was subsequently evaluated on the test set to ensure robust and unbiased performance assessment.

3.5 Evaluation metrics

To assess the performance of the proposed fraud detection models, a comprehensive set of evaluation metrics was utilized. Fraud detection is inherently a class-imbalanced problem, where the fraudulent transactions (class 1) are significantly outnumbered by legitimate transactions (class 0). Consequently, aggregated evaluation metrics such as overall accuracy may be misleading, as they can mask poor performance on the minority class. To address this issue, class-specific metrics were calculated and emphasized, particularly for the minority (fraudulent) class, as it holds critical importance in the context of fraud detection. These metrics include precision, recall, sensitivity, and f1-score, calculated separately for each class, in addition to the confusion matrix.

3.5.1 Confusion matrix

The confusion matrix serves as the foundation for evaluating classification performance. Confusion matrix consists of the following metrics.

- True Positives (TP): Number of fraudulent transactions correctly classified as fraud.

- True Negatives (TN): Number of legitimate transactions correctly classified as legitimate.
- False Positives (FP): Number of legitimate transactions incorrectly classified as fraud (also known as Type I errors).
- False Negatives (FN): Number of fraudulent transactions incorrectly classified as legitimate (also known as Type II errors).

3.5.2 Accuracy

Accuracy measures the proportion of correctly classified transactions to the total number of transactions is shown in Equation (37):

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (32)$$

While accuracy provides a general overview of model performance, it is not a reliable metric in class-imbalanced settings, as it may be dominated by the majority class.

3.5.3 Precision

$$\text{Precision} = \frac{TP}{TP+FP} \quad (33)$$

Precision quantifies the proportion of predicted fraud cases that are true frauds. High precision for class 1 ensures a lower false positive rate as shown in Equation (38), minimizing the misclassification of legitimate transactions as fraud.

3.5.4 Recall

$$\text{Recall} = \frac{TP}{TP+FN} \quad (34)$$

Recall measures the proportion of actual fraud cases that are correctly identified.

High recall for class 1 ensures a low false negative rate, minimizing the number of missed fraudulent transactions. Formula is shown in Equation (39).

3.5.5 F1-score

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (35)$$

The f1-score is the harmonic mean of precision and recall, providing a balanced metric that considers both false positives and false negatives. The formula is shown in Equation (40).

3.6 XAI techniques

This chapter explores a range of methodologies for feature importance and model explainability utilized in this study. From techniques rooted in impurity reduction in decision trees to advanced explainability frameworks like SHAP, LIME, and DeepLIFT, each method contributes unique strengths to the analysis. These methods address global and local interpretability needs, providing insights into both dataset-level patterns and individual predictions. The integration of these approaches ensures a comprehensive understanding of model behavior, uncovering critical features and dependencies that influence the detection of fraudulent activity.

The following sections discuss the working principles, implementation details, and relevance of each method in the context of the study, demonstrating how they collectively enhance the transparency and accountability of the applied ML and DL models.

3.6.1 Impurity based feature importance

Impurity based feature importance method (Breiman, 2001) is a widely used technique for calculating feature importance, particularly in ensemble learning models that rely on decision trees. This method is rooted in the principles of impurity reduction during the training process of decision trees. In decision tree algorithms, each feature is evaluated at potential split points to determine its ability to reduce impurity within a subset of data. Impurity is typically measured using the Gini index or entropy, which quantify the heterogeneity of the target variable within a node. When a feature is selected for a split, it divides the data into two or more subsets, ideally reducing the impurity of the child nodes relative to the parent node. The reduction in impurity achieved by this split is attributed to the feature. Over the course of training, each feature's contribution to impurity reduction is aggregated across all the trees in the ensemble. Features that consistently produce splits with significant reductions in impurity are assigned higher importance scores.

3.6.2 Shapley Additive Explanations (SHAP)

SHAP, introduced by Lundberg and Lee (2017), is a unified framework for interpreting model predictions based on cooperative game theory. By calculating Shapley values, SHAP quantifies the contribution of each feature to a given prediction. This approach guarantees consistency and fairness, ensuring that features contributing more to the prediction receive higher importance scores.

The SHAP methodology can be applied in both model-agnostic and model-specific contexts. For instance, TreeSHAP is an efficient implementation designed for tree-based models, while DeepSHAP extends the framework to neural networks. In this study, SHAP was particularly useful in understanding the behavior of both

tree-based classifiers and DL models used for fraud detection. Global SHAP values provided insights into feature importance across the dataset, identifying key drivers of fraudulent activity. Local SHAP explanations, on the other hand, highlighted the specific contributions of features to individual predictions, enabling a fine-grained understanding of why a transaction was classified as fraudulent. The application of SHAP facilitated an exploration of feature interactions, revealing non-linear dependencies that traditional feature importance methods might overlook.

3.6.3 Local Interpretable Model-agnostic Explanations (LIME)

LIME, developed by Ribeiro et al. (2016), takes a different approach to interpretability by approximating complex models locally with simpler, interpretable models. For a specific instance, LIME generates perturbations of the input data and evaluates the model's predictions on these perturbed samples. Using these results, it constructs a surrogate interpretable model (such as a linear regression) that approximates the black-box model's behavior in the vicinity of the instance of interest.

This study leveraged LIME to provide local explanations for individual transactions flagged as fraudulent. LIME's ability to isolate and rank the contributions of features for a specific instance made it highly effective in understanding edge cases and false positives. The interpretability of LIME stems from its surrogate model, which directly maps features to predicted outcomes. This simplicity is critical for stakeholders such as fraud analysts and regulatory bodies, who require transparent and comprehensible justifications for model decisions. Furthermore, as a model-agnostic tool, LIME was applicable to all types of models

used in the study, ensuring consistency in the explanation process across different architectures.

3.6.4 Permutation importance

The permutation importance (Altmann et al., 2010) method is a model-agnostic approach to assessing the importance of features by evaluating the impact of feature disruption on model performance. This method relies on the working principle that a feature's importance can be inferred from the extent to which its relationship with the target variable influences the model's predictions.

To compute permutation importance, the values of a single feature are shuffled randomly across all samples, effectively breaking any meaningful association between that feature and the target variable. The model is then re-evaluated on the modified dataset, and the resulting performance is compared to the baseline performance obtained with the original data. The difference between the baseline performance and the performance after permutation provides a quantitative measure of the feature's importance. Features that cause a significant drop in model performance when permuted are considered more important.

This method is particularly suitable for evaluating feature importance in complex models such as DNNs, where traditional techniques like impurity-based importance may not be directly applicable. Permutation importance captures the effect of features in a non-linear and high-dimensional context, reflecting their true influence on the model's behavior.

3.6.5 Deep Learning Important Features (DeepLIFT)

DeepLIFT, introduced by Shrikumar et al. (2017), is a gradient-based method tailored for DL models. Unlike traditional gradient methods that measure instantaneous changes, DeepLIFT compares the activation of neurons to a reference input, decomposing predictions into contributions from each input feature. This method captures both direct and indirect effects of input features, making it particularly effective for explaining complex neural networks.

In this study, DeepLIFT was employed to interpret the outputs of DL models. By propagating relevance scores through the network layers, DeepLIFT provided a detailed breakdown of how individual input features influenced the final prediction. This granular view was invaluable for understanding non-linear relationships and dependencies that are characteristic of DL models.

The five XAI methods utilized in this study represent distinct taxonomies of interpretability. By combining these methods, this study addressed both global and local interpretability needs. SHAP was instrumental in identifying features most influential across the dataset, LIME provided detailed insights into individual cases, and DeepLIFT offered an in-depth understanding of the behavior of DL models. This complementary approach enhanced the overall transparency of the ML and DL models, aligning with ethical standards and regulatory requirements in financial applications.

CHAPTER 4

RESULTS

This section presents the outcomes of the experiments conducted to develop and evaluate the effectiveness of ML and DL approaches in detecting credit card fraud. It begins with a detailed overview of the EDA and feature engineering process, highlighting key insights and their implications for model development. The results of various algorithms are compared across different sampling methods to address the class imbalance inherent in fraud detection. Furthermore, the impact of hyperparameter tuning on the performance of DL models is analyzed to evaluate whether further optimization could lead to additional improvements in accuracy and robustness. Finally, the application of XAI techniques is explored, providing insights into the decision-making processes of the models and enhancing the interpretability of their predictions. These findings aim to contribute to a comprehensive understanding of the methodology and effectiveness of the proposed fraud detection framework.

4.1 Exploratory data analysis (EDA)

EDA is a critical step in the data preprocessing phase, aimed at uncovering the underlying structure, patterns, and relationships within the dataset. In the context of fraud detection, EDA serves as the foundation for understanding how various features contribute to distinguishing fraudulent transactions from legitimate ones. This process not only identifies potential inconsistencies, outliers, and missing values but also provides insights that guide subsequent steps such as feature engineering, encoding, and transformations.

4.1.1 EDA for private data

For private data, EDA was performed systematically, beginning with the assessment of missing values to ensure data completeness. Following this, categorical and continuous features were analyzed separately to evaluate their individual contributions to the target variable. Crosstabulations, chi-square tests and Cramér's V value were employed to assess the association of categorical features, while correlation analysis and t-tests were conducted to examine the relationships of continuous features with fraud labels. These analyses informed the integration of categorical and continuous variables in a combined evaluation using mutual information scores, highlighting their overall contributions.

The EDA process also included essential data preparation steps such as encoding categorical variables and transforming continuous variables to address issues like skewness and scale inconsistency. The insights gained from EDA directly influenced the creation and refinement of engineered features to enhance the dataset's capacity to capture fraud-related patterns. This systematic approach ensures a robust foundation for the modeling phase, where the prepared dataset will be utilized to build predictive models capable of identifying fraudulent transactions with high precision.

4.1.1.1 Handling missing values

Handling missing values is an essential step in preparing the dataset for analysis, as it ensures data completeness and prevents potential biases that could impact subsequent modeling efforts. In this study, access to the underlying database enabled the resolution of most missing values by querying the source systems directly. This

approach allowed for accurate data recovery without introducing any bias, preserving the dataset's integrity.

One notable exception was the feature V13, a categorical variable that contained missing values for some records. Assigning an existing category to these missing values would have risked introducing bias, as this approach could distort the natural distribution of the feature and affect its relationship with the target variable. Similarly, removing the feature was not a reasonable option, since it would result in the loss of potentially valuable information for identifying fraudulent transactions. To address this challenge, a new category labeled "Unknown" was created and assigned to the missing values in V13. This approach ensured that the feature could be retained in the dataset while maintaining the objectivity of the handling process. By explicitly accounting for missing data as a distinct category, the model can interpret these instances without any assumptions about their association with existing categories.

4.1.1.2 Categorical data analysis

This section examines the categorical features within the dataset, focusing on their distributions and relationships with the target variable. Crosstabulation and chi-square tests were employed to analyze these associations, while Cramér's V was calculated to quantify the strength of these relationships. For high-cardinality features, categories outside the 15 most frequent were consolidated under an "Others" category to ensure interpretability and reduce noise. The heatmaps for crosstab analysis can be found in the Appendix A, figures A1, A2, A3, A4, A5, A6, A7, A8.

4.1.1.2.1 Crosstabulations and chi-square statistics

Table 2 demonstrates the chi-square statistics and p-values of the categorical features. The analysis of binary variable V2 shows that most of the fraudulent transactions fall into a specific subset of categories, with one category containing an overwhelmingly high number of non-fraudulent transactions. The chi-square statistic indicates a strong association between this feature and the target variable, suggesting its relevance in distinguishing fraud. However, its high representation in non-fraudulent transactions also warrants careful consideration to avoid overrepresentation in the model.

Table 2. Chi-square Statistics and P-values for Categorical Variables

Feature	Chi-square	P-value
V2	65721.45	0.0
V6	72283.05	0.0
V8	80068.08	0.0
V9	41203.88	0.0
V12	37626.65	0.0
V13	50871.73	0.0
V14	99.19	0.0
V15	111.13	0.0

The results for V6 demonstrate clear discrepancies across the four categories. Category "2" dominates the non-fraudulent transactions but also accounts for a substantial portion of fraudulent transactions. Conversely, certain categories (e.g., "3") show little to no overlap with fraud cases. The chi-square statistic highlights V6 as a strong predictor of fraud.

For V8, which is narrowed down to 15 categories and an "Others" category due to high cardinality, category "10" is particularly dominant, representing the majority of both fraudulent and non-fraudulent transactions. However, smaller

categories such as "Others" contribute disproportionately to fraudulent transactions, revealing critical insights about potential outlier behaviors. The chi-square test further confirms this variable's importance.

For V9, the analysis indicates that category "3" is overwhelmingly associated with non-fraudulent transactions, with minimal representation in fraud cases. Other seven categories, while less frequent, exhibit stronger associations with fraud, as indicated by the chi-square test. This variability highlights V9 as a feature with nuanced contributions to fraud detection.

V12 is also one of the variables with high cardinality, therefore have 16 categories including "Others" for the less frequent categories. The crosstab for V12 shows significant variability across its categories, with certain values (e.g., "18") displaying a disproportionate representation of fraudulent transactions relative to non-fraudulent ones. Smaller categories also contribute meaningful signals, underscoring the importance of this variable in capturing fraud-related behaviors. The chi-square statistic of 37,700.02 confirms the relevance of V12 to the target variable.

V13, which has eight categories, exhibits a strong association with the target variable, particularly within category "0," which contains the majority of both fraudulent and non-fraudulent transactions. However, smaller categories show a sharper skew toward fraud cases. This feature's chi-square statistic underscores its predictive potential.

Both V14, which has 16 categories due to the aggregation of smaller categories, and binary variable V15 show weaker associations with the target variable compared to other features. While specific categories exhibit some variability, the overall patterns suggest that these features contribute less distinctively

to fraud detection. For example, V14 has a chi-square statistic of 117.32 and V15 shows limited separation across its categories.

4.1.1.2.2 Cramér's V analysis

The results of the Cramér's V analysis provide a clear ranking of categorical variables by their strength of association with the target variable. As shown in figure 2, V8 and V12 demonstrate the strongest relationships, followed by V6, V2, and V13. Variables such as V14 and V15 exhibit weaker associations, suggesting they may be less impactful in distinguishing fraud.

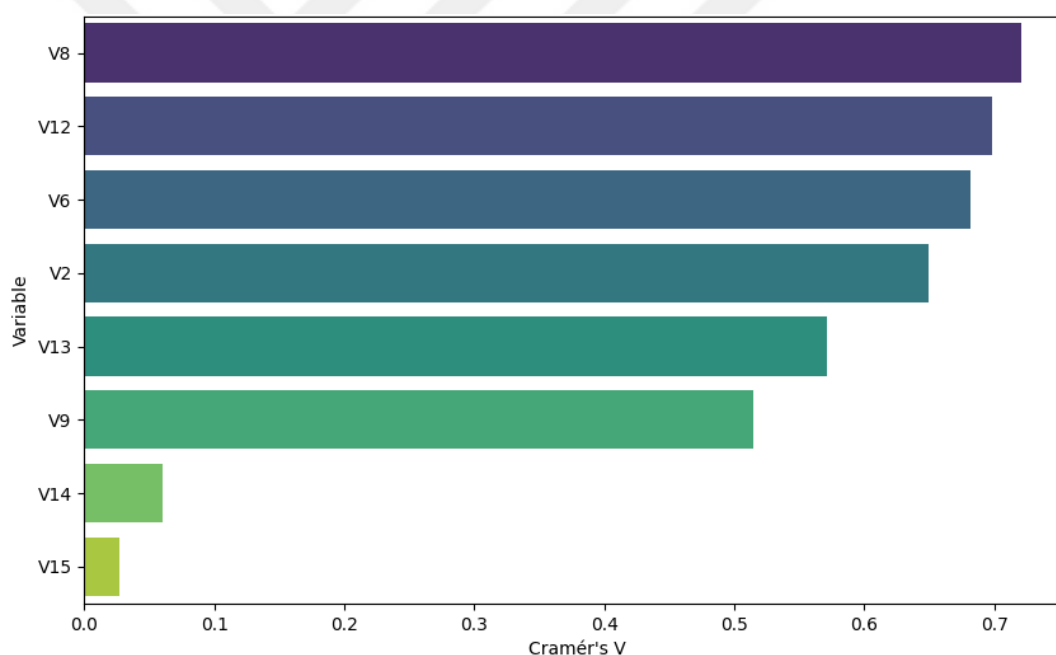


Figure 2. Cramér's V analysis of categorical variables

These findings reveal that certain categorical features, particularly V8 and V12, provide critical signals for fraud detection. The observed imbalance across categories and the high chi-square statistics supports the inclusion of these features in subsequent modeling stages. Conversely, features with weaker associations may

require additional transformations or be deprioritized in the modeling process. The analysis of categorical features underscores the importance of leveraging both high-frequency categories and outlier patterns, ensuring a balanced approach to capturing complex fraud-related behaviors.

4.1.1.3 Continuous data analysis

To gain insights into the distribution and variability of continuous features, histograms and boxplots were employed. These visualizations provide an overview of central tendencies, dispersion, and the presence of outliers, which are particularly relevant in the context of fraud detection, where anomalies often signal fraudulent activity. To evaluate the relationships between continuous features and their relevance to the target variable, correlation analyses were conducted. Pearson correlation coefficients were used to assess linear relationships between the continuous variables, while Point-biserial and Spearman correlation coefficients were calculated to examine the associations of these variables with the binary target variable. To further assess the relationships between the continuous features and the binary target variable, a t-test was conducted.

4.1.1.3.1 Histogram and boxplot analysis

Figure 3 illustrates the overall distribution of the continuous variables with histograms, revealing distinct patterns across the dataset. Features such as V1 display a bimodal distribution, with clusters concentrated around specific ranges, suggesting a structured pattern of transaction behaviors. In contrast, V11 exhibits a heavy skew, with the majority of values clustered near the lower end of the range. Such skewed distributions are typical in financial datasets, reflecting the prevalence of smaller

transaction amounts. V16, however, presents a more normal-like distribution, with a clear central peak and symmetric tails, indicative of relatively consistent feature across the dataset.

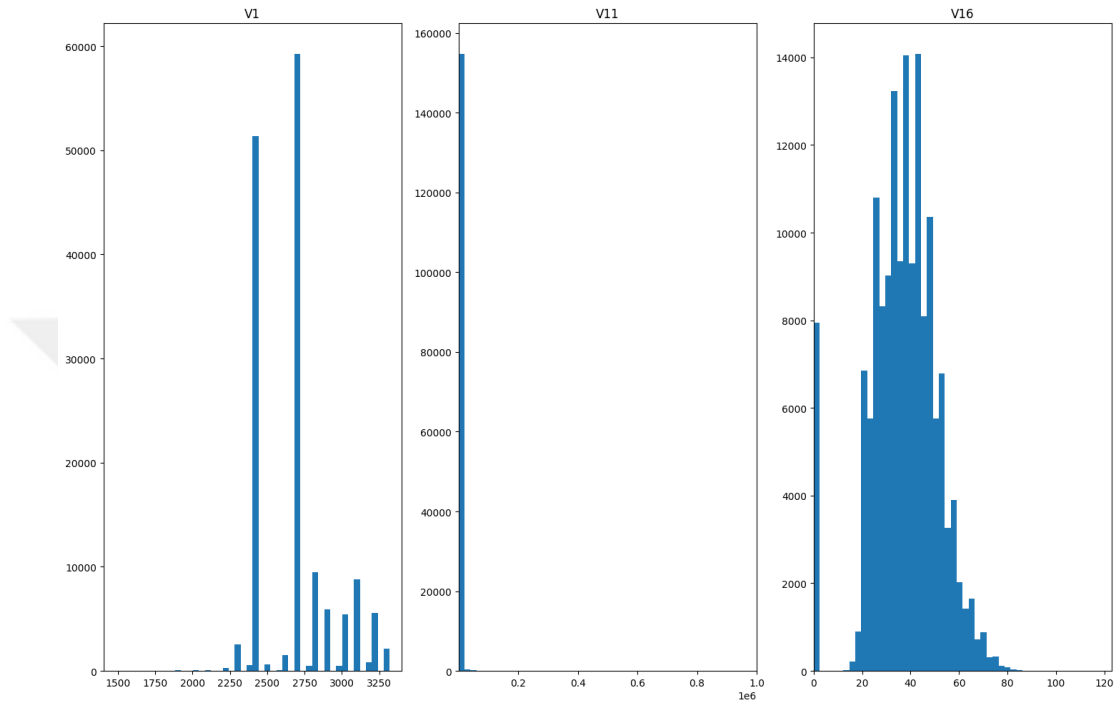


Figure 3. Histogram analysis of continuous variables

Figure 4 shows the boxplots which further complement the histograms by highlighting the presence of outliers. Outliers are a critical aspect of this dataset, as fraudulent transactions often deviate significantly from typical behaviors. For instance, V11 shows a substantial number of extreme values far exceeding the interquartile range, while V1 and V16 exhibit fewer but still prominent outliers. These deviations from the norm are retained in the dataset, as they likely capture the anomalous patterns essential for effective fraud detection. Removing these outliers would risk discarding the very signals that differentiate fraudulent transactions from legitimate ones.

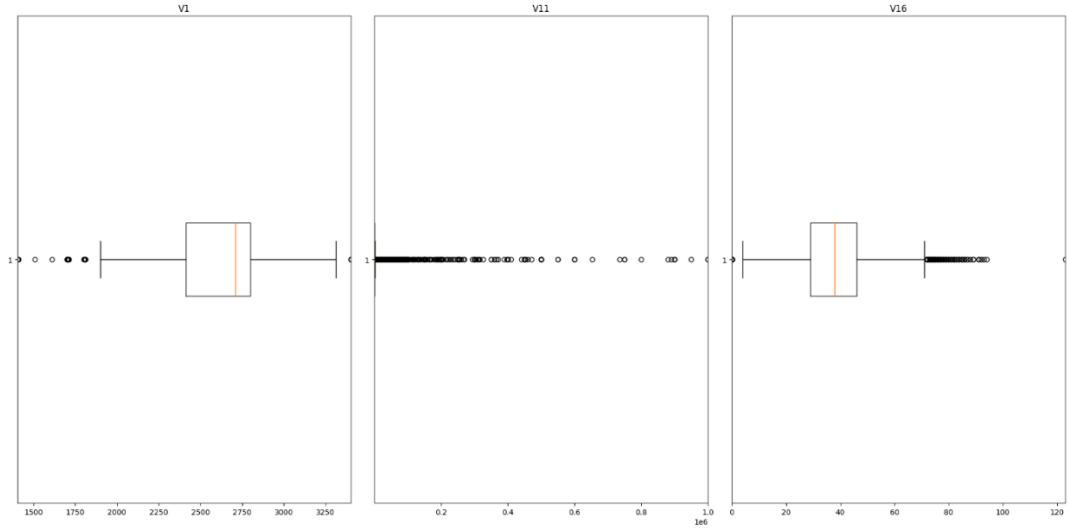


Figure 4. Boxplot analysis of continuous variables

The interplay between the histograms and boxplots underscores the heterogeneity of transaction behaviors within the dataset. While some features show tightly clustered values indicative of consistent patterns, others demonstrate broader variability or heavy skewness. Such diversity highlights the need for robust modeling approaches that can account for these variations while leveraging the presence of anomalies to enhance detection performance.

4.1.1.3.2 Correlation analysis

The Pearson correlation analysis shown in figure 5 reveals that the relationships among the continuous variables are generally weak. For instance, the correlation between V1 and V11 is nearly zero (-0.0053), indicating no linear relationship. Similarly, V1 and V16 exhibit a weak negative correlation (-0.12), as do V11 and V16 (-0.076). This lack of strong correlations suggests that the continuous variables capture independent aspects of transaction and customer behavior, thereby reducing the risk of multicollinearity in the modeling process. The independence of these

variables is advantageous for constructing predictive models, as it ensures that no single feature dominates the others in terms of variance contribution.

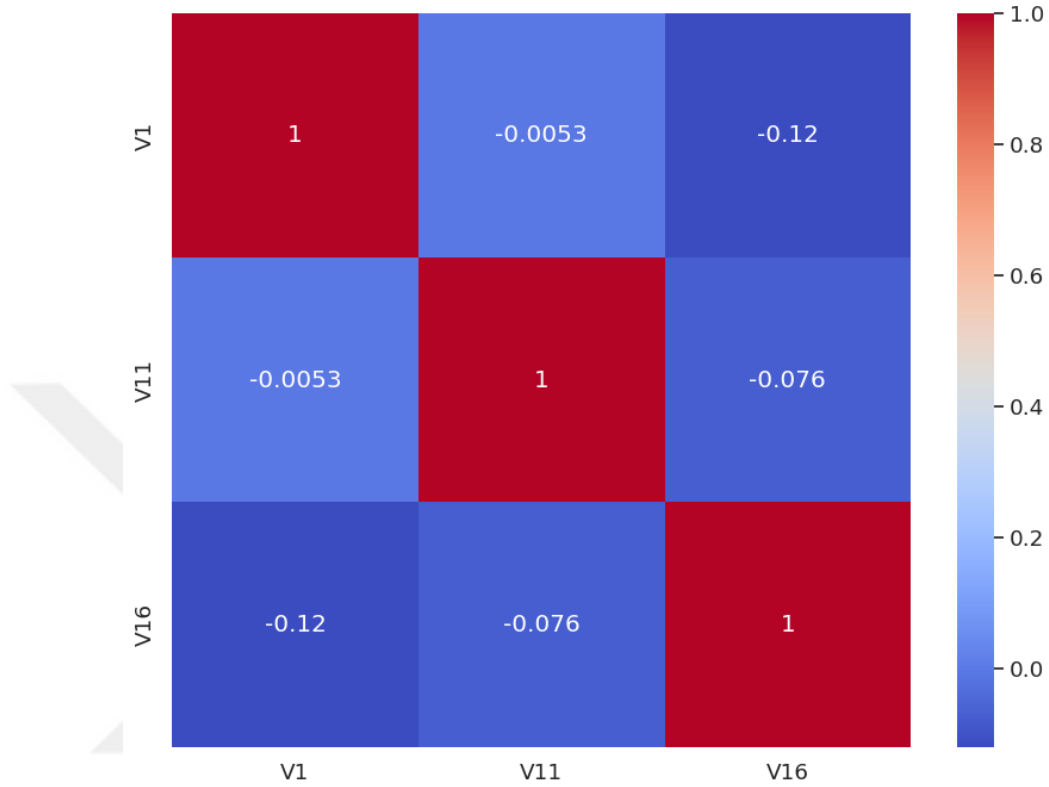


Figure 5. Pearson correlation analysis of continuous variables

The point-biserial correlation analysis shown in figure 6 highlights the strength and direction of the relationships between continuous variables and the binary fraud label. V16 exhibits the strongest positive significant correlation with the target variable, even though the relationship is weak (correlation ~ 0.045). This suggests that V16 contains a slight signal related to fraudulent transactions. V1, in contrast, shows a weak negative correlation with the target variable (correlation ~ -0.035). This indicates that higher values of V1 may be marginally less associated with fraud. V11 demonstrates minimal correlation with the target variable, reflecting its limited direct predictive utility when considered in isolation.

