https://doi.org/
https://orcid.org/0000-0003-2622-0864

VILNIUS UNIVERSITY

Dalia Breskuvienė

Feature Conversion for a Better Imbalanced Data Classification: A Financial Fraud Detection Case

DOCTORAL DISSERTATION

Technological Sciences,
Informatics Engineering (T 007)

VILNIUS 2025

The dissertation was prepared between 2021 and 2025 at Vilnius University.

Academic supervisor – Prof. Habil. Dr. Gintautas Dzemyda (Vilnius University, Natural Sciences, Informatics Engineering T 007).

This doctoral dissertation will be defended at a public meeting of the Dissertation Defence Panel:

Chairman – Prof. Dr. ...Name Surname... (Vilnius University, Natural Sciences, Informatics – N 009).
Members:
Prof. ...Name Surname... (..., Natural Sciences, Informatics – N 009).
Prof. Dr. ...Name Surname... (..., Natural Sciences, Informatics – N 009).
Prof. Habil. Dr. ...Name Surname... (..., Natural Sciences, Informatics – N 009).
Dr. ...Name Surname... (Tallinn University of Technology, Estonia, Natural Sciences, Chemistry – N 003).

The dissertation shall be defended at a public meeting of the Dissertation Defense Panel at ..... a.m. on ...th ..... 20.. in room 203 of the Institute of Data Science and Digital Technologies of Vilnius University.
Address: Akademijos g. 4, LT-04812, Vilnius, Lithuania
Tel. +370 5 210 9300; e-mail: info@mii.vu.lt

The text of this dissertation can be accessed at the Library of Vilnius University, as well on the website of Vilnius University: *https://www.vu.lt/lt/naujienos/ivykiu-kalendorius*.

https://doi.org/
https://orcid.org/0000-0003-2622-0864

VILNIAUS UNIVERSITETAS

Dalia Breskuvienė

Požymių konversija siekiant gerinti nesubalansuotų duomenų klasifikavimą: finansinio sukčiavimo atvejis

DAKTARO DISERTACIJA

Technologijos mokslai,
Informatikos Inžinerija (T 007)

VILNIUS 2025

Disertacija rengta 2021–2025 metais Vilniaus universitete.

Mokslinis (-ė) vadovas (-ė):
prof. dr. Gintautas Dzemyda (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 009).

Gynimo taryba:
Pirmininkas (-ė) – prof. dr. ...Vardas Pavardė... (Vilniaus universitetas, gamtos mokslai, informatika – N 009).
Nariai:
prof. ...Vardas Pavardė... (..., gamtos mokslai, informatika – N 009).
prof. dr. ...Vardas Pavardė... (..., gamtos mokslai, informatika – N 009).
prof. habil. dr. ...Vardas Pavardė... (..., gamtos mokslai, informatika - N 009).
dr. ...Vardas Pavardė... (Talino technikos universitetas, Estija, gamtos mokslai, chemija – N 003).

Disertacija ginama viešame Gynimo tarybos posėdyje 20.. m. .... d. ... val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje. Adresas: Akademijos g. 4, LT-04812, Vilnius, Lietuva, tel. +370 5 210 9300; el. paštas: info@mii.vu.lt.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje ir Vilniaus universiteto interneto svetainėje adresu: *https://www.vu.lt/lt/naujienos/ivykiu-kalendorius*.

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to my supervisor, Professor Habl. Dr. Gintautas Dzemyda for his support and encouragement over the years. He has taught me how to write academic articles and often says, "Write every article as if it is your last one." Professor Dzemyda is not only my supervisor but also my mentor. His guidance has shaped my scientific thinking, strengthened my research skills, and inspired me to pursue academic goals. His patience, insightful feedback, and constant belief in my abilities have been invaluable throughout my doctoral journey. I am deeply grateful for the opportunity to learn from him.

I want to thank Nicholas Clark for planting the seed in my mind to pursue my dissertation and instilling in me the belief that I can do this.

I also express my sincere gratitude to the reviewers, Prof. Dr. Virginijus Marcinkevičius and Dr. Linas Petkevičius, for their valuable time, insightful comments, and constructive feedback on my dissertation. Their suggestions have contributed significantly to improving the quality and clarity of this work. I want to thank all the professors and scientists at the Institute of Data Science and Digital Technologies for their high-quality lectures, ad-hoc assistance with dissertation improvements, and engaging discussions during lunch.

I also want to express my gratitude to Audrius Šapola for connecting me with many incredible people who shared their knowledge with me. One of these people is Nigel Krishna Iyer. Nigel opened my eyes to a broader understanding of fraud detection. He taught me that in this field, not everything is black and white; there are also gray areas.

I want to express my special thanks to my husband, Kęstas, for his unwavering support throughout this journey. He has taken care of our daughters and served as the first reviewer of my articles and dissertation. Without him, I would not have accomplished this.

Finally, I am immensely grateful to my parents for their constant encouragement and belief in me. Their steadfast support and confidence have been crucial in tackling challenging tasks and pushing through difficult times. Their faith in my abilities has been a constant source of motivation, laying the way for my achievements. Thank you for inspiring me to strive for excellence.

ABSTRACT

Fraud detection remains a critical challenge in the financial sector, requiring innovative approaches to detect and prevent losses caused by increasingly sophisticated fraudulent activities. This dissertation addresses several aspects of improving fraud detection: using clustering as a preprocessing step, encoding strategies for imbalanced data, and feature selection importance. First, we propose a clustering-based classification method to increase the recall in credit card fraud detection. By optimizing feature selection and the number of clusters to form more homogeneous subsets for training and strategically undersampling each cluster, we improved the recall from 0.845 to 0.867, statistically significantly reducing the number of misclassified fraudulent cases by 13.9%. Second, we investigate the impact of categorical feature encoding on model performance. Through experiments on datasets with less than 1% fraud prevalence and the application of six encoding methods, we find that target-based encoding, especially James-Stein and Weight of Evidence (WOE), significantly outperform alternatives like CatBoost encoding in imbalanced settings. Our results highlight the importance of careful preprocessing, especially when dealing with high-cardinality categorical features and the curse of dimensionality. Finally, we introduce FID-SOM (Feature Selection for Imbalanced Data Using SOM), a novel feature selection method tailored for highly imbalanced datasets. Leveraging self-organizing maps, the FID-SOM identifies and ranks features on the basis of their contribution to best-matching units' weight vector attribute variability, enabling effective dimensionality reduction without losing critical information. The experimental results show that FID-SOM can match or surpass traditional feature selection techniques in fraud detection tasks. Our findings offer a comprehensive framework to enhance machine learning-based fraud detection in real-world, large-scale, and highly imbalanced datasets.

AI	Artificial Intelligence. 19, 33
AUC-ROC	Area Under the Receiver Operating Characteristic (ROC) Curve. 43, 44, 46, 102, 110–112, 116, 163
BMU	Best Matching Unit. 15, 95, 97–99, 106, 107, 161–163
CART	Classification And Regression Trees. 65, 83, 91, 94
CCFD	Credit Card Fraud Detection. 19, 28, 34, 40, 44, 56
CCPA	California Consumer Privacy Act. 52, 152
CNP	Card-Not-Present. 32, 34
CP	Card-Present. 33, 34
DT	Decision Tree. 33, 40, 41
EFB	Exclusive Feature Bundling. 85
F1 score	*F1* score is a harmonic mean of $precision$ and $recall$. 14, 42, 43, 64, 71, 72, 91, 94, 102, 113, 114
FCE	Fuzzy Combination Entropy. 49
FID-SOM	Feature Selection for Imbalanced Data Using SOM. 95, 99, 100, 104, 107, 108, 110–113, 115, 116, 118, 162–164, 166
FP	False Positive. 19, 42–44, 59
G-Mean	Geometric Mean. 43, 102, 107, 113
GBDT	Gradient Boosting Decision Trees. 85
GDPR	General Data Protection Regulation. 52, 152
i.i.d.	independent and identically distributed. 44, 45
IG	Information Gain. 83, 85
k-NN	k-Nearest Neighbors. 40, 41
LR	Logistic Regression. 33, 40, 41, 51, 102
MCC	Matthews Correlation Coefficient. 43, 102, 113
ML	Machine Learning. 19, 20, 22, 33, 34, 36, 37, 40, 44, 48, 50, 52, 54, 55, 63, 70, 73, 76, 78, 81, 86, 91, 93–95, 102, 103, 107
NB	Naive Bayes. 40, 41
RF	Random Forest. 40, 41, 84, 91, 94, 102, 113, 114

RFE	Recursive Feature Elimination. 101, 108, 110–113
RUS	Random Undersampling. 37, 64
SG	Skip-Gram. 47
SMOTE	Synthetic Minority Over-sampling Technique. 37, 55, 151
SOM	Self-Organizing Map. 15, 22, 95, 96, 98–100, 105, 106, 113, 115, 160–164
SVM	Support Vector Machine. 33, 36, 40, 41
TNR	True Negative Rate. Also called Specificity. 42, 43
TP	True Positive. 42
TPR	True Positive Rate. Also called Sensitivity or Recall. 42–44
UniChi2	univariate feature selection method based on χ^2-test. 101, 108, 110–113
UniF	Univariate feature selection method based on F-test. 100, 108, 110–113
UniMI	Univariate feature selection method based on Mutual Information. 101, 108–113
WOE	Weight Of Evidence. 15, 80, 81, 88–94, 159
XGBImp	XGB Importance method. 101, 108, 110–113

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Fraud represents a widespread issue with clear economic consequences, including reduced financial health and stability of private companies [7], decreased quality of public services [6], reduced disposable income for individuals [30], and diminishment of essential resources for charitable organizations [51]. It has a significant impact on the quality of life in all sectors and countries. Various fraud scenarios differ in terms of the magnitude of loss and the complexity of techniques involved.

This dissertation concentrates on credit card fraud detection from the perspective of financial institutions. Credit card fraud is a form of identity theft, in which another person's credit card information is illegally obtained and used for purchases or cash withdrawals from the account without the owner's knowledge or permission [83]. There are numerous methods of stealing credit card data. One relatively simple yet effective technique is sending links to individuals in order to encourage them to make purchases or transfers. Figure 1 illustrates a fraud case that corresponds to a typical data phishing scenario aimed at stealing banking login credentials and performing unauthorized transactions. Such a process starts with an SMS message impersonating a trusted institution, such as a government agency or a bank, urging the recipient to urgently review the provided notice via a link. By clicking the link, the user is redirected to a fraudulent website that visually resembles a genuine login page, with the goal of harvesting their login credentials. Once the details are entered, they fall into the hands of fraudsters, granting access to the real bank account. This enables criminals to initiate unauthorized transactions, resulting in financial losses. This example demonstrates how easily credit card data can be stolen and misused, often without the victim even realizing it. Once fraudsters obtain such data, financial institutions face the challenge of detecting and stopping them within milliseconds.

This phishing example (see Figure 1) illustrates how easily stolen credit card data can be obtained and misused. The moment a fraudster initiates a transaction using stolen credentials, the clock starts ticking: financial institutions must detect and assess the risk in real-time. Automated fraud detection systems have only milliseconds to analyze the transaction and decide whether to approve, flag, or block it before it is authorized and the funds are released.

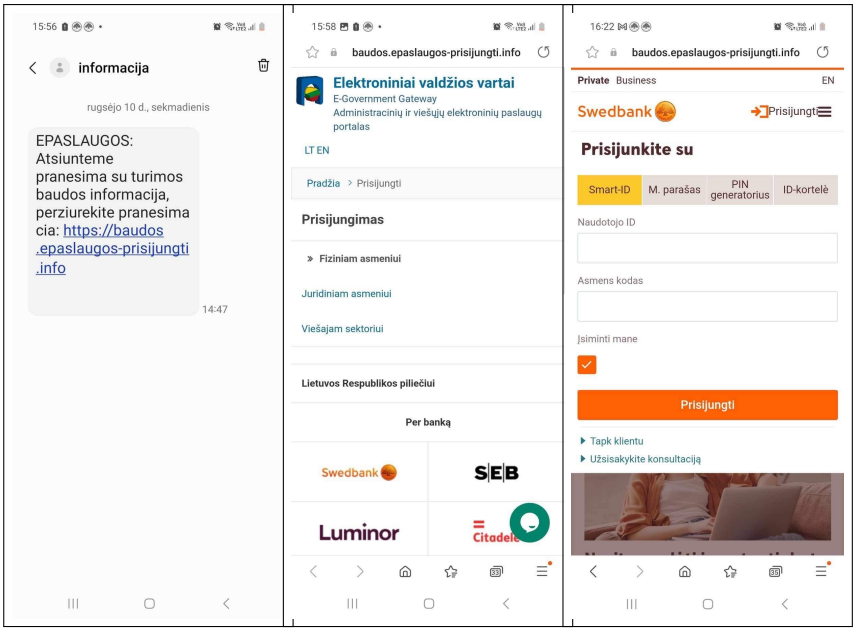


Figure 1: Three steps to lose your money: SMS to phishing link, phishing webpage, money transfer

This dissertation does not focus on the vast market of trading stolen credit card data or techniques shared on the dark web. This area is too broad to explore in detail and isn't directly relevant to preventing this type of fraud; what is crucial for detecting and preventing stolen credit card fraud is the fact that the perpetrator possesses these details and intends to use them. The method by which they acquired this information is intriguing but will not aid in stopping them once they are on your platform. Indeed, the stolen credit card market has evolved to a point where "fraud end-to-end services" are accessible (for example, fraudsters can buy a package including a stolen credit card, a matching virtual private network, and a corresponding aged/spoofed email) [117].

In this dissertation, we characterize credit card fraud as the stage where a criminal already possesses the sensitive card details (e.g., card number, expiration date, cardholder name, CVV) and attempts to use them to make purchases or subscribe to services for financial gain. The goal is to intervene at the transaction stage - after the data has been compromised, but before the transaction is authorized.

In order to tackle the ever-growing sophistication of fraudsters, financial institutions need to leverage the potential of Artificial Intelligence (AI) and Machine Learning (ML) algorithms. By utilizing advanced AI and ML capabilities, these institutions can proactively detect and prevent fraudulent activities, safeguarding their customers and reputation. ML algorithms emerge in the fraud prevention area [68], [85]. However, there are several challenges when applying ML to credit card fraud data, with the most significant obstacle being data imbalance. For instance, most transactions are legitimate when working with transactional data, while less than 1% of transactions are fraudulent.

Moreover, Credit Card Fraud Detection (CCFD) using ML suffers from concept drift [89], high-dimensional categorical features [12], a lack of public databases for research purposes, and even some performance measures can be misleading when used for imbalanced data [35]. The fraud detection algorithm operates under tight time constraints to distinguish between legitimate and fraudulent transactions. Every financial institution establishes its own specific protocols for the duration for which a transaction may be halted for evaluation. In most cases, this timeframe is measured in milliseconds [18].

Research Area

This dissertation primarily focuses on the optimization of imbalanced data in classification tasks such as financial fraud detection, by applying categorical data encoding and feature selection methods in order to enhance detection efficiency and interpretability. Furthermore, it investigates the potential of feature conversion into a new numerical space as an intermediate step within the feature selection process.

Research Problem

Detecting financial fraud is particularly challenging due to one of the most critical issues in this domain - severe class imbalance - along with the need to identify the most informative features and efficiently encode categorical variables. Traditional feature selection and categorical encoding methods do not ensure accurate and reliable machine learning based fraud detection, often leading to an excessive number of FP or missed fraud cases.

Actuality

Financial fraud is a deliberate act of deception committed with the intent to secure unlawful gain or to cause losses to another party. This definition corresponds with the legal framework provided in Article 3(2) of Directive (EU) 2017/1371 on the protection of the Union's financial interests through criminal law [43]. Detecting financial fraud is one of the most critical challenges in the modern financial sector, as even a single undetected case may cause substantial financial losses and damage an institution's reputation.

Although various ML based methods for imbalanced data classification exist, they often fail to effectively detect fraud when the imbalance is severe, i.e., when fraudulent cases represent only a very small fraction of the total transaction flow. Therefore, there remains a strong need to develop more advanced methods capable of operating effectively under conditions of pronounced data imbalance. The method proposed in this dissertation is highly relevant, as it improves fraud detection efficiency and can be broadly applied beyond the financial sector, including in medical data analysis and cybersecurity. The research findings have direct practical significance, as the method has been validated on real financial transaction data and can be integrated into existing risk management systems of financial institutions.

Research Object

The object of this research is the process of financial fraud detection, with particular emphasis on feature selection methods and feature conversion into another numerical space. The study analyzes how various feature selection methods can help to select informative features more effectively in order to improve the efficiency of imbalanced data classification using machine learning algorithms for credit card fraud detection.

Research Aim and Objectives

The aim of this research is to develop a method for rationally reducing the existing feature set in order to improve the classification efficiency of imbalanced data, with the ultimate goal of enhancing credit card fraud detection.

The objectives of the dissertation are as follows:

- to propose a consistent machine learning strategy based on clustering data into meaningful groups, balancing these groups, and applying individually tailored classification methods in order to improve fraud case recognition.
- to evaluate the impact of target-based and target-agnostic encoding methods on imbalanced data classification.
- to assess the applicability of the Self-Organizing Map (SOM) as an intermediate step in the feature selection process, both for clustering transactions and for feature conversion into a new numerical space, with the aim of achieving optimal feature selection in the studied dataset.
- to develop an imbalanced data–oriented method to refine and reduce the existing feature set in order to achieve better classification results.
- to propose an experimental framework that ensures model adaptability to evolving fraud behavior patterns by applying time-based transaction data splits (training on earlier data than testing) and conduct experiments with publicly available annotated datasets to demonstrate the effectiveness of the proposed approach.

Research Methods

This dissertation applies a systematic methodological approach to the study of credit card fraud detection. The following methods were employed:

- A comprehensive literature review was conducted on credit card fraud detection methods, covering fraud typologies, challenges of imbalanced data, approaches to fraud detection, and other related topics.
- Various categorical data encoding techniques were compared. Their impact on fraud detection performance was evaluated in order to identify optimal preprocessing strategies.

- Theoretical insights were combined with empirical experimentation to assess the effectiveness of different feature selection methods for fraud detection.
- The proposed feature conversion approach was tested using benchmark datasets. During the experiments, model training, validation, and performance evaluation were carried out with multiple evaluation metrics to determine the effectiveness of the method in classifying transactions.

Scientific Novelty

This dissertation introduces a novel ML-based method for improving the feature space of financial transactions, specifically designed for highly imbalanced datasets. The key scientific contribution is the development of a new feature conversion method that enables more efficient data preparation and improves fraud detection accuracy. The method employs nonlinear transformations to better separate fraud-related characteristics, thereby enhancing the performance of classification models. Comprehensive experiments conducted on both real-world and synthetic financial data demonstrate that the proposed approach significantly improves classification metrics, particularly in scenarios where fraud cases are extremely rare. Furthermore, the method extends the applicability of SOM, providing new opportunities to address challenges related to the classification of imbalanced data.