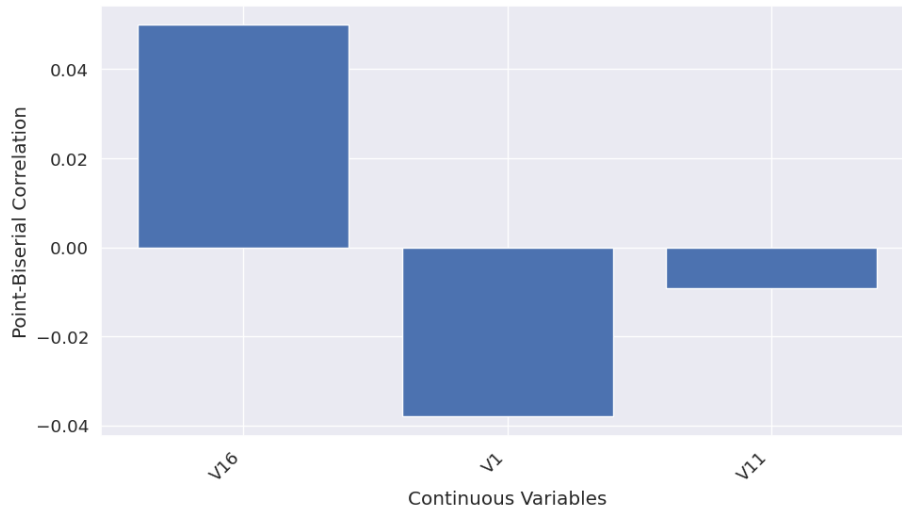


Figure 6. Point-biserial correlations between continuous variables and label

Figure 7 demonstrates the Spearman correlation coefficients, which measure monotonic relationships, are consistent with the point-biserial results. The ranking of the correlations remains unchanged, further confirming that V16 is the most strongly associated feature among the continuous variables, followed by V1 and V11. The weak correlations overall highlight the subtle nature of the relationships between these features and fraudulent transactions, emphasizing the need for advanced modeling techniques to extract meaningful patterns.



Figure 7. Spearman correlations between continuous variables and label

The correlation analysis underscores the complexity of the dataset. The weak relationships among continuous features suggest minimal redundancy, allowing each variable to contribute unique information to the models. Meanwhile, the weak but consistent correlations with the target variable point to the potential value of these features in identifying fraud, particularly when used in combination with other predictors. These findings justify including all continuous features in the modeling process since their combined contributions could improve detection performance.

4.1.1.3.3 T-test

To evaluate the statistical significance of differences in feature distributions between classes, an independent t-test was conducted. The results indicate that features V1, V11, and V16 exhibit statistically significant differences between the two groups. Feature V1 ($t = 14.99$, $p \approx 8.18 \times 10^{-51}$) and V16 ($t = -19.74$, $p \approx 1.32 \times 10^{-86}$) show the strongest effects, with extremely small p-values, suggesting a substantial divergence between the two classes. Feature V11 ($t = 3.61$, $p \approx 0.0$) also demonstrates a significant difference, though with a comparatively smaller effect. Table 3 demonstrates the t-test and p-values of each continuous variable.

Table 3. T-test Statistics and P-values of Continuous Variables

Feature	T-Test	P-value
V1	14.99	$8.18 \times 10^{(-51)}$
V11	3.61	0.0
V16	-19.74	$1.32 \times 10^{(-86)}$

When we compare the mean differences visually by boxplots in figure 8, feature V1 shows a higher mean for fraudulent transactions compared to non-fraudulent ones, as reflected in the box plots. Although the variability in non-

fraudulent transactions is greater, t-test confirms that the differences in means are statistically significant, suggesting the feature captures meaningful distinctions between the two groups. For V11, most of the values are concentrated near zero across both classes, with the presence of extreme outliers in non-fraudulent transactions. Despite the skewed distribution, the t-test results indicate a significant difference in means which underscores the importance of this feature in capturing variations associated with fraudulent behavior. V16 presents a relatively uniform distribution across both classes, with slightly higher median and mean values for fraudulent transactions. The consistent variability across the two groups and the statistically significant difference in means highlight its potential as an effective predictor in identifying fraudulent transactions.

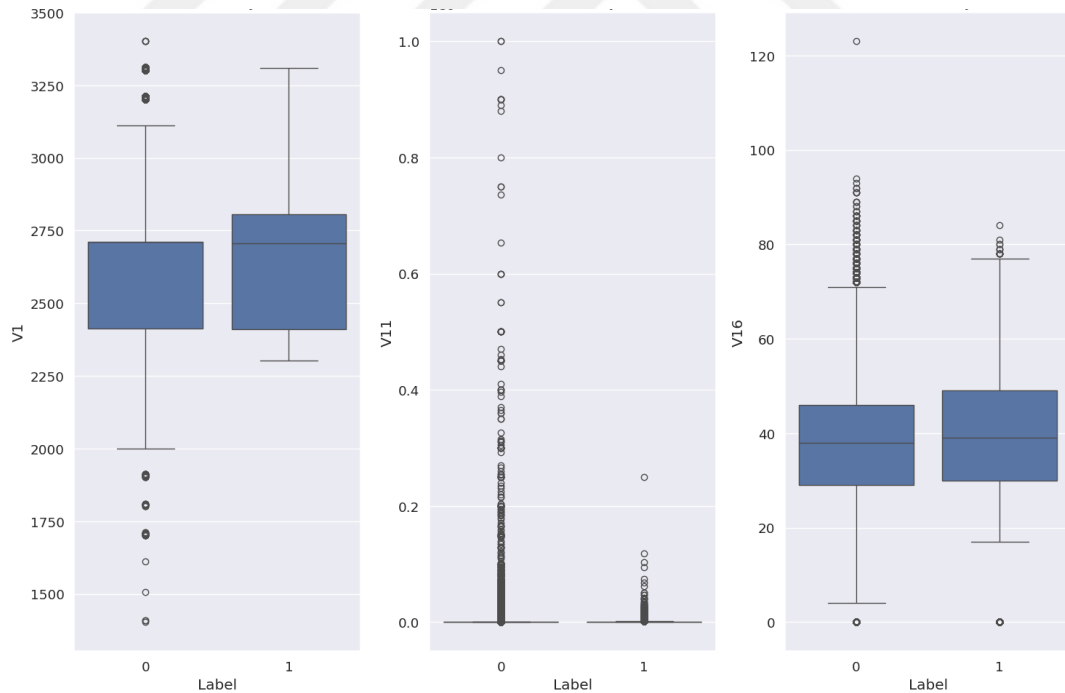


Figure 8. Boxplots of continuous variables by label

Overall, the t-test results emphasize the value of these continuous features in the fraud detection process. The significant differences in means suggest that these

features provide distinct signals that can be leveraged to improve model performance. However, the presence of skewness and outliers in certain features highlights the need for robust modeling techniques to fully utilize their predictive power.

4.1.1.4 Combined data analysis: Mutual information scores

To gain a comprehensive understanding of the dataset, the categorical and continuous features were analyzed together to assess their collective contribution to distinguishing fraudulent transactions. By integrating both types of features, we aim to capture the full range of relationships between the variables and the target label. This holistic approach ensures that key patterns, which may not be evident when analyzing features in isolation, are fully leveraged to inform subsequent modeling efforts.

Mutual information scores were calculated to evaluate the overall contribution of both categorical and continuous features in relation to the fraud label. Mutual information measures the dependency between two variables, capturing both linear and non-linear relationships. This provides a more comprehensive understanding of the predictive power of each feature in the dataset.

The results which are shown in figure 9 reveal that certain categorical features exhibit the strongest associations with the target variable. Among these, V8 stands out with the highest mutual information score, indicating that it provides the most significant contribution to distinguishing between fraudulent and non-fraudulent transactions. Features such as V2, V6, and V12 also demonstrate strong scores, underscoring their importance in capturing patterns related to fraud. These features

likely encode complex relationships and categorical patterns that are highly indicative of fraudulent activity.

In contrast, continuous features such as V1, V16, and V15 exhibit relatively lower mutual information scores. While these features were found to be statistically significant in earlier analyses, their lower mutual information values suggest that their contribution to the overall prediction task may be more subtle or less direct. Nonetheless, their inclusion remains valuable as they may capture complementary signals that enhance model performance when combined with other features.

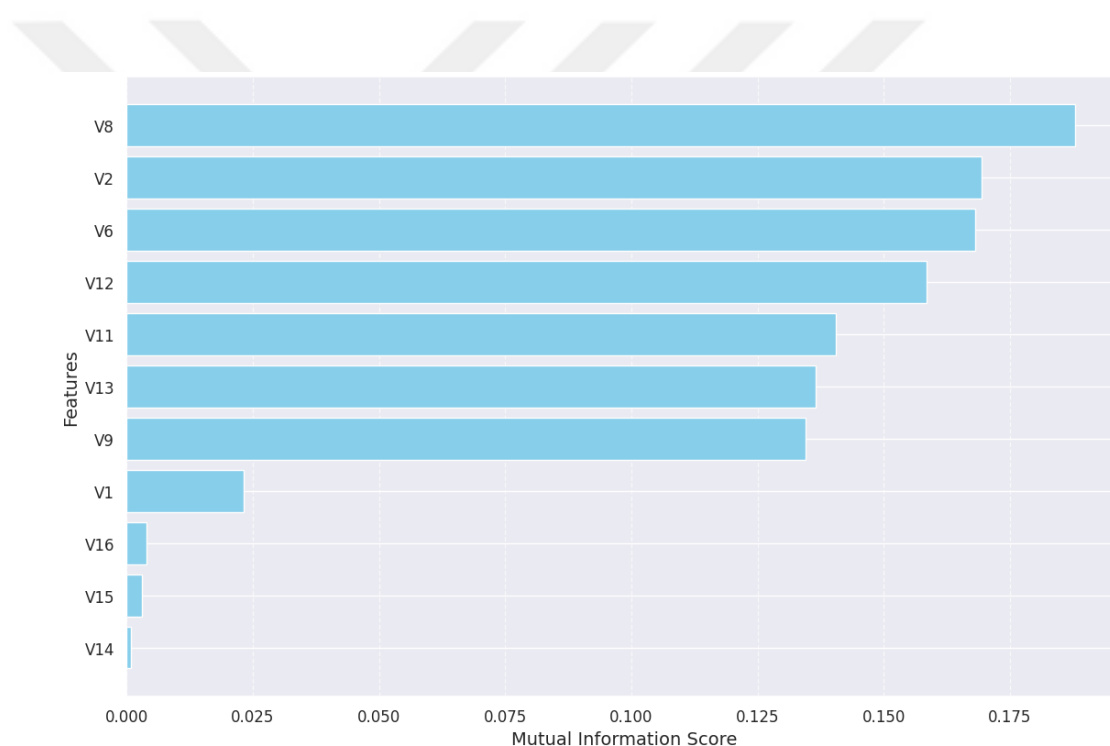


Figure 9. Mutual information scores of the variables

This analysis underscores the importance of leveraging both categorical and continuous features to capture the complex dynamics of fraud detection. Categorical variables, particularly those with high mutual information scores, play a dominant role in distinguishing fraudulent patterns, while continuous features offer complementary details. This comprehensive approach to feature evaluation ensures

that no critical information is overlooked and supports the development of robust predictive models.

4.1.1.5 Data transformations and encoding for private data

Data transformations and encoding were applied to prepare the dataset for modeling by ensuring that all features were in a format suitable for ML algorithms. These steps aim to address issues such as categorical variable representation, skewness in continuous features, and scale inconsistencies across variables.

For categorical variables, different encoding techniques were employed based on the nature of the variable. Binary variables were transformed using label encoding, which assigns 0 and 1 to represent the two categories efficiently to avoid unnecessary dimensionality increases. For non-binary categorical variables, one-hot encoding was applied to create separate binary columns for each category. This approach preserves the categorical nature of these features without introducing artificial ordinal relationships.

Continuous variables were evaluated for skewness, as heavily skewed distributions can negatively impact the performance of ML models. Highly skewed continuous features were subjected to a log transformation to reduce skewness and bring their distributions closer to normal. This transformation ensures that extreme values in the dataset do not disproportionately influence the models. After log transformations, all continuous variables were standardized to ensure they have a mean of zero and a standard deviation of one. Standardization ensures that features contribute equally to the learning process, preventing those with larger scales from dominating the model. This step is particularly important when combining features with varying units or magnitudes.

These transformations and encoding steps ensured that the final dataset was numerically stable, balanced, and ready for feature engineering and modeling. By addressing the inherent inconsistencies and biases in raw data, this process lays the groundwork for robust and effective fraud detection.

EDA highlighted several limitations in the dataset that can be addressed through feature engineering. While categorical features showed significant associations with the target variable, high-cardinality variables such as V8 and V12 pose challenges for direct inclusion due to computational complexity and potential overfitting. Additionally, these variables often lack behavioral context, such as frequent spending patterns or geographical trends, which are critical for detecting deviations linked to fraud. Continuous features exhibited generally weak correlations with the target variable, suggesting their individual contributions are limited and may not fully capture temporal or contextual dynamics. To overcome these weaknesses, feature engineering focuses on creating enriched features that encapsulate key behaviors, reduce dimensionality for complex variables, and enhance the dataset's ability to identify anomalous patterns indicative of fraud.

4.1.2 EDA for public data

EDA conducted on the Kaggle credit card fraud detection dataset aims to uncover underlying patterns, relationships, and trends within the data that may influence the performance of predictive models. The analysis focuses on both statistical and visual insights, utilizing a variety of plots and metrics to better understand the data's structure and the relationship between features and the target variable. There are no missing values in the dataset; thus, the handling of null values process was not carried out.

To examine the relationship between continuous variables and the target variable, the point-biserial correlation was computed for each feature. The analysis in figure 10 revealed that certain features, such as V17, V14, and V12, exhibit the strongest negative correlations with the target variable, while others, such as V11 and V4, display moderate positive correlations. The Spearman correlation was also evaluated to capture potential monotonic relationships. The results in figure 11 align with the point-biserial analysis, confirming that some variables are more influential in distinguishing fraudulent transactions from non-fraudulent ones.

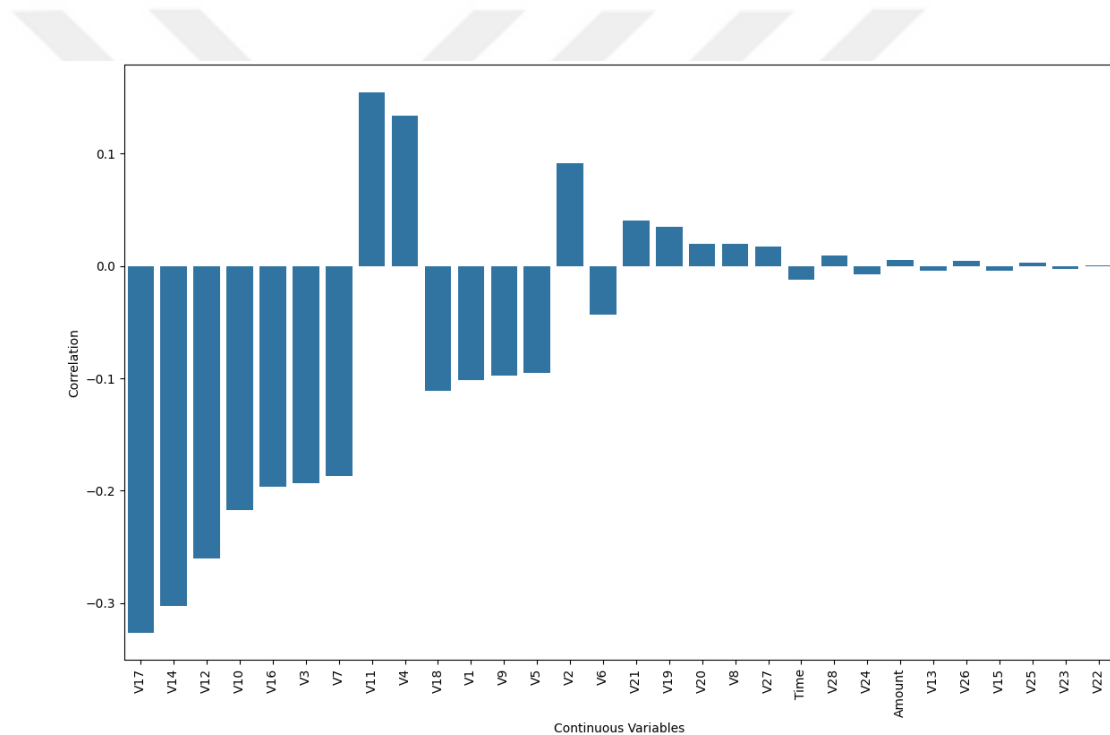


Figure 10. Point-biserial correlation of public dataset variables with label

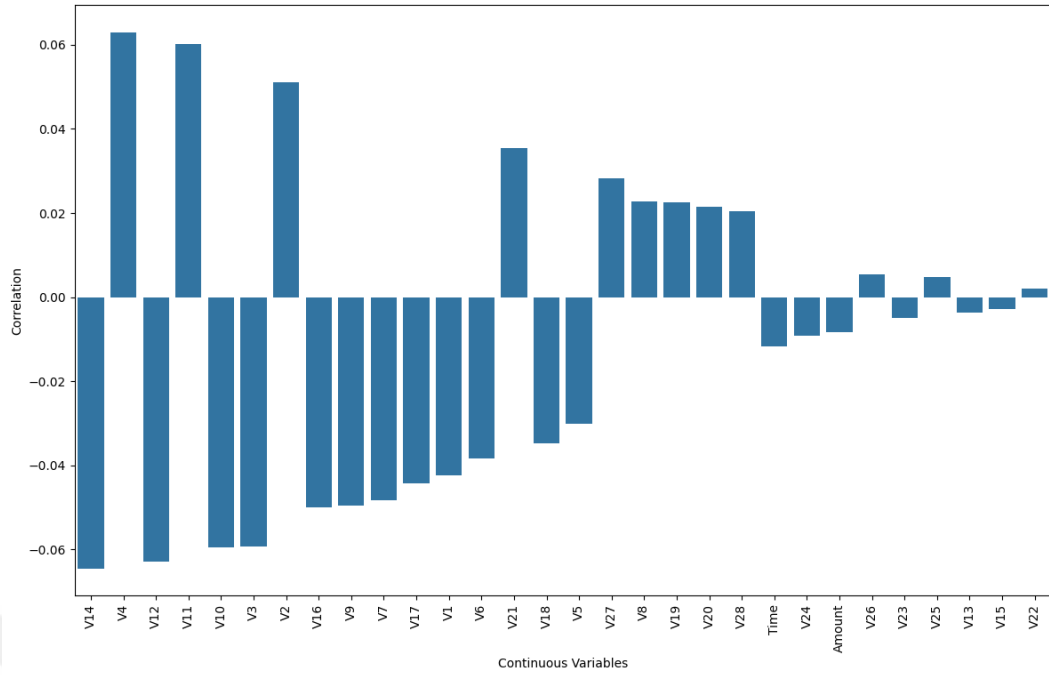


Figure 11. Spearman correlation of public dataset variables with label

To quantify the relationship between each feature and the target variable, mutual information scores were calculated and shown in figure 12. Features such as V17, V14, and V12 consistently ranked highest in their association with the target variable. This analysis underscores the relevance of these features in identifying fraudulent patterns, supporting their inclusion in downstream modeling efforts.

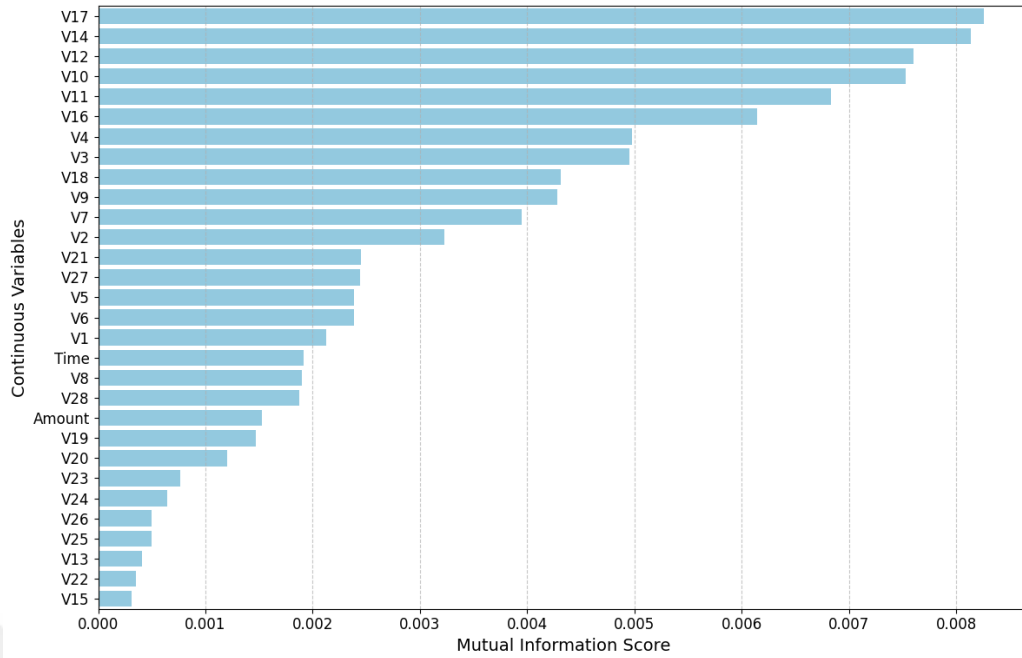


Figure 12. Mutual information scores of public dataset variables

Overall, the EDA for public dataset demonstrates the relationships of features with the classes. No further encoding or data transformation was made since the data is already PCA-transformed. The findings provide a comprehensive understanding of the dataset and establish a strong foundation for developing robust predictive models tailored to the fraud detection task. The visualizations and correlation metrics serve as critical tools for model optimization, ensuring that the insights from the data are effectively leveraged in subsequent analyses.

4.2 Feature engineering for private data

To enhance the predictive power of the models, various features were engineered based on transactional data, customer profiles, and behavioral patterns. These features were designed to capture statistical, behavioral, and geographical aspects of customers' transaction activity, which are critical for distinguishing fraudulent behavior from legitimate transactions. Another reason to create these features was to

address the limitations of the raw data, particularly the challenge of incorporating high-cardinality categorical variables, while preserving their essential and valuable information. By extracting essential details into more manageable forms, valuable signals were preserved without creating excessive computational complexity.

In this feature set, the first group of engineered variables captures transaction amounts at the customer level. Maximum (MaxAmount), minimum (MinAmount), and average (AvgAmount) transaction amounts were calculated for each customer. They provide insights into the typical transaction patterns of customers, helping to detect anomalies such as unusually large or small transactions that deviate significantly from historical behavior.

Annual transaction count (AnnualTranCount) was also calculated to measure the total number of transactions a customer performs in one year. Sudden spikes or drops in transaction frequency can indicate suspicious behavior. In addition, transaction count in the transaction country (TranCountryTranCount) was introduced to capture how many times a customer had transacted in the same country as the current transaction. This feature focuses on unusual geographic activity, such as a purchase in a country where the customer had little to no prior history.

Number of credit cards (NumofCreditCards) and number of debit cards (NumofDebitCards) were calculated to capture the total number of credit and debit cards linked to each customer. They help model the customer's financial behavior, as fraudulent activities often involve accounts with atypical numbers of cards or accounts that are rarely used.

Residency and Merchant Country features, which contain many unique categories, are infeasible to include directly in the dataset due to the computational complexity of one-hot encoding. To derive simplified yet informative representations

of these variables, a binary feature (ResidencyBoolean) was created. It indicates whether a transaction occurred within the customer's country of residence.

Transactions made outside the customer's usual geographical region are often flagged as higher risk and may require further scrutiny. By adding this feature to the dataset, related raw features were dropped from the final dataset.

Lastly, Merchant Category Codes (MCCs) describe the type of business that a transaction occurred and has hundreds of categories which makes it impractical to use directly in the dataset. A feature which identifies whether a transaction falls within the customer's top 5 most frequent MCCs was created (IsTop5MCC) in order to include MCCs in the dataset in an informative and valuable way without increasing the cardinality. Fraudulent transactions often deviate from typical purchasing patterns, including spending at unfamiliar merchants. After adding this feature, corresponding transactional feature was removed from the final dataset.

These engineered features were integrated into the final dataset to enrich the representation of customer behavior, offering the models additional context to detect deviations indicative of fraud. Their inclusion was guided by domain knowledge and exploratory data analysis, ensuring their relevance and contribution to improving model performance.

4.3 EDA for engineered features

A subsequent exploratory analysis was conducted to evaluate their relationships and contributions with the engineered features integrated into the dataset. This phase focused on assessing the correlations among engineered features to ensure they provided non-redundant information. The same encoding and transformation

methods applied to the original features were also used for the engineered features to maintain consistency across the dataset.

4.3.1 Correlation analysis with engineered features

The correlation analysis of the engineered features reveals valuable insights into their relationships and highlights redundancy among some variables. In figure 13, most features exhibit low to moderate correlations, indicating that they capture distinct aspects of customer behavior and transactional patterns. However, a few strong correlations suggest potential redundancy.

The most notable observation is the high correlation between AnnualTranCount and TranCountryTranCount (correlation = 0.93). This strong relationship indicates significant overlap in the information captured by these two features. Both variables describe transaction frequency, but TranCountryTranCount specifically focuses on the number of transactions occurring in the country of transaction of a data point. To avoid redundancy and ensure each feature contributes unique information to the model, AnnualTranCount was removed from the dataset since TranCountryTranCount focuses on a more specific information with a geographical aspect of the spending behavior.

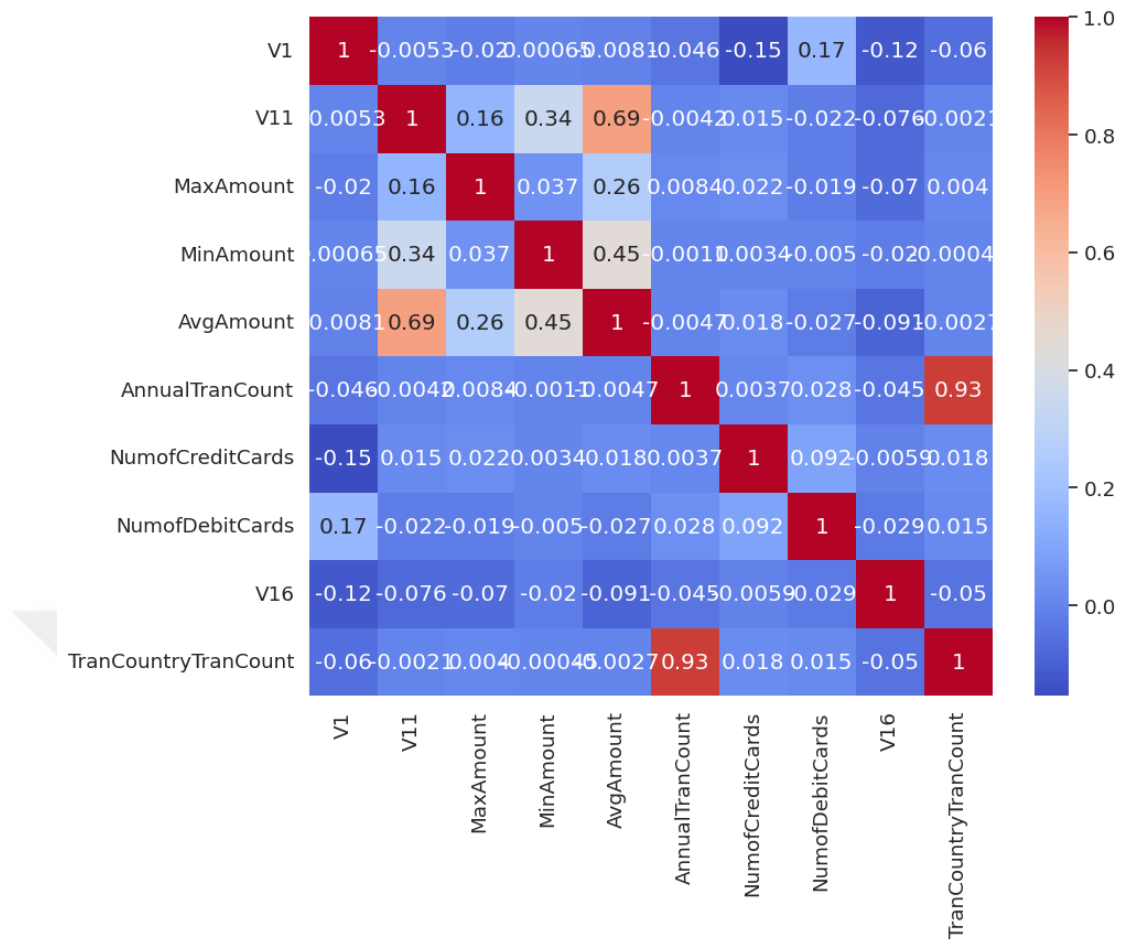


Figure 13. Pearson correlations of continuous variables with feature engineering

Other engineered features, such as MaxAmount, MinAmount, and AvgAmount, show moderate correlations (e.g., MinAmount and AvgAmount with a correlation of 0.45). These relationships are expected, as these features are derived from similar transactional data. Despite their correlations, they provide complementary statistical insights into customer spending behavior and are therefore retained for modeling. The remaining engineered features, such as NumofCreditCards and NumofDebitCards, exhibit low correlations with both each other and the target variable, emphasizing their distinct contributions to capturing customer profile characteristics.

This analysis ensures that the final dataset consists of features that minimize redundancy while maximizing the diversity of information, a critical step in building robust and effective fraud detection models.

4.3.2 Encoding and data transformation for engineered features

To maintain consistency and compatibility with the original features, the same encoding and transformation methods were applied to the engineered features.

Binary features, such as `ResidencyBoolean`, were label-encoded to represent their two categories as 0 and 1, ensuring simplicity and efficiency without increasing dimensionality. For categorical variables, one-hot encoding was applied to convert each category into a binary column, preserving their categorical nature while avoiding the introduction of ordinal relationships.

For continuous engineered features, log transformation was applied to those exhibiting skewed distributions, bringing their values closer to a normal distribution and reducing the influence of extreme outliers. Subsequently, all continuous features were standardized to ensure they had a mean of zero and a standard deviation of one, allowing them to contribute equally during the modeling process.

These steps ensured that the engineered features were fully integrated into the dataset, both in terms of structure and scale, and were ready for use in the modeling phase. The combination of consistent preprocessing and careful feature selection ensures that the dataset is optimized to capture the complex patterns associated with fraudulent transactions effectively. To ensure consistency and completeness, all exploratory data analysis steps applied to the raw features—such as correlation analysis, chi-square tests, and mutual information scores—were also performed on the final dataset, which includes both raw and engineered features. The resulting

visualizations and analyses, demonstrating the enhanced dataset's properties, are provided in the Appendix B, figures B1, B2, B3, B4, B5, B6, B7, B8, B9.

4.4 Model performance comparison

This section presents the evaluation results of the SL and USL models discussed in the Methodology section across different sampling techniques by focusing on their performance on the private and public fraud detection dataset. The evaluation metrics mentioned in the Methodology section are employed to assess the performance of the models on the fraud detection dataset. Results are ordered by the model performance on f1-score, recall and precision of Class 1 in an ascending order. Following the application and evaluation of all models, hyperparameter tuning was selectively performed on the top-performing DL models to refine their parameters and improve their predictive performance for private dataset.

The datasets were first divided into an 80-20 train-test split, where 80% of the data was used for training the models and 20% for validation on an unseen test set. This split ensured that the models were tested on previously unseen data to provide an accurate assessment of their real-world applicability. For the ML models, 5-fold stratified cross-validation was applied to the training set. This technique was used to validate the consistency of the models' performance, distinguishing between one-time results and consistent patterns.

In the case of DL models, cross-validation was not utilized due to the computational burden of training across multiple folds. Instead, the models were trained for a maximum of 50 epochs, with early stopping based on validation performance to prevent overfitting. The unseen test set was subsequently used to validate these models and provide a fair comparison with the ML counterparts. Using

cross-validation for ML models and early stopping for DL models ensured a comprehensive and computationally efficient evaluation framework.

4.4.1 Model performances for private dataset

The following sections discuss the results of model evaluations, focusing on the impact of different sampling techniques, model-specific performance, and their implications for fraud detection with private dataset.

4.4.1.1 Model performances on imbalanced private dataset

The performances of the ML models were evaluated on the imbalanced fraud detection dataset without any modifications to the class distribution. This evaluation aimed to benchmark the models under real-world conditions where the dataset is imbalanced, with legitimate transactions vastly outnumbering fraudulent ones. The results were analyzed using accuracy and class-specific metrics, such as precision, recall, and f1-score on test data to highlight the models' ability to detect fraudulent transactions.

4.4.1.1.1 Performance comparison of SL models on private imbalanced dataset

Supervised models encompass both ML algorithms and DL models. Table 4 presents the performance of various ML models on an imbalanced dataset, where Class 0 represents legitimate transactions and Class 1 denotes fraudulent transactions.

Random Forest and XGBoost achieved the highest levels of accuracy (0.99).

Random Forest slightly outperforms XGBoost across other metrics, with a precision of 0.95 and an f1-score of 0.95 for fraudulent transactions, whereas XGBoost yielded a precision of 0.93 and f1-score of 0.94 for Class 1. Logistic Regression and SVM,

while slightly behind, performed respectably with an accuracy of 0.97 and 0.98 respectively, and an f1-score of 0.82 and 0.88 for Class 1.

Table 4. Performance Comparison of ML Models on Private Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	0.98	0.83	0.98	0.81	0.98	0.82	0.97
SVM	0.99	0.88	0.99	0.87	0.99	0.88	0.98
XGBoost	1.00	0.93	0.99	0.95	0.99	0.94	0.99
Random Forests	1.00	0.95	0.99	0.95	1.00	0.95	0.99

To provide a more comprehensive evaluation of the models, we further analyzed the performance of Random Forest, the best-performing ML model on the imbalanced data, on the transactional data without engineered features. This comparison aims to highlight the impact of feature engineering on the model's ability to detect fraudulent transactions. Figure 14 illustrates a comparative analysis of the results from two datasets and FE stands for feature engineering.

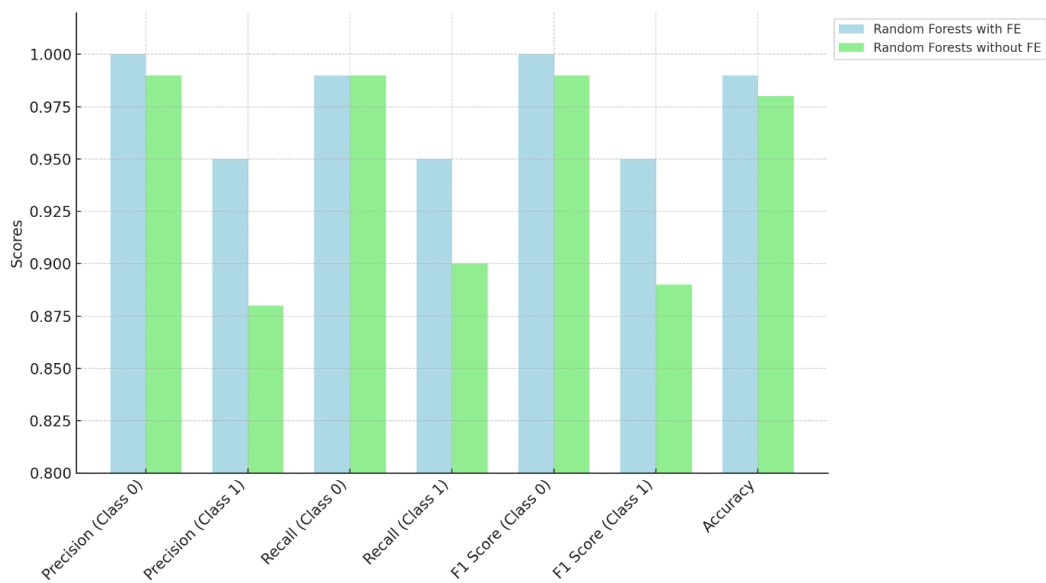


Figure 14. Performance comparison of Random Forests with and without FE

The results of the Random Forest model applied to transactional data without engineered features reveal a notable decline in performance compared to the results with feature engineering. When feature engineering is utilized, the model achieves a precision of 1.00 for Class 0 and 0.95 for Class 1, while the precision for Class 1 drops to 0.88 without feature engineering. Similarly, the recall for Class 1 decreases from 0.95 with feature engineering to 0.90 without it. This reduction is further reflected in the f1-score for fraudulent transactions (Class 1), which decreases from 0.95 with feature engineering to 0.89 without it. Overall, the accuracy of the model declines slightly from 0.99 with feature engineering to 0.98 without feature engineering. These findings emphasize the importance of feature engineering in enhancing the detection of fraudulent transactions and improving the overall performance of the model.

Among DL models, TabTransformer achieved the best overall performance with an f1-score of 0.93 and a recall of 0.94 for fraudulent transactions as shown in table 5. MLP and DNN also performed strongly, each achieving an f1-score of 0.92 for fraudulent transactions, while LSTM and GRU closely followed with an f1-score of 0.91. Other architectures, including RNN, CNN, and BiLSTM, demonstrated comparable results, all achieving an accuracy of 0.98 with f1-scores ranging from 0.89 to 0.91 for fraudulent transactions. Despite their similar accuracy, these DL models showed slight differences in their ability to balance precision and recall.

Table 5. Performance Comparison of DL Models on Private Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RNN	0.99	0.90	0.99	0.89	0.99	0.89	0.98
CNN	0.99	0.88	0.99	0.90	0.99	0.89	0.98
LSTM	0.99	0.90	0.99	0.91	0.99	0.90	0.98
BiLSTM	0.99	0.91	0.99	0.91	0.99	0.91	0.98
GRU	0.99	0.90	0.99	0.92	0.99	0.91	0.98
DNN	0.99	0.91	0.99	0.92	0.99	0.92	0.98
MLP	0.99	0.92	0.99	0.93	0.99	0.92	0.99
TabTransformer	0.99	0.92	0.99	0.94	0.99	0.93	0.99

4.4.1.1.2 Performance comparison of USL models on private imbalanced dataset

RBM and Autoencoders were applied to the dataset without using labels during training. Their performance on the imbalanced dataset was compared to the SL approaches to assess their applicability in real-world fraud detection scenarios and the results are shown in table 6.

Table 6. Performance Comparison of USL Models on Private Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	0.96	0.16	0.61	0.73	0.75	0.26	0.62
Autoencoder	0.99	0.36	0.85	0.88	0.91	0.52	0.85

RBM achieved an overall accuracy of 0.62, with precision and recall scores for fraudulent transactions (Class 1) of 0.16 and 0.73, respectively. The f1-score for fraudulent transactions was 0.26, indicating significant challenges in identifying fraudulent cases effectively. Autoencoder outperformed the RBM, achieving an accuracy of 0.85. Its precision and recall for fraudulent transactions were 0.36 and 0.88, respectively, with an f1-score of 0.52 for Class 1.

4.4.1.2 Model performances for private dataset with RU

The results of the SL and USL models on the undersampled dataset are presented in this section. By reducing the majority class, undersampling created a balanced dataset, enabling the models to focus equally on detecting fraudulent and non-fraudulent transactions. The following subsections detail the outcomes for SL and USL models, providing insights into their ability to adapt to the reduced dataset size and class distribution.

4.4.1.2.1 Performance comparison of SL models on private dataset with RU

The SL models displayed noticeable differences in performance when trained on the undersampled dataset compared to the imbalanced data. Table 7 presents the results of ML models. Logistic Regression achieved an accuracy of 0.94, which was slightly lower than its accuracy of 0.97 on the imbalanced dataset. The recall for fraudulent transactions (Class 1) improved significantly from 0.81 to 0.96 due to the balanced dataset. However, precision for Class 1 dropped from 0.83 to 0.60, resulting in a decline in the f1-score for Class 1 from 0.82 to 0.74.

Table 7. Performance Comparison of ML Models on Private Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.60	0.94	0.96	0.97	0.74	0.94
SVM	1.00	0.67	0.95	0.97	0.98	0.79	0.96
XGBoost	1.00	0.75	0.97	0.99	0.98	0.86	0.97
Random Forests	1.00	0.77	0.97	0.99	0.98	0.86	0.97

SVM maintained an accuracy of 0.96, close to its performance on the imbalanced dataset. The recall for Class 1 increased from 0.87 to 0.97, indicating

better sensitivity to detecting fraud. On the other hand, precision for Class 1 dropped from 0.88 to 0.67, causing the f1-score for Class 1 to fall from 0.88 to 0.79. These results suggest that while the model identified more fraudulent transactions, it also produced more false positives.

XGBoost and Random Forest continued to perform well, maintaining an accuracy of 0.97, slightly lower than their results on the imbalanced dataset. Both models exhibited an improvement in recall for Class 1, increasing from 0.95 to 0.99. However, their precision for Class 1 decreased, with XGBoost dropping from 0.93 to 0.75 and Random Forest dropping from 0.95 to 0.77. Despite these changes, the f1-scores for Class 1 maintained a notable performance at 0.86, highlighting their ability to handle the undersampled dataset effectively.

The performance of DL models on the undersampled dataset, presented in table 8, shows a shift in their ranking compared to the results on the imbalanced data. This change highlights how the balanced dataset influences the models' ability to classify fraudulent and non-fraudulent transactions.

Table 8. Performance Comparison of DL Models on Private Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
GRU	1.00	0.63	0.94	0.98	0.97	0.77	0.95
TabTransformer	1.00	0.63	0.94	0.99	0.97	0.77	0.95
RNN	1.00	0.67	0.95	0.96	0.97	0.79	0.96
CNN	0.99	0.71	0.96	0.92	0.98	0.80	0.96
BiLSTM	1.00	0.69	0.96	0.96	0.98	0.80	0.96
MLP	1.00	0.69	0.96	0.98	0.98	0.81	0.96
LSTM	1.00	0.71	0.96	0.96	0.98	0.82	0.96
DNN	1.00	0.73	0.97	0.97	0.98	0.83	0.97

GRU and the TabTransformer achieved the lowest overall accuracy among the DL models, both at 0.95. While GRU maintained strong recall for Class 1 (0.98), its precision dropped to 0.63, leading to an f1-score of 0.77 for Class 1.

TabTransformer exhibited identical precision and recall metrics to GRU, but it was the best performer on the imbalanced dataset among DL models, where it had a higher f1-score and accuracy. This decline in performance suggests that the TabTransformer struggled to leverage the reduced dataset effectively.

RNN and CNN both achieved an accuracy of 0.96. While the recall for Class 1 was consistent at 0.96 for both models, the CNN demonstrated better precision at 0.71 compared to the RNN's 0.67. This difference resulted in a higher f1-score of 0.80 for the CNN versus 0.79 for the RNN. On the imbalanced dataset, RNN showed stronger performance relative to CNN, but the undersampling appears to have reversed this order, favoring CNN. BiLSTM matched the accuracy of RNN and CNN at 0.96, with precision and recall for Class 1 at 0.69 and 0.96, respectively, yielding an f1-score of 0.80. Decrease in precision and increase in recall pattern applies to BiLSTM as well.

LSTM and MLP both achieved an accuracy of 0.96, but LSTM demonstrated a slightly better f1-score. MLP had a recall of 0.98 for Class 1 compared to LSTM's 0.96, but LSTM achieved a higher precision of 0.71 compared to MLP's 0.69. Despite the increase in recall and decrease in other metrics, they keep a similar performance ranking among DL models compared to results on imbalanced data.

DNN achieved the highest accuracy among the DL models at 0.97. The precision for Class 1 was 0.73, with a recall of 0.97, resulting in an f1-score of 0.83. This performance not only strengthens its ranking as the top-performing DL model

on the undersampled dataset but also reflects its ability to adapt well to the balanced class distribution.

4.4.1.2.2 Performance comparison of USL models on private dataset with RU

The performance of the USL models on the undersampled dataset revealed some changes in their behavior for both Class 0 and Class 1 compared to the imbalanced dataset. The results of the models are shown in table 9..

Table 9. Performance Comparison of USL Models on Private Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	0.93	0.10	0.41	0.66	0.57	0.17	0.44
Autoencoder	0.99	0.37	0.85	0.88	0.91	0.52	0.85

For RBM, all evaluation metrics for both Class 0 and Class 1 exhibited a noticeable decline when RU was applied, compared to the performance observed on the imbalanced dataset. Accuracy dropped significantly from 0.62 on the imbalanced dataset to 0.44 on the undersampled dataset. The f1-score for Class 0 dropped from 0.75 to 0.57, for Class 1 decreased from 0.26 to 0.17, reflecting the model's reduced ability to classify both legitimate and fraudulent transactions correctly. The substantial decline in performance indicates that RBM struggled to detect meaningful patterns in the data.

The performance of Autoencoder with RU remained almost the same with the imbalanced dataset. For Class 1, precision increased slightly from 0.36 to 0.37, while all other metrics remained stable. This indicates that the application of random undersampling had minimal impact on the overall performance of the Autoencoder, with only a marginal improvement in precision for Class 1.

4.4.1.3 Model performances for private dataset with SMOTE

This section presents the results of the SL and USL models on the oversampled dataset. By augmenting the minority class using SMOTE, oversampling addressed the class imbalance, aiming to improve the detection of fraudulent transactions while preserving the overall data structure. The following subsections explore the performance of SL and USL models, highlighting their ability to utilize the oversampled data and adapt to the increased representation of the minority class.

4.4.1.3.1 Performance comparison of SL models on private dataset with SMOTE

Under the SL models section, the application of SMOTE yields notable differences in model performance metrics compared to the imbalanced dataset and RU. These differences highlight both advantages and trade-offs depending on the model and metric. Table 10 presents the results of ML models with SMOTE.

Table 10. Performance Comparison of ML Models on Private Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.61	0.94	0.96	0.97	0.75	0.94
SVM	1.00	0.75	0.97	0.97	0.98	0.85	0.97
XGBoost	0.99	0.93	0.99	0.92	0.99	0.93	0.99
Random Forests	1.00	0.90	0.99	0.97	0.99	0.94	0.99

For Logistic Regression, SMOTE improves recall for Class 1, increasing it from 0.81 (imbalanced dataset) to 0.96. This indicates that the model better identifies instances of the minority class. However, the precision for Class 1 declines significantly from 0.83 to 0.61, suggesting an increase in false positives. The f1-

score for Class 1 decreases from 0.82 to 0.74 when compared to the imbalanced dataset and shows similar trends to RU.

For SVM, SMOTE enhances recall for Class 1, which improves from 0.87 (imbalanced dataset) to 0.97. However, the f1-score for Class 1 drops from 0.88 to 0.85 compared to the imbalanced dataset due to the decrease in the precision for Class 1 from 0.88 to 0.75. The higher f1-score observed with SMOTE, compared to RU, highlights its advantage in effectively balancing precision and recall.

XGBoost produces similar results to imbalanced dataset with SMOTE. Precision for Class 1 remains stable at 0.93 with SMOTE, while there is a slight decrease in the recall from 0.95 to 0.93. Compared to the undersampled dataset, SMOTE demonstrates significantly higher results for precision but slightly lower performance on recall of the fraud class. Overall, XGBoost benefits from SMOTE in terms of sensitivity to the minority class while maintaining balanced performance.

Random Forests maintain high performance with a recall for Class 1 is 0.97 which is slightly better than the imbalanced dataset results (0.95) whereas it is slightly worse than the RU results (0.99). Precision for Class 1 decreases from 0.95 (imbalanced dataset) to 0.90 under SMOTE, but this is a significant increase when compared to RU (0.77).

Among DL models, even though their performance are comparatively worse than the others, CNN and RNN demonstrate some improvements with SMOTE as shown in Table 11. Both models achieved a Class 1 recall of 0.95 and 0.97, respectively, and f1-scores of 0.84 and 0.85. These results represented an improvement over their performance with RU, where CNN and RNN had Class 1 f1-scores of 0.80 and 0.79, respectively. However, their performance with SMOTE lagged behind their results on the imbalanced dataset, where both achieved Class 1

f1-scores of 0.89. The overall accuracy for both models remained consistent at 0.97 under SMOTE.

Table 11. Performance Comparison of DL Models on Private Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
CNN	0.99	0.76	0.97	0.95	0.98	0.84	0.97
RNN	1.00	0.76	0.97	0.97	0.98	0.85	0.97
GRU	1.00	0.81	0.98	0.97	0.99	0.88	0.98
LSTM	1.00	0.82	0.98	0.96	0.99	0.89	0.98
BiLSTM	1.00	0.82	0.98	0.97	0.99	0.89	0.98
DNN	1.00	0.87	0.99	0.95	0.99	0.91	0.98
MLP	0.99	0.89	0.99	0.95	0.89	0.92	0.98
TabTransformer	1.00	0.91	0.99	0.96	0.99	0.93	0.99

The GRU, LSTM, and BiLSTM models, as recurrent architectures, exhibited similar performance trends on the SMOTE-sampled dataset. For GRU, the Class 1 f1-score significantly increased from 0.77 with undersampling to 0.88 with SMOTE, while for LSTM and BiLSTM, the scores improved from 0.82 and 0.80, respectively, to 0.89. These values were slightly lower than the f1-scores achieved on the imbalanced dataset (GRU: 0.91, LSTM: 0.90, BiLSTM: 0.91), reflecting a minor trade-off in performance for the majority class due to synthetic data generation. Overall accuracy for all three models remained consistent at 0.98 across SMOTE and the imbalanced dataset, further confirming the effectiveness of SMOTE in enabling recurrent models to better detect minority class instances without compromising their generalization capability.

DNN and MLP performed better than most of the other DL models across all sampling techniques. The DNN model achieved a Class 1 f1-score of 0.91 with

SMOTE, a slight decrease from 0.92 on the imbalanced dataset and an increase from 0.83 with undersampling. MLP achieved an f1-score of 0.92 for Class 1 with SMOTE which reflects a slight decrease compared to its performance with the imbalanced dataset (0.93) but a significant increase from 0.81 with undersampling.

Among the models, TabTransformer achieved the best performance, with a Class 1 precision of 0.91, a recall of 0.96, and an f1-score of 0.93. This represents similar results to its performance on the imbalanced dataset with a slight improvement over Class 1 Recall from 0.94 to 0.96, a slight decrease in Class 1 Precision from 0.91 to 0.91 and f1-scores are consistent at 0.93. On the other hand, it demonstrates substantial improvement compared to RU, where the f1-score for Class 1 was only 0.77. The overall accuracy of TabTransformer remained high on SMOTE at 0.99, matching its performance with the imbalanced dataset and surpassing the 0.95 accuracy obtained with undersampling.

4.4.1.3.2 Performance comparison of USL models on private dataset with SMOTE

In USL models, the evaluation of RBM and Autoencoder models highlights the influence of sampling techniques on performance. RBM showed moderate improvements with undersampling but struggled with low precision for fraudulent cases. In contrast, Autoencoder with SMOTE achieved better overall metrics, including increased precision, f1-scores, and accuracy, demonstrating SMOTE's effectiveness in handling data imbalance and improving fraud detection.

Table 12. Performance Comparison of USL Models on Private Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	0.94	0.11	0.48	0.67	0.63	0.19	0.49
Autoencoder	0.99	0.41	0.88	0.87	0.93	0.56	0.88

For the RBM model, all evaluation metrics showed a decline compared to the imbalanced dataset but improved relative to the undersampled dataset. The model achieved an accuracy of 0.49, with f1-scores of 0.63 and 0.19 for Class 0 and Class 1, respectively. It demonstrated better performance in identifying fraudulent transactions, as evidenced by a recall of 0.67 for Class 1 compared to 0.48 for Class 0. However, the model was more precise in identifying legitimate transactions, achieving a precision of 0.94 for Class 0 and 0.11 for Class 1.

Autoencoder's results with SMOTE showed notable improvements compared to both the imbalanced and undersampled datasets. Precision for Class 1 increased to 0.41, while remaining stable at 0.99 for Class 0. Recall for Class 0 improved slightly from 0.85 to 0.88, while Class 1 experienced a minor decrease from 0.88 to 0.87. f1-scores improved for both classes, rising from 0.91 to 0.93 for Class 0 and from 0.52 to 0.56 for Class 1. Additionally, overall accuracy increased from 0.85 to 0.88. These results suggest that SMOTE effectively enhances the model's ability to balance performance across both classes, particularly by improving the detection of fraudulent transactions.

4.4.1.4 Model performances on private dataset with majority voting using RU

To address the limitations associated with RU, a majority voting approach combined with this sampling technique. By leveraging multiple subsets, this approach aims to mitigate the potential biases and instability caused by training on a single

undersampled dataset. However, training each DL model ten times to implement this method is not computationally feasible, given the significant resources and time required. As a result, this strategy was exclusively applied to ML models. Table 13 presents the performance of these models under this approach.

Table 13. Performance Comparison of ML Models on Private Dataset with Majority Voting

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.59	0.93	0.97	0.96	0.73	0.94
SVM	1.00	0.67	0.95	0.97	0.98	0.80	0.96
XGBoost	1.00	0.76	0.97	0.99	0.98	0.86	0.97
Random Forests	1.00	0.76	0.97	0.99	0.98	0.86	0.97

Logistic Regression achieved a class 1 precision of 0.59, recall of 0.97, and an f1-score of 0.73, with an overall accuracy of 94%. While the precision for class 1 is relatively lower compared to other models, the high recall indicates its effectiveness in identifying fraudulent cases. SVM demonstrated a more balanced performance with a class 1 precision of 0.67, recall of 0.97, and an f1-score of 0.80, achieving an accuracy of 96%. This indicates that SVM was more effective in maintaining precision while retaining a high recall compared to Logistic Regression.

XGBoost and Random Forests both achieved the highest metrics among the models, with identical class 1 precision (0.76), recall (0.99), and f1-score (0.86). Their overall accuracy was also the highest at 97%, highlighting their robustness and effectiveness when leveraging the majority voting strategy. The consistency between these two models suggests that ensemble-based techniques benefit more from the use of multiple undersampled subsets.

Overall, Random Forests outperformed all other models with each sampling technique on the private dataset among the ML models. XGBoost also performed well, with slightly lower performance compared to Random Forests. When we compare the performance of Random Forests across the sampling techniques, it showed the best performance on the imbalanced dataset. Among the DL models, TabTransformer demonstrated the highest performance on the private dataset, yielding results comparable to those of Random Forests, with the exception of the undersampled dataset. These results highlight the potential of both ML and DL approaches in fraud detection.

4.4.1.5 Hyperparameter tuning for DL models

Hyperparameter tuning is a critical process in DL, enabling models to achieve their optimal performance by identifying the best combination of parameters. In this study, hyperparameter tuning was applied exclusively to DL models, as these architectures exhibit a high sensitivity to hyperparameter configurations due to their complex structures. In contrast, ML models, which performed robustly without extensive tuning, have simpler parameter spaces that do not typically yield significant gains from exhaustive optimization.

Among the DL models evaluated, hyperparameter tuning was limited to the top two performers: DNN and TabTransformer. This selection was driven by their superior predictive performance on the imbalanced dataset, as determined by the evaluation metrics detailed in the Methodology chapter. The imbalanced dataset was prioritized for hyperparameter tuning because it consistently produced the highest performance across all sampling techniques, including undersampling, SMOTE, and

Majority Voting. This highlights the imbalanced dataset's integrity and richness in capturing the underlying patterns necessary for effective fraud detection.

By focusing on the best-performing models and the imbalanced dataset, computational resources were efficiently allocated to optimize the architecture with the greatest potential for improvement. This approach ensures that the study's findings remain both methodologically rigorous and practically relevant, emphasizing the importance of refining models where performance gains are most impactful.