Practical Significance

Imbalanced data remains a significant challenge in machine learning, particularly in real-world applications where the minority class is of primary interest. Many domains, such as fraud detection, customer churn prediction, medical diagnosis, and anomaly detection in cybersecurity, exhibit this property. In financial fraud detection, the ability to accurately identify rare fraudulent transactions while minimizing false positives is crucial for reducing financial losses and operational costs. This research contributes to the field by developing and evaluating feature selection method that enhance classification performance under severe class imbalance. The proposed approach aims to improve fraud

detection accuracy, optimize computational efficiency, and provide practical insights for financial institutions deploying fraud detection models in high-risk environments.

The research emphasized the importance of effective data preparation and selection of measurements to avoid obtaining overfitted or misleading outcomes in research related to credit card fraud detection; was constructed replicating real world scenarious based on work experience in financial institutions and literature review; proposed a feature selection method which aims to outperform other feature selection methods, increase classifiers accuracy and contributes to reducing prediction time which is critical in credit card detection process.

Statements to be Defended

Fraud detection in financial transactions remains a challenging task, primarily due to the severe class imbalance and the dynamically evolving nature of fraudulent behavior. The main defended statements of this dissertation are as follows:

- The proposed cluster-specific classification strategy, which incorporates individual cluster balancing and classification, significantly improves fraud detection Recall and reduces the number of legitimate transactions incorrectly classified as fraudulent.
- Target-based encoding methods, compared to traditional target-agnostic encoding schemes, demonstrate superior performance in classification tasks characterized by class imbalance and high categorical feature cardinality, as they are capable of capturing meaningful statistical relationships often lost by traditional approaches.
- The proposed FID-SOM (Feature Selection for Imbalanced Data Using SOM) method, based on competitive learning principles, employs SOM as a feature conversion mechanism by using the variance of BMU weight vectors to assess feature importance, thereby enhancing the selection process and significantly improving fraud detection.
- To ensure that models remain adaptable to evolving fraud behavior patterns, it is necessary to apply time-based transaction dataset

splits (training on earlier data than testing). This approach allows for evaluating model adaptability in predicting future events in financial fraud detection.

Approbation and Publications of the Research

The results of this research have been presented and validated through publications in two peer-reviewed international journals indexed in Q1 and Q3, as well as in a peer-reviewed book chapter and conference abstracts. The findings have also been disseminated within the scientific community through participation in both international and national conferences. The following list provides an overview of the research contributions, including journal articles, book chapters, and conference presentations. Among these, particular attention should be drawn to the article "Enhancing Credit Card Fraud Detection: Highly Imbalanced Data Case", published in the Q1-quartile journal Journal of Big Data, which was nominated for the Vilnius University Rector's Award.

Publications

Articles published in international journals that are included in Clarivate's Web of Science database:

1. Breskuvienė, Dalia; Dzemyda, Gintautas. Enhancing credit card fraud detection: highly imbalanced data case // Journal of Big Data. ISSN 2196-1115. 2024, vol. 11, iss. 1, sp. [182]. DOI: 10.1186/s40537-024-01059-5.

2. Breskuvienė, Dalia; Dzemyda, Gintautas. Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions // International Journal of Computers Communications & Control. Oradea: Agora University. ISSN 1841-9836. eISSN 1841-9844. 2023, vol. 18, iss. 3, art. no. 5433, p. [1-17]. DOI: 10.15837/ijccc.2023.3.5433.

Chapter in the peer-reviewed scientific book:

1. Breskuvienė, Dalia; Dzemyda, Gintautas. Imbalanced data classification approach based on clustered training set // Data Science in Applications / Editors: Dzemyda, G., Bernatavičienė, J., Kacprzyk, J. Cham: Springer, 2023. ISBN 9783031244520. eISBN

9783031244537. p. 43-62. (Studies in Computational Intelligence, ISSN 1860-949X, eISSN 1860-9503; vol. 1084). DOI: 10.1007/978-3-031-24453-7_3.

Conference abstracts:

- Breskuvienė, Dalia; Dzemyda, Gintautas. Highly imbalanced data case: pattern-guided feature selection to detect financial fraud // DAMSS: 15th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 28-30, 2024. Vilnius: Vilniaus universiteto leidykla, 2024. eISBN 9786090711125. p. 12-13. DOI: 10.15388/DAMSS.15.2024.
- Breskuvienė, Dalia; Dzemyda, Gintautas. What is a concept drift, and does it affect machine learning performance? // DAMSS: 14th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 30 - December 2, 2023. Vilnius: Vilniaus universiteto leidykla, 2023. eISBN 9786090709856. p. 14. DOI: 10.15388/DAMSS.14.2023.
- Breskuvienė, Dalia; Dzemyda, Gintautas. Autoencoder for fraudulent transactions data feature engineering // DAMSS: 13th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, December 1–3, 2022. Vilnius: Vilniaus universiteto leidykla, 2022. ISBN 9786090707944. eISBN 9786090707951. p. 11. DOI: 10.15388/DAMSS.13.2022.
- Breskuvienė, Dalia; Dzemyda, Gintautas. Clustering-based optimization in fraud detection classifier training // EURO 2022: [32nd European Conference on Operational Research (EURO XXXII)], Espoo, Finland, July 3-6, 2022: Abstract Book. Espoo: Aalto University, 2022. ISBN 9789519525419. p. 152. Available online: https://www.euro-online.org/conf/admin/tmp/program-euro32.pdf.

International conferences

1. Breskuvienė, Dalia. Adapt or fall behind: A deep dive into machine learning techniques for detection of the evolving fraud in the financial realm // 13th Annual Counter Fraud, Cybercrime and

Forensic Accounting Conference, June 12–13, 2024, Portsmouth, UK.

2. Breskuvienė, Dalia. Clustering-based optimization in fraud detection classifier training // EURO 2022: [32nd European Conference on Operational Research (EURO XXXII)], Espoo, Finland, July 3-6, 2022.

National conferences

1. Breskuvienė, Dalia. Highly imbalanced data case: pattern-guided feature selection to detect financial fraud // DAMSS: 15th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 28-30, 2024.

2. Breskuvienė, Dalia. What is a concept drift, and does it affect machine learning performance? // DAMSS: 14th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 30 - December 2, 2023.

3. Breskuvienė, Dalia. Autoencoder for fraudulent transactions data feature engineering // DAMSS: 13th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, December 1–3, 2022.

Python scripts used in the researched

- Breskuvienė, Dalia. Enhancing credit card fraud detection: highly imbalanced data case. National Open Access Research Data Archive (MIDAS). DOI: 10.18279/MIDAS.261094.

- Breskuvienė, Dalia. Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions. National Open Access Research Data Archive (MIDAS). DOI: 10.18279/MIDAS.261073.

- Breskuvienė, Dalia. Cluster-specific training strategy for credit card fraud detection. National Open Access Research Data Archive (MIDAS). DOI: 10.18279/MIDAS.259980.

Outline of the Thesis

This doctoral thesis consists of an introduction, four chapters, conclusions, and a summary in the Lithuanian language. The introduction section provides an introduction to the research and an overview of the dissertation. Chapter 1 presents a literature review that describes differences in financial fraud types, addresses imbalanced data handling techniques, reviews feature encoding, and feature selection methods. Chapter 2 introduces an imbalanced data classification approach based on a clustered training set, including transactions undersampling. Chapter 3 explores categorical feature encoding techniques to improve classifier performance when dealing with fraudulent transactions. Chapter 4 presents the novel FID-SOM method for feature selection in fraud detection and discusses experimental results. The general conclusions are summarized in the last chapter.

153 bibliographic references are included at the end of the thesis. The dissertation consists of 168 pages, 30 figures, and 20 tables.

1. LITERATURE REVIEW OF FINANCIAL FRAUD DETECTION IN THE PRESENCE OF HIGH CLASS IMBALANCE

This chapter presents a review of literature relevant to the detection of financial fraud, with an emphasis on credit card fraud. The review is structured to provide a clear and logical look at key topics in the field. Section 1.1 begins by defining financial fraud and categorizing its various forms to establish a broader context. Section 1.3 examines the challenges posed by imbalanced datasets, defining characteristics of datasets in the detection of credit card fraud. This section discusses techniques for balancing the dataset as well as the classifiers commonly utilized to address imbalanced dataset challenges. In addition, this section critically evaluates common issues in research on fraud detection, including limitations in model testing, early handling of data imbalance, and selection of appropriate performance metrics. Section 1.4 investigates feature encoding methods, focusing on handling high-cardinality categorical features in imbalanced datasets. Section 1.5 explores feature selection strategies tailored for imbalanced data, highlighting their significance in improving model performance. Section 1.7 examines the availability of data on research related to CCFD. It covers the constraints faced and outlines an alternative approach to enable the continuation of the research. Finally, Section 1.8 consolidates the literature review findings, summarizing key insights and identifying research gaps addressed in this dissertation.

1.1. Definition of Financial Fraud

Financial fraud is anything that involves lying to gain a benefit. The costs of identifying fraud are transferred to society by increasing customer inconvenience and higher prices for goods and services [81]. Although this dissertation focuses specifically on credit card fraud, for a broader understanding of the landscape, various types of financial fraud are described (see Figure 1.1), each characterized by distinct methods and impacts. This categorization was developed by the author to summarize and structure common fraud types discussed in the literature.

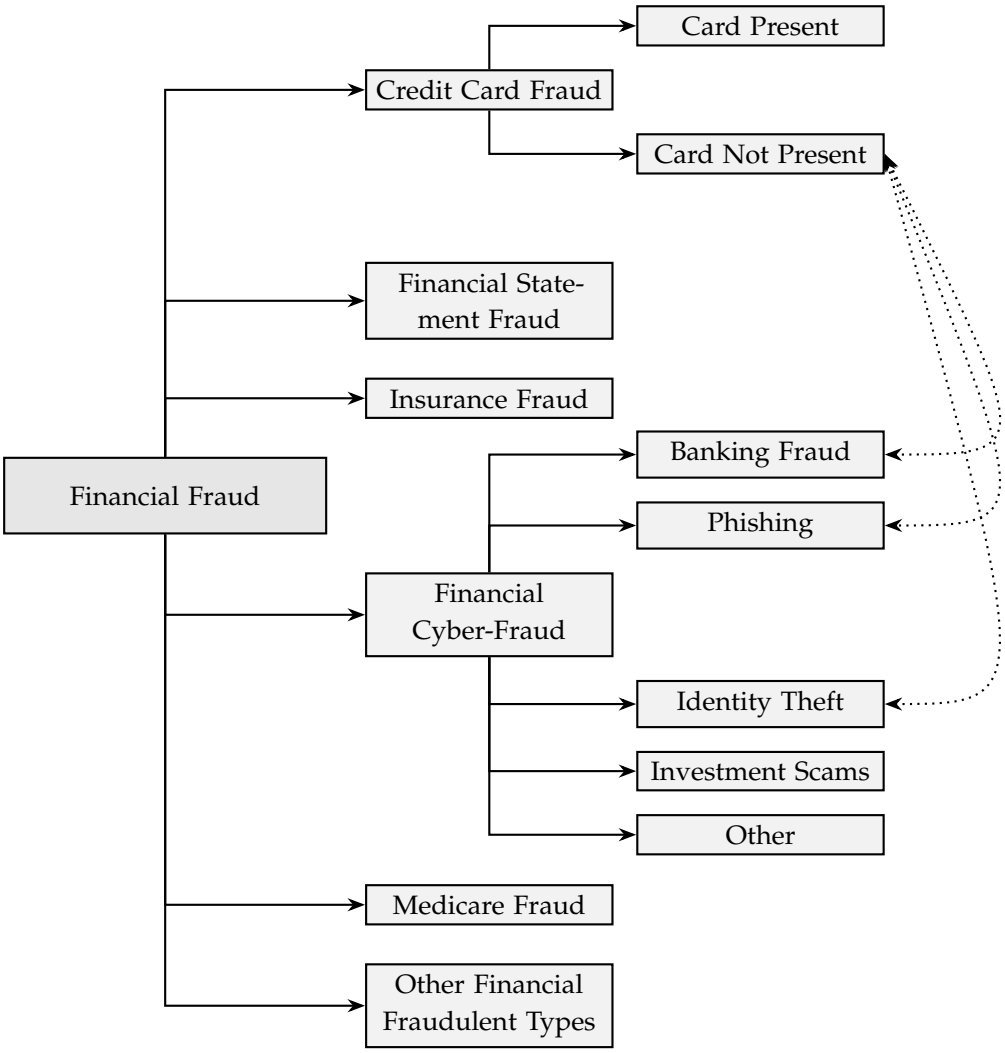


Figure 1.1: Hierarchical representation of various types of financial fraud.

Financial Statement Fraud

Financial statement fraud involves intentionally misrepresenting, deleting, or manipulating the financial information companies present to the public, their investors, and other stakeholders [125]. This type of fraud is typically committed by companies looking to appear more profitable or financially stable than they are, influencing the decisions of investors, lenders, and other financial partners. These manipulations can significantly distort a company's financial health, misleading stakeholders about its financial performance, leading to misinformed decisions, and potentially causing legal consequences.

Insurance Fraud

In general terms, insurance fraud refers to any act committed with the intent to obtain payment from an insurer fraudulently. This definition is widely used by law enforcement and organizations such as the Federal Bureau of Investigation (FBI) or National Insurance Crime Bureau (NICB). This type of fraud can vary in severity from exaggerated claims to outright false statements. It can be committed by applicants, policyholders, or professionals who provide services to claimants. Insurance fraud can be divided into following types [137]:

- Hard Fraud, which occurs when someone deliberately plans or invents a loss, such as a collision, auto theft, or fire, that allows them to file a fraudulent claim against an insurance policy. It is premeditated and calculated.
- Soft Fraud, also known as opportunistic fraud, involves policyholders exaggerating otherwise legitimate claims. For example, they may add more damage than actually occurred in an accident or claim for items that were not actually stolen during a burglary.

Insurance fraud can impact various types of insurance, such as health, auto, life, and property insurance. Each type can involve different specific schemes. The consequences of insurance fraud are not limited to the insurance industry. They ripple through the economy, affecting the general public by increasing the cost of premiums. It is often pursued legally and can result in severe penalties including fines and imprisonment.

Medicare Fraud

Medicare fraud [62] is a type of healthcare fraud that involves the submission of false or misleading information to the Medicare program in order to receive unauthorized benefits or payments. It can be committed by various parties, including healthcare providers, patients, and medical equipment suppliers. Common examples of Medicare fraud include billing for services that were never provided, such as charging for medical procedures, tests, or equipment that the patient never received; upcoding, which involves billing for more expensive services than were actually provided in order to receive higher reimbursement; and unbundling, which involves submitting multiple bills for procedures that should be grouped and billed as a single complex service. Fraud can also include falsifying diagnoses to justify unnecessary tests or treatments, bribery and giving or accepting bribes for patient referrals or product use, identity theft, using another person's Medicare number to submit false claims, and double-billing, where the same service or procedure is billed multiple times.

Medicare fraud not only leads to significant financial losses for the government but also can harm patients by exposing them to unnecessary treatments and compromising the quality of care.

Financial Cyber-Fraud

Financial cyber-fraud is an illegal activity involving networks and Internet capabilities to trick individuals or organizations into gaining economic advantage. This type of fraud encompasses a wide array of activities, including:

- Banking Fraud [13] includes hacking into bank accounts or payment card systems to transfer funds illegally or make unauthorized withdrawals or transactions.
- Phishing [20], a common form of financial cyber-fraud, operates by scammers sending deceptive emails or messages that mimic reputable organizations. Their aim is to illicitly acquire sensitive data, including passwords, credit card numbers, and banking information.

- Identity Theft [65] occurs when cybercriminals use stolen personal information to access financial accounts, open new accounts, or make unauthorized transactions in someone else's name.
- Investment Scams [76] occur when cyber fraudsters promote nonexistent opportunities to invest in stocks, cryptocurrencies, or other financial markets, promising high returns.
- There are Other types of Financial Cyber-Fraud such as Malware and Ransomware Attacks [129]. Malicious software is used to infiltrate and damage systems, steal personal data, or lock out users from their systems until a ransom is paid.

The ramifications of financial cyber-fraud are far-reaching, impacting not only individual victims but also large organizations. These crimes can result in substantial financial losses and reputational damage. To combat this, it is crucial to implement robust cybersecurity measures, conduct public awareness campaigns, and establish stringent legal frameworks to deter cyber criminals and safeguard potential victims.

Credit Card Fraud

Credit card fraud is a form of identity theft that involves unauthorized use of a credit card or credit card information to make purchases, withdraw funds, or conduct other transactions without the cardholder's permission. It can result in financial losses for both the cardholder and the issuing bank. It can damage the cardholder's credit score or influence the financial institution's reputation. Common types of credit card fraud include:

- Card-Not-Present (CNP) fraud occurs when the fraudster uses stolen credit card information to make online, phone, or mail-order purchases without physically possessing the card. The part of CNP is:
  - Phishing: Fraudsters trick individuals into providing their credit card details by posing as a trustworthy entity through emails, phone calls, or fake websites.
  - Account Takeover: The fraudster gains access to the cardholder's online account, changes the account, and makes unauthorized transactions.

- Card-Present (CP) fraud involves using a stolen or counterfeit physical credit card for in-person transactions at stores or ATMs.

It is not always easy to classify the type of fraud that has occurred. For instance, phishing, which is a form of cyber-fraud, is also considered credit card fraud from the bank's perspective.

1.2. Tailored Approaches to Fraud Detection

The different origins of fraud require different methods to prevent and detect it. One of the techniques to fight fraud is ML and AI. However, one algorithm does not fit all. Each type of fraud presents unique challenges and patterns, necessitating tailored approaches to identify and mitigate fraudulent activities effectively.

For instance, detecting cyber fraud involves monitoring unusual online activities, such as unauthorized access attempts, phishing emails, or suspicious transactions. Techniques like anomaly detection and natural language processing [42] can be used to flag potentially fraudulent activities. Anomaly detection algorithms identify patterns that deviate from typical user behavior, while natural language processing helps recognize malicious content in emails and websites.

Financial statement fraud, on the other hand, requires a different approach. This type of fraud involves manipulating financial reports to present a false picture of a company's financial health. Detection methods include forensic accounting techniques, and data analytics [110]. The literature review [5] showed that the most popular technique in published papers is supervised learning, specifically Support Vector Machine (SVM) [31] and Decision Tree (DT) [109]. On the other hand, Logistic Regression (LR) can outperform an artificial neural network, bagging, decision trees, and stacking in some cases [105].