The hyperparameter tuning for the DNN model was conducted using Optuna, focusing on six key hyperparameters: the number of units in the four existing layers, the learning rate, and the batch size. The tuning ranges were carefully designed to allow a thorough yet computationally efficient exploration. Table 14 shows the range and values that were tried for each hyperparameter. The optimization metric used was the f1-score for class 1, which balanced precision and recall. The hyperparameter tuning process identified the optimal configuration for the DNN model as having 130 units in the first layer, 70 units in the second layer, 25 units in the third layer, and 12 units in the fourth layer, with a learning rate of 0.0011 and a batch size of 128.

Table 14. Hyperparameter Values for DNN

Hyperparameter	Range/Value	Optimal Value
First Layer Units	100 to 150 (step=10)	130
Second Layer Units	50 to 80 (step=10)	70
Third Layer Units	20 to 40 (step=5)	25
Fourth Layer Units	10 to 20 (step=2)	12
Learning Rate	$1 \times 10^{(-4)}$ to $5 \times 10^{(-3)}$ (log)	0.0011
Batch Size	[128, 256, 512]	128

Before tuning, the DNN model demonstrated robust performance, with an f1-score of 0.92 for class 1, precision of 0.91, and recall of 0.92. These metrics reflected a well-balanced ability to detect fraudulent cases while minimizing false positives. After tuning, the model's f1-score for class 1 remained stable at 0.92, with a slight increase in precision to 0.92 and recall holding steady at 0.92. The overall accuracy improved marginally from 0.98 to 0.99, and the ROC AUC score saw a minor increase from 0.9966 to 0.9969. These results are presented in Table 15. Despite these adjustments, the improvements were very limited, likely due to the model already being well-optimized for the dataset. This strong baseline performance may have constrained the potential for further gains through hyperparameter tuning, as the existing architecture and parameters were already effectively capturing the dataset's underlying patterns.

Table 15. DNN Performance Before and After Hyperparameter Tuning

Performance Metrics	Before Tuning	After Tuning
Class 1 F1-Score	0.92	0.92
Class 1 Recall	0.92	0.92
Class 1 Precision	0.91	0.92
Accuracy	0.98	0.99
ROC AUC	0.9966	0.9969

The hyperparameter tuning for the TabTransformer model, conducted using Optuna, focused on optimizing its architectural and training parameters to maximize its predictive performance on the imbalanced dataset. The tuned hyperparameters included the number of attention heads (num_heads), embedding dimension (embed_dim), the number of transformer layers (num_layers), the dimensionality of the multi-layer perceptron (mlp_hidden_dim), the learning rate, and the batch size. The ranges for these hyperparameters were carefully defined to balance exploration

and computational efficiency. Table 16 summarizes the range and values evaluated for each hyperparameter. The f1-score for class 1 was used as the optimization metric to prioritize the model's ability to detect minority class instances effectively. After the tuning process, the optimal hyperparameters for the TabTransformer model were determined to include 16 attention heads, an embedding dimension of 64, 3 transformer layers, an MLP hidden dimension of 254, a learning rate of 0.0002, and a batch size of 256. This configuration effectively balances model complexity with training efficiency.

Table 16. Hyperparameter Values for TabTransformer

Hyperparameter	Range/Value	Optimal Value
Number of Attention Heads (num_heads)	[2, 4, 8, 16]	16
Embedding Dimension	$2 \times \text{num_heads}$ to $8 \times \text{num_heads}$ (step=num_heads)	64
Number of Transformer Layers	2 to 8	3
MLP Hidden Dimension	32 to 256	254
Learning Rate	$1 \times 10^{(-5)}$ to $5 \times 10^{(-2)}$ (log)	0.0002
Batch Size	[64, 128, 256, 512]	256

Before tuning, the TabTransformer achieved an f1-score of 0.93 for class 1, with a precision of 0.92 and recall of 0.94. Other metrics, such as an overall accuracy of 0.99 and a ROC AUC score of 0.9972, indicated its strong baseline performance. After applying the optimized hyperparameters, the f1-score for class 1 remained at 0.93, while precision slightly decreased to 0.90, and recall improved to 0.96. The overall accuracy remained stable at 0.99, and the ROC AUC score exhibited a negligible decline to 0.9971. These results, which are presented in table 17, highlight the minimal impact of hyperparameter tuning, which can be attributed to the model's already high performance prior to optimization.

Table 17. TabTransformer Performance Before and After Hyperparameter Tuning

Performance Metrics	Before Tuning	After Tuning
Class 1 F1-Score	0.93	0.93
Class 1 Recall	0.94	0.96
Class 1 Precision	0.92	0.90
Accuracy	0.99	0.99
ROC AUC	0.9972	0.9971

The limited improvements underscore the diminishing returns of hyperparameter tuning when working with models that are well-calibrated and inherently suited to the problem, as was the case for the TabTransformer. Nevertheless, the tuning process ensured that the model's configuration was systematically refined, validating its robustness and stability for detecting fraudulent cases with high reliability. The identified hyperparameters serve as a validated configuration for future applications of the TabTransformer in similar fraud detection tasks.

4.4.2 Model performances for public dataset

The subsequent sections analyze model evaluation outcomes, emphasizing the effects of various sampling methods, individual model performance, and their significance in fraud detection using a public dataset.

4.4.2.1 Model performances on imbalanced public dataset

The ML models were assessed using the imbalanced public fraud detection dataset, maintaining the natural class imbalance. This analysis served to evaluate model performance in scenarios reflecting real-world conditions, where legitimate transactions significantly outnumber fraudulent ones. The same evaluation metrics

used for the private dataset were applied to ensure consistency in highlighting the models' ability to detect fraudulent transactions.

4.4.2.1.1 Performance comparison of SL models on public imbalanced dataset

As shown in table 18, all four algorithms attained an overall accuracy of 1.00 on the test set. However, this high accuracy results from the substantial class imbalance, despite the model's relatively lower ability to detect fraudulent instances. Logistic Regression achieved a precision of 0.91 for class 1, coupled with a recall of 0.62 and an f1-score of 0.73, suggesting somewhat reduced sensitivity relative to the other methods. SVM obtained a higher recall of 0.69 and an f1-score of 0.80 for class 1, while Random Forest and XGBoost both reached a recall of 0.79 for the same class, leading to f1-scores of 0.86 and 0.87, respectively. Precision for class 1 was also highest with XGBoost (0.98), followed by Random Forest and SVM at 0.96, and Logistic Regression at 0.91. Across all four models, precision, recall, and f1-scores for class 0 were consistently perfect (1.00), underscoring these classifiers' success for identifying the majority class.

Table 18. Performance Comparison of ML Models on Public Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class	Class 1	Class	Class	Class	Class	
Logistic Regression	1.00	0.91	1.00	0.62	1.00	0.73	1.00
SVM	1.00	0.96	1.00	0.69	1.00	0.80	1.00
Random Forests	1.00	0.96	1.00	0.79	1.00	0.86	1.00
XGBoost	1.00	0.98	1.00	0.79	1.00	0.87	1.00

DL models demonstrate diverse outcomes depending on their architecture as presented in table 19. Precision for class 0 was consistently perfect (1.00) across all

models, indicating the complete absence of false positives in predicting non-fraudulent transactions. For class 1, precision exhibited variation, with the LSTM model achieving the highest score of 0.98, followed by the RNN and BiLSTM models with scores of 0.96 and 0.94, respectively. In contrast, the TabTransformer model demonstrated the lowest precision for class 1, with a score of 0.84, reflecting its relatively higher rate of false positives in fraud detection.

Table 19. Performance Comparison of DL Models on Public Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class	Class 1	Class	Class	Class	Class	
TabTransformer	1.00	0.84	1.00	0.61	1.00	0.70	1.00
RNN	1.00	0.96	1.00	0.73	1.00	0.83	1.00
GRU	1.00	0.87	1.00	0.79	1.00	0.83	1.00
DNN	1.00	0.88	1.00	0.82	1.00	0.85	1.00
CNN	1.00	0.87	1.00	0.84	1.00	0.85	1.00
LSTM	1.00	0.98	1.00	0.77	1.00	0.86	1.00
BiLSTM	1.00	0.94	1.00	0.79	1.00	0.86	1.00
MLP	1.00	0.92	1.00	0.80	1.00	0.86	1.00

The recall for class 0 was also perfect (1.00) across all models, highlighting their effectiveness in correctly identifying non-fraudulent transactions. However, recall for class 1 varied significantly. The CNN and BiLSTM models achieved the highest recall (0.84), indicating their superior ability to capture fraudulent transactions. The TabTransformer model, with a recall of 0.61 for class 1, performed the poorest among all models in identifying fraudulent cases. The remaining models, including RNN, GRU, and MLP, demonstrated moderate recall values for class 1, ranging from 0.73 to 0.80.

The f1-score for class 0 was consistently high (1.00) due to the perfect precision and recall achieved by all models. For class 1, the f1-score ranged from 0.70 for the TabTransformer model to 0.86 for the LSTM, BiLSTM, and MLP models. These results indicate that the LSTM, BiLSTM, and MLP models achieved a better balance between precision and recall for the minority class, making them more effective at handling the imbalanced nature of the dataset. In contrast, the TabTransformer model's lower f1-score for class 1 highlights its difficulty in addressing the imbalance effectively.

4.4.2.1.2 Performance comparison of USL models on public imbalanced dataset

USL models, including the Restricted Boltzmann Machine (RBM) and Autoencoder, were also applied to the imbalanced dataset. The results in table 20 indicate that while both models achieve a perfect precision of 1.00 for Class 0, the Autoencoder outperforms the RBM in nearly every other metric. In particular, the Autoencoder attains a higher recall for Class 0 (0.97 vs. 0.86) and an improved precision for Class 1 (0.06 vs. 0.01). This improvement for Class 1 is reflected in an F1-score of 0.11 for the Autoencoder, which, although still relatively low, surpasses the F1-score of 0.02 obtained by the RBM. Furthermore, the Autoencoder achieves an overall accuracy of 0.97, substantially exceeding the 0.86 accuracy of the RBM. These findings suggest that the Autoencoder is better suited for capturing class distinctions under the current experimental conditions, despite the challenge posed by imbalanced performance across classes.

Table 20. Performance Comparison of USL Models on Public Imbalanced Dataset

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	1.00	0.01	0.86	0.88	0.92	0.02	0.86
Autoencoder	1.00	0.06	0.97	0.86	0.99	0.11	0.97

These results highlight the varying effectiveness of ML and DL models on the imbalanced dataset. Among ML models, XGBoost and Random Forests demonstrate strong performance, while among DL approaches, MLP and BiLSTM models show considerable potential for detecting fraudulent transactions. The overall findings underscore the challenges posed by the imbalanced nature of the dataset, while also showcasing the strengths and limitations of different model architectures in addressing this issue. Furthermore, the USL models' performance emphasizes the need for tailored strategies when applying such approaches to highly imbalanced datasets.

4.4.2.2 Model performances for public dataset with RU

In this section, the performance of SL and USL models on the undersampled dataset is analyzed. Undersampling was employed to balance the dataset by reducing the majority class, ensuring that the models could allocate equal attention to detecting both fraudulent and legitimate transactions. The subsequent subsections provide a comprehensive evaluation of the models, focusing on their ability to adapt to the smaller dataset and the revised class distribution.

4.4.2.2.1 Performance comparison of SL models on public dataset with RU

The results of ML models on the public dataset with RU are presented in table 21.

The overall accuracy of all models remained relatively high—ranging from 0.96 to

0.99—yet the trade-offs in class-level metrics became more pronounced compared to the imbalanced dataset. All algorithms showed a marked increase in recall for the minority class, with values exceeding 0.87 for every model; however, this improvement came at the expense of much lower precision for the minority class, which diminished the corresponding F1-scores. For Logistic Regression, recall for class 1 rose from 0.62 in the imbalanced setting to 0.90 after undersampling, yet its precision for the same class dropped from 0.91 to 0.04, resulting in a decrease in the F1-score from 0.73 to 0.08. Similar trends were observed with the other algorithms, where heightened recalls above 0.87 for class 1 were offset by precision values that did not exceed 0.10. Overall, these findings underscore the typical trade-off associated with undersampling: although models are better able to identify minority-class instances, they produce more false positives, thus reducing their precision and moderating gains in class-specific F1-scores.

Table 21 Performance Comparison of ML Models on Public Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.04	0.96	0.90	0.98	0.08	0.96
XGBoost	1.00	0.05	0.96	0.91	0.98	0.09	0.96
Random Forests	1.00	0.07	0.98	0.88	0.99	0.12	0.97
SVM	1.00	0.10	0.99	0.87	0.99	0.19	0.99

With RU, Class 1 precision decreased significantly, while modest improvements were observed in Class 1 recall in the DL models. The substantial decline in Class 1 precision resulted in a slight to moderate decrease in accuracy and a dramatic reduction in Class 1 f1-score across the models. Among the DL models, CNN achieved the best performance with a Class 1 precision of 0.07, recall of 0.87,

and f1-score of 0.12. LSTM, RNN, and BiLSTM produced similar results, with minor variations compared to CNN. TabTransformer demonstrated the poorest performance, with a Class 1 F1-score of 0.01 on the undersampled dataset. These findings indicate that DL models struggled to achieve a balance between precision and recall for Class 1 under RU.

Table 22. Performance Comparison of DL Models on Public Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
TabTransformer	1.00	0.01	0.66	0.95	0.80	0.01	0.66
MLP	1.00	0.03	0.94	0.89	0.97	0.06	0.94
GRU	1.00	0.03	0.95	0.90	0.97	0.07	0.95
DNN	1.00	0.05	0.96	0.88	0.98	0.09	0.96
LSTM	1.00	0.05	0.97	0.88	0.98	0.09	0.97
RNN	1.00	0.05	0.97	0.88	0.98	0.09	0.97
BiLSTM	1.00	0.05	0.97	0.89	0.98	0.09	0.97
CNN	1.00	0.07	0.98	0.87	0.99	0.12	0.98

In comparison to the imbalanced dataset, RU yields a general improvement in recall scores for the minority class among ML models, with Random Forests and SVM showing the most notable gains. However, this comes at the cost of overall precision and f1-scores, particularly for DL models, which struggle to balance performance for undersampled data. While models such as CNN demonstrate high recall, their lack of precision highlights the challenges of achieving a balanced detection rate. These trends emphasize the importance of selecting appropriate models and techniques tailored to the resampling strategy and the unique characteristics of the dataset.

4.4.2.2.2 Performance comparison of USL models on public dataset with RU

Performances of RBM and Autoencoder are demonstrated in table 23. Their performance declined across nearly all evaluation metrics when compared to the imbalanced dataset. RBM achieved a recall of 0.51 and an f1-score of 0.01 for Class 1, while the Autoencoder delivered a recall of 0.89 and an f1-score of 0.05 for the minority class. Overall performance of the USL models remained inadequate especially for the minority class with RU.

Table 23. Performance Comparison of USL Models on Public Dataset with RU

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	1.00	0.01	0.81	0.51	0.89	0.01	0.81
Autoencoder	1.00	0.02	0.93	0.89	0.96	0.05	0.93

These findings highlight the trade-offs introduced by the RU technique.

While some improvements were observed for some evaluation metrics, most of the models struggled to generalize effectively for the undersampled data. The results emphasize the need for balancing data resampling strategies with appropriate model selection to achieve robust fraud detection outcomes.

4.4.2.3 Model performances on public dataset with SMOTE

This section evaluates the performance of SL and USL models on the dataset enhanced through oversampling. SMOTE was employed to generate synthetic samples for the minority class, effectively addressing the class imbalance and improving the models' capacity to identify fraudulent transactions. The subsequent subsections provide a detailed assessment of how the models utilized the oversampled data and adapted to the increased representation of the minority class.

4.4.2.3.1 Performance comparison of SL models on public dataset with SMOTE

Under SMOTE, each model showed a clear shift toward better detection of minority-class instances than in the imbalanced scenario and a more balanced precision–recall trade-off than under random undersampling, as can be interpreted by table 24.

Logistic Regression and SVM attained higher yet still suboptimal f1-scores for Class 1, at 0.12 and 0.16, respectively, relative to RU. Their overall accuracy, 0.97 and 0.98, respectively, was lower than that of the imbalanced dataset due to a significant drop in Class 1 precision but remained higher than that of the undersampled dataset due to the improved balance between precision and recall for Class 1. XGBoost and Random Forests demonstrated substantial improvements in Class 1 precision, reaching 0.74 and 0.90, respectively. This led to markedly better performance compared to random undersampling and comparable performance to the imbalanced dataset, with F1-scores of 0.79 and 0.86 and accuracy of 1.00.

Table 24. Performance Comparison of ML Models on Public Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.07	0.97	0.89	0.99	0.12	0.97
SVM	1.00	0.09	0.98	0.83	0.99	0.16	0.98
XGBoost	1.00	0.74	1.00	0.84	1.00	0.79	1.00
Random Forests	1.00	0.90	1.00	0.82	1.00	0.86	1.00

Performances of DL models under SMOTE are presented in table 25. They display substantial improvements compared to the imbalanced and undersampled dataset in some metrics. The MLP and DNN models achieve notable performance, with MLP achieving a recall of 0.78 and an f1-score of 0.77 for the minority class, while DNN achieves a recall of 0.84 and an f1-score of 0.77. Other architectures,

including TabTransformer, LSTM and BiLSTM, also demonstrate good performance, with f1-scores of 0.76, 0.75 and 0.74, respectively.

Table 25. Performance Comparison of DL Models on Public Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RNN	1.00	0.47	1.00	0.81	1.00	0.59	1.00
GRU	1.00	0.47	1.00	0.83	1.00	0.60	1.00
CNN	1.00	0.61	1.00	0.85	1.00	0.71	1.00
BiLSTM	1.00	0.66	1.00	0.83	1.00	0.74	1.00
LSTM	1.00	0.67	1.00	0.84	1.00	0.75	1.00
TabTransformer	1.00	0.72	1.00	0.81	1.00	0.76	1.00
MLP	1.00	0.76	1.00	0.78	1.00	0.77	1.00
DNN	1.00	0.71	1.00	0.84	1.00	0.77	1.00

Compared to RU, SMOTE achieves more balanced performance, enabling models like SVM, XGBoost, DNN, and MLP to demonstrate robust detection capabilities while maintaining precision and f1-scores. This improvement highlights SMOTE's effectiveness in mitigating the challenges of class imbalance and enhancing fraud detection outcomes.

4.4.2.3.2 Performance comparison of USL models on public dataset with SMOTE

Despite the use of oversampling, performances of RBM and Autoencoder remains insufficient for the precision of the minority class as shown in table 26. RBM achieved a recall of 0.21 for the minority class, with an f1-score of 0.01, while the Autoencoder displayed better performance with a recall of 0.86 and an f1-score of 0.10. Their overall accuracy remained at 0.93 and 0.97, respectively, demonstrating limited capacity to generalize effectively under SMOTE.

Table 26. Performance Comparison of USL Models on Public Dataset with SMOTE

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
RBM	1.00	0.01	0.94	0.21	0.97	0.01	0.93
Autoencoder	1.00	0.05	0.97	0.86	0.99	0.10	0.97

4.4.2.4 Model performances on public dataset with majority voting using RU

Majority Voting with RU was applied exclusively to ML models, delivering varying levels of success in detecting fraudulent transactions and the results are presented in table 27. Logistic Regression, XGBoost and Random Forests demonstrated minor improvements in Class 1 f1-scores with 0.10, 0.10 and 0.13, respectively, over single RU. SVM achieved a recall of 0.88 for the minority class, while its f1-score remained at 0.19. The performance of majority voting was inferior to that of other sampling techniques across most evaluation metrics and demonstrated not much improvement compared to the single RU.

Table 27 Performance Comparison of ML Models on Public Dataset with Majority Voting

Models	Precision		Recall		F1-Score		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Logistic Regression	1.00	0.05	0.97	0.89	0.98	0.10	0.97
Random Forests	1.00	0.05	0.97	0.91	0.98	0.10	0.97
XGBoost	1.00	0.07	0.98	0.88	0.99	0.13	0.98
SVM	1.00	0.11	0.99	0.88	0.99	0.19	0.99

Overall, the evaluation of SL and USL models across different sampling techniques highlights critical insights into the effectiveness of ML and DL approaches for fraud detection. On the imbalanced dataset, models such as XGBoost and MLP achieved superior performance, with high precision and recall, while USL

approaches struggled to balance precision and recall. RU improved recall for some models, particularly for SVM and Random Forests, but significantly decreased the precision of fraud class. SMOTE proved to be more effective sampling technique than undersampling for enhancing recall without sacrificing overall precision, particularly benefiting models like Random Forests, MLP, and DNN.

4.5 Explainability of best performing models

In this study, XAI methods were applied to enhance the interpretability of the models, ensuring that their decisions could be understood and validated. To achieve this, XAI techniques were applied to one best-performing ML model (Random Forests) and one DL model (TabTransformer), chosen based on their predictive performance. For the Random Forest model, explainability was achieved through feature importance analysis, global SHAP, showing the overall importance of each feature, and local SHAP, detailing how individual features influenced specific predictions, and LIME to offer an alternative localized view. For the TabTransformer model, explainability methods tailored for DL were applied. Feature importance with permutation, DeepLIFT, to generate both global explanations specifically for all fraud instances in the dataset and local explanations to some specific predictions, and LIME were applied to the TabTransformer. By employing a diverse range of explainability techniques, the study aimed to provide a comprehensive understanding of the decision-making processes of the selected models. The following sections present the results of these XAI methods, offering insights into the key factors driving model predictions.

4.5.1 XAI for Random Forests

The global explanations for the Random Forest model, derived from feature importance and SHAP analysis, reveal key insights into the model's decision-making process. The feature importance plot in figure 15 demonstrates that ResidencyBoolean is the most influential feature, followed closely by TranCountryTranCount and the transactional feature V13_1. Other highly important features include V2, V6_1, and V6_2, alongside engineered features such as MaxAmount, MinAmount, and AvgAmount. The predominance of ResidencyBoolean and TranCountryTranCount suggests that demographic and transactional patterns play critical roles in identifying fraudulent activity. The transactional features, denoted by "V" prefixes, further highlight the relevance of specific transaction-related variables, suggesting that subtle patterns in transactions contribute significantly to the model's predictions.

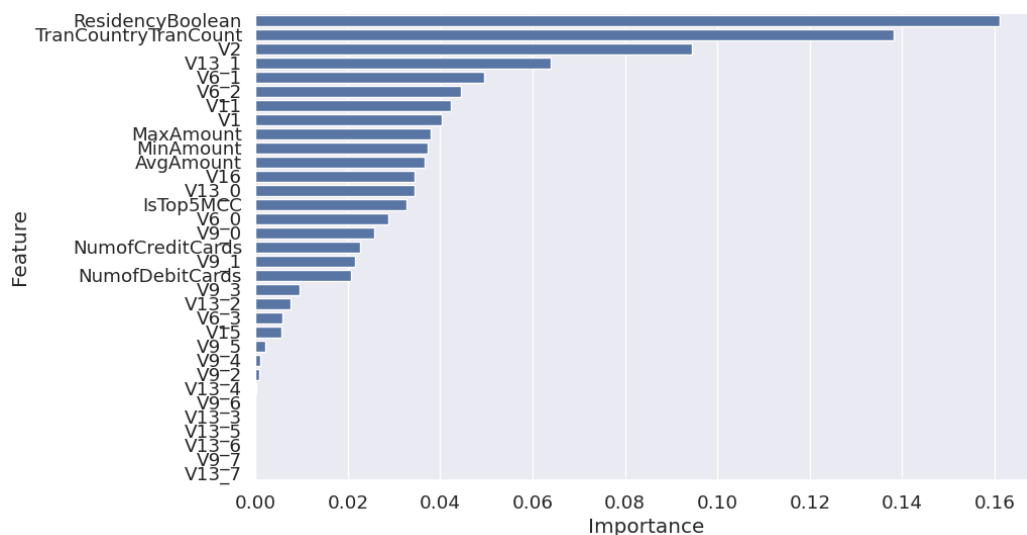


Figure 15. Feature importance of Random Forests

The SHAP summary plot in figure 16 complements this analysis by providing a more nuanced view of how each feature impacts the model's predictions across the

dataset. In this visualization, ResidencyBoolean, TranCountryTranCount and V13_1 again emerge as the most impactful features, with clear distinctions in their SHAP values for low and high feature values. For instance, a high value of ResidencyBoolean appears to have a strong negative impact on the prediction of fraud, whereas a low value increases the likelihood of a positive classification. This means that when a transaction occurred somewhere else than the customer's residency (ResidencyBoolean=0), the possibility of this transaction being fraud increases. Similarly, TranCountryTranCount consistently influences the likelihood of fraud: when customers have fewer transactions in a particular country (i.e., those transactions are less common for them), it generally increases the probability of fraud detection. Additionally, higher values of V13_1 tend to contribute positively to the model's output, whereas lower values correspondingly pull the prediction in the opposite direction. As previously discussed, this means that if the transaction's security type is unknown, it is highly likely to be fraudulent.

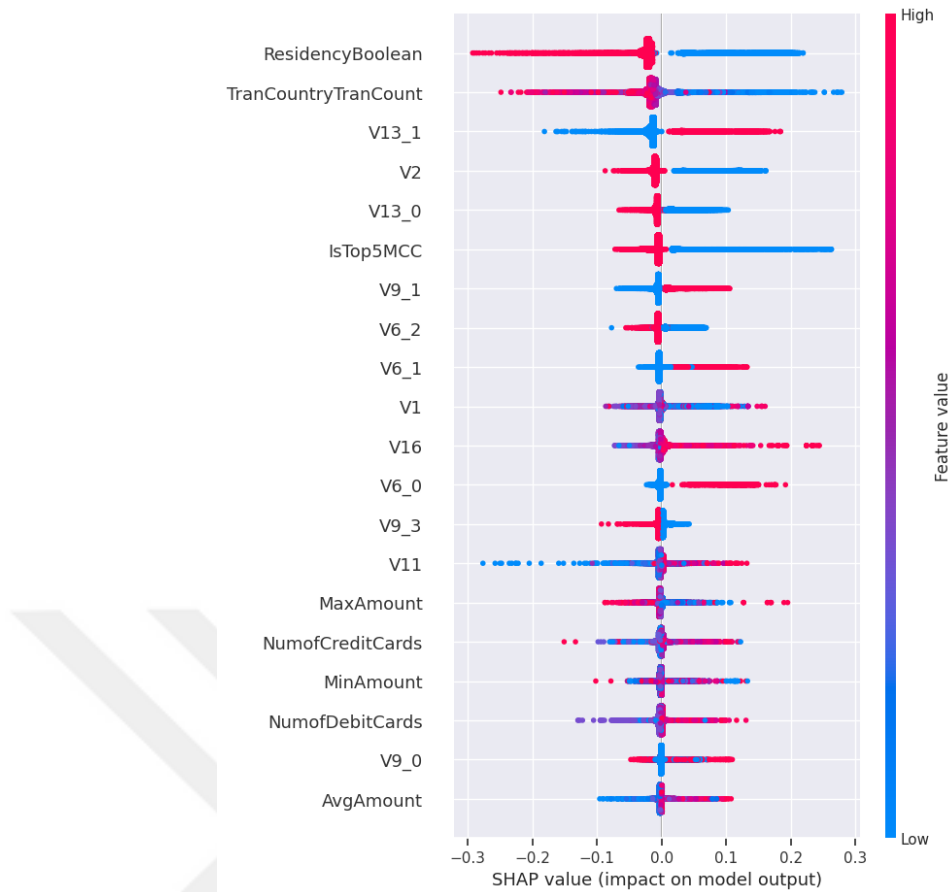


Figure 16. SHAP summary plot of Random Forests

Interestingly, several transactional features exhibit scattered SHAP value distributions, indicating that their contributions are context-dependent and vary significantly between individual cases. The local SHAP and LIME graphs provide detailed insights into how individual features contribute to the fraud predictions for three specific instances in the dataset. These explanations help understand the unique role of features in influencing the model's output for specific cases.