Medicare fraud is another distinct category that necessitates specialized detection methods. This fraud can involve billing for services not provided, upcoding, or unbundling procedures to receive higher reimbursements. ML methods such as Neural Networks [69] used to combat Medicare fraud might include supervised learning techniques [61] that analyze historical claims data to identify suspicious patterns. These models can flag claims that deviate from expected billing practices, such as excessive treatments or services that are not medically necessary.

CCFD differs significantly from other types of fraud detection, such as Medicare fraud or financial statement fraud, due to the unique characteristics and behaviors associated with each type of fraud. We explore these distinctions in detail.

- Nature of Transactions. Credit card fraud typically involves high-frequency, low-value transactions that occur in real time. This necessitates rapid detection and response mechanisms to prevent further unauthorized use [120], [135].

- Behavioral Analysis. Detection systems focus heavily on analyzing transaction patterns and consumer behavior. Techniques such as anomaly detection [68] and real-time monitoring are employed to identify unusual activities, like spending spikes, geographical inconsistencies, or atypical purchasing behavior.

- Data Volume and Variety. Credit card transactions generate vast amounts of data, including time stamps, merchant details, transaction amounts, and geolocations. Advanced ML algorithms are used to sift through this data to identify fraud patterns.

- Technological Measures. The industry leverages technologies like EMV (Europay, Mastercard, and Visa - a payment method based on a technical standard for smart payment cards, payment terminals, and automated teller machines which can accept them) chips [142], [47], tokenization, and secure 3D protocols (e.g., 3D Secure) to add layers of security. CP fraud persists, but it is clear that the blend of chip and PIN technology for CP transactions along with the vastness of online activities and commerce has placed CNP fraud in the spotlight [32] (see Figure 1.2).

- Additionally, the fraud detection algorithm has a minimal time span to classify transactions. Each financial institution has its guidelines and rules on how much time a transaction can be stopped for classification, whether it is legitimate or fraudulent. Here, we usually are talking in milliseconds [18].

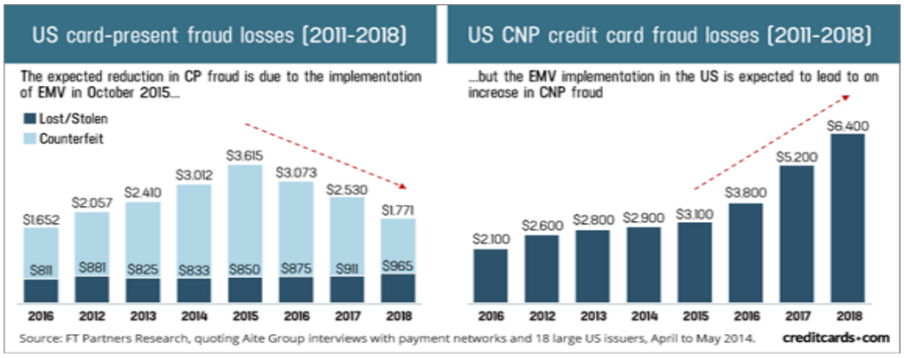


Figure 1.2: Illustration of concept drift: changes in fraudulent activity patterns following the implementation of EMV [32]

To achieve effective credit card fraud detection, researchers encounter numerous additional challenges. The most important ones are:

- Highly imbalanced data - less than 1% of financial fraud cases [34];
- Financial fraud data often present categorical features with high cardinality [12];
- Initial data classification (labeled data) is not always accurate (verification latency problem) [35];
- Concept drift [89], [E.2];
- Data availability for research purposes [4].

1.3. Challenges and Solutions in Learning from Imbalanced Data

Among the many challenges in the detection of credit card fraud, the issue of data imbalance is one of the most critical. Fraudulent transactions, being rare events, create a significant disparity between the classes in the dataset. The following section provides a review of the literature regarding the concept of imbalanced data and its implications for fraud detection systems.

Fraudulent cases are called Minority class as there are much fewer instances, while legitimate transactions are called Majority class. The

binary dataset is defined as imbalanced when one of two classes is much more prevalent in the data than the other one. As fraudulent transactions are a rare event that leads to a sharply imbalanced dataset. Standard ML algorithms treat datasets as roughly balanced, which can cause inaccurate results if used with imbalanced datasets [25]. The imbalance data classification problem can be solved on the data level by balancing the training dataset or on the algorithm level by adjusting the ML algorithm. One of the algorithm-level solutions is modifying the classification threshold by the relevant percent to use those algorithms efficiently [108]. Some ML algorithms such as SVM or XGBoost have parameters to set weights on different classes.

1.3.1. Techniques for Balancing Dataset: Undersampling and Oversampling

A widespread way to do optimization on the data-level is to resample the training dataset. Researchers use various oversampling and/or undersampling techniques for better ML performance. The analysis of the Clarivate within the Web of Science Core Collection indicates that oversampling is currently considerably more favored in the research community, as demonstrated in Figure 1.3.

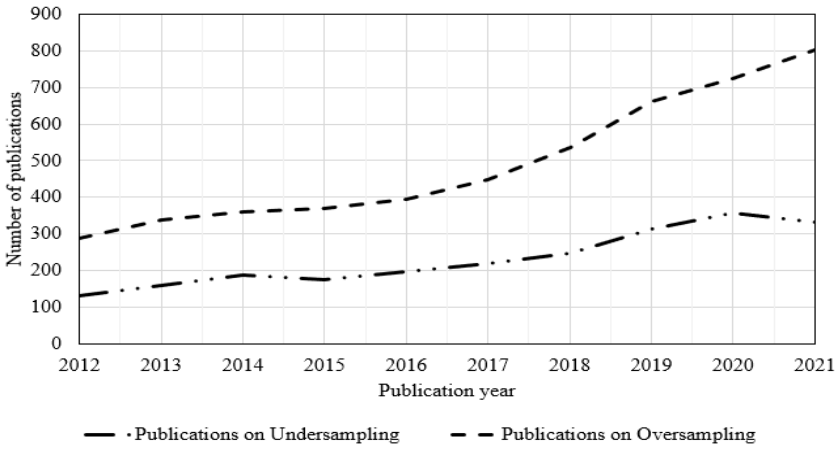


Figure 1.3: Number of publications on undersampling and oversampling techniques, based on the author's research and analysis using the Web of Science Core Collection.

However, both come with advantages and disadvantages. In order to apply resampling methods, the main question that needs to be answered is what share of Minority and Majority classes is the optimal one. The experiment described in [101] uses the oversampling method with approximately a 30/70 split on traffic accident data. A popular oversampling technique is the Synthetic Minority Over-sampling Technique (SMOTE) [22]. It creates synthetic Minority class instances by choosing some of the nearest Minority neighbors, and it generates new samples using the interpolation method between the Minority instances that lie together. Generated data usually do not have accurate probabilistic distribution and are not diverse enough. The paper [149] recommends using "Binary imbalanced data classification based on diversity oversampling using extreme learning machine autoencoder" and "Binary imbalanced data classification based on diversity oversampling by a generative adversarial network". The authors conclude that experimental results show promising performance on imbalanced data classification. However, oversampling techniques require more extensive computational power to generate additional data rows.

In the era of big data, generating large amounts of additional artificial data does not make sense. In this case, researchers try to find optimal data balance using undersampling methods. Many articles have been published on the topic of undersampling [82], [79], [146], [153]. Regardless, a comparison of the undersampling and oversampling techniques showed that the oversampling approach (SMOTE) behaved more robustly than the undersampling (Random Undersampling (RUS)) method under noisy conditions [74]. The experimental results [133] suggest using oversampling rather than undersampling. However, the experimental outcomes were not explicit because undersampling showed better results in several ML models in the same experiment. The most significant disadvantage of undersampling is the data loss, which can create a non-representative dataset.

The study [130] examines the average error rate, recall, and precision across 100 runs per class ratio and dataset, finding that these metrics reach minimum values when the datasets are balanced (50%:50%). In contrast, the metrics are highest when the datasets are imbalanced (10%:90% or 90%:10%) due to increased certainty in variables relative to the target variable. However, this may result in bias since instances are

predominantly associated with the majority class, potentially leading to an unfair advantage for the majority class during evaluation. On the other hand, it was uncertain whether balancing occurred prior to or after the split.

Despite extensive research efforts, several methodological limitations persist in the existing literature in the domain of fraud detection. These issues often arise from data preparation, model evaluation, and performance assessment. Specifically, certain practices in the literature, such as using random splits for training and testing data, introduce the risk of data leakage, leading to overly optimistic performance evaluations. Additionally, the sequence of balancing the dataset prior to the split of the test and train has been observed, which violates the principles of proper experimental design and may result in an overfitting bias. Furthermore, the selection of inappropriate performance metrics, such as accuracy, in highly imbalanced datasets does not reflect the true capability of models to detect rare fraudulent instances. The following subsections elaborate on these recurring issues, highlighting their implications and potential treatments.

1.3.2. Eliminating the Class Imbalance Problem at the Initial Step

Fraud detection is inherently a highly imbalanced problem. Despite this, many studies employ experimental pipelines that fail to accurately reflect this real-world challenge [64], [147]. A common yet flawed approach involves balancing the dataset before splitting it into training and testing subsets. While this may simplify model development and yield higher performance metrics, it introduces several critical issues. Challenges of Balancing Before Splitting are as follows:

- Artificially Eliminating the Imbalance Problem. Balancing the dataset before splitting ensures that the training and testing subsets have a uniform class distribution in both datasets. However, this does not align with real-world scenarios [95] where predicted data remains imbalanced. As a result, the trained model is optimized for a balanced distribution that it will never encounter in practice, leading to significant performance degradation when deployed.

- Data Leakage [138]. Balancing techniques such as oversampling

(e.g., SMOTE) often introduce synthetic data or select samples from the dataset that may inadvertently leak information between training and testing sets. This leakage inflates performance metrics such as accuracy and F1-score, providing a false sense of the model's efficacy.

- Generalization Issues [59]. A model trained on artificially balanced data often overemphasizes the minority class, which may result in an increased number of false positives. In fraud detection, this can have significant operational costs, such as alerting legitimate users unnecessarily or straining fraud investigation resources.

1.3.3. Classifiers Used for Imbalanced Data

Certain classifiers and techniques are particularly effective when dealing with highly imbalanced datasets. Scientific studies [57], [128], [122], [106] have identified several classifiers that perform well in handling class imbalance:

- Ensemble Methods such as Extremely Randomized Trees [50]. This method is robust and can effectively handle many features. Studies [57] have shown that it is beneficial for handling imbalanced data, especially when combined with sampling techniques. Another example is Gradient Boosting Machines, especially XGBoost [24] or CatBoost [107], which are highly effective for imbalanced datasets [57], [128]. The algorithm includes parameters that can be tuned to address the imbalance.

- Cost-Sensitive Learning refers to training models while considering the cost associated with misclassifications. One significant advantage of cost-sensitive learning is its ability to adjust the learning algorithm directly to handle the imbalances without altering the data distribution. Some studies have shown that cost-sensitive algorithms can outperform experiments using sampling methods [122], [106].

Furthermore, the systematic reviews of the literature presented in [28], [15], and [2], which covered publications from the periods 2020–2021, 2019–2021, and 2010–2021, respectively, showed that the

classifiers most frequently used are those listed in Table 1.1. Each entry in the table represents the number of studies employing a specific classifier (numerator) out of the total studies reviewed (denominator) for each reference. The aggregated "Total" column provides an overview of the classifier's overall adoption across the reviewed literature. For instance, Random Forest (RF), with its strong adaptability and effectiveness, was used in 88 out of 233 studies, making it the most frequently applied ML method. Due to their relatively recent introduction, algorithms such as XGBClassifier and CatBoost do not yet have high usage scores. Nonetheless, they are considered state-of-the-art, particularly in non-academic contexts like Kaggle competitions, where their ability to handle complex, imbalanced datasets has proven highly effective. This analysis emphasizes the continued popularity of traditional classifiers while also shedding light on new trends towards the adoption of modern algorithms.

Table 1.1: Distribution of classifiers across reviewed papers (2010–2021)

Classifier	**[28]**	**[15]**	**[2]**	**Total**
RF	11/20	74/181	3/32	88/233
SVM	6/20	56/181	5/32	67/233
LR	9/20	52/181	1/32	62/233
DT	8/20	49/181	3/32	60/233
Naive Bayes (NB)	5/20	42/181	4/32	51/233
k-Nearest Neighbors (k-NN)	7/20	39/181	0/32	46/233
XGBClassifier	0/20	18/181	0/32	18/233
CatBoost	0/20	3/181	0/32	3/233

The recent systematic review of AI-enhanced techniques in CCFD [55], which examined research published between 2019 and 2024, identified 621 relevant publications in Scopus and 300 in Web of Science, underscoring the intensifying academic interest and rapid methodological advancements in this domain. While the full review encompassed a broad range of studies, quantitative method distributions were not explicitly reported. Table 1.2 presents a condensed summary of the usage of the classifier based on the subset of articles explicitly detailed in the cited paper. This distribution was calculated and compiled by the author through a manual count and categorization of the methods.

As shown in Table 1.2, the most frequently used methods were LR,

Table 1.2: Distribution of classifiers across reviewed papers (2019–2024)

Category	Method	Estimated # of Papers
Machine Learning	LR	11
	RF	9
	DT	5
	NB	5
	SVM	5
	k-NN	7
Ensemble / Hybrid	XGBoost	3
	Gradient Boosting	2
	Ensemble methods	2
Deep Learning	Convolutional Neural Network (CNN)	8
	Long Short-Term Memory (LSTM)	8
	Neural Network (NN)	3
	Multilayer Perceptron (MLP)	2
	AutoEncoder	2
	Recurrent Neural Network (RNN)	2

RF, and k-NN among machine learning models, while CNN and LSTM dominated the deep learning category. These findings align with the conclusions of Hafez et al. [55], who emphasized the effectiveness of these models for high-dimensional and sequential fraud detection tasks (the order or timing of events matters), despite not reporting specific frequency counts in the review.

1.3.4. Evaluation Metrics When Classifying Imbalanced Data

Finding a correct metric to measure the model's performance is an additional issue. The classifier outcome can be grouped into four buckets, as shown in Figure 1.4.

Traditional classifiers are built to improve accuracy and the percentage of correctly labeled values for the test data, which is unsuitable for an imbalanced dataset. For instance, if the bank has 0.5% of fraudulent transactions, then the model which labels every transaction as non-fraudulent would have an accuracy of 99.5%, yet it would completely fail to detect any actual fraud, making it practically useless. This illustrates that accuracy is a misleading metric in highly imbalanced datasets, as it does not reflect the model's ability to identify the minority class—fraudulent transactions in this case. Research conducted by Bekkar et al. [11] demonstrated that employing accuracy as a perfor-

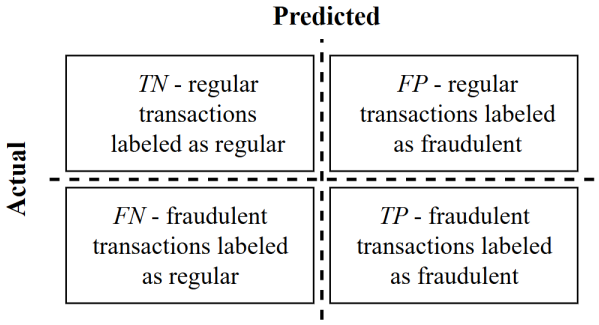


Figure 1.4: Confusion matrix used in this thesis' experiments

mance metric in situations with imbalanced data can often be deceptive. One of the alternative measures used in such a case could be the F1 score as suggested in [60]. The F1 score is a harmonic mean of Precision and Recall, and the input of Precision and Recall have the same preference. Precision is a measure of quality, and Recall is a measure of quantity. Higher Precision implies that an algorithm produces more relevant outcomes than irrelevant ones. In contrast, high Recall indicates that an algorithm produces most of the relevant results (nevertheless, irrelevant values are also returned). The ideal value of the F1 score is 1, and the poorest is 0. The formula of F1 score is presented in Table 1.3, where TP is a prediction results that correctly indicates the presence of a fraudulent transactions (True Positive), FP - a prediction result which wrongly indicates that a fraudulent transaction is present (FP) and FN - a prediction result which wrongly indicates that a fraudulent transaction is absent (False Negative)

Table 1.3: Selected metrics in binary classification

Metric	**Notation**	**Formula**	**Range**
Recall (Sensitivity)	*TPR*	$\frac{\text{TP}}{\text{TP}+\text{FN}}$	[0,1]
Precision	—	$\frac{\text{TP}}{\text{TP}+\text{FP}}$	[0,1]
Specificity	*TNR*	$\frac{\text{TN}}{\text{TN}+\text{FP}}$	[0,1]
F1 score	*F1*	$\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$	[0,1]

Research conducted by [37] aligns with earlier findings, revealing that several widely used binary classification metrics (Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC), accuracy, and TNR) may not be optimal for selecting the superior model in cases of imbalanced data. The study highlights the efficacy of multiple classification metrics such as TPR, the Critical success index, Sokal & Sneath index, Faith index, Matthews Correlation Coefficient (MCC), Geometric Mean (G-Mean), and F1 score for model evaluation when the success rate is low. For scenarios with a high success rate, metrics such as TNR, MCC and G-Mean are more fitting for performance assessment. Notably, the G-Mean accurately reflects the classification performance balance between majority and minority classes. The formulas for some metrics, which are not explicitly covered above, are presented in Table 1.4

Table 1.4: Additional metrics in binary classification

Metric	**Notation**	**Formula**	**Range**
Accuracy	*ACC*	$\frac{TP+TN}{TP+TN+FP+FN}$	[0,1]
Critical success index [119]	*CSI*	$\frac{TP}{TP+FP+FN}$	[0,1]
Sokal & Sneath index [124]	*SSI*	$\frac{TP}{TP+2\times FP+2\times FN}$	[0,1]
Faith index [37]	*FAITH*	$\frac{TP+0.5\times TN}{TP+FP+FN+TN}$	[0,1]
Matthews Correlation Coefficient [152]	*MCC*	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$	[-1,1]
G-Mean [80]	*GM*	$\sqrt{TPR \times TNR}$	[0,1]

The Matthews Correlation Coefficient (MCC) is a measure commonly used to assess the quality of binary classification models, especially when dealing with imbalanced datasets. It takes into account true positives, true negatives, FP, and false negatives and provides a value that ranges from -1 to +1, with +1 indicating a perfect prediction, 0 indicating random prediction, and -1 indicating complete disagreement between prediction and observation.

The G-Mean, also known as the geometric mean or balanced accuracy, is a statistical measure used to evaluate the performance of classification models, particularly in situations where class imbalance exists. It offers a balanced perspective by considering sensitivity (recall) and specificity.

AUC-PR [36] is a performance metric used to evaluate the effective-

ness of classification models, especially in scenarios where class imbalance exists or when the focus is on positive instances. The Precision-Recall curve plots precision against recall as the classification threshold changes. Precision represents the proportion of correctly predicted positive instances among all instances predicted as positive, while recall is the proportion of correctly predicted positive instances among all actual positive instances.

AUC-ROC [45] is widely used performance metric for binary classification models. The ROC curve plots TPR against the FP rate as the classification threshold changes.

Over the years, researchers have raised questions about whether traditional measurements are sufficient to be used when classifying imbalanced data. New measurements are suggested, such as the weighted AUC-ROC [144], the adjusted F-measure [91], or the Bayes imbalance impact index [88] as an example. This study will employ conventional metrics, as they are well-established and commonly recognized in the field of classification of imbalanced data. These metrics facilitate direct comparisons with previous research, thus contextualizing the effectiveness of the proposed method. Using new metrics can complicate comparisons, given their potential lack of validation or acceptance in the research community.

1.3.5. Pitfalls in Model Testing on Temporally Structured Fraud Data

Credit card, investment, or any other type of fraud data has a concept drift property [17], [35], [89]. Concept drift refers to the phenomenon in which the underlying statistical properties of the data distribution change over time. This can happen for various reasons, such as changes in user behavior, fraud patterns, or market conditions. These changes can be gradual or abrupt (see Figure 1.5), challenging the conventional ML assumption that the data used for training and testing remain static.

The classical train-test split assumption of independent and identically distributed (i.i.d.) samples does not hold well for time series data, particularly in domains like fraud detection, where concept drift is a common challenge [54], [104]. Concept drift violates the assumption that training and testing data originate from the same distribution, complicating model evaluation and performance estimation.

Datasets for CCFD are inherently temporal, meaning observations

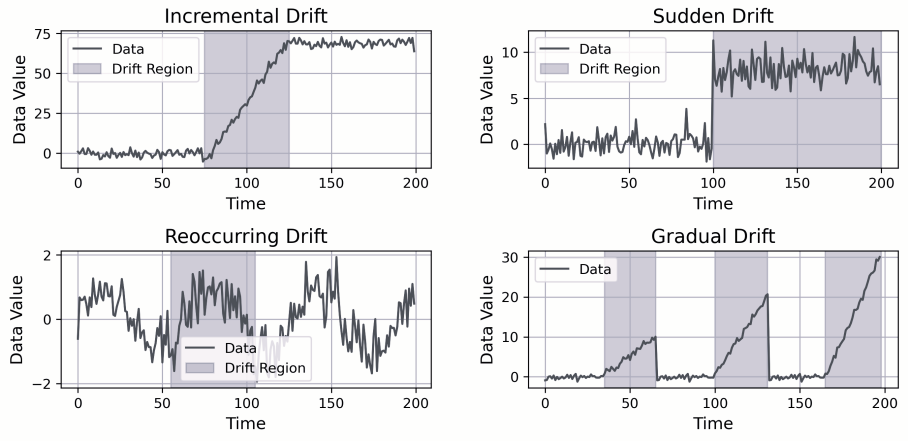


Figure 1.5: Types of Concept Drift. Visualization designed by the author.

exhibit sequential dependencies. This temporal structure creates correlations between data points that are close in time. Applying a standard train-test split, where data is randomly shuffled and partitioned, risks introducing temporal leaks. Such leaks can lead to unrealistic correlations between training and testing sets, resulting in overly optimistic performance estimates [139].

A significant challenge in fraud detection research is the inconsistency in data splitting strategies across studies, which makes comparing results difficult. Many papers fail to specify the splitting ratio or whether a random or time-based split was used. In several cases, fraud detection studies evaluate model performance using random splits ([84], [66]), assuming that the data is i.i.d. over time. However, real-world credit card transaction data often exhibits temporal dependencies and non-stationarity. Consequently, models trained on one time period may perform poorly when applied to another due to shifts in transaction patterns, fraudulent behaviors, or user habits.

Addressing these challenges is crucial, as random shuffling of time-series data can lead to data leakage, ultimately distorting model evaluation and hindering its real-world applicability. To mitigate the effects of concept drift and temporal dependencies, various methodological solutions have been proposed. A summary of existing approaches is provided in the conference poster [E.2], which reviews a range of adaptive strategies such as incremental learning, transfer learning, and ensemble-

based methods. These techniques aim to maintain model relevance over time by continuously updating the learning process or integrating multiple models trained on different time segments. The poster highlights that overlooking concept drift can lead to reduced model accuracy and increased false positives.

1.4. Data Transformation: Encoding Categorical Variables

This section provides an in-depth analysis of key and influential academic studies focusing on encoding high-cardinality categorical features across datasets of different sizes and applications. Following this, the impact of high-cardinality feature encoding when dealing with highly imbalanced datasets, where the minority class contains less than 1% samples from the whole dataset, will be reviewed.

1.4.1. High-Cardinality Categorical Features Encoding

Comprehensive research on high-cardinality feature encoding for classification and regression problems using balanced datasets is presented in the paper [100]. The authors compare seven encoding techniques using five machine-learning algorithms on 24 datasets. Datasets used in the research are binary or multi-class and relatively balanced compared to fraudulent transaction datasets. Chosen datasets differ in size: the smallest is less than a thousand entries, and the biggest is more than a million. The datasets consist of 1 to 20 categorical features, each with over 10 levels (distinct values of a particular feature). The highest number of levels for a feature varies from 14 to 30 114. The article suggests that target-based encoders outperform target-agnostic encoding techniques.

Uyar et al. [132] compared automatically calculated techniques against expert judgment. Feature encoding techniques were investigated in IVF (in-vitro fertilization) implantation prediction. The suggested frequency-based encoding technique outperforms expert judgment.

A special case was presented in [123], where the Bayesian encoding technique was developed for WeWork's lead scoring engine. The company faces a high cardinality feature problem as they have categorical features with more than 300k categories. The authors state that the AUC-ROC metric improved from 0.87 to 0.97. However, when researchers compared performance on the publicly available dataset, the developed

solution was not so impressive. Due to high-cardinality, these types of features are sometimes excluded from the modeling scope. However, [19], [94] showed that the model's performance increases statistically significantly when they are included.

1.4.2. Features Encoding for Imbalanced Data

The paper [12] investigates the impact of feature encoding techniques on highly imbalanced fraudulent transaction dataset. The data used for the research is from a major French bank, and Data Protection Law does not allow sharing it. In this case, replicating the experiment is not possible. However, the results and conclusions inspire more profound research. Another study on real data [113] proposes a way to encode categorical features by applying Word2Vec embedding, which is usually used for word and sentence encoding. The outcome of the research was a 50% reduction in memory usage and slightly improved performance.

Johnson and Khoshgoftaar published several papers regarding high-cardinality categorical features encoding on Medicare Fraud Prediction [70], [71]. The dataset used in the research is highly imbalanced as in 56 million rows, only 0.06% are fraudulent. With paper [70], researchers showed that semantic embedding performs significantly better than the traditional one-hot encoder, and Skip-Gram (SG) embedding performs best overall. One-hot encoding is a technique used to transform categorical data into numerical data, and it defines categorical data as binary vectors. In this method, each category is represented as a binary vector with a length equal to the total number of categories. The vector contains 1 in the position corresponding to the category and 0 elsewhere. SG embedding is a neural network trained to predict the surrounding words given a target word. The experiments in [71] showed that One-hot encoding is unsuitable for high-cardinality features when using ensemble learners.

While categorical encoding is often treated as a purely technical preprocessing step, recent research has brought attention to its ethical implications. Mougan et al. [96] demonstrate that encoding protected categorical attributes using one-hot or target encoding can introduce both irreducible and reducible bias. Their findings show that regularization techniques such as smoothing or adding Gaussian noise, can effectively mitigate these biases without significantly sacrificing model

performance. In a complementary study, Valentim et al. [134] highlight that the data preparation phase, including encoding strategies, has a significant impact on fairness outcomes in software systems. These studies underscore the need to incorporate fairness-aware practices into feature encoding, particularly in sensitive domains such as financial fraud detection.

The above-mentioned studies demonstrate that one-hot encoding is generally unsuitable for high-cardinality categorical features in imbalanced datasets due to the increased dimensionality and sparse representation, which may dilute the signal from the minority class. In such cases, more compact and information-rich encodings are preferred. Techniques such as target encoding, frequency encoding, or embedding-based methods (e.g., Word2Vec, Skip-Gram) have demonstrated improved performance. These methods preserve semantic relationships between categories while reducing dimensionality, making them more effective when the minority class is underrepresented.

1.5. Feature Selection Techniques

The evolution of technology and the increasing complexities of digital transactions have given rise to sophisticated fraudulent activities, necessitating novel and intelligent solutions for detecting and preventing such cyber threats. This section contains an overview of the feature selection methods when working with imbalanced datasets, especially in fraud detection applications.

Several studies have highlighted that including too many features can negatively impact machine learning performance. Empirical studies have shown that adding too many features - particularly weakly relevant or noisy ones - can increase model variance and degrade performance [97]. In contrast, well executed feature selection improves both predictive accuracy and execution efficiency, which is particularly valuable in time-sensitive and resource-constrained domains [98].

1.5.1. General Approaches to Feature Selection

Feature selection techniques are commonly categorized into three main groups: filters, wrappers, and embedded methods [27], [10]. The review paper [86] delves into the significance of feature selection in ML

and data mining. It highlights contemporary challenges that are of particular importance. These challenges include feature selection for high-dimensional data with small sample sizes, dealing with large-scale data, and ensuring secure feature selection. Despite these challenges, several noteworthy trends in feature selection have surfaced, such as stable feature selection, multi-view feature selection, distributed feature selection, multi-label feature selection, online feature selection, and adversarial feature selection. The paper goes on to explore recent advancements in these areas. For each trend, it examines the current issues, presents existing solutions, and discusses them. Beyond these trends, the paper also introduces diverse applications of feature selection. These applications span fields including bioinformatics, social media analysis, and multimedia retrieval, showcasing practical relevance.

An alternative approach [145] to arranging feature selection methods involves distinguishing between global and instance-wise feature selection strategies. The primary objective of global feature selection is to identify a singular feature selector applicable to all data samples, focusing on minimizing the number of features while retaining the capacity for discriminative predictions. On the other hand, instance-wise feature selection involves calculating distinct selectors for each instance, resulting in enhanced performance compared to the global feature selection approach. The article [145] suggests group-wise feature selection, which occupies an intermediate position between global feature selection and instance-wise feature selection.

The paper [33] highlights the importance of feature selection in reducing data processing complexity, particularly in the context of high-dimensional data. The study introduces the concept of Fuzzy Combination Entropy (FCE) to address the limitations of classical combination entropy, especially in handling continuous features. The paper presents the development of FCE based on fuzzy λ-similarity relation, incorporating fuzzy rough sets and combination entropy. Furthermore, the concepts of global and local feature correlations are defined, leading to the design of a feature selection method, FSmFCE. Experimental findings demonstrate the algorithm's ability to preferentially select a smaller feature set while maintaining commendable classification performance.

1.5.2. Feature Selection for Imbalanced Data

When working with imbalanced datasets, where one class is significantly more prevalent than the other, feature selection becomes an even more complex task. Imbalanced datasets can introduce biases and negatively affect the performance of ML models.

Researchers propose many different approaches for feature selection when working with imbalanced data. The work by Yin et al. [148] suggests a feature selection technique that centers around class decomposition. The suggested approach initially subdivides majority classes into more manageable pseudo-subclasses characterized by relatively balanced sizes. Subsequent feature selection operates on these newly decomposed data to calculate feature goodness metrics. Moreover, the study introduces a feature selection method reliant on the Hellinger distance [29]. It measures distribution divergence, offering greater resilience to imbalanced class distributions [148].

Another example is when neighborhood rough set theory is employed for feature selection [23]. The empirical findings showed the effectiveness of RSFSAID (Rough-Set-based Feature Selection Algorithm for Imbalanced Data) across binary and multiclass datasets. Nevertheless, in most scenarios, the information about the minority class holds greater significance. The noise within the minority class might impact the classifier's generalization ability when utilizing the chosen features.

The paper [63] introduces a feature selection technique for imbalanced data, utilizing a new regularization method called IR-LDA to enhance classification performance by emphasizing the minority class. The method employs cosine similarity to address feature redundancy issues and incorporates the regularization into the global feature selection framework, improving classifier performance and reducing feature redundancy.

In summary, feature selection plays a crucial role in enhancing model performance, reducing computational complexity, and improving interpretability, particularly in high-dimensional and imbalanced datasets. General approaches such as filter, wrapper, and embedded methods continue to evolve, with recent advances addressing challenges like scalability, multi-label classification, and adversarial robustness. Additionally, new paradigms - such as instance-wise and group-wise feature selection - offer more adaptive and context-aware solutions. In

the specific context of imbalanced data, tailored methods that prioritize the representation of the minority class, such as class decomposition, Hellinger distance-based metrics, neighborhood rough sets, and specialized regularization techniques, have shown promising results. These strategies help mitigate the negative effects of imbalance and ensure more reliable feature selection for fraud detection and similar applications.

1.6. A Review of Fraud Detection and Class Imbalance Studies in Lithuania

Research on fraud detection and imbalanced data has evolved into substantial and diverse work, with outputs spanning more than a decade and increasing in volume in recent years through both national and international collaborations. Early studies in tax fraud concentrated on conceptual prevention frameworks integrating economic, legal, and behavioural dimensions [126], [114], while financial statement fraud was approached through statistical ratio analysis and LR to detect anomalous reporting patterns [73]. Subsequent work expanded to emerging digital threats, including money laundering detection via decision tree models on synthetic financial datasets [140], classification of fraudulent e-commerce platforms using combined textual, and structural features [67], and detection of online advertising fraud with neural network architectures incorporating embedding layers and behavioural feature engineering [49].

The methodological challenge of extreme class imbalance - common to fraud data where illicit cases form a tiny fraction of the total - is a recurring theme. Research on network intrusion detection has adapted resampling techniques such as SMOTE alongside ensemble methods to improve minority class recall without excessive false positives [16], offering lessons transferable to fraud analytics. Other contributions include multiple outlier detection tests for parametric models [9] and fuzzy-logic-driven feature selection [136] to improve classifier sensitivity to rare-event patterns. More recent studies extend into cybersecurity, such as malware detection [141]. Taken together, this body of work underscores the importance of combining domain-specific feature engineering with imbalance-aware machine learning to achieve robust detection performance in fraud and related anomaly detection tasks.

1.7. Data for Fraud Detection Research

The quality, completeness, and representativeness of the data directly influence the performance and reliability of the developed models. This section addresses various challenges associated with data availability in the research area of financial fraud detection.

Financial fraud is one of those areas where access to data is very limited. Many large-scale datasets are susceptible and restricted by laws such as General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA). These regulations limit research investigations and create bottlenecks in ML development.

In this case, synthetic data is a promising technology that helps to solve privacy, fairness, data augmentation, and many other issues. To accelerate scientific research, necessary datasets can be available with high volume, velocity, and variety.

The definition of synthetic data proposed by [72] is "Synthetic data is data that has been generated using a purpose-built mathematical model or algorithm, with the aim of solving a (set of) data science task(s)."

Various data types can be synthesized, including Tabular, Image, and Audio data. In this research, we are interested in how tabular data can be generated and its fairness. This kind of data consists of rows and columns. The synthesis of such data demands the simultaneous modeling of each column distribution and has row- and table-wise constraints.

Synthetic data plays a vital role in research and developments where data availability is limited by not only laws but also its nature to be rare. A great application of synthetic dataset is presented [21]. This synthetic dataset is used to generate realistic cyber data for ML classifiers for network intrusion detection systems [21]. The article concludes that their chosen generative methods, CTGAN and TVAE, generated synthetic cyber data reasonably well. Yet, ML models trained with only synthetic data resulted in low classification recall. In addition, the authors suggest having at least 15% actual data when training the model.

1.7.1. Synthetic Datasets for Credit Card Fraud Detection

Erik Altman generated a dataset [3], which aims to allow researchers and developers to work on the data that represents the buying habits of

U.S. citizens. This dataset is like a virtual world with customers, merchants, and fraudsters. The model creates features so that main statistics like mean and standard deviation would be the same as in the actual population. However, it is not enough to have only means and standard deviations. The author selects characteristic values for individuals by stochastic sampling, generally from a Gaussian distribution. The advantage of other synthetic datasets like [87] is that an individual's activities are related. For instance, if an individual is in travel mode, he/she will have different spending behavior. Similarly, the same logic applies if the purchase happens on weekdays or weekends, and much more evidence that this dataset reflects the actual population can be found in [3]. A noteworthy point is that the virtual credit card dataset captures real-world banking actions, such as the production of the card chip. This chip technology was widely adopted in the U.S. in 2014, replacing magnetic stripe technology, and subsequently made "card-present" fraud more difficult, where the fraudster physically presents a stolen credit card to a merchant. Exploratory data analysis of this dataset revealed that fraudsters tend to purchase specific goods on a preferred day and month. They are interested in different deceptions. As shown in Figure 1.6 and Figure 1.7, fraudulent transactions are more concentrated within a specific time window, whereas non-fraudulent transactions are more widely distributed throughout the day. Additionally, fraudsters tend to attack older people.

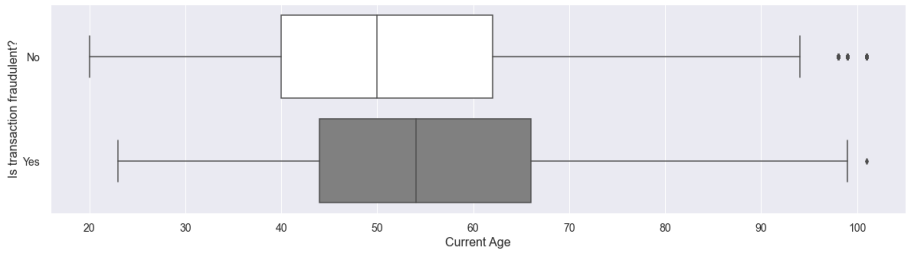


Figure 1.6: Fraudster attacks by age

In this virtual world, merchants represent many real-world retailers' behavior, such as McDonald's, WallMart, or luxury goods shops. Retailers' profit is generated depending on their type. So, fraudsters' manners are generated based on the merchant's service. For future reference, we will refer to this dataset as DataSet1.