For the prediction at first index, the model's output score was 0.94, indicating a strong likelihood of fraud. According to SHAP, the features $\text{ResidencyBoolean} = 0.0$, $V2 = 0.0$, $\text{TranCountryTranCount} = -1.95$ (has a range between -3.35 and 4.76 after transformation), and $V13_1 = 1.0$ had significant contributions to this prediction as shown in figure 17. Notably, ResidencyBoolean and $V2$ had the largest positive

impacts, as they are positioned prominently on the positive side of the x-axis. This means their values increased the model's confidence in classifying the instance as fraudulent. LIME also confirms that $V13_1 > 0.0$ and $TranCountryTranCount \leq -0.46$ had the largest positive contributions to the model's prediction of fraud as can be seen in figure 18. Overall, this instance is classified as fraudulent because the transaction occurred in a country different from the customer's residency and abroad, conducting a transaction in that country is uncommon for the customer, and the security type associated with the transaction is unknown.

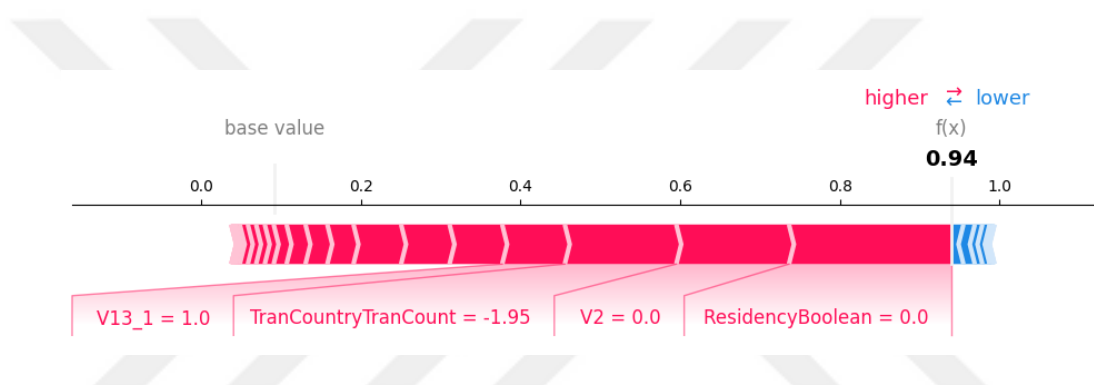


Figure 17. Local SHAP results for the first instance

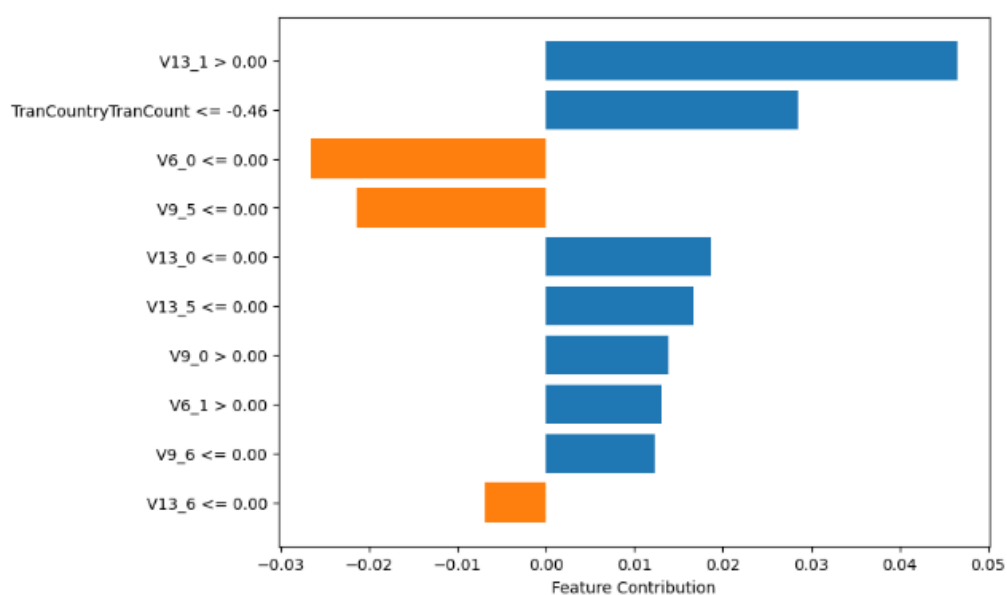


Figure 18. Local LIME results for the first instance

At the second instance, the model output was 1.00, indicating an almost certain classification of fraud. SHAP results in figure 19 demonstrate that the features ResidencyBoolean = 0.0, V2 = 0.0, TranCountryTranCount = -2.35, V13_1 = 1.0, and V9_0 = 1.0 were the key drivers of this prediction. According to figure 20, LIME results show that $V13_1 > 0.0$ and $TranCountryTranCount \leq -0.46$ again had the most significant positive impact, strongly supporting the model's decision to classify this instance as fraud. Features such as $V6_0 \leq 0.0$ and $V9_7 \leq 0.0$ had notable negative contributions, counteracting the fraud classification to a degree. We can conclude that, for this instance, in addition to locational and security characteristics common with instance at index 1, making the transaction with saved card data collectively reinforces the high fraud likelihood.

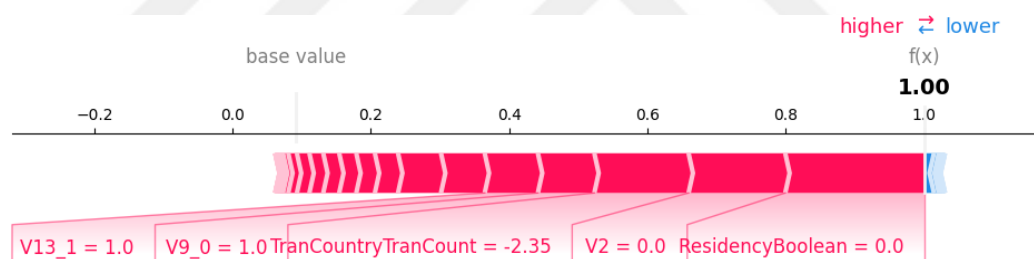


Figure 19. Local SHAP results for the second instance

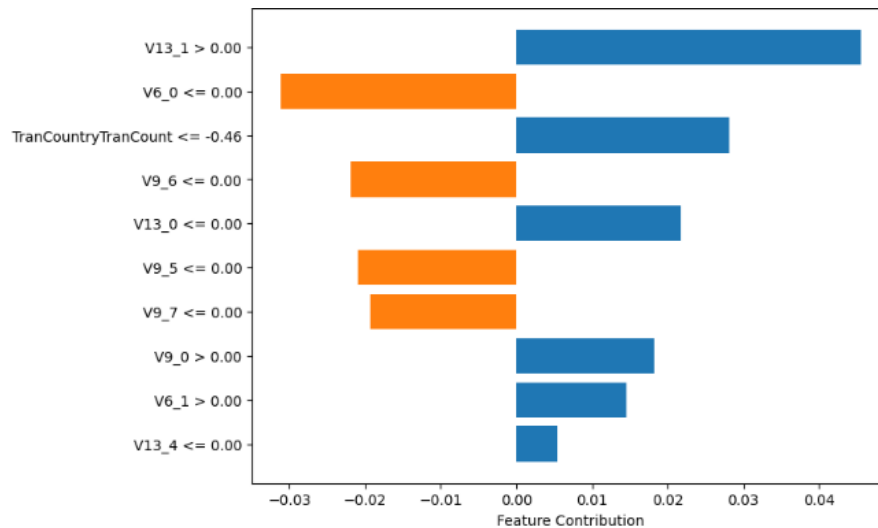


Figure 20 Local LIME results for the second instance

For the third instance, the model's output score was 0.94, again reflecting a strong likelihood of fraud. SHAP results in figure 21 show that the main contributors in this instance were IsTop5MCC = 0.0, V9_5 = 1.0, V11 = 2.63, and MaxAmount = 2.19 (has a range between -5.68 and 6.35 after transformation). LIME results in figure 22 support the high positive contribution of V9_5 > 0.0 along with V6_0 <= 0.0. The most important reason for classifying this instance as fraud was that the merchant category of the transaction differed from the top 5 merchant categories the customer typically shops from and the transactional characteristics such as transaction type and service provider.

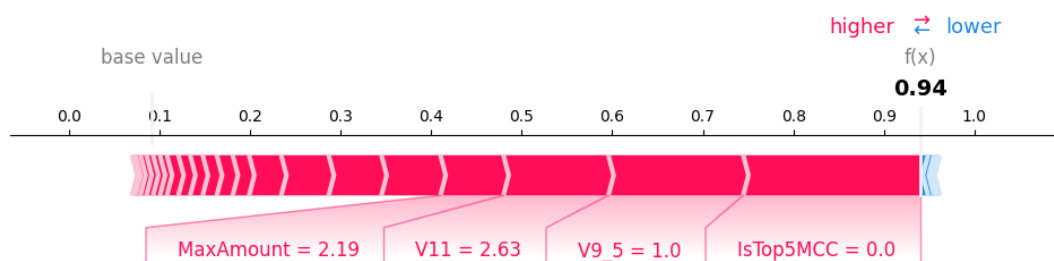


Figure 21. Local SHAP results for the third instance

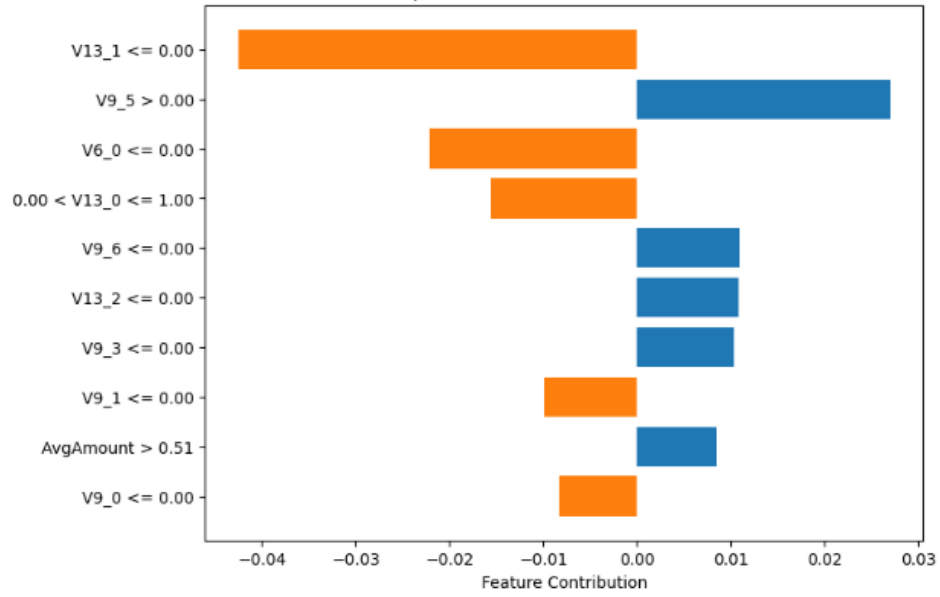


Figure 22. Local LIME results for the third instance

In conclusion, across these instances, ResidencyBoolean, TranCountryTranCount and V13_1 consistently appears as a significant factor in the model's predictions, demonstrating its importance in identifying fraudulent behavior. The LIME explanations provide a complementary perspective to SHAP, offering an intuitive understanding of how specific feature values influence the model's predictions for individual instances. The alignment between the feature importance, SHAP and LIME results underscores the reliability of the Random Forest model in prioritizing meaningful variables. Together, these explanations provide a comprehensive understanding of the model's behavior, offering critical insights into the factors most indicative of fraudulent activity.

4.5.2 XAI for TabTransformer

The feature importance analysis for the TabTransformer model in figure 23 reveals significant insights into the predictive behavior of the model for fraud detection.

Based on the permutation importance results, the feature ResidencyBoolean emerged

as the most influential predictor, exhibiting a notably higher importance score compared to other features. This highlights the critical role of a transaction occurring in a different country than the customer's residency, a scenario that often correlates with fraudulent activity. The second most important feature, TranCountryTranCount, measures the frequency of transactions conducted in transaction countries and further underscores the relevance of cross-border transactional behavior in identifying fraudulent cases. Additional features, such as V13_1, NumofDebitCards, and MaxAmount, also demonstrated moderate importance, indicating their role in capturing patterns associated with transactional volume, account activity, and spending behavior. The remaining features exhibited progressively lower importance, suggesting minimal impact on the model's predictions relative to the top-ranked features.

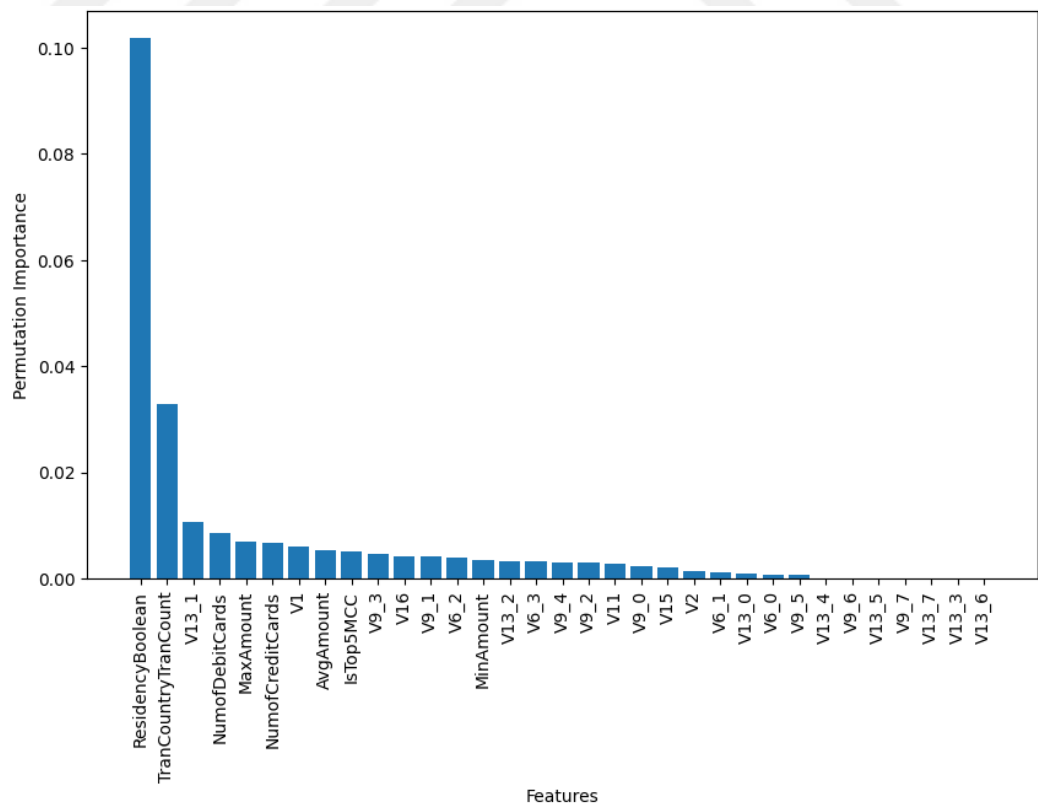


Figure 23. Feature importance of TabTransformer

DeepLIFT, calculated exclusively for fraud instances, provides a complementary perspective on the model's interpretability. Unlike the permutation importance results, DeepLIFT identified V9_1 as the most impactful feature, followed by IsTop5MCC and MinAmount as shown in figure 24. These results suggest that customer spending habits are critical factors associated with fraudulent activity. Interestingly, while ResidencyBoolean ranked highest in the permutation importance analysis, it held a relatively lower rank in the DeepLIFT results, reflecting its localized significance rather than a broader, global impact across all fraudulent cases. Similarly, TranCountryTranCount and MaxAmount, which were prominent in the permutation importance analysis, appeared less influential according to DeepLIFT. Conversely, features such as V9_1, which ranked lower in the permutation importance analysis, gained prominence in the DeepLIFT results.

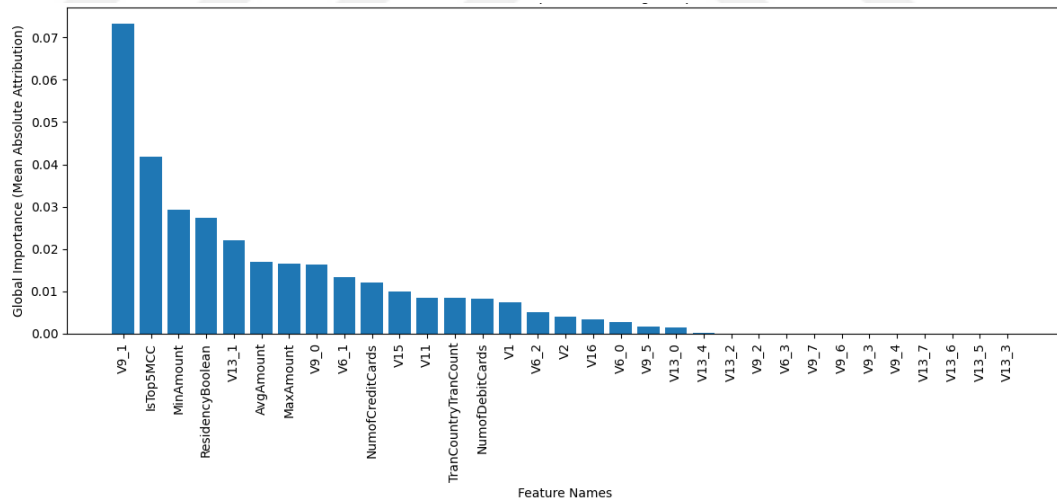


Figure 24. Global feature importance with DeepLIFT for TabTransformer

These findings underscore the importance of employing multiple interpretability techniques to gain a comprehensive understanding of the model's behavior. By combining the overall feature ranking from permutation importance

with the fraud-specific global insights provided by DeepLIFT, a more holistic interpretation of the TabTransformer model's decision-making process can be achieved. This dual approach not only highlights the key features driving fraud predictions but also provides valuable insights for improving model transparency and trustworthiness in high-stakes applications such as fraud detection. Additionally, exploring local explanations for specific instances using DeepLIFT and LIME offers a detailed view of how feature contributions vary across individual predictions. This instance-level perspective uncovers subtle patterns that complement global analyses, enhancing the interpretability of the model's decisions.

In figure 25, TranCountryTranCount, with a feature value of -1.81, has the strongest positive attribution for the first instance, meaning its low value strongly supports the model's classification of this instance as fraudulent. This indicates that the transactional behavior captured by this feature aligns with a fraud-related pattern. In contrast, MinAmount=0.99 (has a range between -1.47 and 9.74 after transformation) and V16=1.47 have negative attributions, pulling the prediction away from fraud. Features like IsTop5MCC=1 and MaxAmount=-0.83 have positive attributions, meaning their values align with patterns that increase the likelihood of fraud. The ranking provided by LIME in figure 26 underscores the importance of TranCountryTranCount as the most influential feature for this instance, as in the DeepLIFT analysis. However, additional features such as V13_5 and V9_3 gain prominence in the LIME explanation, suggesting variability in the local context that might not be as evident in the global attributions provided by DeepLIFT. When both graphs are analyzed, we can conclude that customer's spending habits like the locations of transaction and minimum, maximum and average amounts they spent,

and transactional features like the type and security of the transaction have greater impact in classifying this feature as fraud.

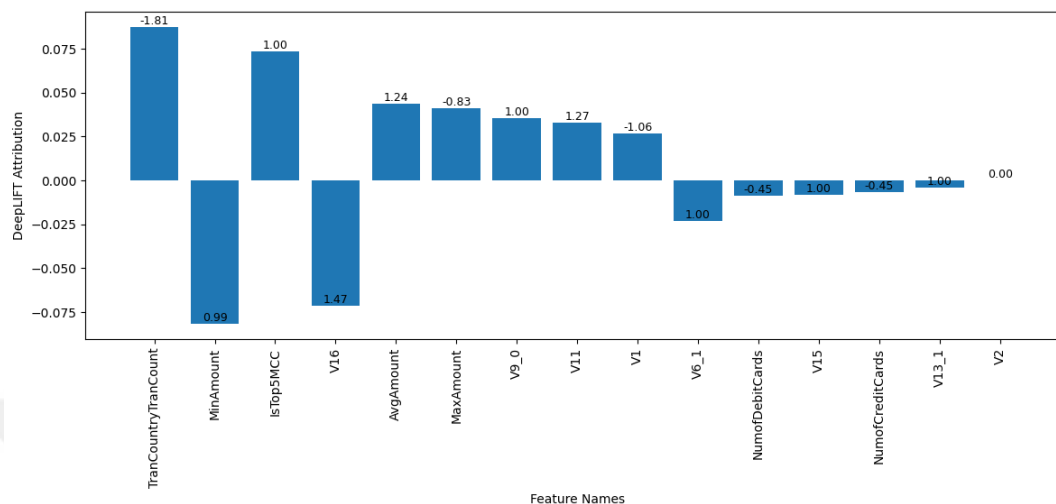


Figure 25. Local feature importance with DeepLIFT for the first instance

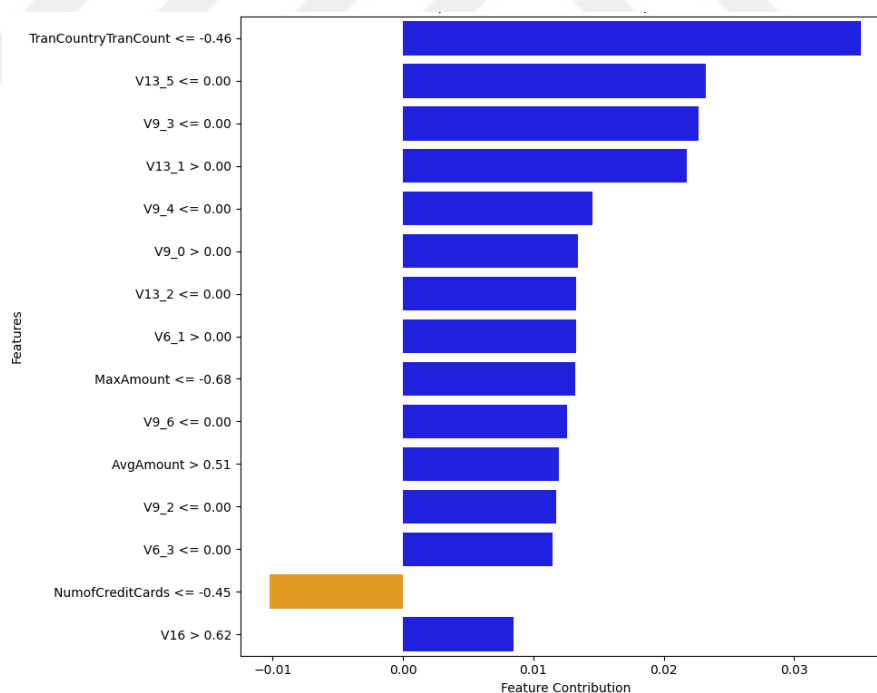


Figure 26. Local feature importance with LIME for the first instance

In the second instance, a distinctly different pattern emerged as shown in figure 27. The feature V16 had the most significant negative attribution, pulling the prediction away from fraud. Meanwhile, V13_1, V9_1, and V6_0 being 1, had the strongest positive contributions in classifying this instance as fraudulent. Interestingly, TranCountryTranCount, which generally has a positive impact for its low values, showed a negative contribution in this case, reducing the likelihood of fraud. In LIME explanations, V13_1 and TranCountryTranCount emerge as the most influential features positively contributing to the model's output as shown in figure 28. Their strong impact signifies that the values of these features align well with the characteristics associated with the predicted class. Meanwhile, features such as V9_5 and NumofCreditCards show notable negative contributions, suggesting that their values counteract the prediction, aligning more closely with the characteristics of the alternative class.

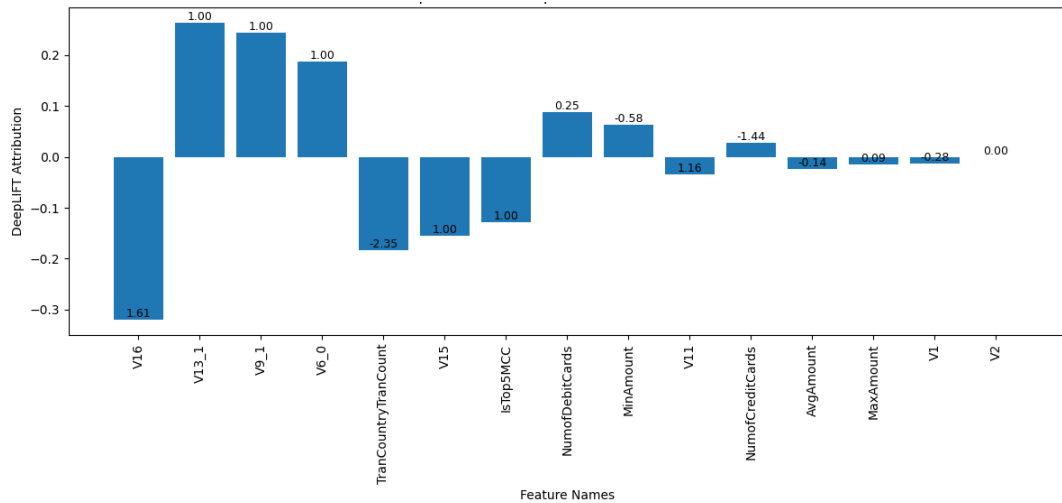


Figure 27. Local feature importance with DeepLIFT for the second instance

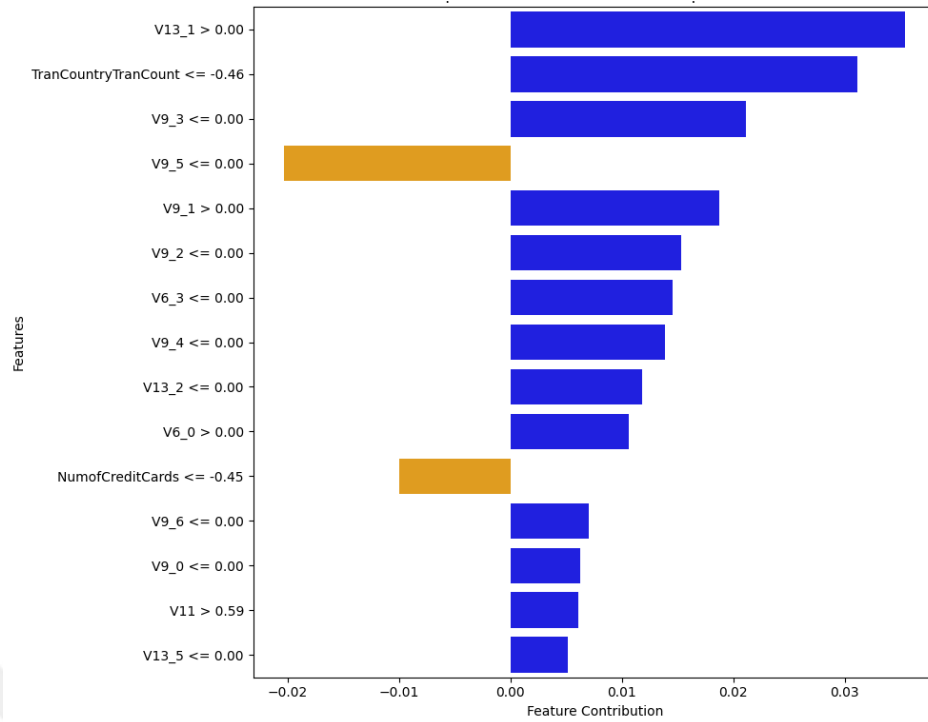


Figure 28. Local feature importance with LIME for the second instance

For the third instance, TranCountryTranCount again showed the highest positive contribution for the fraud classification, both in DeepLIFT and LIME analyses, in figures 29 and 30 respectively. However, negative contributions from features such as AvgAmount = 1.78 (has a range between -5.25 and 7.61 after transformation), MinAmount = -0.91 and IsTop5MCC = 1 played a critical role in opposing the fraud classification. According to LIME, V9 and V13 for different categories have significant positive impact on classifying this feature as fraud. In that context, customer spending habits depending on location and amount, and transactional features such as the type and security of the transaction decides the class of this instance as fraud again.