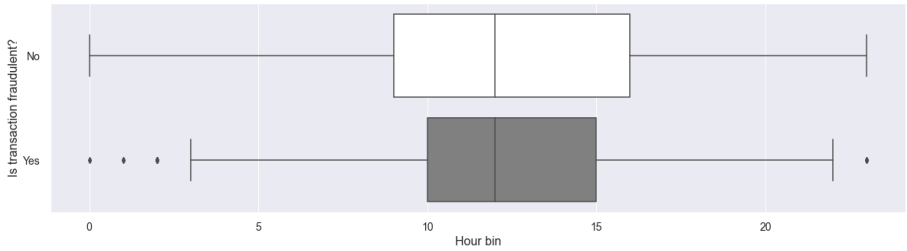


Figure 1.7: Fraudster attacks by hour

Additionally, the second data source used in the experiments is synthetic and was created with the Sparkov Data Generation tool [58]. Syntethic data was generated through a process that simulates credit card transactions over time. This process involves creating realistic transaction patterns for individual users and user segments based on various behavioral characteristics, such as spending frequency, transaction amounts, and merchant categories. The generated data includes normal transactions as well as fraudulent ones, ensuring a diverse and imbalanced dataset that mimics real-world fraud scenarios. The simulation incorporates temporal aspects, such as recurring payments and anomalies that may indicate potential fraud. For future reference, we will refer to this dataset as DataSet2.

Table 1.5 provides a comparison between two synthetic datasets utilized for fraud detection, focusing on variations in class distribution, dataset dimensions, and feature quantities. In the first synthetic dataset, 99.86% of transactions are labeled as non-fraudulent compared to 0.14% that are fraudulent, encompassing a total of 3 445 553 entries with 25 features. On the other hand, the second synthetic dataset records a slightly reduced percentage of non-fraudulent transactions at 99.48%, with the fraudulent ones constituting 0.52%. It includes 1 852 394 entries and 11 features. These distinctions indicate differences in data complexity and the representation of fraud across the two datasets, which might influence the effectiveness of fraud detection models.

1.8. Conclusions of the Chapter

The challenges posed by imbalanced data in ML are significant and pervasive across various domains. The literature review highlights

Table 1.5: Summary statistics of the synthetic datasets utilized

Category	Synthetic DataSet1 [3]	Synthetic DataSet2 [58]
Not Fraud (Percentage)	99.86%	99.48%
Fraud (Percentage)	0.14%	0.52%
Number of instances	3 445 553	1 852 394
Number of features	25	11

several effective techniques to address these challenges, emphasizing the importance of data resampling methods, such as oversampling the minority class and undersampling the majority class. Advanced techniques like SMOTE and its variations have shown promise in generating synthetic samples to balance datasets more effectively. Moreover, ensemble methods, including bagging and boosting, have been instrumental in enhancing the performance of classifiers on imbalanced datasets. These techniques work by creating multiple models and aggregating their predictions, which mitigates the bias towards the majority class often observed in single classifiers. Classifiers specifically designed or adapted for imbalanced data, such as cost-sensitive learning algorithms, have also demonstrated effectiveness. These algorithms adjust their learning process to account for the imbalance by assigning higher misclassification costs to the minority class, thereby improving prediction performance for rare but critical cases.

Although substantial progress has been made in tackling the issue of imbalanced data, several gaps still exist in the related literature:

1. Additional empirical studies are needed to systematically compare the effectiveness of different high-cardinality categorical feature encoding techniques across various types of fraud detection datasets and ML models. These studies can offer more detailed recommendations to practitioners about the best encoding strategies to use. The interaction between advanced ML models, such as ensemble methods and neural networks, and feature encoding techniques needs further exploration. The ML model effectiveness depends on feature encoding techniques. Understanding how these models manage the encoded features can result in more efficient and effective preprocessing workflows in fraud detection.

2. Given that feature encoding can impact model fairness, particularly in sensitive financial areas, there is an urgent need for research that tackles these ethical issues and suggests guidelines for fair and unbiased encoding practices. The ethical concerns and potential biases introduced by different encoding methods are not well explored, particularly in the context of CCFD.

3. Feature selection becomes a vital focus in detecting credit card fraud because of the vast number of features created by financial institutions. Choosing the smaller number of most relevant features not only decreases computational load but also ensures rapid model predictions. Speed is essential for real-time fraud detection to prevent interruptions in customers' payment activities. As reviewed above, credit card fraud prediction must happen in milliseconds.

4. The current evaluations in academic research of CCFD are often not meaningful, as typically, they involve a random shuffling of data. Such approaches do not accurately reflect real-world conditions when models need to be trained on historical data and tested on more recent data.

2. IMBALANCED DATA CLASSIFICATION APPROACH BASED ON A CLUSTERED TRAINING SET

The literature review highlighted the present research gaps in the field of imbalanced data classification, especially concerning credit card detection. Starting with this Chapter, we outline proposed strategies and methods to address the aforementioned research gaps. Sections of this chapter have been included in the book "Data Science in Application" [B.1]. The findings from paper [B.1] were showcased at the Lithuanian Actuarial Society Seminar. Additionally, these results were shared at the international conference [C.1].

This section presents a comprehensive strategy to improve classification performance on imbalanced datasets, which involves training multiple classifiers on clustered training data, utilizing techniques such as k-means for clustering. For clarity, the terms *strategy* and *approach* are used interchangeably throughout this section to refer to the proposed method. The approach begins with the preparation of an imbalanced dataset described in detail in Subsection 2.4, which is then split into training, validation, and test subsets. The training data is standardized and clustered using a technique such as k-means, with the optimal number of clusters determined using the Silhouette Score, which evaluates intra-cluster cohesion and inter-cluster separation. Once the clusters are established, the training set is divided into subsets, with each subset containing the data points assigned to a specific cluster. Within each cluster, the majority and minority classes are separated, and a cluster-specific undersampling ratio is determined to balance the class distribution locally. Each resulting balanced cluster subset is then used to train a dedicated classifier, forming a specialized sub-model. Validation data is used to fine-tune each sub-model. During inference, the Euclidean distance between each test sample and the training cluster centroids is computed to assign the sample to its closest cluster. The corresponding sub-classifier is then activated to produce the prediction. This localized, distance-based classification strategy allows the model to adapt to the intrinsic structure of the data and has the potential to significantly improve performance on rare class detection. The entire process is illustrated in Figure 2.1.

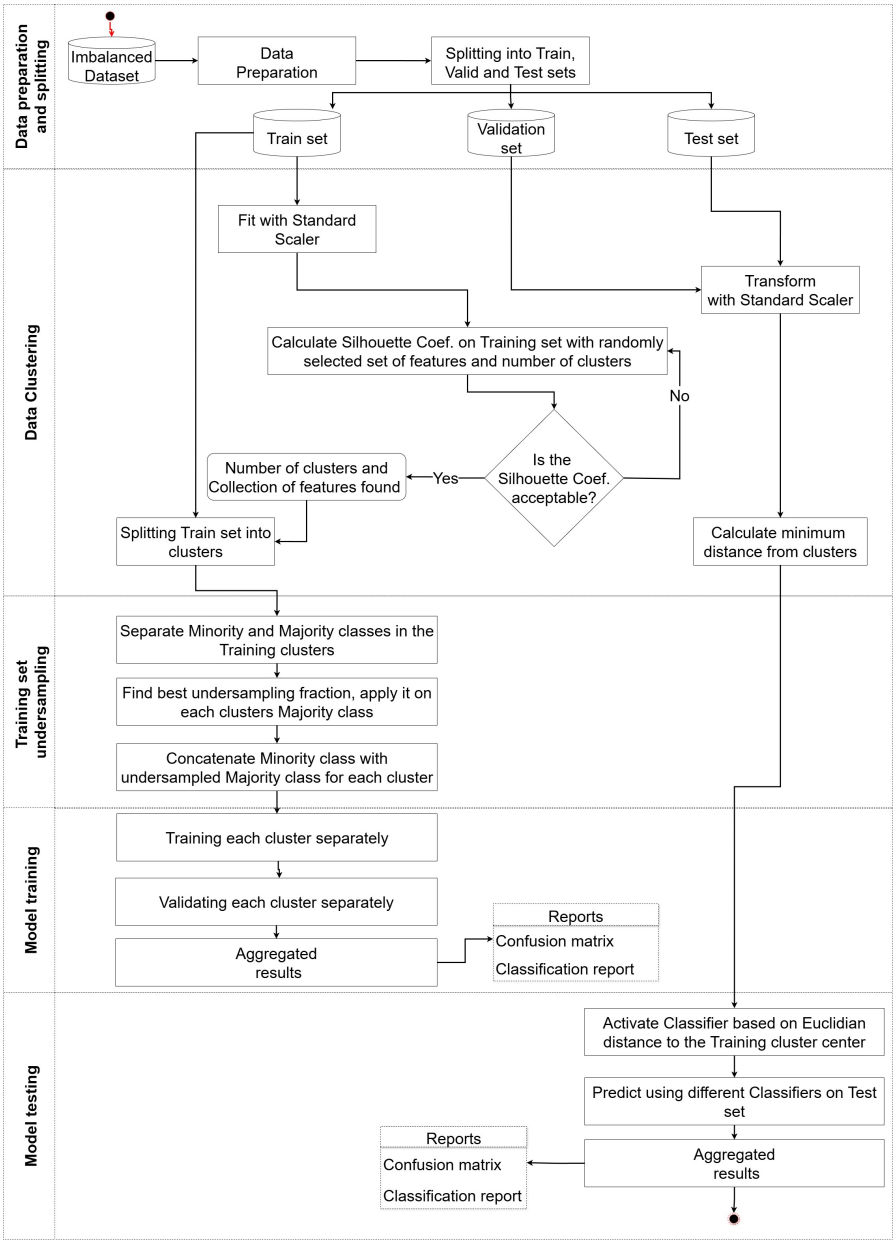


Figure 2.1: Cluster-Specific strategy for imbalanced data classification

The performance of the proposed strategy is evaluated using $recall$, which is particularly important in the context of the detection of credit card fraud. In this setting, the fraudulent class is considered the positive class, while regular transactions represent the negative class. $Recall$ measures the proportion of actual fraudulent transactions that the model correctly identifies. A high $recall$ indicates that the model successfully detects most fraud cases, which is critical for reducing financial losses and limiting the impact of fraud. While we acknowledge that relying solely on $recall$ can be misleading, for example, a model that labels all transactions as fraudulent would achieve perfect $recall$ but perform poorly overall, our main focus in this approach remains on maximizing $recall$, given the asymmetric cost of misclassification. In fraud detection, FP (regular transactions incorrectly flagged as fraud) are generally more acceptable than false negatives (fraudulent transactions missed by the model), since the latter result in direct financial loss. Nevertheless, we do not ignore other performance aspects. An illustration of the confusion matrix used for evaluation is provided in Figure 1.4.

We divide the dataset into three subsets: training, validation, and test. The test set consists of the most recent data, which comes from a later time period than the training and validation sets. In contrast, the training and validation sets are both drawn from the same historical time window but are split randomly, using a fixed proportion such as 80% for training and 20% for validation. This ensures that the model is validated on data it has not seen during training, while still being tested on future, unseen data. The split strategy is illustrated in Figure 2.2.

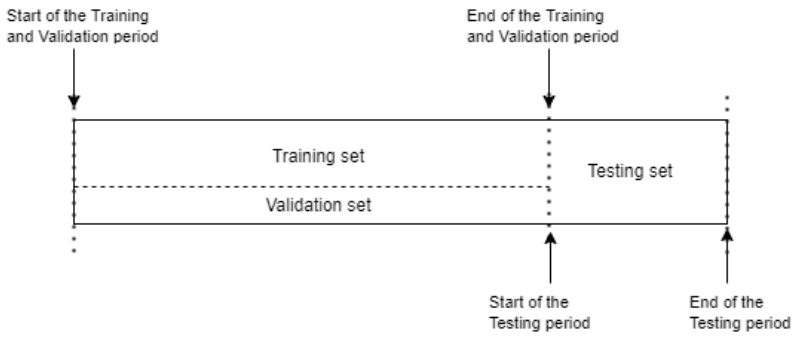


Figure 2.2: Suggested dataset split into Train, Validation, and Test sets

2.1. Splitting Financial Transactions into Homogeneous Clusters

Consider a multidimensional dataset represented as an array X containing n data points, where each data point X_i ($i = 1, \ldots, n$) is a vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ in $\mathbb{R}^m$. These data points are observations of objects or phenomena influenced by m different features $(x_1, x_2, \ldots, x_m)$. Some of these features are numerical, while others are categorical. Furthermore, each data point is associated with a class label y_i, where y_i indicates the class to which the sample X_i belongs.

In our specific context, these features describe various aspects of customers' financial behavior. We have categorized these data points into two classes, where 0 represents the Majority (Regular or Legitimate transactions), and 1 signifies the Minority (Fraudulent transactions). Therefore, the target variable y assumes values $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$.

This chapter focuses on splitting the initial training set into smaller clusters using the k-means clustering algorithm. The k-means algorithm is a widely utilized unsupervised learning technique that clusters data by separating instances into k clusters by reducing within-cluster sum-of-squares (see Formula 2.1). This ensures that instances within each cluster exhibit maximal similarity while maintaining distinct separation between clusters.

$$\min \sum_{i=1}^{k} \sum_{X_j \in C_i} \|X_j - \mu_i\|^2, \tag{2.1}$$

where μ_i is the mean of points in C_i.

In order to use the k-means algorithm, it is necessary to specify the number of clusters. Even though there is no single way to determine the optimal number of clusters, it can be done visually or using the Silhouette score.

A well-known method for visually determining the number of clusters is the *Elbow method*. It helps data scientists to select the optimal number of clusters by drawing the line with the distortion score (sum of square errors) or other relevant scores on the vertical axes and the number of clusters on the horizontal axes. In this case, the "error" is the distance between each data point and the cluster center, which may be either a calculated centroid or an actual representative data point. If

the line chart corresponds to an arm, then the *elbow* indicates the point where the model fits the best.

When using the elbow method, identifying the optimal number of clusters can sometimes be challenging – for example, when the curve is nearly linear or contains multiple fluctuations, making the "elbow" point ambiguous. In such cases, the Silhouette score can serve as a complementary, quantitative guideline for assessing clustering quality.

The Silhouette score evaluates how similar a data point is to its own cluster compared to other clusters. A higher score indicates better-defined and more separated clusters. The score for a data point X_i is computed as [112]:

$$s(X_i) = \frac{b(X_i) - a(X_i)}{\max\{a(X_i), b(X_i)\}}, \tag{2.2}$$

where:

- $a(X_i)$ is the mean *intra-cluster distance*, i.e., the average distance between X_i and all other points in the same cluster C_I:

$$a(X_i) = \frac{1}{|C_I| - 1} \sum_{X_j \in C_I, j \neq i} d(X_i, X_j), \tag{2.3}$$

- $b(X_i)$ is the mean *nearest-cluster distance*, i.e., the smallest average distance between X_i and all points in any other cluster C_J:

$$b(X_i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{X_j \in C_J} d(X_i, X_j). \tag{2.4}$$

Here, $d(X_i, X_j)$ denotes the Euclidean distance between data points X_i and X_j, and $|C_I|$ is the number of points in cluster C_I.

The Silhouette score ranges from -1 to 1. A score close to 1 indicates that the sample is well matched to its own cluster and far from other clusters, a value around 0 suggests overlapping clusters, and negative values imply that the sample may have been assigned to the wrong cluster [103].

We used the Silhouette score to select relevant features and the number of clusters. Additionally, we evaluated the results by plotting an *elbow* graph. We suggest empirically checking the feature combinations and the number of clusters until choosing those that satisfy

the expectations. The criterion for selecting relevant features and the optimal number of clusters is the highest Silhouette score for various combinations of features. The pseudo code for selecting features for clustering is provided in Algorithm 1. The goal of the algorithm is to find the best combination of variables (up to a predefined maximum) and the optimal number of clusters (within a specified range) that together yield the highest silhouette score. We explore cluster counts from $k_{\min}$ to $k_{\max}$. For each value of k, we evaluate the clustering quality of multiple feature subsets using the silhouette score as the objective. Instead of exhaustively testing all possible feature combinations - for instance, it would be in total 16 515 combinations for a dataset with 27 features - we apply a random sampling strategy. The number 16 515 results from evaluating all combinations of 1, 2, or 3 features out of 27 available variables:

$$\binom{27}{1} + \binom{27}{2} + \binom{27}{3} = 27 + 351 + 2925 = 3303, \tag{2.5}$$

and since this is repeated for each of the 5 values of k (from 2 to 6), the total number of possible evaluations becomes is $3303 \times 5 = 16515$.

Evaluating all combinations would be computationally prohibitive. Therefore, we randomly sample 56 feature combinations per value of k, across all subsets of size 1 to 3, using a fixed random seed for reproducibility. This results in a total of 280 evaluations.

For each sampled subset, k-means clustering is applied and the silhouette score is computed. The best-performing subset for each k is recorded. Finally, the overall best configuration, defined by the highest silhouette score across all k values and feature combinations, is selected as the final model. This strategy balances computational feasibility with exploratory coverage, while still optimizing for both intra-cluster compactness and inter-cluster separation in a data-driven and interpretable manner.

It is important to mention that the k-means algorithm is sensitive to the amplitude of the feature values, so it is necessary to use the scaling method before the k-means algorithm.

Algorithm 1 Greedy clustering algorithm with random subsets

Require: Dataset D,
Require: feature list F,
Require: max subset size m,
Require: $k_{\min}$, $k_{\max}$
Require: h number combinations to test
Ensure: Best k and corresponding best feature subset

Initialize $results \leftarrow [\,]$, $vars\_map \leftarrow \{\}$
for $k \leftarrow k_{\min}$ to $k_{\max}$ **do**
    $best\_score \leftarrow -1$, $best\_features \leftarrow \emptyset$
    $C \leftarrow$ all combinations of F of size 1 to m
    Sample h combinations $C_{sample} \subset C$ with fixed random seed
    for all $features \in C_{sample}$ **do**
        Fit KMeans on $D[features]$ with k clusters
        Compute Silhouette score s on clustered data
        if $s > best\_score$ **then**
            $best\_score \leftarrow s$
            $best\_features \leftarrow features$
        end if
    end for
    Append $best\_score$ to $results$
    Store $best\_features$ in $vars\_map[k]$
end for
$k^* \leftarrow \arg\max(results)$
Fit final KMeans on $D[vars\_map[k^*]]$ with k^* clusters
Return $best\_features[k^*]$

2.2. Cluster-Specific Class Balancing via Undersampling

After determining the features used for clustering and the number of clusters k, the initial training set is divided into smaller k training subsets, which are used as training sets for individual ML models. Since the original dataset is highly imbalanced, each cluster remains imbalanced after partitioning. To address this, we apply a cluster-specific undersampling strategy to reduce the size of the majority class within each cluster. We deliberately chose undersampling over oversampling, reasoning that undersampling is computationally more efficient. Moreover, we suggest

using cluster-based undersampling rather than oversampling, based on the reasoning that clustering has already grouped similar samples together, making it possible to remove redundant majority-class instances within each cluster. This reduction is expected to sharpen the model's focus on the minority class while preserving the diversity of the majority class. In the following sections, we empirically evaluate this choice and demonstrate its effectiveness through comparative experiments.

We propose to use an individual RUS strategy for each training cluster. In our case, the undersampling means leaving all points of the Minority class (fraudulent cases) and removing some percentage of points from the Majority class (regular transactions).

The validation set is utilized to individually determine the best-performing undersampling percentage for each cluster. Let us fix some undersampling percentages for the clusters. When the undersampling of the training set is performed, k sub-classifiers are trained. We go through all validation dataset points and apply one of the sub-classifier for decision. The criterion for the selection of a proper classifier is the minimal Euclidean distance between the validation set point and the corresponding cluster center of training data. We check the best performing undersampling percentage for each training cluster by calculating the F1 score on the validation data. While our primary goal is to improve $recall$, we use the F1 score in selecting the best performing resampling percent, because otherwise we could end up having an unacceptable number of regular transactions labeled as fraudulent.

2.3. Cluster-Specific Classification Using eXtreme Gradient Boosting

For training sub-classifiers within each cluster, we employ XGBoost, which stands for eXtreme Gradient Boosting [24]. The XGBoost classifier algorithm starts by initializing the model with a single Decision Tree called the base learner. On the other hand, XGBoost also supports other types of base learners, such as linear models. The base learner is typically a shallow Decision Tree with few nodes, which serves as a weak learner. The model then calculates the gradient of the loss function with respect to the predictions made by the base learner. This gradient represents the direction in which the model needs to update the predictions to reduce the loss. The XGBoost classifier constructs a new Decision Tree to

correct the errors of the base learner. The construction of this tree is done greedily by iteratively adding nodes that minimize the loss function. The tree is built by selecting the best-split point at each node based on the gradient of the loss function. Once the new tree is constructed, the XGBoost classifier updates its predictions by adding the new tree's predictions to the previous trees' predictions. This process is repeated for a fixed number of iterations or until the model converges to acceptable performance. XGBoost includes several regularization techniques to prevent overfitting, such as $L1$ and $L2$ regularization and tree pruning.

The XGBoost predicted value is as given below [24]:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(X_i), f_k \in F, \tag{2.6}$$

where K is the number of Decision Trees, $f_k(X_i)$ is the function of input in the k-th Decision Tree, and F is the set of all possible Classification And Regression Trees (CART).

The loss function of the XGBoost consists of training error and regularization:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{2.7}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2, \tag{2.8}$$

where l is the loss function, n is a number of training examples, T is the number of the leaf nodes, w is vector of leaf weights (scores assigned to each leaf), γ is the leaf penalty coefficient, and λ controls the scale of w.

As the model is trained in an additive way, we can rewrite the loss function as

$$\pounds^{(k)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{k-1} + f_k(X_i)) + \Omega(f_k). \tag{2.9}$$

Using second-order approximation (an estimate of the second derivative of the loss function with respect to each parameter), we can optimize the loss function:

$$\tilde{\pounds}^{(k)} = \sum_{i=1}^{n} [l(y_i, \hat{y_i}^{k-1}) + g_i f_k(X_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_k), \tag{2.10}$$

where $g_i = \partial_{\hat{y}^{(k-1)}} l(y_i, \hat{y_i}^{(k-1)})$ and $h_i = \partial^2_{\hat{y}^{(k-1)}} l(y_i, \hat{y_i}^{(k-1)})$. The solutions for the optimal values of w based on [24] is

$$w_j = \frac{G_j}{H_j + \lambda}, \tag{2.11}$$

Substituting the optimal leaf weights w_j into the approximated loss function yields the minimized loss (also referred to as the *gain score*) for the k-th tree:

$$L = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T, \tag{2.12}$$

where $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$, and I_j is the instance set of leaf j.

This greedy optimization makes XBGoost a fast algorithm but does not necessarily lead to the optimal solution.

2.4. Performance Assessment of the Proposed Cluster-Specific Strategy

The suggested strategy is tested on DataSet1 [3], which was described in Section 1.7.1. Several steps were applied to preprocess this dataset:

- Joining tables. The published dataset is separated into three files. The file containing customer-related information has 2 000 rows, the card-related file contains 6 146 rows, and the transaction-related information file contains more than 24 million rows. After joining everything into one dataset using a left join strategy, it contains more than 24 million rows and 45 features.

- Columns filtering. Columns like Apartment, Merchant State, or Zip were removed because they contained many null values, and it would be complicated to fill them in. Furthermore, they do not bring significant value overall, as the data in other columns provides equivalent information.

- Encoding. Categorical variables with fewer than six unique values were encoded using a one-hot encoding approach, which creates a binary column for each category. The remaining categorical features were encoded using a label encoding technique, where each category was assigned a unique numeric value. It is important

to note that this method is not ideal, as label encoding may unintentionally introduce ordinal relationships between categories, thereby assigning disproportionate weights to the feature values. Further research and experimentation are needed to identify more suitable encoding strategies for these variables.

- Rows filtering. The initial dataset contains information from 1991 to 2020, with a growing number of transactions each year. For the experiment, we selected data from 2014 to ensure a manageable dataset size while still capturing modern transaction patterns relevant for modeling. That year was chosen as a representative point in the data range, balancing recency and computational feasibility.
- Dataset splitting to Train, Validation, and Test. Data from 2014 to 2018, including, was used for training and validation, and data from 2019 to 2020 was used for testing. The dataset for training and validation was split randomly using a 30/70 share. The prepared training set contains 28 features (list of the features can be found in Appendix) and 5 969 329 rows, of which fraudulent cases are 0.125%. This dataset can be called extremely imbalanced. The feature engineering was performed on variables like *Expires_Date* or *Acct_Open_Date* to calculate how many days the card is valid.

2.4.1. Finding the Best Collection of Features and Number of Clusters

After preprocessing, our dataset had different types of features. Some of them, like "Amount" or "Yearly Income - Person" are float; some, like "Current Age" or "Day", are integer, and others like "CardType Debit" or "Gender Male" are binary. Binary features create problems when using the k-means algorithm [99]. Application of k-means clustering and the Euclidean distance for binary data is a controversial topic in the research literature. However, we have chosen this way and the experiments have proved its suitability. We applied standardization to the dataset, re-scaling each feature separately so that it has a mean of zero and a standard deviation of one. This transformation retains the distributional characteristics of the original binary feature values in the standardized form.

There are at least a few approaches to cluster the transactions of the training set into separate subsets. One way of clustering could be based

on intuition and experience. For example, to have clusters of young males with higher credit limits, young females with higher credit limits, etc. It is natural to think that fraud cases could happen to older people with middle or low-level incomes, as shown in Figure 1.6 and Figure 2.3.

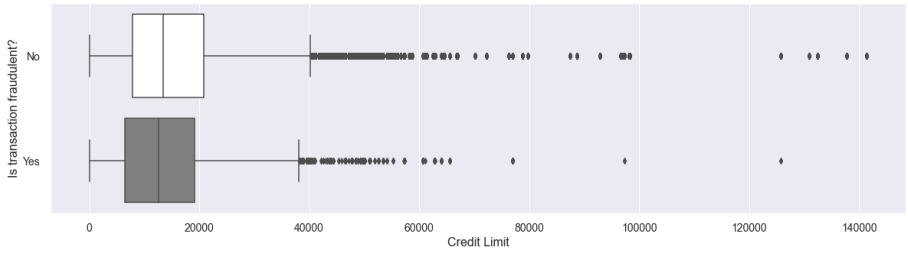


Figure 2.3: Fraudster attacks by Credit Limit

As suggested previously, we are using the Silhouette Score to evaluate the goodness of the clustering. For instance, when splitting clusters by age, gender, and credit limit, we got a Silhouette score equal to 0.3992, which is not very high and implies that clusters' borders are close to each other. We have tried more than 280 combinations of features and a number of clusters using Algorithm 1, and the best one with a score of 0.862248 was *[CardType_Debit, HasChip_YES, Use_Chip_Swipe_Transaction]* with four clusters. An interesting fact is that all three features used for clustering are binary. In our case, features mean:

- *CardType_Debit* feature marks if a transaction was done using a Debit card.
- *HasChip_YES* feature marks if a transaction was done with the card which has a chip. A debit or credit card can have a chip that holds an integrated microchip along with the traditional magnetic stripe. The chip gives customers more security because they are harder to skim.
- *Use_Chip_Swipe_Transaction* feature marks the transactions that are done by swiping the card through the card reader and following its instructions.

To determine the optimal number of clusters for k-means clustering, we employed the *Elbow Method*, a widely used heuristic that evaluates clustering performance based on the distortion score (within-cluster sum

of squares). Specifically, the KElbowVisualizer from the Yellowbrick library was applied to the standardized training dataset. This visualizer fits k-means models with varying numbers of clusters k and plots the resulting distortion scores, allowing for the identification of an inflection point - the *elbow* - where the marginal gain in clustering performance begins to diminish. As shown in Figure 2.4, the elbow occurs at $k = 4$. This indicates that four clusters achieve a suitable balance between model complexity and within-cluster compactness, suggesting an appropriate choice for subsequent clustering analyses.

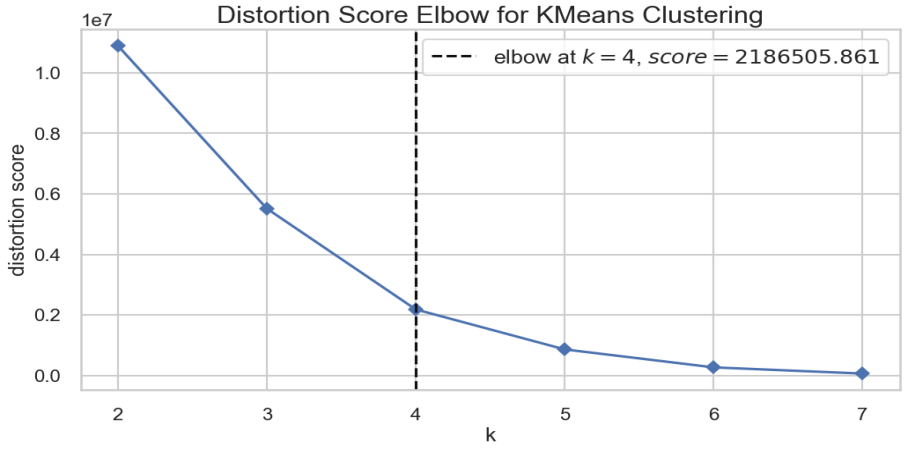


Figure 2.4: *Elbow* method applied to evaluate optimal number of clusters for k-means clustering with features *CardType_Debit, HasChip_YES, Use_Chip_Swipe_Transaction* on DataSet1

In Figure 2.4, distortion score - the mean sum of squared distances to centers - is marked on the y - axis while the number of clusters is on the x-axis. The dotted vertical line marks the *elbow* point found using the "knee point detection algorithm" [118].

After splitting the training set into four clusters, we can notice that they are not equal by size or by the share of the fraudulent cases, as shown in the table below (see Table 2.1).

Since clustering was done based on the three features, it is possible to plot cluster centers in 3D. We can see in Figure 2.5 that one of the clusters is located much further than the others and that this cluster has the lowest number of data points.

Cluster No	Number of data points	Share of fraudulent transactions
1	1 522 231	0.19%
2	596 897	0.11%
3	2 533 953	0.15%
4	1 316 248	0.02%

Table 2.1: Training set clusters characteristics

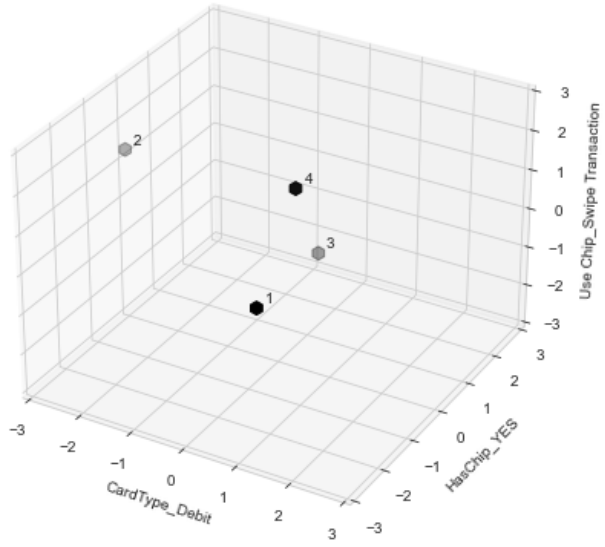


Figure 2.5: Centers of the clusters of the DataSet1

2.4.2. Undersampling and Model Fitting

The XGBoost classifier was chosen as a machine learning model for each cluster. XGBoost - Extreme gradient boosting - is a widely used ML algorithm and usually achieves state-of-the-art results in competitions like Kaggle. It is built on a gradient-boosting decision tree algorithm. XGBoost is a part of the ensemble methods of the supervised ML algorithms family. In this study, default hyperparameters of XGBoost were used for model training, as the primary objective was to evaluate the effectiveness of the overall strategy rather than to optimize model parameters for the given dataset.

The validation set is used to select the best-performing undersampling percentage for each cluster individually. We have chosen undersampling percent, and after resampling, sub-classifiers were trained. We go through all validation dataset points and use one of the sub-classifier for the decision. The criterion for selecting an appropriate classifier is the minimal Euclidean distance between the validation set point and the corresponding cluster center of training data. We measure the best performing undersampling percent for each training cluster by computing the F1 score on the validation data. We repeated this procedure 99 times by checking undersampling percentages from 1 to 99.

We see in Table 2.2 that there is no linear or direct relationship between undersampling percent (the percentage value that is left in the Majority class), the share of fraudulent cases, or the size of the cluster. However, we can see that the worst-performing cluster (C4) has the lowest share of fraudulent cases, and to achieve better results, it requires a low undersampling percentage.

Metrics	C1	C2	C3	C4
Train set size	1 522 231	596 897	2 533 953	1 316 248
Validation set size	653 420	255 489	1 084 892	564 483
Share of fraud (%)	0.19	0.11	0.15	0.02
Undersampling percent	87	91	49	7
Share of fraud in Train set after sampling (%)	0.22	0.12	0.30	0.27
F1 score of Validation set	0.85	0.77	0.82	0.40
Recall of Validation set	0.75	0.63	0.72	0.31

Table 2.2: Cluster-Specific undersampling settings and validation performance metrics

Plotted results (see Figure 2.6) show that the undersampling percent and F1 score do not have a linear dependency, and the F1 score has fluctuations.

Our baseline performance, without applying the training strategy proposed in this section, achieved a recall of 0.69. Despite varying the random seed, the XGBoost classifier produced identical results across runs (see Figure 2.7). This consistency arises because, under default parameter settings, XGBoost operates deterministically. To evaluate the effect of clustering, the training set was divided into four clusters, and sub-classifiers were trained using different random seeds. Within each random seed, we evaluated 99 different values of the undersampling

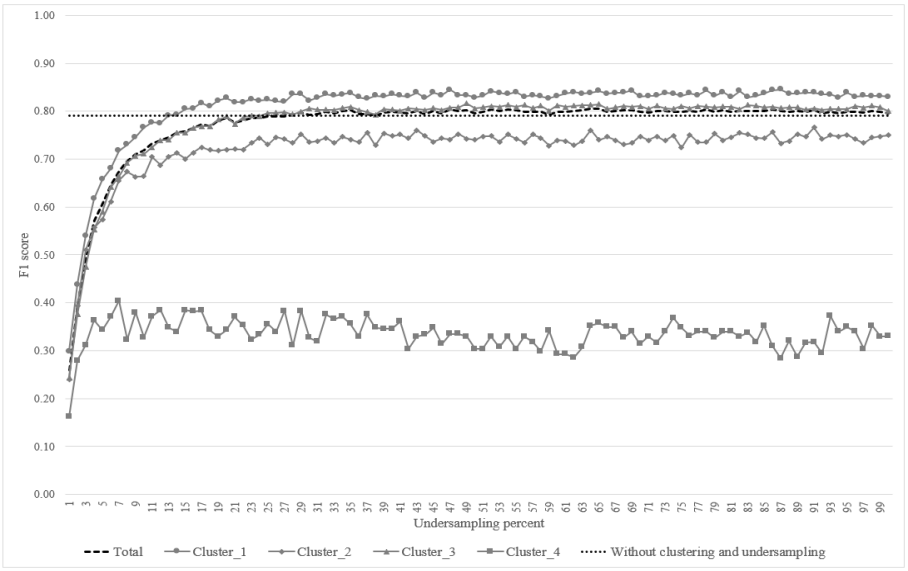


Figure 2.6: F1 score with different undersampling percentage

percentage. The average recall across the different random seeds increased to 0.71. With every ten runs, we got improved results compared to the baseline (see Figure 2.7). There was a slight variation between the results of the runs, although it was negligible.

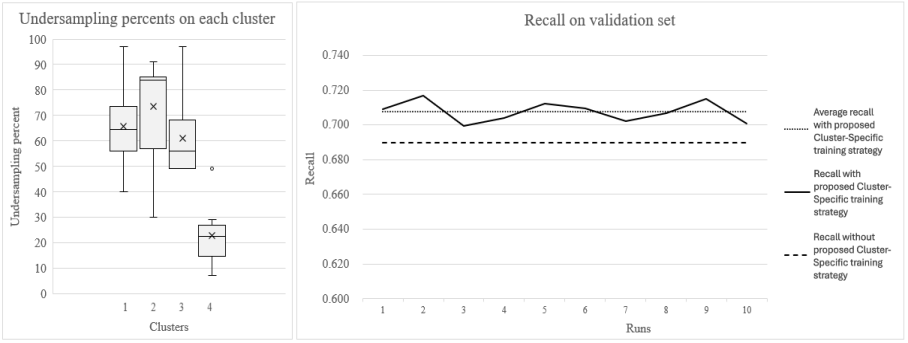


Figure 2.7: Cluster-Specific undersampling rates and *Recall* improvement across different random seeds

Figure 2.7 shows that undersampling percent varies a lot for each cluster with different runs. However, the trend that the cluster with the lowest number of fraudulent cases (in our case, cluster C4) has the lower undersample percent is obvious.

2.4.3. Classification Results

The aim of this strategy is to evaluate whether clustering-based undersampling of the majority class, combined with training separate classifiers for each cluster, can improve classification performance on highly imbalanced datasets compared to standard classification approaches.

The key evaluation step is measuring performance on the test dataset, which represents future fraudulent transactions. The procedure for the test dataset follows the same steps as for the validation dataset. Specifically, the test data are standardized, after which the classifier responsible for making the prediction is selected based on the smallest Euclidean distance between the test instance and the corresponding cluster center from the training data.

To ensure the reliability of the results, predictions were repeated ten times. For comparison, we also calculated the $recall$ on the test dataset without applying the proposed training strategy, thereby establishing a performance baseline.