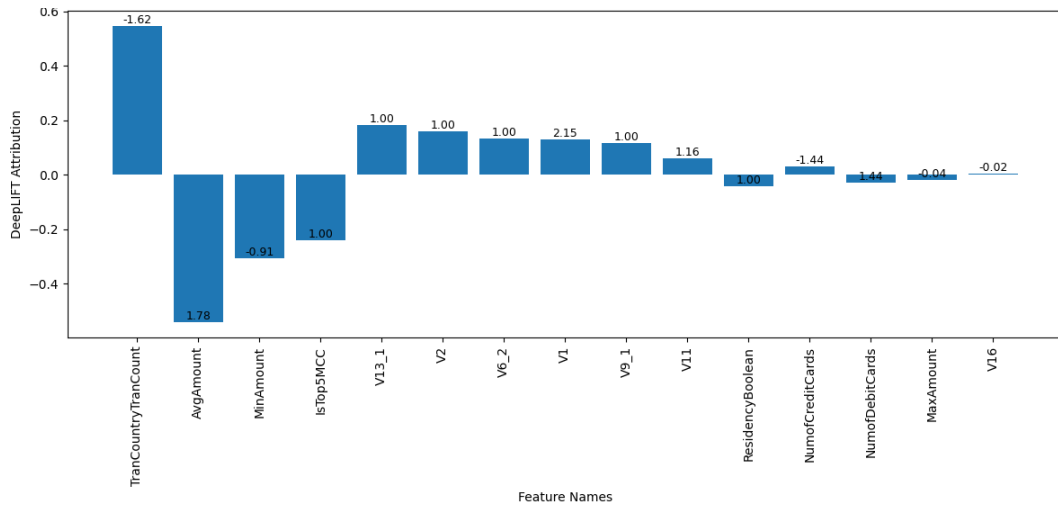


Figure 29. Local feature importance with DeepLIFT for the third instance

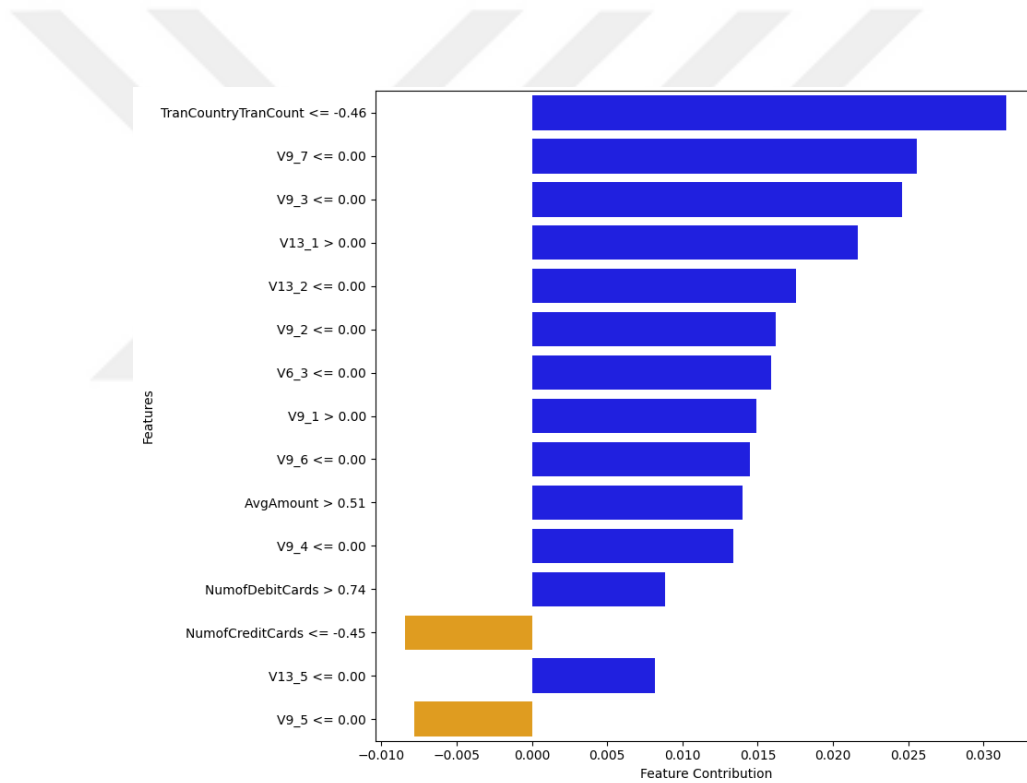


Figure 30. Local feature importance with LIME for the third instance

In conclusion, the explainability analysis of the TabTransformer model demonstrates the effectiveness of using diverse interpretability techniques to uncover key insights into the model's predictions for fraud detection. The permutation importance method highlighted ResidencyBoolean and TranCountryTranCount as

critical global features, emphasizing the role of cross-border transactional behavior in identifying fraud. In contrast, DeepLIFT provided fraud-specific perspectives, showcasing the importance of features like V9_1 and IsTop5MCC in capturing customer spending patterns. Although the results vary across explanation tools and instances for local explanations, TranCountryTranCount has the strongest positive effect on fraud classification along with different categories of V13 and V9 mostly. Together, these methods highlight the interplay between global and local interpretability, reinforcing the significance of features related to customer location, transaction frequency, spending behavior, security and type of the transaction in fraud classification.

CHAPTER 5

DISCUSSION

The discussion chapter aims to interpret the findings of this research within the broader context of fraud detection and AI. The primary aim of this study was to evaluate the performance of ML and DL algorithms in detecting fraudulent transactions within the context of imbalanced datasets, using real-world private bank data, assess the most effective sampling techniques for handling class imbalance, understand how XAI techniques can enhance model interpretability while maintaining performance, evaluate the impact of engineered features on fraud detection, and compare results between private bank data and publicly available datasets. This chapter begins by summarizing the key findings and interpreting them in the context of existing literature. It then discusses the practical, theoretical, and methodological implications of the results, followed by a comparison with previous studies in the field.

The SL models, particularly Random Forest and XGBoost among ML models and TabTransformer and MLP among DL models, effectively handled the imbalanced dataset, demonstrating their suitability for real-world fraud detection scenarios. While ML models offered greater computational efficiency, DL models also proved to be effective in capturing complex patterns in the data. The USL models showed limitations in detecting fraudulent transactions on the imbalanced dataset.

Compared to the imbalanced dataset, the undersampled dataset improved the recall for fraudulent transactions across all models. However, this improvement in recall was accompanied by a decrease in precision for Class 1. The reduction in the number of non-fraudulent transactions may lead to the loss of valuable patterns or

relationships present in the majority class, which could help the models distinguish between fraudulent and non-fraudulent transactions. As a result, models may misclassify more non-fraudulent transactions as fraudulent, leading to a drop in precision. Tree-based models, including Random Forest and XGBoost, and DL models such as LSTM and DNN demonstrated the best performance on the undersampled data. For USL models, the undersampled dataset led to a noticeable decline in Class 0 performance for both models, particularly in precision and f1-score. While precision for Class 1 improved, recall either declined or showed minimal improvement, resulting in little change to the f1-score for Class 1. These results highlight the difficulty USL models face in maintaining performance across both classes when applied to undersampled datasets.

The application of SMOTE resulted in improved recall for Class 1 across all models, demonstrating its efficacy in addressing class imbalance. However, the reduction in precision for Class 1 introduces a trade-off, leading to slightly lower f1-scores for some models compared to imbalanced data. On the other hand, the performance gains observed in SMOTE, particularly in Class 1 metrics, highlight its superiority over RU and its ability to enhance model performance compared to the imbalanced dataset in some cases. Random Forests and XGBoost among ML models and TabTransformer and MLP among DL models achieved the best performance on SMOTE data. In USL models, SMOTE shows limited effectiveness with improvements in Class 1 precision often offset by declines in Class 0 precision and other metrics. RBM and Autoencoder both struggle to address class imbalance, showing reduced performance in identifying fraudulent instances and overall accuracy of 46% and 50%, respectively. These findings suggest that SMOTE may not be a suitable approach for USL models in this context.

The results of applying the majority voting approach using RU to ML models show very little improvement in some metrics compared to single RU. Accuracy of all four models stayed at the same level as in RU. Improvements and drawbacks compared to imbalanced dataset are the same as in section single RU, such as declines in Class 0 precisions and increases in Class 1 recalls. Ensemble models, such as Random Forests and XGBoost, maintain robust metrics across all sampling techniques, but the majority voting method shows no clear advantage over the imbalanced dataset, single RU or SMOTE in addressing the challenges of fraud detection.

Overall, Random Forests outperformed all other models with each sampling technique. XGBoost, TabTransformer, MLP and DNN also performed well, with slightly lower performance compared to Random Forests. Random Forests emerged as the best-performing model is a result that aligns closely with existing literature. Random Forests are widely recognized for their robustness in handling imbalanced datasets and their ensemble-based approach excels in capturing complex relationships between features without requiring extensive preprocessing. While recent literature increasingly highlights the use of DL models for large and complex fraud detection datasets, the success of Random Forests in this study suggests that simpler models can remain highly competitive, particularly when the dataset lacks sequential features or when computational efficiency is a priority. This finding underscores the continued relevance of Random Forests as a practical, interpretable, and effective solution for fraud detection, complementing the growing body of research advocating their application in this field.

USL models, on the other hand, performed poorly under each sampling method. RBMs and Autoencoders are designed to detect sequential dependencies and

learn interdependent relationships between variables. However, our data lacks time and customer ID columns, which prevents the identification of patterns over time or deviations in customer behavior. Furthermore, these models rely on strong correlations or interdependencies among features to learn meaningful latent representations. In the absence of such relationships or with anomalies that are not distinctly represented, these models face challenges in distinguishing normal from fraudulent transactions. The static nature of the data, combined with the lack of sequential insights, appears to align better with feature-driven algorithms such as tree-based models, which do not rely on temporal information.

When we compare the results of Random Forests across the sampling techniques it showed the best performance on the imbalanced dataset as most of the other models did. Class 0 results maintain a more stable result set across different sampling methods, but Class 1 Results fluctuate notably. Despite the improved recall, decrease in precision hinders the performance of RU or SMOTE. This might be controversial to the findings of the earlier research in the literature. One potential reason for the decrease in precision under RU could be the loss of informative patterns in the majority class. RU, by reducing the number of majority class samples, might remove instances that are critical for distinguishing between fraudulent and non-fraudulent transactions, leading to higher false positive rates. Similarly, while SMOTE generates synthetic samples for the minority class, it may inadvertently introduce noise or unrealistic patterns that compromise the model's ability to generalize, particularly when the generated data points do not reflect the true underlying distribution.

Explainability analysis reveals critical insights into the influence of demographic, transactional, and spending behavior-related features on model

predictions. ResidencyBoolean and TranCountryTranCount consistently emerged as key global predictors, highlighting the relevance of location-based patterns related to customer spending behavior and their association with fraudulent activity. Local explanations offered by SHAP and LIME further emphasize the transaction-dependent contributions of features like V13_1 and V9 categories in individual fraud cases. For the TabTransformer, DeepLIFT provided complementary insights, underscoring the significance of customer-specific patterns in spending habits. The integration of global and local interpretability methods illustrates the nuanced interplay of features driving fraud predictions, thereby enhancing model transparency and trustworthiness in high-stakes applications.

The explainability analysis underscores the critical role of engineered features in detecting fraudulent instances, as they consistently emerged as the most influential factors in the models' decision-making processes. This finding is further corroborated by the comparative results of Random Forests on datasets with and without feature engineering, which show a 6% increase in the f1-score of Class 1 when engineered features were included. Additionally, EDA highlights that these engineered features exhibit significantly higher correlations with the label compared to most transactional features. This observation aligns with existing literature emphasizing the importance of domain-specific feature engineering in enhancing fraud detection systems. However, the relevance of this finding extends beyond performance metrics; it emphasizes the necessity of integrating expert knowledge into data preprocessing to capture nuanced patterns in fraud detection, offering both theoretical and practical implications for the field.

The results of this study demonstrate that the public dataset yielded similar outcomes to the private dataset in terms of identifying the best-performing models

and observing consistent trends across sampling techniques. Tree-based models showed the best performance and showed comparable rankings and similar trade-offs between precision and recall in both datasets. This consistency suggests that the patterns and relationships captured by the models are generalizable across different contexts, validating the robustness of the methods applied. Moreover, the similar trends across sampling techniques reinforce the notion that these approaches, particularly RU and SMOTE, impact decision boundaries in a predictable manner, regardless of the dataset's origin. These findings contribute to the literature by demonstrating the transferability of model performance and methodological insights between private and public datasets, emphasizing the practical utility of these models for broader fraud detection applications.

The theoretical implications of this research are rooted in its exploration of the intricate dynamics between sampling techniques and model performance. This study demonstrates that the methodological insights gained from private datasets are transferable to public datasets, thereby bridging the often-challenging gap between research and real-world applications. This finding not only reinforces the robustness of the tested methods but also underscores the universal applicability of the approaches across diverse data sources. From a practical standpoint, these insights offer valuable guidance for practitioners tasked with developing fraud detection systems. The research highlights the importance of creating domain-specific features which captures the common and rare behaviors of customers emerges as a pivotal step in enhancing model performance and interpretability. Furthermore, by integrating explainability into the modeling process, the study provides a roadmap for enhancing stakeholder trust and model adoption in high-stakes applications, such as fraud detection in financial institutions.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

This study comprehensively evaluated the performance of ML and DL models in detecting fraudulent transactions across imbalanced datasets. Leveraging private bank data and a publicly available dataset, the research identified Random Forests and TabTransformer as the most effective models, with feature engineering enhancing their performance and XAI adds interpretability to the model results. The transferability of these findings across datasets highlights the robustness of the methodologies employed and their relevance for real-world fraud detection.

While the study yielded significant insights, it is not without limitations. The lack of temporal and sequential data restricted the applicability of USL models such as RBMs and Autoencoders. Additionally, computational constraints restricted the exploration of more resource-intensive models and prevented extensive hyperparameter optimization for all evaluated algorithms. Furthermore, hyperparameter tuning and XAI could be applied to the public dataset to improve the results and interpret the feature importance of this dataset.

Future studies could incorporate temporal or sequential data to improve the effectiveness of USL models and allow for the exploration of time-series approaches such as RNNs, GRUs, and LSTMs. Hybrid models combining SL and USL techniques may also offer enhanced performance by leveraging the strengths of both methodologies. Expanding the methodology to other domains such as healthcare fraud, insurance claims, or cybersecurity could test the generalizability of the findings. Investigating diverse fraud scenarios would provide a broader understanding of the models' applicability and limitations. Integrating larger, multi-

regional datasets and experimenting with cutting-edge techniques like GAN for synthetic data generation could further enhance the robustness and scalability of the approaches. Evaluating the cost-effectiveness and scalability of these models in operational fraud detection pipelines remains a crucial avenue for future research. Practical implementations should focus on optimizing computational efficiency while maintaining high performance.

This study has contributed to fraud detection research by highlighting the impact of feature engineering and explainability on model performance and demonstrating the transferability of findings across diverse datasets. Random Forests and TabTransformer consistently outperformed other models, showcasing their robustness across sampling techniques. The integration of explainable AI tools provided transparency into model decisions, reinforcing trust and accountability. By addressing class imbalance and model interpretability, this research offers actionable insights for developing practical and effective fraud detection systems. The findings lay the groundwork for future advancements in ML applications for high-stakes domains.