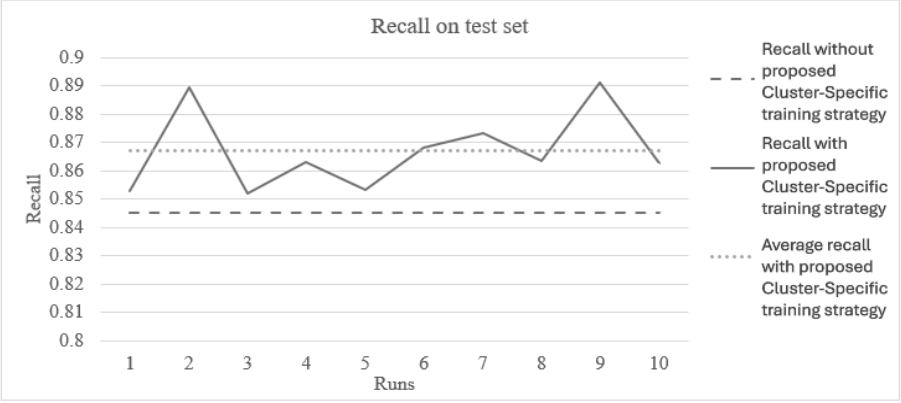


Figure 2.8: Cluster-Specific runs on the Testing set with different random seeds

Figure 2.8 shows that experimental results imply that clustering-based classification with optimal undersampling improved the ML performance. When predicting fraudulent transactions with the XGBoost classifier with no training strategy, the $recall$ is 0.845, and our strategy managed to increase the performance significantly to 0.867.

By comparing the absolute numbers (see Figure 2.9), we see that the

classification of fraudulent cases that were labeled as regular decreased from 323 to 278, i.e. by 13.9%.

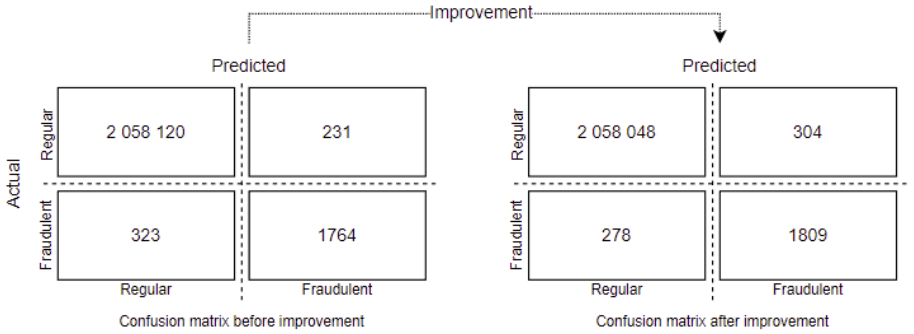


Figure 2.9: Confusion matrix before and after applying the Cluster-Specific training strategy

To evaluate whether the observed differences in classification performance between models were statistically significant, we applied a two-proportion z-test. This classical hypothesis test is appropriate for comparing the proportions of correctly classified instances between two independent models, especially when applied to large datasets. In our context, the test was used to determine whether the improved model detected a statistically significantly higher proportion of fraudulent transactions than the baseline model.

The test assumes that both models made predictions independently on the same test set, which contained n actual fraud cases, and that each prediction (correct or incorrect) follows a Bernoulli distribution. When sample sizes are large, the difference between two sample proportions approximates a normal distribution due to the Central Limit Theorem, justifying the use of the z-test [1].

We formulate the hypotheses as follows:

$$H_0 : \text{TPR}_{\text{baseline}} = \text{TPR}_{\text{improved}}$$
$$H_1 : \text{TPR}_{\text{baseline}} < \text{TPR}_{\text{improved}}$$

where TPR denotes the true positive rate (i.e., the recall for the fraud class). The null hypothesis states that both models achieve equal recall, while the alternative hypothesis posits that the improved model achieves a higher recall. Because our goal was to detect an improvement, we employed a one-tailed test at the 95% confidence level.

The test statistic is computed using the unpooled-variance formulation:

$$Z = \frac{\tilde{p} - \tilde{p}_0}{\sqrt{\frac{\tilde{p}_0(1-\tilde{p}_0)}{n} + \frac{\tilde{p}(1-\tilde{p})}{n}}} \tag{2.13}$$

where $\tilde{p}$ and $\tilde{p}_0$ are the observed proportions of correctly classified fraud cases for the improved and baseline models, respectively, and n is the number of fraudulent instances in the test set.

For comparison, the test statistic can also be expressed using the pooled-variance version as presented by Agresti [1, Ch. 7], which assumes equal proportions under the null hypothesis:

$$Z = \frac{\tilde{p} - \tilde{p}_0}{\sqrt{\frac{2\hat{p}(1-\hat{p})}{n}}} \tag{2.14}$$

where $\hat{p} = \frac{x_1+x_2}{2n}$ is the pooled estimate of the true proportion.

In contrast to the pooled approach, our chosen formulation uses an unpooled variance estimate, which does not assume $p = p_0$. This method calculates variance separately for each model and is considered more conservative, especially when there is no strong reason to assume equal underlying success rates. Patwary et al. [102] showed that unpooled tests tend to result in lower Type I error rates under realistic conditions, making them more reliable for evaluating model improvements without equality assumptions.

In our case,

$$\tilde{p_0} = 1764/2087 = 0.8452$$
$$\tilde{p} = 1809/2087 = 0.8668$$
$$Z = 1.9849$$

Using Z-score table with α = 0.05, we have p value = 0.0256 (Z-Table). In this case, we can conclude that the obtained increase in the classifier performance is significant.

While McNemar's [92] test is widely recognized as the appropriate choice for comparing the performance of two classification models evaluated on the same test instances, particularly when the focus lies on pairwise prediction disagreements, our objective was different. In this work, we employed the two-proportion z-test to assess whether

the improved model achieved a statistically significantly higher true positive rate (i.e., recall for the fraud class) than the baseline model. Rather than analyzing the symmetry of disagreements, we were primarily interested in detecting an improvement in the overall proportion of correctly identified fraudulent transactions.

Nevertheless, we acknowledge that since both models were evaluated on an identical test set, McNemar's test would offer a complementary perspective, particularly for evaluating the statistical significance of per-instance prediction differences. Incorporating McNemar's test as an additional robustness check is a valuable direction for future work.

2.5. Conclusions of the Chapter

Fraud detection is an activity that prevents fraudsters from obtaining financial assets. The goal of the research is to increase the quality of ML predictions in fraudulent cases and to decrease false negative cases in prediction. Fraudulent data, such as credit card transactions, are imbalanced data. In this case, standard ML algorithms cannot reach the expected levels of quality. This chapter investigates and proposes a Cluster-Specific classification strategy to improve $recall$. $Recall$ was chosen as it is crucial in detecting credit card fraud since it evaluates a model's capability to correctly identify particularly the fraudulent activities. The idea in the proposed approach lies in the undersampling of each cluster and further training the sub-classifiers by the undersampled data. It means that each ML model can be created separately based on the cluster data. Which sub-classifier will be activated to predict the label depends on the Euclidean distance of the particular unseen data point to the training set cluster center. One of the sub-classifiers makes the decision on the dependence of a particular transaction on a regular or fraudulent class. For the experimental evaluation, we use a credit card transaction database. Our baseline $recall$ is 0.845, obtained after the direct training of the XGBoost classifier. By applying the proposed approach, we improved the $recall$ to 0.867. Moreover, the classification of fraudulent cases that were labeled as regular decreased from 323 to 278, i.e., by 13.9%, which is significant. Furthermore, we found that the prediction becomes higher when the training set is properly split into clusters and balanced separately for each cluster.

From a global perspective, these results suggest that clustering can

serve as a powerful pre-processing step in fraud detection pipelines. By ensuring that each cluster is adequately balanced before training, the model gains a better understanding of fraudulent patterns within specific transaction segments. This insight opens avenues for exploring more sophisticated clustering methods, such as density-based clustering or deep clustering techniques, which could further enhance model performance by identifying nuanced patterns within fraudulent data. Additionally, feature encoding and selection play a pivotal role in optimizing fraud detection systems. The research should focus on identifying the most informative features for classification. Feature encoding techniques, such as target-based or target-agnostic, can further refine cluster definitions and improve the discriminatory power of sub-classifiers.

Overall, the results of this chapter prepare the way for future advancements in fraud detection methodologies by emphasizing the importance of data-driven pre-processing strategies, such as clustering, feature selection, and encoding. These insights can significantly contribute to the ongoing efforts to combat financial fraud and develop more efficient, scalable, and interpretable machine-learning solutions.

3. CATEGORICAL FEATURE ENCODING FOR IMPROVED CLASSIFIER PERFORMANCE WHEN DEALING WITH IMBALANCED DATA OF FRAUDULENT TRANSACTIONS

Fraudulent transaction data tend to have several categorical features with high cardinality. Data preprocessing becomes complicated if categories in such features do not have an order or meaningful mapping to numerical values. Even though many encoding techniques exist, their impact on highly imbalanced massive datasets is not thoroughly evaluated. Therefore, it is necessary to investigate the influence of different encoding methods, such as target encoding or One-Hot encoding, on model performance in the context of fraud detection. Particularly in highly imbalanced datasets, where fraudulent cases represent only a small fraction of the data, improper encoding can either introduce noise or cause models to overfit.

In this chapter, we aim to systematically evaluate encoding strategies tailored to high-cardinality categorical features in large-scale imbalanced datasets, using fraud detection as a representative case. Our goal is to identify robust preprocessing pipelines that minimize bias and support improved detection of rare fraudulent activities.

Parts of this chapter are published in the international research journal with a citation index in the Clarivate Web of Science (CA WoS) database [A.1]. The results were presented at the international conference [D.1], as well.

3.1. Overview of Feature Encoding Techniques Used for Comparison

Most ML algorithms are built for numerical data. Hence researchers and developers must decide how to encode categorical variables. Various encoding techniques exist for this purpose. They can be grouped based on their relation to the target. Namely, target-based and target-agnostic. Target-based encoding methods use information from the target variable when transforming the categorical feature, for example, replacing categories with the mean of the target variable for each category. This allows them to potentially capture useful patterns but can also lead to target leakage if not used carefully. In contrast, target-agnostic encoding methods, such as One-Hot or Hashing encoders, do not use any information from the target variable and instead focus on representing the

categories in a purely data-driven way. Another way to group encoding techniques is based on their impact on dataset dimensionality. Encoders like One-Hot or Hashing encoders are the ones that increase the dimensionality of the dataset. This section analyzes four target-based and two target-agnostic techniques presented in Figure 3.1.

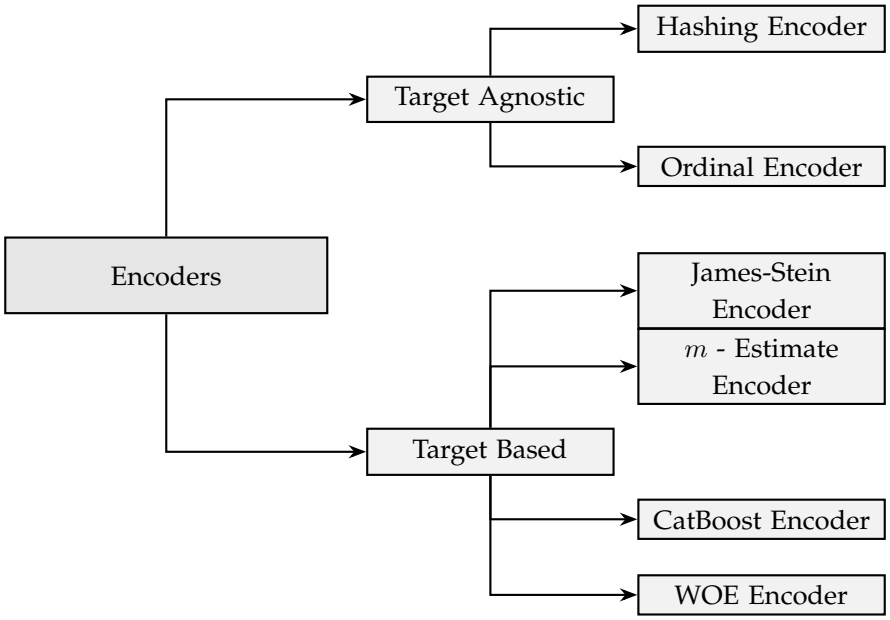


Figure 3.1: Hierarchical representation of various types of encoders.

3.1.1. m-estimate Encoder

The m-estimate encoder (*mee*) is a target-based encoder. It has one hyperparameter - m, representing the power of regularization where a higher value of m results in stronger shrinking. Recommended values for m are in the range of 1 to 100. The formula to compute estimated values for a category is [93]:

$$S_i = \frac{n_{iY} + p_{prior} m}{n_i + m}, \tag{3.1}$$

where S_i is the encoded value for category i; n_i is the number of times the category i appears in the dataset; n_{iY} is the number of times the binary target has value 1 ($Y = 1$) when the category is i; p_{prior} is a prior probability of $Y = 1$ without considering categories.

3.1.2. James-Stein Encoder

The James-Stein encoder (*jse*) is a target-based encoder as well. Initially, the James-Stein estimator was not meant to be used for binary classification and was defined only for normal distributions. In our case, we want to apply it for binary classification, so firstly, we convert the mean target value to the log-odds ratio.

The James-Stein encoder is a method for shrinking mean estimates only when the variances of those means are assumed to be equal. However, this assumption is often only valid when the sample sizes of each group are equal. In most real-world scenarios, sample sizes and variances of the means are not equal, which makes it difficult to determine the appropriate course of action. For the execution of the James-Stein encoder, we use the Scikit-learn library *Category Encoders*, which has implemented a binary version of the James-Stein encoder proposed in the paper [151].

3.1.3. CatBoost Encoder

The CatBoost encoder (*cbe*) [107] uses the same formula as the *m*-estimate encoder. However, the critical difference lies in how the encodings are computed to prevent target leakage. The *m*-estimate encoder uses the entire dataset, including the current row, which can lead to information leakage from the target variable into the features. To address this, Prokhorenkova et al. [107] proposed a method within the CatBoost algorithm that avoids such leakage by using permutations of the training data. The idea is to compute the category statistics for each row using only the preceding rows, ensuring that the target value of the current observation does not influence its own encoding. For execution, we use the Scikit-learn library *Category Encoders*, where the implementation is time-sensitive (it does not use random permutation).

3.1.4. Weight of Evidence Encoder

The Weight of Evidence encoder (WOE) [52] is a statistical measure that quantifies the strength of the relationship between a categorical variable and a binary target variable. The WOE encoder for a particular category is calculated by taking the natural logarithm of the ratio of

the percentage of observations in that category that belong to the class $Y = 0$ to the percentage of observations in that category that belong to the class $Y = 1$.

$$WOE_i = ln\frac{p_{i(Y=0)}}{p_{i(Y=1)}}, \tag{3.2}$$

where $p_{i(Y=0)}$ is percentage of $Y = 0$ when the category is i; $p_{i(Y=1)}$ is percentage of $Y = 1$ when the category is i.

3.1.5. Ordinal Encoder

The Ordinal Encoder (*oe*) is a target-agnostic method that assigns integer values to each unique category in a feature, typically ranging from 0 to $k - 1$, where k is the number of distinct categories. It does not use any information from the target variable. This method is simple and efficient but introduces an artificial ordering among categories, which may not reflect any true relationship. Such ordering can negatively impact models that interpret numeric input as ordered or continuous.

A similar technique is the Label Encoder, which also maps categorical values to integers. The key difference is in their typical applications:

- Ordinal Encoder is designed for input features where a meaningful order between categories exists (e.g., "low", "medium", "high").
- Label Encoder is typically used for target variables in classification tasks, converting class labels to integers without implying order.

Although both Ordinal Encoder and Label Encoder assign integer values to categorical data, they are implemented differently in scikit-learn. Notably, only Ordinal Encoder supports handling unseen categories via dedicated parameters, whereas Label Encoder will raise an error if an unknown category is encountered during prediction.

3.1.6. Hashing Encoder

Feature hashing is a technique for converting categorical features into numerical features for ML models. It works by mapping each category of a categorical feature to an integer within a pre-determined range. The output range is usually smaller than the input range, meaning multiple

categories may be mapped to the same integer. Such conditions are called collisions. However, collisions are often rare in practice and do not significantly affect performance.

Feature hashing (*he*) is similar to One-Hot encoding but with a few key differences. One of the main advantages of feature hashing is that it allows for control over the output dimensions. Additionally, feature hashing can be faster and more memory-efficient than One-Hot encoding, especially when dealing with large datasets.

This encoder applies the hashing trick [143] to a categorical feature and then encodes the resulting integers as numerical features. The basic idea behind the hashing trick is to use a hash function to map the input data to a fixed-size output space. The hash function takes the input data (e.g., a word or categorical feature) as input and produces a hash value that is an integer between 0 and a predefined maximum value.

3.2. Overview of Machine Learning Algorithms Used for Comparison

To evaluate the impact of encoding techniques on the classification algorithms, we select different models in terms of framework, used loss function, regularization, complexity, and speed. We compare ensemble learning models with non-linear and linear models. Ensemble learning can be visually explained as a judgment of the crowd when the decision is taken by voting. A real-life example of a crowd decision can be a famous TV show named "Who Wants to be a Millionaire". The idea of the show was to answer fifteen questions in a row correctly and win one million dollars. The participant had a chance to ask for help from an intelligent friend or the audience. The intelligent friend was right almost 65% of the time. Unexpectedly, the audience of random people was correctly answering 91% of the time [127]. Ensemble learning can improve the efficiency of a model compared to a single model by reducing the risk of overfitting and underfitting when combining multiple models. It is also less sensitive to outliers, and noise [115]. Ensemble learning has a subgroup called gradient-boosting, with examples like XGBoost, LightGBM, and CatBoost. Ensemble learning is usually built on Decision trees.

3.2.1. Decision Tree

The abbreviation CART is used for "Classification And Regression Trees," which was introduced by Breiman [14]. CART is a Decision Tree algorithm that recursively partitions data into smaller subsets, represented by nodes, with the final subsets being represented by leaf nodes. For each partition, the best splitting feature is selected. This algorithm typically employs Shannon entropy [121] and the Gini index [14] to identify the best feature to split data.

Shannon entropy quantifies the degree of uncertainty (impurity) in a dataset. In the context of classification, a partition with low entropy is considered relatively pure, meaning that the majority of the samples share the same label. In contrast, a partition with high entropy indicates that the class labels are more evenly distributed, with no clear majority class.

$$Entropy(D) = -\sum_{i=1}^{c} p(i) \log_2 p(i), \tag{3.3}$$

where $Entropy(D)$ is the Shannon entropy of dataset D, c is the number of classes, and $p(i)$ is the probability that a randomly chosen sample from D belongs to class i. If D is fully pure (i.e., it contains only one class), then the entropy equals zero. Information Gain (IG) is then used to determine whether a given split leads to a decrease in overall entropy:

$$IG = Entropy(D) - \sum_{j=1}^{k} \frac{n_j}{n} Entropy(D_j), \tag{3.4}$$

where k is the number of partitions (child nodes) created by the split, n_j is the number of samples in subset D_j, and n is the total number of samples in D. Each D_j is the subset of D containing all samples that share the same value (or fall within the same interval, in the case of continuous features) for the splitting feature. A greater reduction in entropy corresponds to higher IG, indicating a better split point.

The Gini index [14] is another measure for evaluating the purity of a split:

$$Gini(D) = 1 - \sum_{i=1}^{c} p^2(i), \tag{3.5}$$

where $p(i)$ is the probability that a randomly chosen sample from D belongs to class i. The Gini index equals zero when one class has probability 1 and all others have probability 0. For a given split, the *weighted Gini index* is calculated as:

$$wGini = \sum_{j=1}^{k} \frac{n_j}{n} Gini(D_j), \tag{3.6}$$

where k is the number of partitions (or subsets), n is the total number of samples in D, n_j is the number of samples in subset D_j, and $Gini(D_j)$ is the Gini impurity of subset D_j. A lower weighted Gini value indicates a better split.

Decision trees can grow very large when applied to large datasets, which often leads to overfitting. To mitigate this, one can specify a minimum number of samples required in a leaf node or set a maximum tree depth. Another technique, known as pruning, prevents the tree from growing to its full depth by removing nodes or branches that do not significantly improve predictive performance. Pruning produces a smaller and simpler tree that is less likely to overfit and more likely to generalize well to unseen data. The main advantages of decision trees are their interpretability, ability to handle both numerical and categorical features, and competitive predictive performance on tabular datasets.

3.2.2. Random Forest

Breiman introduced a RF algorithm in 2001 cite breiman2001random. It is one of the most widely used ensemble learning algorithms, primarily due to its simplicity and predictive power. Fernández-Delgado et al [46] showed that RF can beat other classifiers from 17 families under different kinds of problems by using 121 databases from UCI. On the other hand, this research does not provide insights and experiments on highly imbalanced datasets.

RF is a machine-learning algorithm that combines multiple Decision Trees to make a final prediction. The number of Decision Trees in a RF is a hyperparameter that can be set before training the model. Each Decision Tree in the RF is trained on a random subset of the training data and a random subset of the features. This process is repeated multiple times to create a diverse set of Decision Trees. The final result is obtained during prediction by aggregating the predictions of all the

individual trees in the forest. The aggregated prediction is either by taking the majority of individual predictions (for classification) or the mean of the predicted values (for regression).

3.2.3. LightGBM - Light Gradient Boosting Machine

Gradient Boosting Decision Trees (GBDT) is a machine learning algorithm combining Decision Trees and gradient boosting to create an ensemble model. Gradient boosting iteratively improves a model's predictions by adding new models to the ensemble, each focusing on previously misclassified examples. GBDT faces challenges when dealing with large data samples, and they can require a large amount of memory, especially when the number of features or trees is high. For each feature, GBDT requires scanning through all data instances to calculate the IG for every potential split point.

Reducing the number of data instances or features seems like a simple solution to address this issue. However, it is not a trivial task. No weight is assigned to the data instance in the GBDT, and the gradient of the loss function is used to update the model in each iteration instead. Data instances with more significant gradients have a more considerable impact on constructing the Decision Tree. It means that they also have a more significant influence on the computation of IG, even though no exact weight is assigned to each data instance. This conclusion is one of the prominent uniqueness of the LightGBM [75].

Thus, when undersampling the data instances, we should better keep those instances with large gradients to maintain the accuracy of IG estimation and only randomly drop those instances with slight gradients. The paper [75] proves that the mentioned strategy can increase IG estimation accuracy better than uniformly random sampling. This approach is called Gradient based One Side Sampling.

Additionally, LightGBM implemented Exclusive Feature Bundling (EFB) algorithm. The authors design an efficient algorithm to solve the optimal bundling problem by reducing it to a graph coloring problem and solving it using a greedy algorithm with a constant approximation ratio which means that the solution it produces is always within a constant factor of the optimal solution.

LightGBM can encode categorical features inside the algorithm. However, in this research, we do not use this option and instead feed

already encoded data to achieve the research goal.

3.2.4. CatBoost - Category Boosting

CatBoost is another boosting algorithm released in 2017 [38] after XGBoost and LightGBM [107]. CatBoost can automatically handle categorical features by combining One-Hot and integer encoding if needed. It also uses target encoding to deal with high-cardinality categorical features. However, the novelty of this method is that it addresses and suggests solutions for solving the prediction-shifting problem. The solution is called ordered boosting.

CatBoost addresses prediction shifts by creating new datasets at each boosting step, which are independent of the previous datasets, to obtain unshifted residuals. This is accomplished by applying the current model to new training examples. No instances may be used for training the previous models to ensure unbiased residuals for all training examples. In this case, CatBoost maintains a set of models that differ in the examples used for training. When calculating the residual for a particular example, CatBoost uses a model that was trained without that example. The random permutation of the training examples is used to achieve this.

3.3. Impact Assessment of Feature Encoding Methods

In this research, we focus on large-scale datasets with an imbalance ratio below 1% and categorical features of high cardinality. For the analyses presented here, we rely on the two datasets described in Chapter 1: the first, hereafter referred to as DataSet1, and the second, generated using the Sparkov Data Generation tool [58], hereafter referred to as DataSet2.

Our goal is to find the best-fitting categorical encoder for highly imbalanced massive data, so we are not hyper-tuning selected ML models or changing thresholds. We calculate results using cross-validation with a stratified split of five-fold. We have performed the cross-validation four times with different seeds. We believe that by using the Grid Search algorithm, we could achieve better results in general. For the encoding algorithms, we use default parameters as well. Our focus is on univariate encoding, where each feature is always encoded separately.

It is assumed that the choice of the encoding method significantly

impacts classification performance on imbalanced datasets in financial fraud detection. To test this assumption, we conduct an experiment to analyze and compare various encoding techniques. Although LightGBM and CatBoost include their own internal encoding mechanisms as part of their optimization strategies, we do not compare these built-in methods with external encoding approaches, as the results would not be directly comparable.

Both datasets have categorical features with different cardinality. The cardinality of each categorical feature is presented in Table 3.1.

Metric	**DataSet1**	**DataSet2**
Training set size	5 969 329	907 672
Categorical feature	**Cardinality**	**Cardinality**
Card Brand	4	–
Card Type	3	–
Has Chip	2	–
Use Chip	3	–
Merchant City	11 391	–
MCC	109	14
Error1_cat	8	–
Error2_cat	5	–
Gender	2	2
City	1 074	894
State	51	51
Job	–	494

Table 3.1: Comparison of categorical feature cardinality between DataSet1 and DataSet2

Additionally, we present an example of differences in encoding technique performance by plotting histograms of encoded values of the categorical feature "State" from DataSet1 (Figure 3.2 - Figure 3.6). The x-axis represents encoded values, and the y-axis shows the number of cases of the appearance of the particular encoded value. The maximum histogram of the Label encoder is much lower than that of the CarBoost encoder. This means that the CatBoost encoder shrinks categorical values, resulting in a much higher number of cases for a particular encoded value.

Upon analyzing the target-based techniques, it is evident that the encoded values display notable variability in terms of their size and

shape, as represented in the histograms. More specifically, the James-Stein encoder demonstrates a compact range of values, while the WOE encoding results in a roughly balanced distribution, though slightly skewed to the negative side. Additionally, the CatBoost encoder produces a highly concentrated distribution of values near zero, with a long right tail, as illustrated in Figure 3.5. The inset histogram highlights this skewness more clearly.

Target-agnostic techniques are not comparable in our case, as the Ordinal encoder does not expand dataset dimensionality while Hashing encoder does. The histogram of the encoded variable "State" with an Ordinal encoder represents the frequency of the values encoded with no logical order.

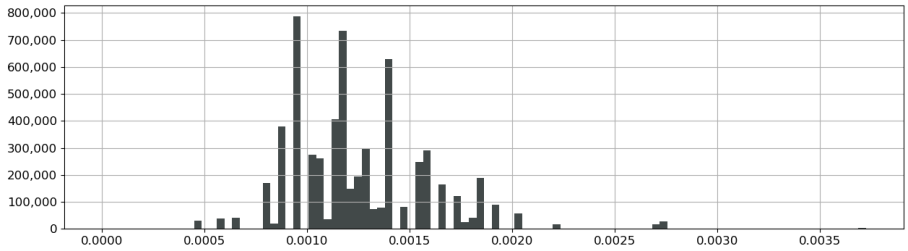


Figure 3.2: Distribution of *State* values after applying the m-estimate encoder

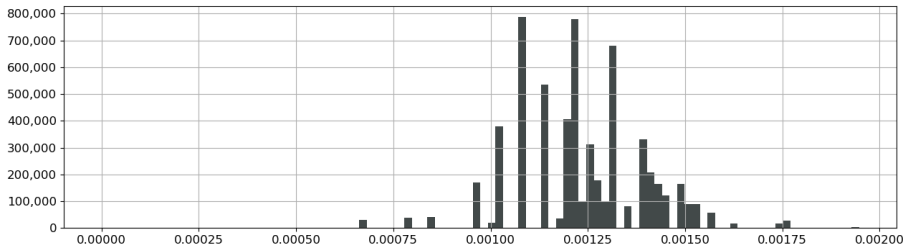


Figure 3.3: Distribution of *State* values after applying the James-Stein encoder

Visual comparison of the feature "State" encoded with different encoders can be challenging, owing to their differing scales. In an attempt to mitigate this challenge, we opt to standardize the encoded values by scaling them within the range of zero and one. Following, we plotted the density function, as depicted in Figure 3.7. This visualization allowed

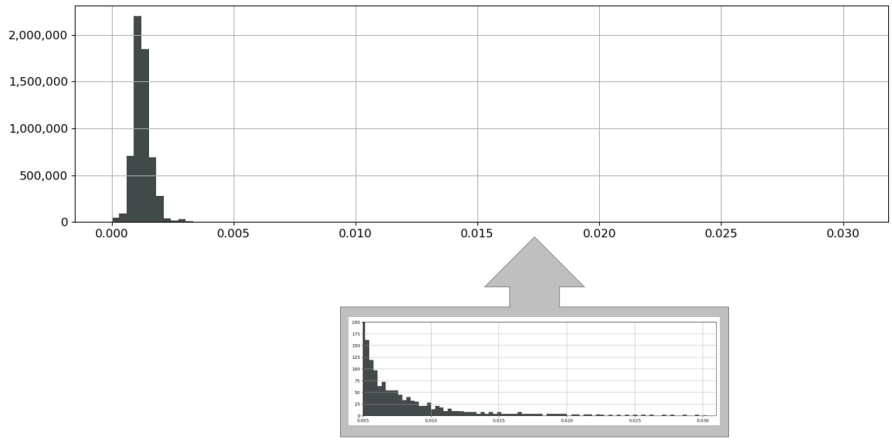


Figure 3.4: Distribution of *State* values after applying the CatBoost encoder

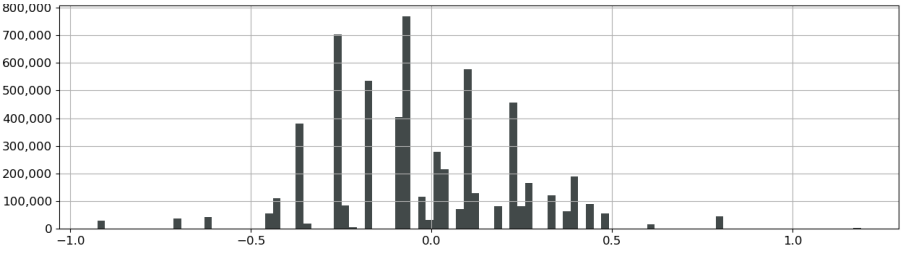


Figure 3.5: Distribution of *State* values after applying the WOE encoder

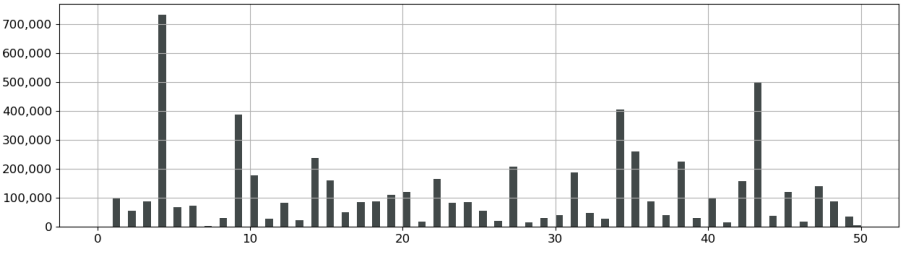


Figure 3.6: Distribution of *State* values after applying the Ordinal encoder

us to easily discern that the James-Stein, m-estimate, and WOE encoders demonstrate similar shapes and density amplitude. However, their primary variation lies in their position along the x-axis. Conversely, values encoded with the CatBoost encoder are characterized by a significant level of skewness, as previously noted. Moreover, we observed that the encoded values with CatBoost appear to be compressed, as evident from the density function plot.

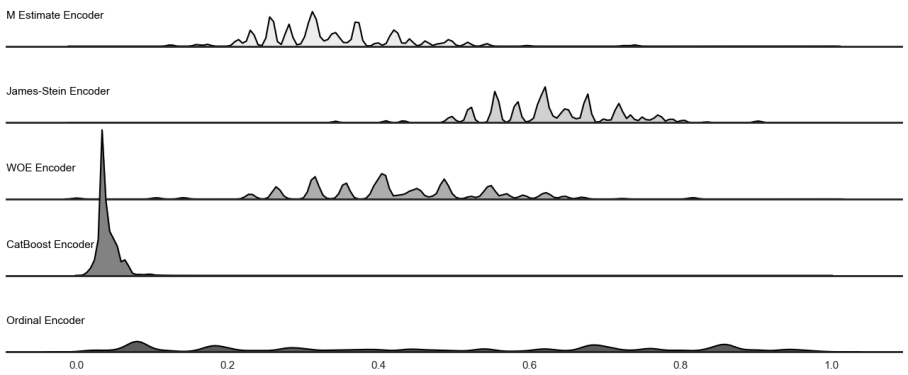


Figure 3.7: Density plots of the normalized encoded feature *State* using different encoding methods

The comparative analysis of encoding techniques across the mentioned datasets reveals distinct performance patterns, underscoring the importance of encoder selection in imbalanced classification tasks. In DataSet1, the James-Stein encoder demonstrates the strongest performance overall (see Table 3.2), achieving the highest mean F1-score of 0.8135 ± 0.0588 with CatBoost and also performing competitively across other classifiers. The m-estimate and WOE encoders similarly exhibit stable and relatively high F1-scores, particularly when used with XGBoost (0.8108 and 0.8006, respectively). In contrast, encoders such as the CatBoost encoder and the Hashing encoder show notably lower mean scores and higher variance, indicating instability and reduced effectiveness. DataSet2 presents a different landscape, where all encoders perform substantially better on average, but the James-Stein and WOE encoders emerge as the top performers. The m-estimate encoder also performs consistently well across models in DataSet2, often achieving mean F1-scores above 0.8.

Dataset	Encoder	CART	CatBoost	LGBM	RF	XGB
DataSet1	jse	**0.7119** ± 0.0056	**0.8135** ± 0.0588	**0.6579** ± 0.0489	0.6609 ± 0.0117	0.8092 ± 0.0063
	woe	0.6998 ± 0.0066	0.7858 ± 0.0102	0.5938 ± 0.0402	**0.6714** ± 0.0126	0.8006 ± 0.0100
	mee	0.7117 ± 0.0058	0.8055 ± 0.0072	0.5517 ± 0.1068	0.6587 ± 0.0138	**0.8108** ± 0.0100
	cbe	0.3486 ± 0.1698	0.5336 ± 0.1270	0.2864 ± 0.1185	0.5120 ± 0.0755	0.5804 ± 0.0980
	oe	0.6887 ± 0.0051	0.7997 ± 0.0084	0.4796 ± 0.1263	0.5568 ± 0.0118	0.7884 ± 0.0073
	he	0.2110 ± 0.0099	0.3710 ± 0.0091	0.1303 ± 0.0143	0.1642 ± 0.0107	0.2186 ± 0.0077
DataSet2	jse	0.8089 ± 0.0107	**0.8954** ± 0.0073	0.7538 ± 0.0305	0.8582 ± 0.0083	0.8997 ± 0.0075
	woe	0.8061 ± 0.0096	0.8951 ± 0.0080	0.7386 ± 0.0464	**0.8589** ± 0.0069	**0.9028** ± 0.0075
	mee	**0.8097** ± 0.0099	0.8951 ± 0.0095	0.7600 ± 0.0217	0.8581 ± 0.0083	0.9017 ± 0.0081
	cbe	0.7001 ± 0.1445	0.8167 ± 0.1067	0.6525 ± 0.0597	0.8028 ± 0.0600	0.8266 ± 0.0610
	oe	0.8012 ± 0.0126	0.8923 ± 0.0091	**0.7711** ± 0.0235	0.8508 ± 0.0088	0.9028 ± 0.0098
	he	0.6161 ± 0.0088	0.7604 ± 0.0081	0.6246 ± 0.0120	0.7070 ± 0.0056	0.7648 ± 0.0094

Table 3.2: F1-score means and standard deviations (mean ± std) for each encoder and classifier across DataSet1 and DataSet2

Visual analysis from the boxplot (see Figure 3.8) further confirms these observations, with tighter interquartile ranges and higher medians for the James-Stein, WOE and m-estimate encoders in both datasets. These results show that target-based encoding methods outperform target-agnostic approaches across both datasets. Although the Ordinal encoder presents competitive results, there is a significant variation in the outcomes. Incorporating label information during the encoding process enables machine learning models to capture the underlying relationships between categorical features and the target variable more effectively, which in turn significantly enhances predictive performance. Furthermore, Table 3.2 reveals that LightGBM is particularly sensitive to the choice of encoding method, as shown by the substantial variability in its F1-scores across encoders. While LightGBM achieves competitive performance with encoders such encoders as James-Stein and Ordinal, its performance drops considerably when paired with CatBoost encoder, Hasing, or even WOE encoders, especially in DataSet1. This variation underscores the critical role that encoding method selection plays in achieving optimal results with LightGBM.

Considering the generalization of the findings, the average F1 score is assessed within each ML category, including Boosting (XGBoost, CatBoost, LGBM), Ensemble (RF), and Non-linear (Decision Tree). The results are displayed in Table 3.3. This table presents the maximum, average, and standard deviation of F1-scores for six different encoders evaluated across three model groups: Boosting, Ensemble (RF), and Non-linear (