APPENDIX A

CROSSTAB HEATMAPS FOR PRIVATE DATA CATEGORICAL VARIABLES

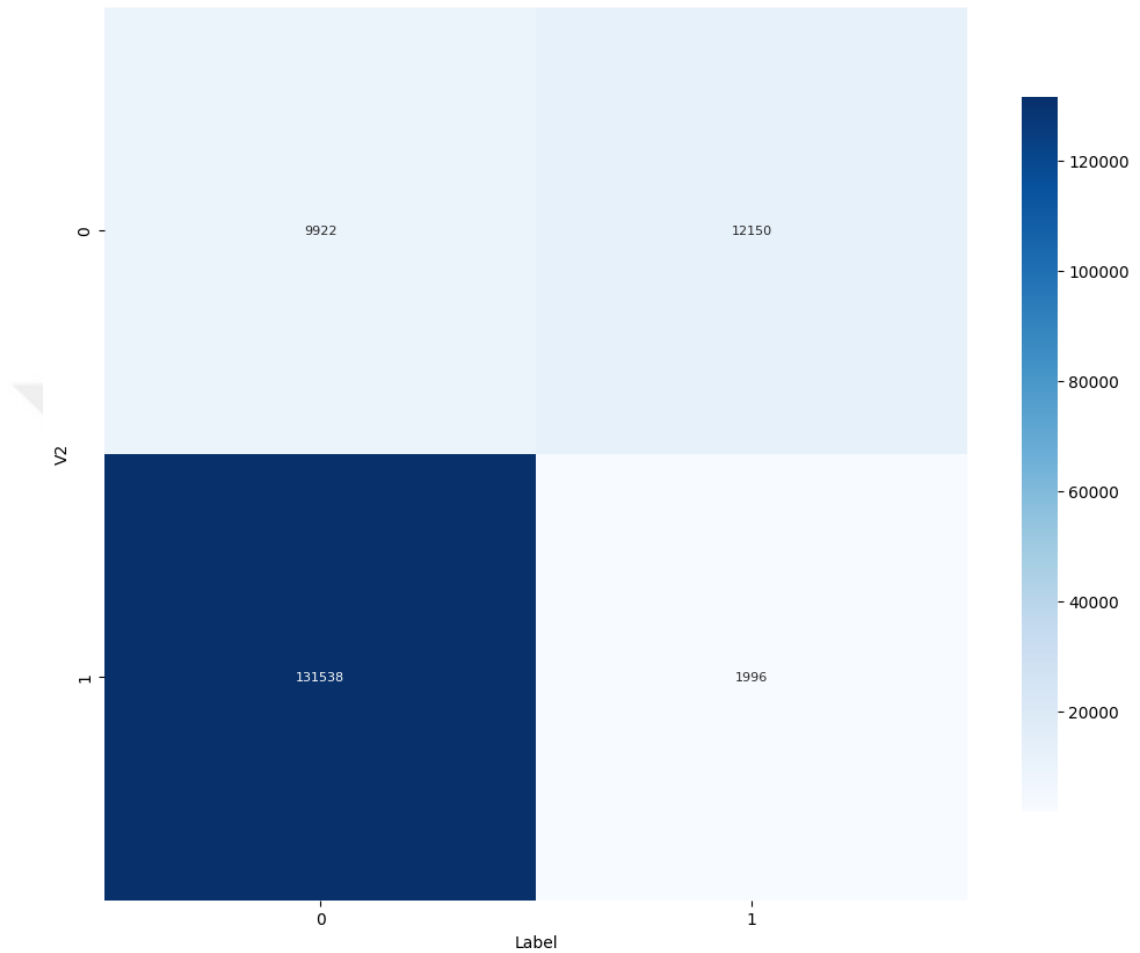


Figure A1. Crosstab for V2 and label

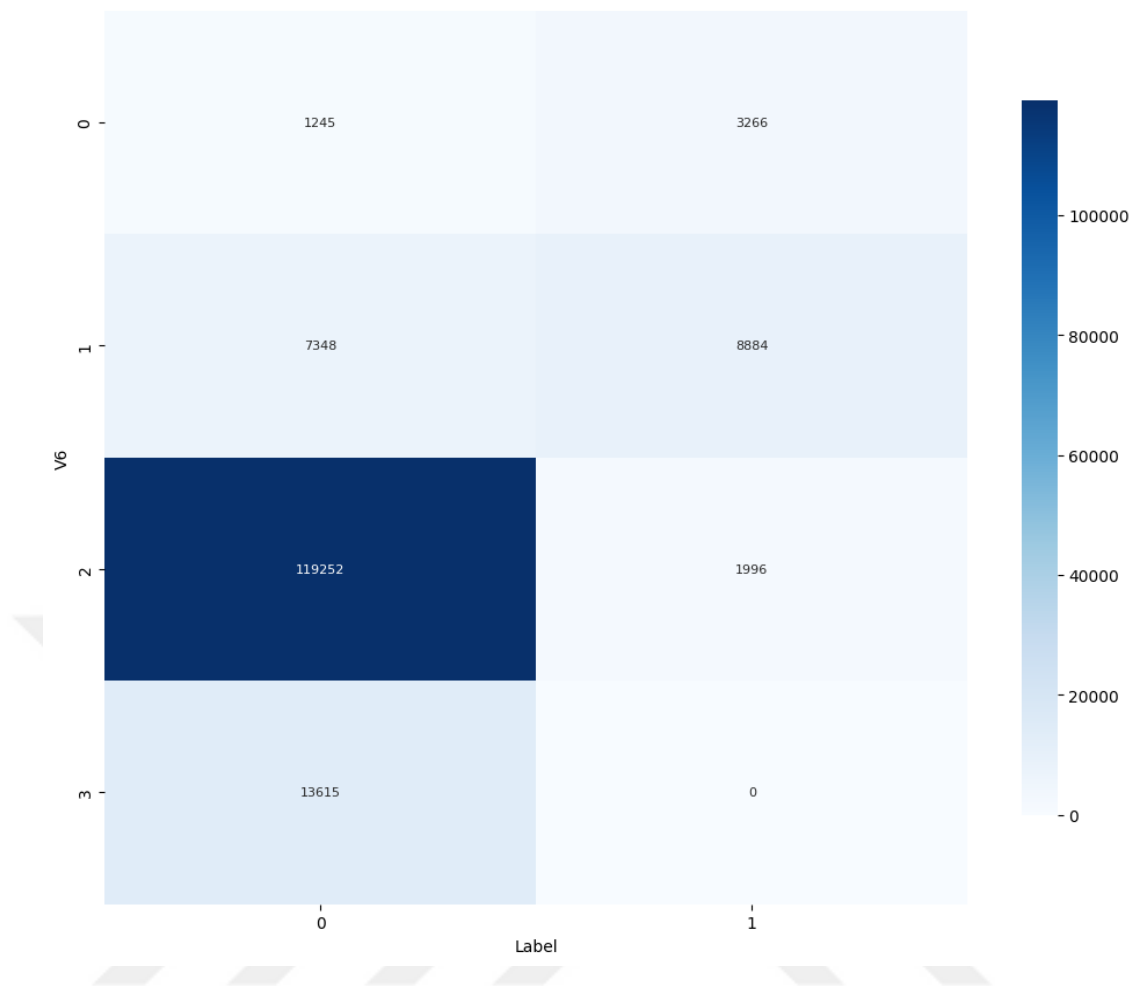


Figure A2. Crosstab for V6 and label



Figure A3. Crosstab for V8 and label

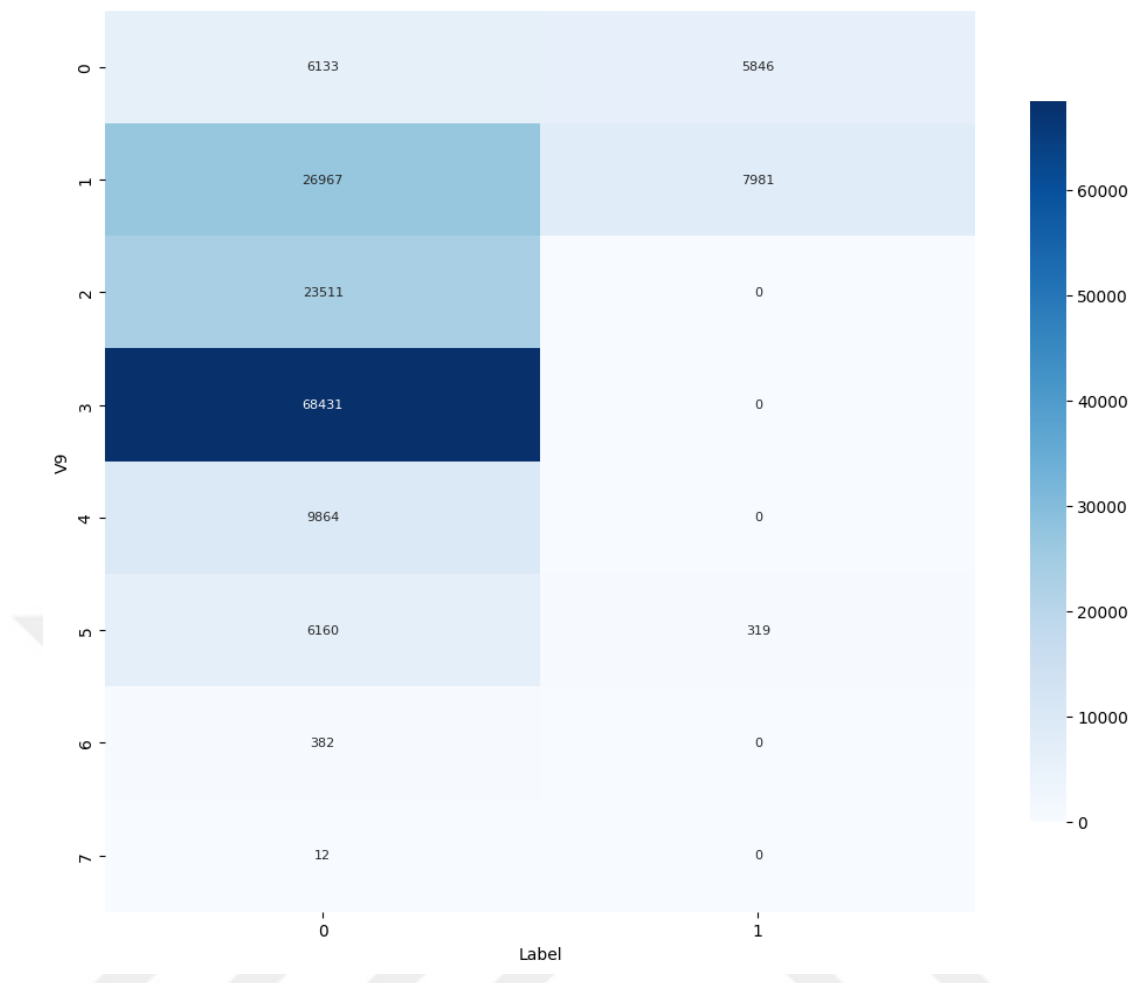


Figure A4. Crosstab for V2 and label

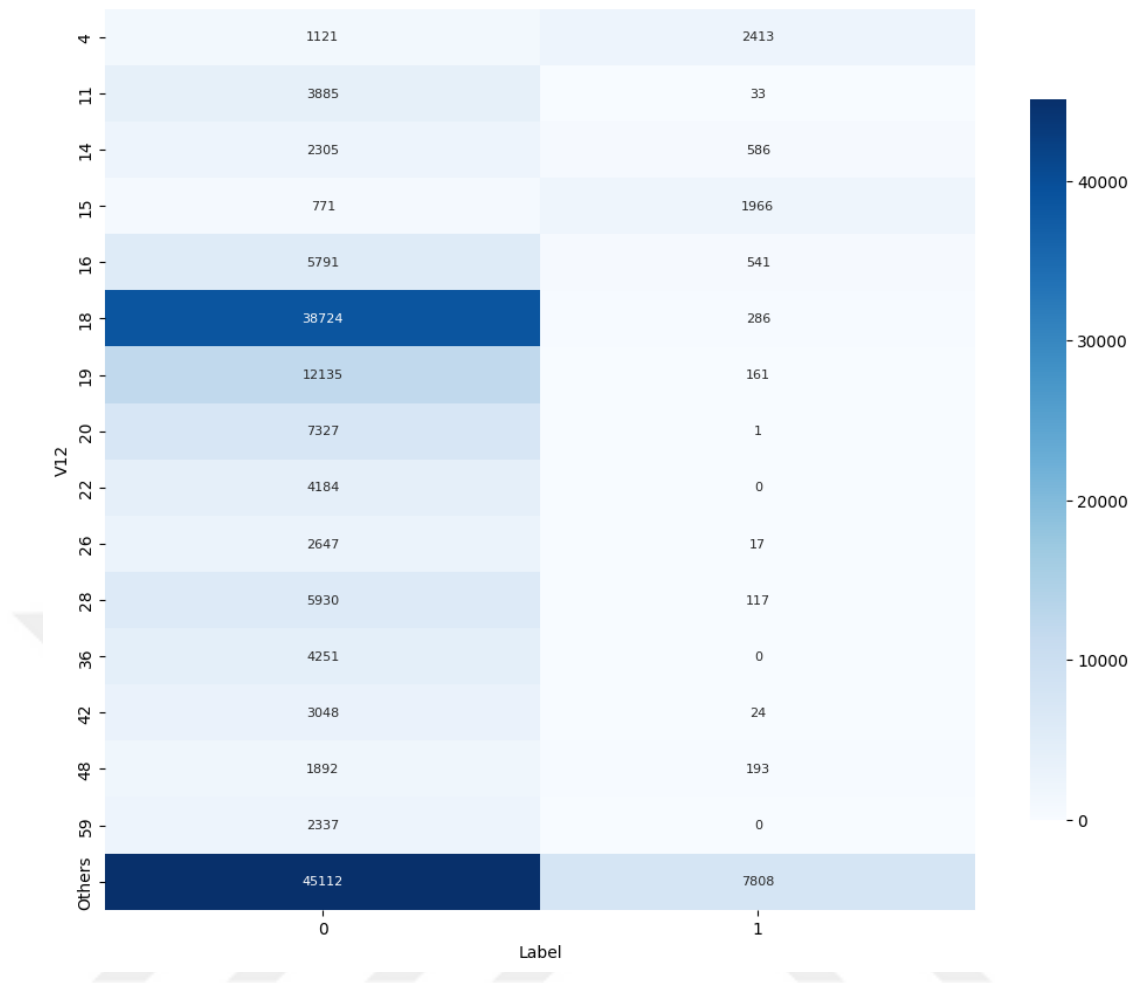


Figure A5. Crosstab for V12 and label

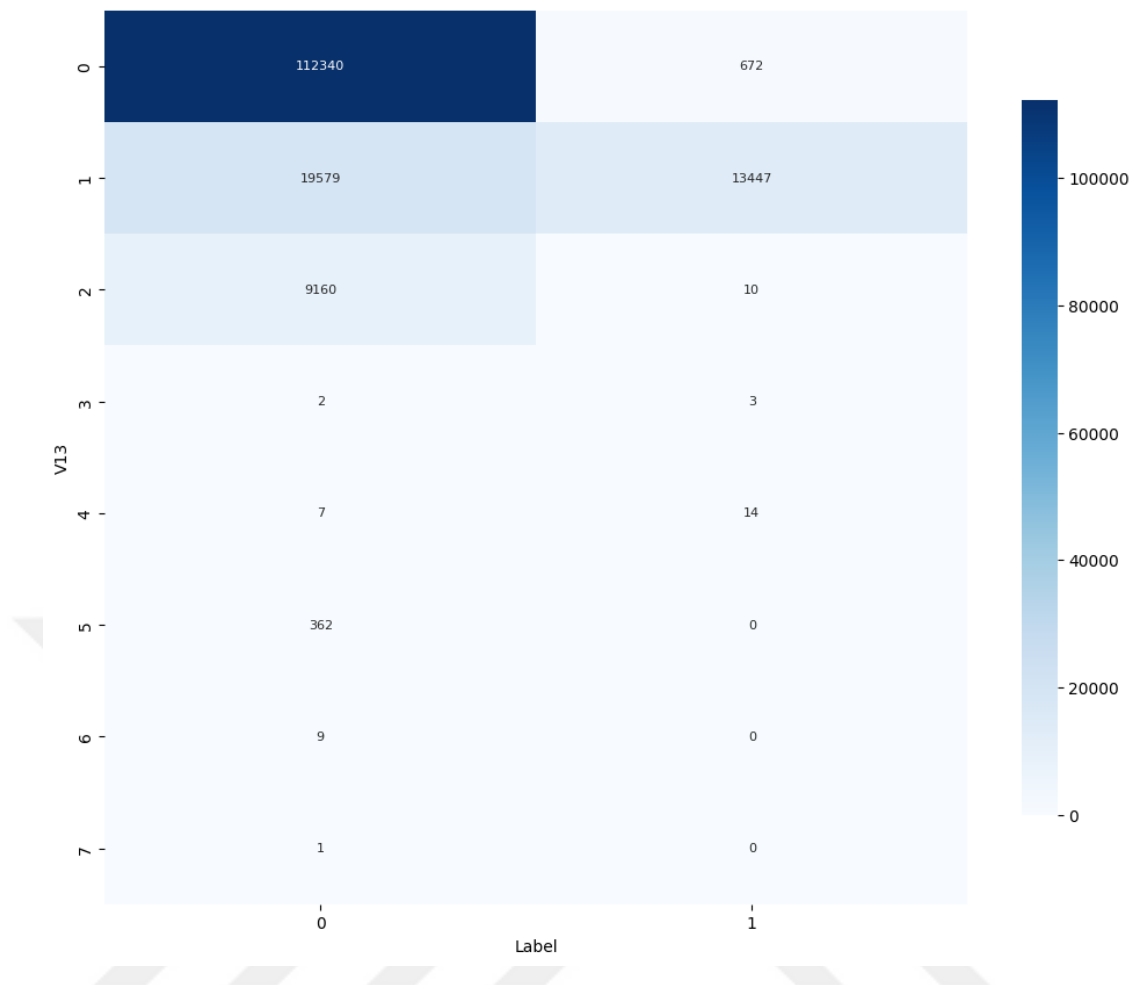


Figure A6. Crosstab for V13 and label

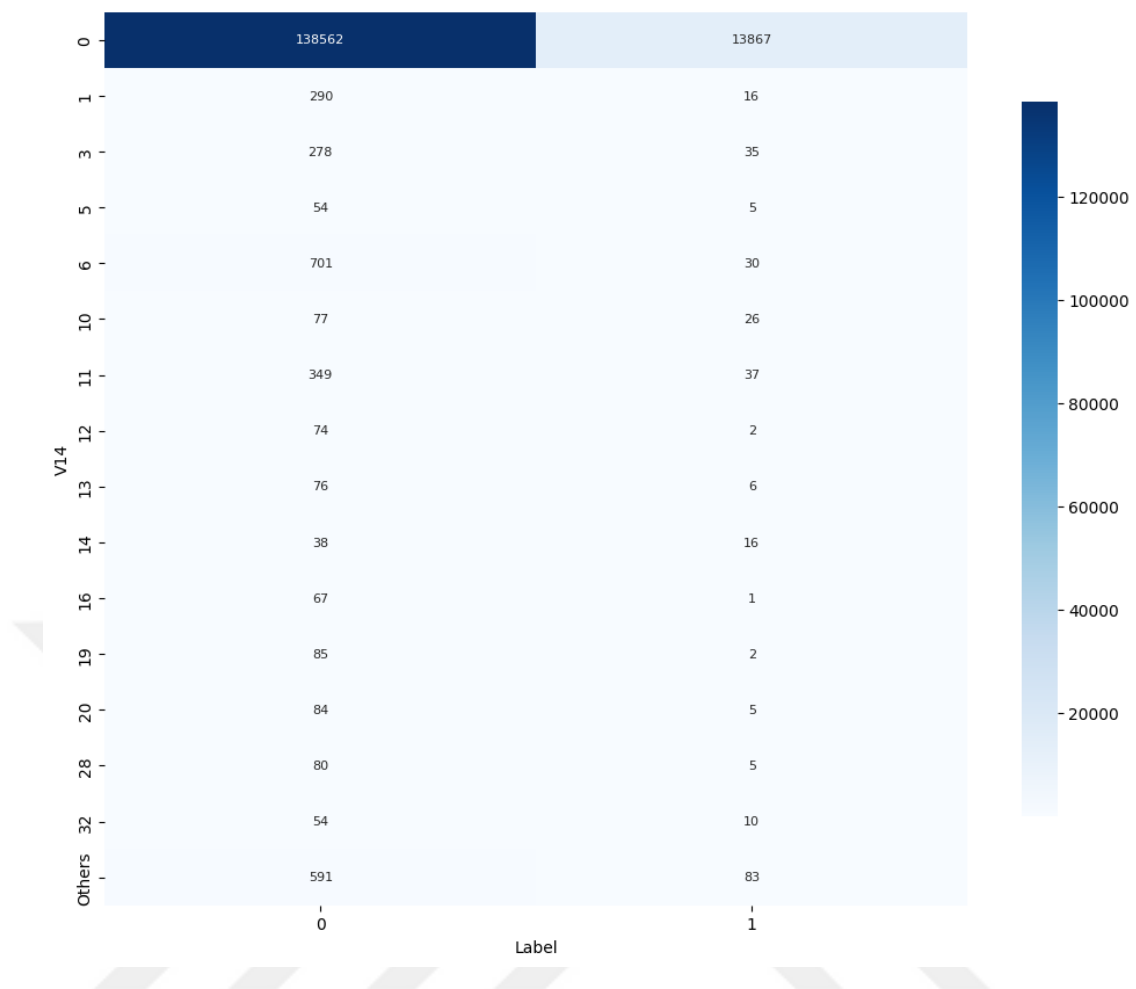


Figure A7. Crosstab for V14 and label

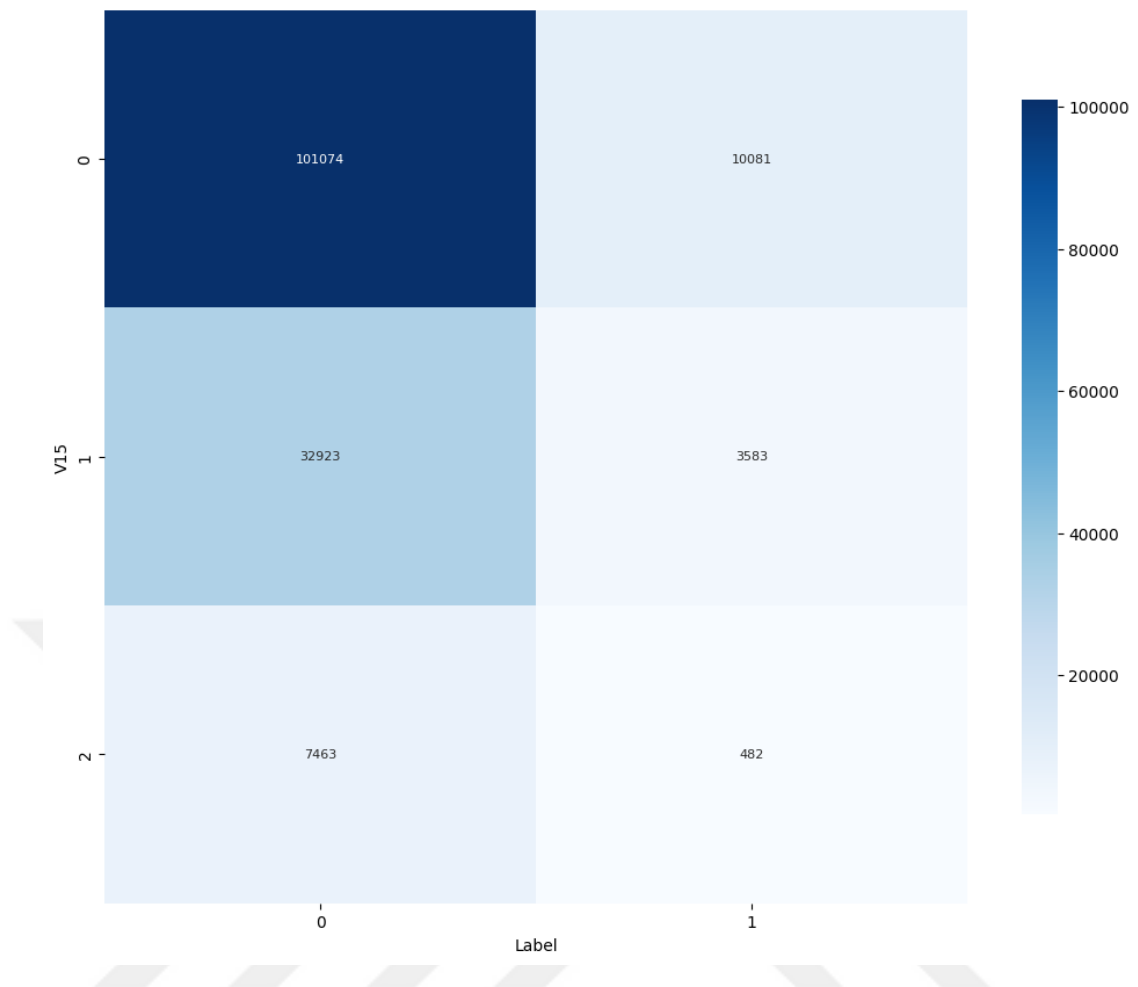


Figure A8. Crosstab for V15 and label

APPENDIX B

EDA RESULTS FOR PRIVATE DATA WITH ENGINEERED FEATURES

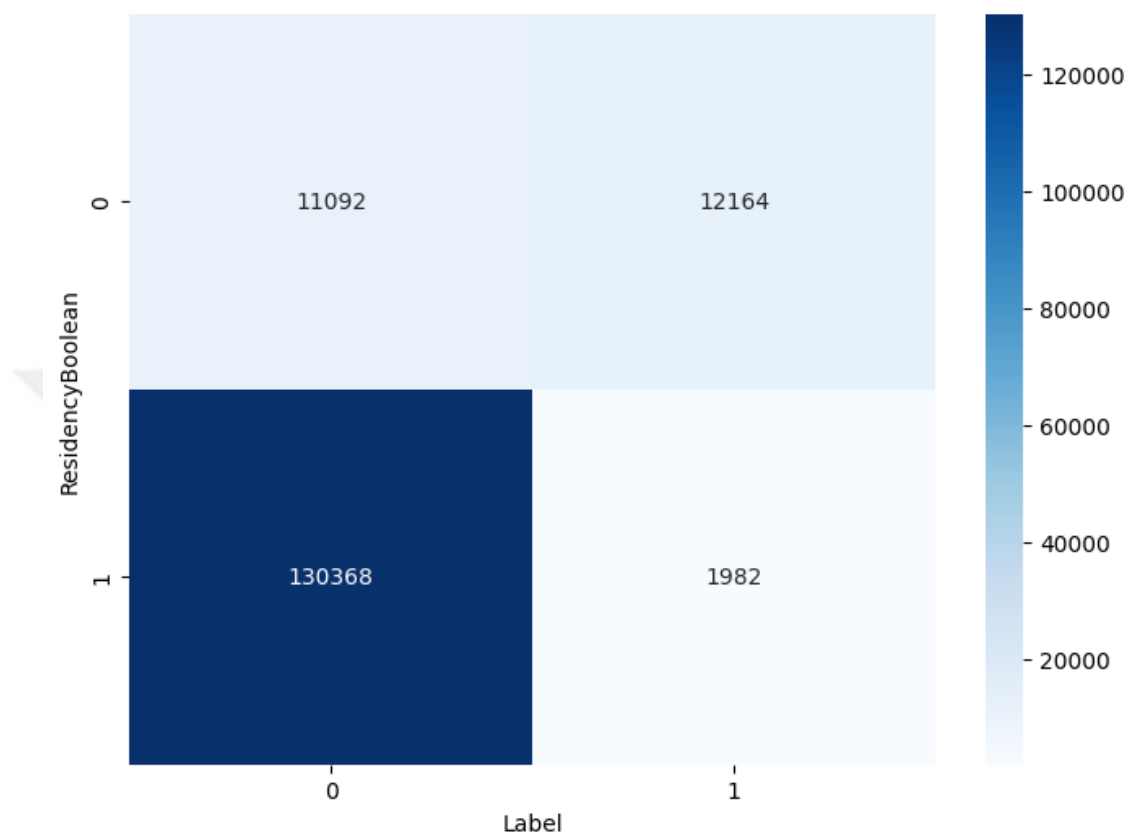


Figure B1. Crosstab for ResidencyBoolean and label

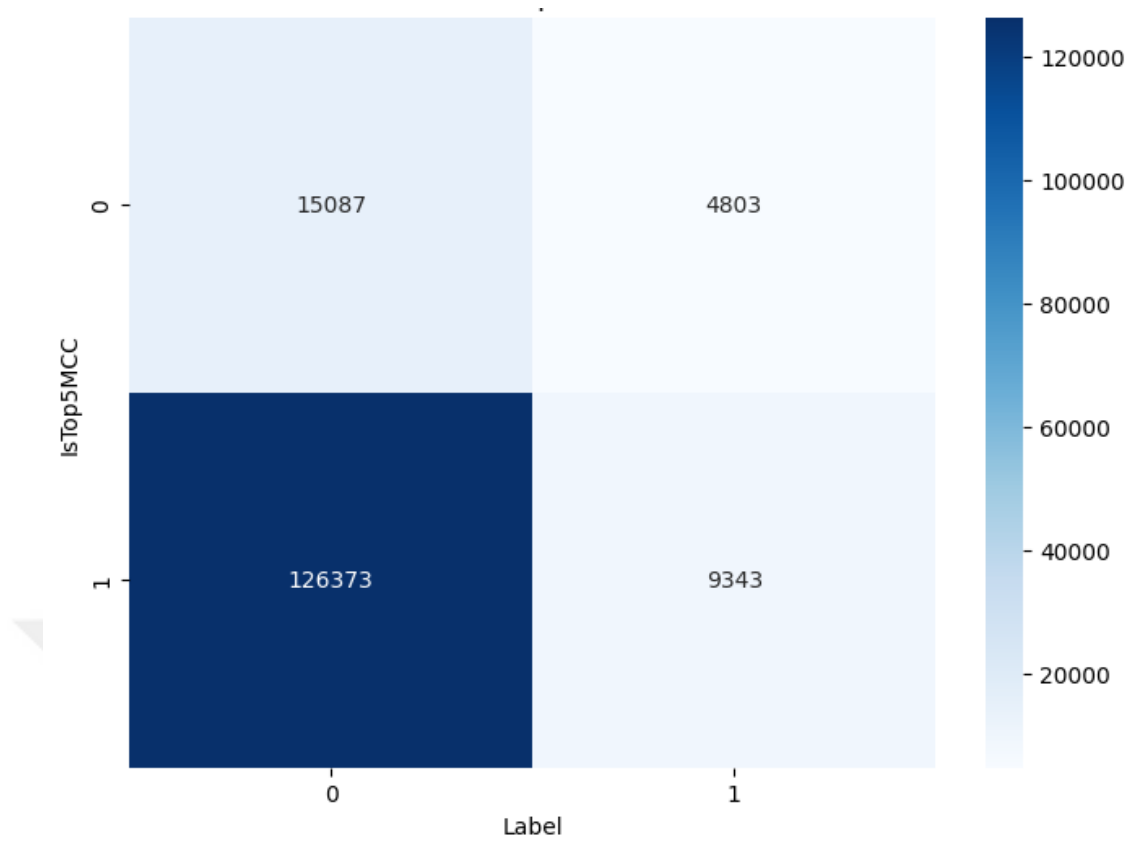


Figure B2. Crosstab for IsTop5MCC and label

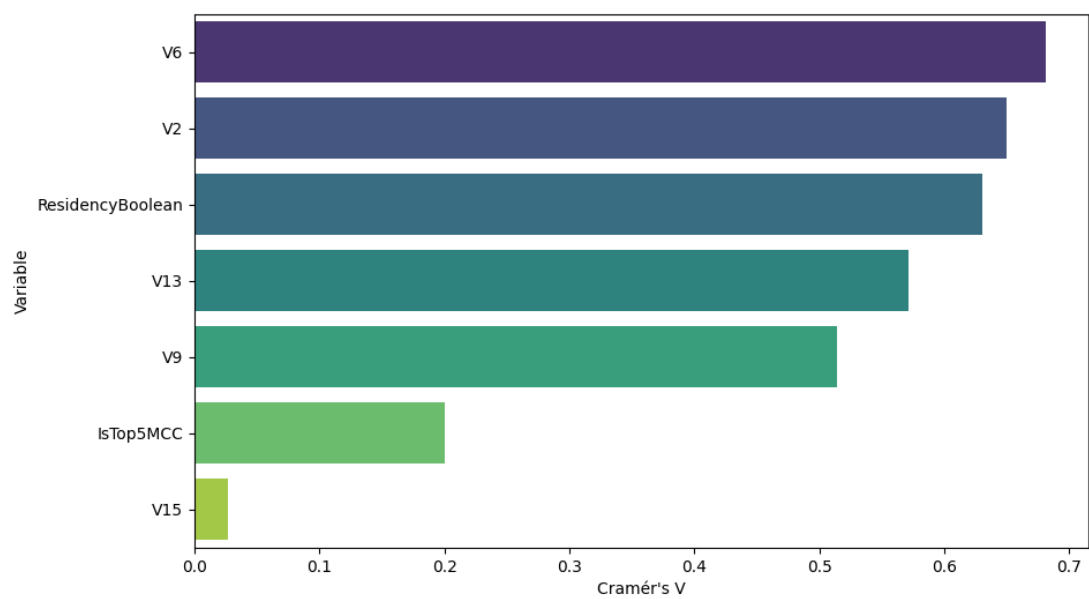


Figure B3. Cramér's V for categorical variables

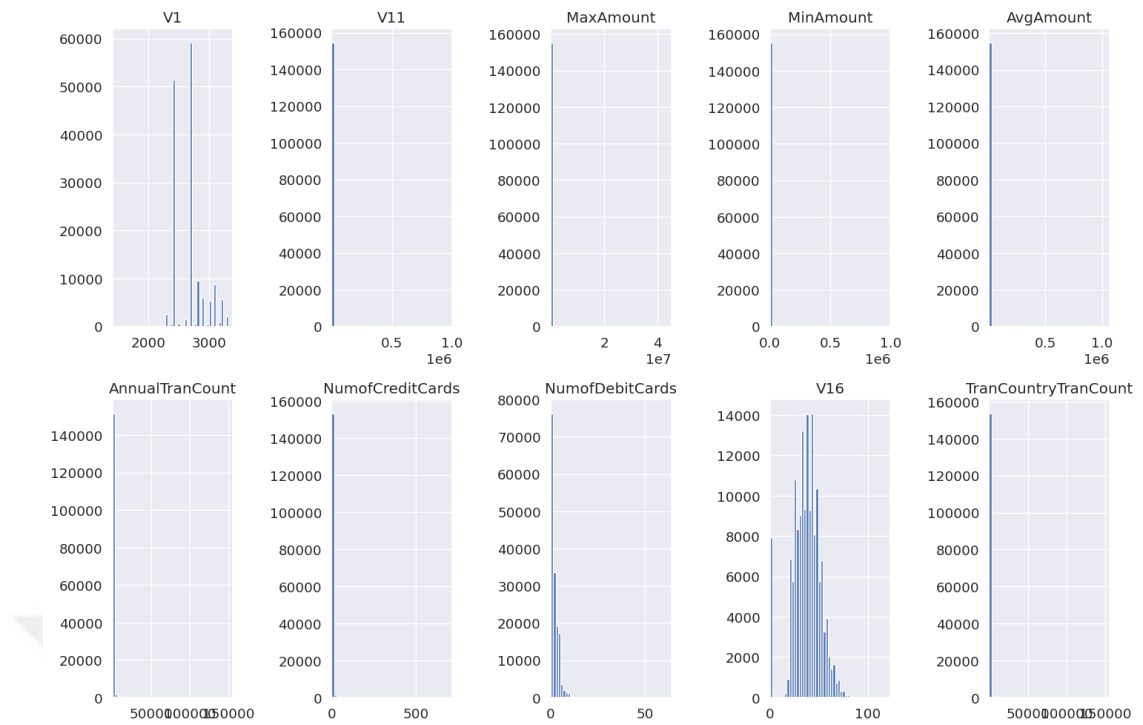


Figure B4. Histograms of continuous variables

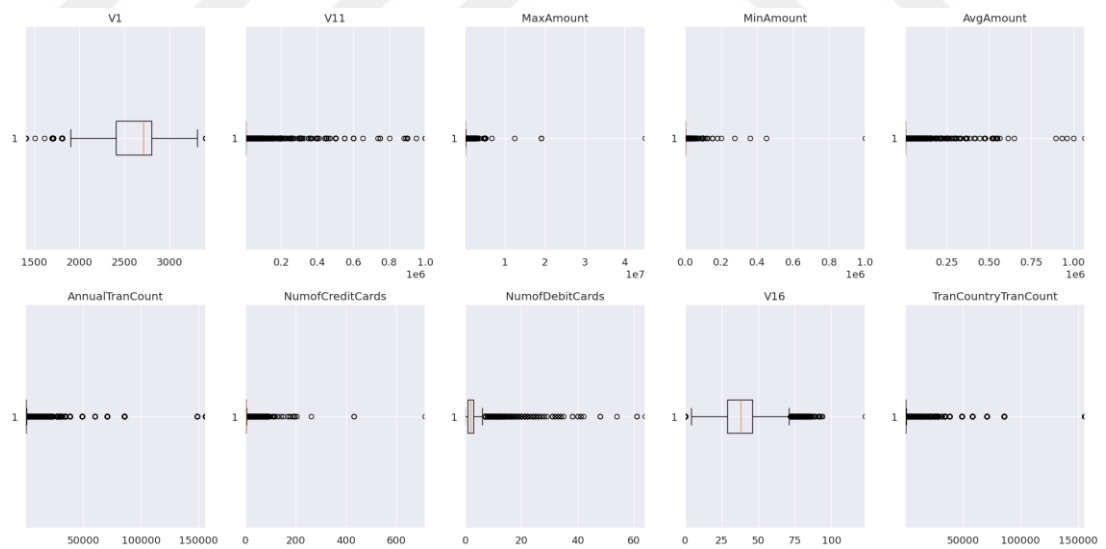


Figure B5. Boxplots of continuous variables

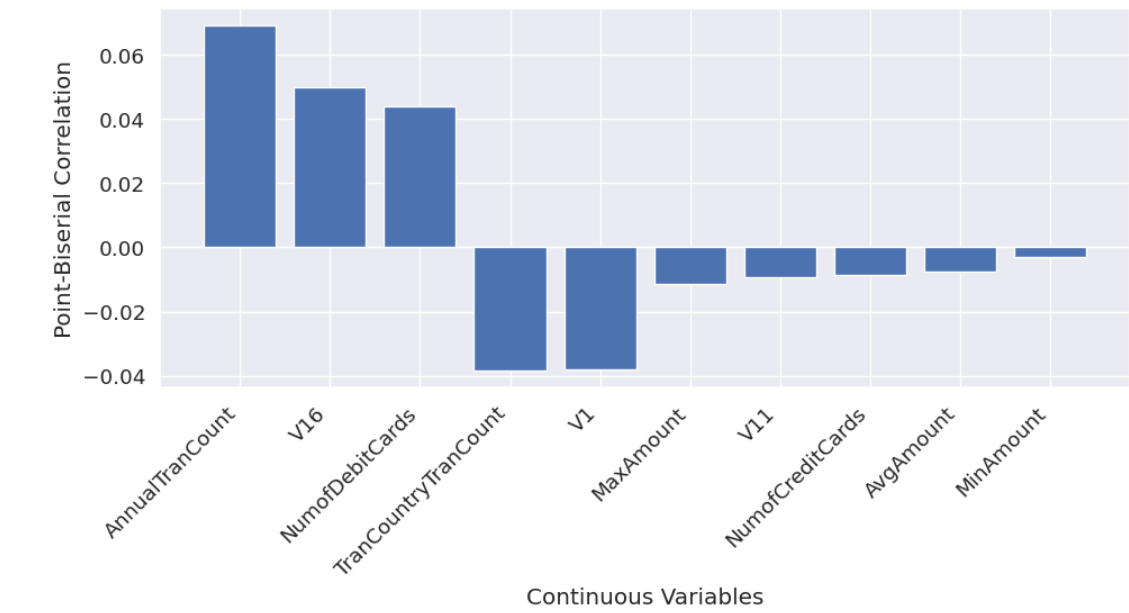


Figure B6. Point-biserial correlations of continuous variables with label

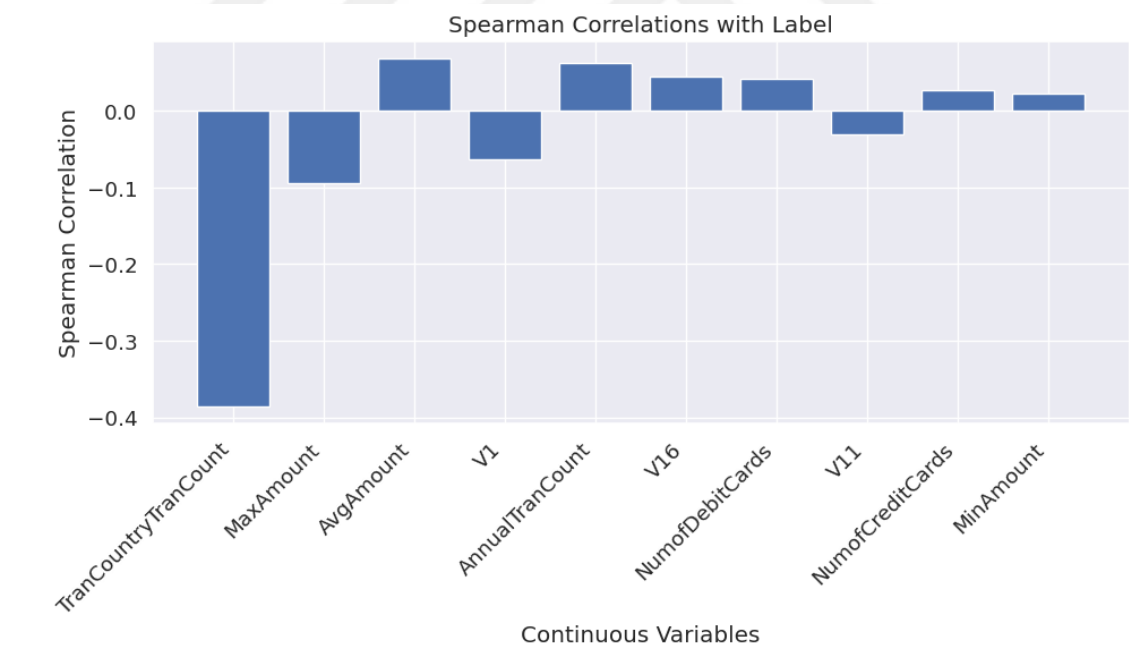


Figure B7. Spearman correlations of continuous variables with label

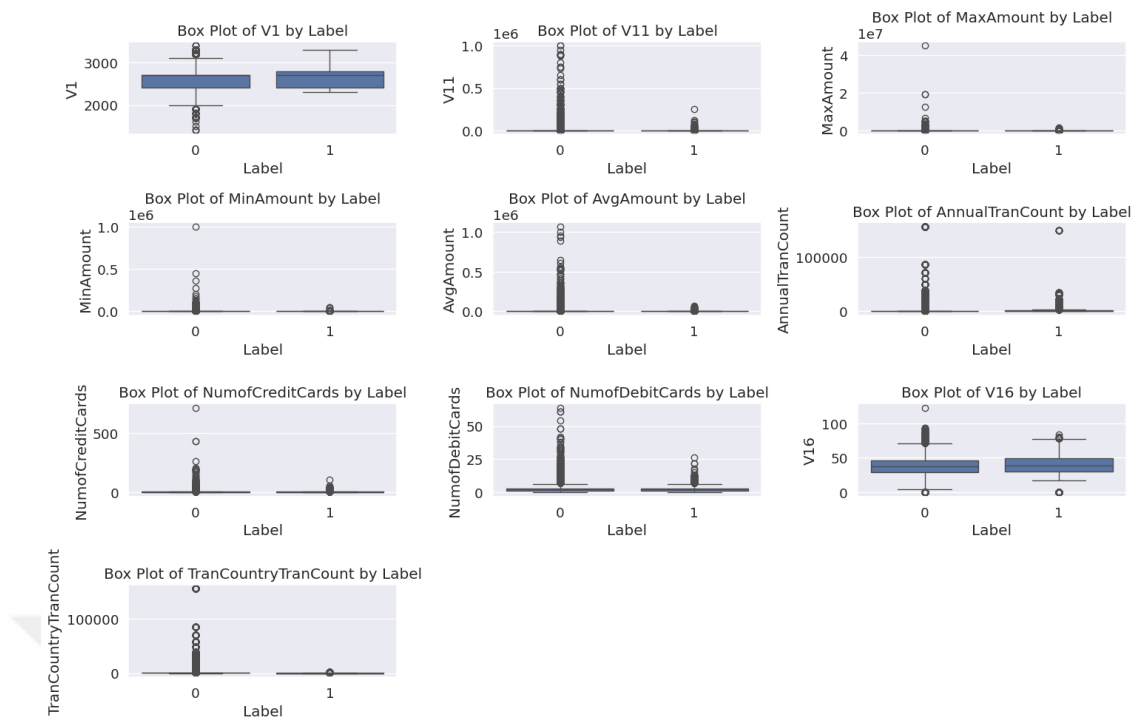


Figure B8. Boxplots of continuous variables by label

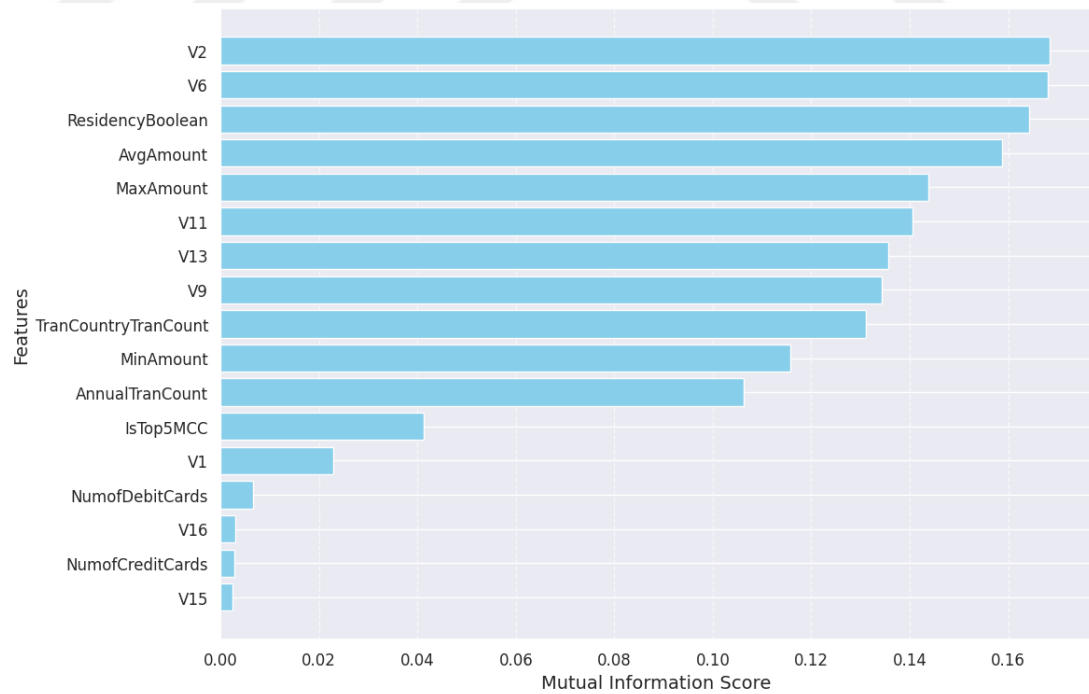


Figure B9. Mutual information scores for features

REFERENCES

- Abraham, A. (2017). Artificial neural networks. In *Handbook of Measuring System Design*. Wiley. <https://doi.org/10.1002/0471497398.mm421>
- Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138-52160. doi: 10.1109/ACCESS.2018.2870052.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902
- Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., & Ahmed, M. (2022). Credit card fraud detection using state-of-the-art ML and DL algorithms. *IEEE Access*, 10, 39700-39715.
- Alharbi, A., Alshammari, M., Okon, O., Alabrah, A., Rauf, H., Alyami, H., & Meraj, T. (2022). A novel text2img mechanism of credit card fraud detection: A DL approach. *Electronics*, 11(5), 756.
- Altman, E. R. (2019). Synthesizing Credit Card Transactions. arXiv:1910.03033.
- Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10), 1340-1347.
- Aras, M. (2023). A multi-modal profiling fraud-detection system for capturing suspicious airline ticket activities. *Applied Sciences*, 13(24), 13121.
- Arrieta, A., Díaz-Rodríguez, N., Ser, J., Bennetot, A., Tabik, S., Barbado, A., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82-115.
- Association of Certified Fraud Examiners. (2023). COSO fraud risk management guide: Second edition executive summary. <https://www.acfe.com/-/media/files/acfe/pdfs/fraud-risk-tools/coso-fraud-risk-management-guide-second-edition-executive-summary.pdf>.
- Association of Certified Fraud Examiners. (2020). Report to the Nations: Global Study on Occupational Fraud and Abuse.
- Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, October). Credit card fraud detection using ML techniques: A comparative analysis. In *2017 International Conference on Computing Networking and Informatics (ICCNI)* (pp. 1-9). IEEE.
- Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51, 134-142.

- Bauder, R., Khoshgoftaar, T. M., & Seliya, N. (2017). A survey on the state of healthcare upcoding fraud analysis and detection. *Health Services and Outcomes Research Methodology*, 17, 31-55.
- Bello, O. A., & Olufemi, K. (2024). Artificial intelligence in fraud prevention: Exploring techniques and applications challenges and opportunities. *Computer Science & IT Research Journal*, 5(6), 1505-1520
- Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. *Proceedings of the 30th International Conference on ML (ICML-13)*, 115-123.
- Bhardwaj, S., & Gupta, S. (2022). Effects of feature selection with ML algorithms in detection of credit card fraud. *International Journal of Engineering Research in Computer Science and Engineering*, 9(7), 46-51.
- Breiman, L. (2001). Random forests. *ML*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Branco, B., Abreu, P., Gomes, A., Almeida, M., Ascensão, J., & Bizarro, P. (2020). Interleaved sequence RNNs for fraud detection. *Proceedings*, 3101-3109.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.
- Chen, J., & Lai, K. (2021). Deep convolution neural network model for credit-card fraud detection and alert. *Journal of Artificial Intelligence and Capsule Networks*, 3(2), 101-112.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>.
- Chen, Z., Hohmann, L., Banjara, B., Zhao, Y., Diggs, K., & Westrick, S. (2020). Recommendations to protect patients and health care practices from medicare and medicaid fraud. *Journal of the American Pharmacists Association*, 60(6), e60-e65.
- Chen, Z., Soliman, W., & Nazir, A. (2021). Variational autoencoders and wasserstein generative adversarial networks for improving the anti-money laundering process. *Ieee Access*, 9, 83762-83785.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>

- Choi, D., & Lee, K. (2018). An artificial intelligence approach to financial fraud detection under IoT environment: A survey and implementation. *Security and Communication Networks*, 5483472.
- Cirqueira, D., Helfert, M., & Bezbradica, M. (2021). Towards design principles for user-centric explainable ai in fraud detection., 21-40.
- Clement, T., Kemmerzell, N., Abdelaal, M., & Amberg, M. (2023). Xair: a systematic metareview of explainable ai (xai) aligned to the software development process. *ML and Knowledge Extraction*, 5(1), 78-108.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *ML*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2), 215–242.
- Coyle, D. and Weller, A. (2020). “explaining” ML reveals policy challenges. *Science*, 368(6498), 1433-1434.
- Cross, C. and Gillett, R. (2020). Exploiting trust for financial gain: an overview of business email compromise (bec) fraud. *Journal of Financial Crime*, 27(3), 871-884.
- Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3784–3797.
- Daliri, S. (2020). Using harmony search algorithm in neural networks to improve fraud detection in banking system. *Computational Intelligence and Neuroscience*, 2020(1), 6503459.
- Das, A. (2020). Opportunities and challenges in explainable artificial intelligence (XAI): A survey.
- Dash, R., Rautray, R., & Dash, R. (2021). A Legendre Neural Network for Credit Card Fraud Detection. *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2* (pp. 411-418). Springer Singapore.
- Debener, J., Heinke, V., & Kriebel, J. (2023). Detecting insurance fraud using supervised and unsupervised ML. *Journal of Risk & Insurance*, 90(3), 743-768.
- Dornadula, S. & Geetha, K. (2019). A survey on credit card fraud detection techniques. *International Journal of Computer Applications*, 178(7), 1-6. doi:10.5120/ijca2019918838.
- Doshi, S., Desai, K., & Shukla, D. (2023). Comparative study of fraudulent activities and various fraud detection techniques. *International Journal for Research in Applied Science and Engineering Technology*, 11(8), 1140-1148.

- Dwivedi, A., Kumar, A., & Gupta, S. (2023). "A Comprehensive Review on Explainable Artificial Intelligence: Challenges and Future Directions." *Artificial Intelligence Review*, 56(1), 1-35. doi:10.1007/s10462-022-10001-9.
- Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A neural network ensemble with feature engineering for improved credit card fraud detection. *IEEE Access*, 10, 16400-16407.
- Farbmacher, H., Löw, L., & Spindler, M. (2022). An explainable attention network for fraud detection in claims management. *Journal of Econometrics*, 228(2), 244-258. <https://doi.org/10.1016/j.jeconom.2020.05.021>.
- Geneva, P. (2020). A comprehensive overview of recurrent neural networks and their applications in natural language processing. *Journal of Artificial Intelligence Research*, 68, 1-25. doi:10.1613/jair.1.12345.
- Gerlach, T., Schmid, M., & Schmid, U. (2022). Evaluating Explainable AI: A Comprehensive Framework for Assessing the Effectiveness of XAI Methods. *Journal of Artificial Intelligence Research*, 73, 1-30. doi:10.1613/jair.1.12345.
- Ghrib, T., Khaldi, Y., Pandey, P. S., & Abusal, Y. A. (2024). Advanced Fraud Detection In Card-Based Financial Systems Using A Bidirectional LSTM-GRU Ensemble Model. *Applied Computer Science*, 20(3), 51-66.
- Gómez, J. A., Arévalo, J., Paredes, R., & Nin, J. (2018). End-to-end neural network architecture for fraud scoring in card payments. *Pattern Recognition Letters*, 105, 175-181.
- Gyamfi, N. K., & Abdulai, J. D. (2018, November). Bank fraud detection using support vector machine. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 37-41). IEEE.
- Górski, L., Ramakrishna, S., & Nowosielski, J. (2020). Towards grad-cam based explainability in a legal text processing pipeline.
- Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73, 220-239.
- Hamal, S., & Şenvar, Ö. (2021). Comparing performances and effectiveness of ML classifiers in detecting financial accounting fraud for Turkish SMEs. *International Journal of Computational Intelligence Systems*, 14(1), 769.
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263–1284.
- He, K., Zhang, X., Ren, S., & Sun, J. (2020). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. doi:10.1109/CVPR.2016.90

- Herland, M., Khoshgoftaar, T. M., & Bauder (2018). R. A. Big data fraud detection using multiple medicare data sources. *J Big Data*, 5 (1), 29.
- Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507. <https://doi.org/10.1126/science.1127647>.
- Hinton, G. E., & Zemel, R. (1993). Autoencoders, minimum description length and Helmholtz free energy. *Advances in neural information processing systems*, 6.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558. <https://doi.org/10.1073/pnas.79.8.2554>.
- Hu, Y., Wang, L., & Chen, Z. (2023). BERT4ETH: A pre-trained transformer model for fraud detection in Ethereum transactions. *Journal of Blockchain Research*, 5(3), 123-139. doi:10.5678/jbr.2023.12345
- Huang, S. Y., Lin, C. C., Chiu, A. A., & Yen, D. C. (2017). Fraud detection using fraud triangle risk factors. *Information Systems Frontiers*, 19, 1343-1356.
- Huang, X., Khetan, A., Cvitkovic, M., & Karnin, Z. (2020). TabTransformer: Tabular data modeling using contextual embeddings. *Advances in Neural Information Processing Systems*, 33, 6472–6482.
- Husnaningtyas, N. (2023). Financial fraud detection and ML algorithm (unsupervised learning): systematic literature review. *Jurnal Riset Akuntansi Dan Bisnis Airlangga*, 8(2), 1521-1542.
- Hussein, A., Ali, A., & Alshahrani, M. "Credit Card Fraud Detection Using Fuzzy Rough Nearest Neighbor and Sequential Minimal Optimization with Logistic Regression." *International Journal of Interactive Mobile Technologies (iJIM)*, 2021.
- Hájek, P., Abedin, M., & Sivarajah, U. (2022). Fraud detection in mobile payment systems using an xgboost-based framework. *Information Systems Frontiers*, 25(5), 1985-2003.
- Ileberi, E., Sun, Y., & Wang, Z. (2022). A ML based credit card fraud detection using the ga algorithm for feature selection. *Journal of Big Data*, 9(1).
- Iranzad, R., & Liu, X. (2024). A review of random forest-based feature selection methods for data science education and applications. *International Journal of Data Science and Analytics*, 1-15.

- Jan, C. L. (2018). An effective financial statements fraud detection model for the sustainable development of financial markets: *Evidence from Taiwan. Sustainability*, 10(2), 513.
- Jan, C. (2021). Detection of financial statement fraud using DL for sustainable development of capital markets under information asymmetry. *Sustainability*, 13(17), 9879.
- Kahn, C., & Roberds, W. (2005). Credit and identity theft. *SSRN Electronic Journal*.
- Kalid, R., Ameer, A., & Alshahrani, M. (2020). A Multiple Classifiers System for Anomaly Detection in Credit Card Data With Unbalanced and Overlapped Classes. *IEEE Access*.
- Kalid, S., Ng, K., Tong, G., & Khor, K. (2020). A multiple classifiers system for anomaly detection in credit card data with unbalanced and overlapped classes. *IEEE Access*, 8, 28210-28221.
- Kanika, K., & Singla, R. (2022). ML Models for Credit Card Fraud Detection: A Comprehensive Review. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 13(5), 124-130. DOI: 10.14569/IJACSA.2022.0130518
- Kasasbeh, B., & Ahmad, H. (2022). Multilayer perceptron artificial neural networks-based model for credit card fraud detection. *Indonesian Journal of Electrical Engineering and Computer Science*, 26(1), 362.
- Kolli, C., & Tatavarthi, U. (2020). Fraud detection in bank transaction with wrapper model and Harris water optimization-based deep recurrent neural network. *Kybernetes*, 50(6), 1731-1750.
- KPMG & Nets. (2020). Fighting fraud with a model of models [White paper]. Retrieved from <https://assets.kpmg.com/content/dam/kpmg/dk/pdf/dk-2020/04/Nets-KPMG-Fighting-Fraud-with-a-Model-of-Models-whitepaper-2020.pdf>
- Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. *Proceedings of the Fourteenth International Conference on ML*, 179–186.
- Kuttiyappan, D., & Rajasekar, V. (2024, March). AI-Enhanced Fraud Detection: Novel Approaches and Performance Analysis. In *Proceedings of the 1st International Conference on Artificial Intelligence, Communication, IoT, Data Engineering and Security, IACIDS 2023*, 23-25 November 2023, Lavasa, Pune, India.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- Leevy, J. (2023). Investigating the effectiveness of one-class and binary classification for fraud detection. *Journal of Big Data*, 10(1).

- Li, K., Yang, T., Zhou, M., Meng, J., Wang, S., Wu, Y., & Tong, Y. (2024, August). SEFraud: Graph-based Self-Explainable Fraud Detection via Interpretative Mask Learning. *In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 5329-5338).
- Lopez-Rojas, E., Axelsson, S., & Baca, D. (2018). Analysis of fraud controls using the paysim financial simulator. *International Journal of Simulation and Process Modelling*, 13(4), 377.
- Lozano-Murcia, C., Romero, F., Serrano-Guerrero, J., & Varela, J. (2023). A comparison between explainable ML methods for classification and regression problems in the actuarial context. *Mathematics*, 11(14), 3088.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30. Retrieved from <https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf>
- Mahesh, K., Afrouz, S., & Areecal, A. (2022). Detection of fraudulent credit card transactions: a comparative analysis of data sampling and classification techniques. *Journal of Physics Conference Series*, 2161(1), 012072.
- Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M., & Zeineddine, H. (2019). An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, 7, 93010-93022.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>
- McKay, M. (2022). AI transparency in a real-world context: what we can learn from past examples of algorithmic and statistical decision-making.
- Mienye, I. D., & Jere, N. (2024). DL for credit card fraud detection: A review of algorithms, challenges, and solutions. *IEEE Access*.
- Mill, E., Garn, W., Ryman-Tubb, N., & Turner, C. (2023). Opportunities in real time fraud detection: an explainable artificial intelligence (XAI) research agenda. *International Journal of Advanced Computer Science and Applications*, 14(5). <https://doi.org/10.14569/ijacsa.2023.01405121>.
- Muter, M., & Molood, M. (2020). A Combined SVM-MPSO-MMPSO Method for Credit Card Fraud Detection. *Journal of Computer Science and Technology*.
- Mytnyk, B., Tkachyk, O., Shakhovska, N., Fedushko, S., & Syerov, Y. (2023). Application of artificial intelligence for fraudulent banking operations recognition. *Big Data and Cognitive Computing*, 7(2), 93.
- Nabrawi, E., & Alanazi, A. (2023). Fraud detection in healthcare insurance claims using ML. *Risks*, 11(9), 160.

- Niu, X., Wang, L., & Yang, X. (2019). A comparison study of credit card fraud detection: supervised versus unsupervised.
- Noviandy, T. R., Idroes, G. M., Maulana, A., Hardi, I., Ringga, E. S., & Idroes, R. (2023). Credit card fraud detection for contemporary financial management using XGBoost-driven ML and data augmentation techniques. *Indatu Journal of Management and Accounting*, 1(1), 29-35.
- Peng, H. and Wang, J. (2022). Unbalanced data processing and ML in credit card fraud detection.
- Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. arXiv:1009.6119.
- Psychoula, I., Gutmann, A., Mainali, P., Lee, S., Dunphy, P., & Petitcolas, F. (2021). Explainable ML for fraud detection. *Computer*, 54(10), 49-59. <https://doi.org/10.1109/mc.2021.3081249>
- Pumsirirat, A. and Liu, Y. (2018). Credit card fraud detection using DL based on auto-encoder and restricted boltzmann machine. *International Journal of Advanced Computer Science and Applications*, 9(1).
- PwC. (2022). Global Economic Crime and Fraud Survey 2022.
- Qin, Z., Liu, Y., He, Q., & Ao, X. (2022). Explainable graph-based fraud detection via neural meta-graph search., 4414-4418. <https://doi.org/10.1145/3511808.3557598>
- Raghavan, P., & El Gayar, N. (2019). Fraud Detection using ML and DL. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (pp. 334-339). IEEE.
- Raj, S. and Portia, A. A. (2011). Analysis on credit card fraud detection methods. *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, 152-156.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. <https://doi.org/10.1145/2939672.2939778>
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>
- Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., & Beling, P. (2018). DL detecting fraud in credit card transactions., 129-134.
- Rudin, C. (2019). Stop explaining black-box ML models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206-215.

- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
<https://doi.org/10.1038/323533a0>
- S.M. Lundberg, S.-I. Lee (2017), A unified approach to interpreting model predictions, in: *Advances in Neural Information Processing Systems*, pp. 4765–4774.
- Sadhukhan, P., Pakrashi, A., & Namee, B. M. (2021). Random walk-steered majority undersampling. <https://doi.org/10.48550/arxiv.2109.12423>
- Schuster, M., & Paliwal, K. K. (1997). Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, 45(11), 2673-2681.
<https://doi.org/10.1109/78.650093>
- Sharma, M., Raj, B., Ramamurthy, B., & Bhaskar, R. (2022). Credit card fraud detection using DL based on auto-encoder. *Itm Web of Conferences*, 50, 01001.
- Sharma, M., Watane, A., Cavuoto, K., Parikh, R., & Sridhar, J. (2023). Fraud claims filed involving practicing ophthalmologists from 1985 through 2020. *Clinical Ophthalmology*, 17, 341-350.
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. *Proceedings of the 34th International Conference on ML*, 70, 3145–3153.
- Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. McClelland, & PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Foundations*, 1, 194–281. MIT Press.
- Thennakoon, A., Bhagyan, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using ML. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE.
- Tiwari, R. (2023). Explainable ai (xai) and its applications in building trust and understanding in ai decision making. *Interantional Journal of Scientific Research in Engineering and Management*, 07(01).
- Tran, L., Tran, T. H., Tran, L., & Mai, A. (2019). Solve fraud detection problem by using graph-based learning methods. *Journal of Engineering and Science Research*, 3(4), 28-31.
- Vaswani, A., Shardlow, M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 30.
- Veigas, K., Regulagadda, D., & Kokatnoor, S. (2021). Optimized stacking ensemble (ose) for credit card fraud detection using synthetic minority oversampling model. *Indian Journal of Science and Technology*, 14(32), 2607-2615.

- Verma, P. and Tyagi, P. (2022). Credit card fraud detection using selective class sampling and random forest classifier. *Ecs Transactions*, 107(1), 4885-4894.
- Weber, P., Carl, K. V., & Hinz, O. (2024). Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Management Review Quarterly*, 74(2), 867-907.
- Wibowo, P. and Fatichah, C. (2021). An in-depth performance analysis of the oversampling techniques for high-class imbalanced dataset. *Register Jurnal Ilmiah Teknologi Sistem Informasi*, 7(1), 63.
- Wolf, T., Chaumond, J., Debut, L., Sanh, V., & Delangue, C. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38-45.
- Ye, H., Lin, X., & Gan, Y. (2019). Detecting financial statement fraud using random forest with smote. *Iop Conference Series Materials Science and Engineering*, 612(5), 052051.
- Yu, X., Zhang, Y., & Li, J. (2024). A comparative study of transformer-based models and traditional ML algorithms for credit card fraud detection. *Journal of Financial Technology*, 12(1), 45-67. doi:10.1234/jft.2024.56789
- Zhang, Z., Zhou, X., Zhang, X., Wang, L., & Wang, P. (2018). A model based on convolutional neural network for online transaction fraud detection. *Security and Communication Networks*, 2018, 1-9.
- Zhu, X., Ao, X., Qin, Z., Chang, Y., Liu, Y., He, Q., ... & Li, J. (2021). Intelligent financial fraud detection practices in post-pandemic era. *The Innovation*, 2(4), 100176.
- Zhu, Y. (2022). Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection.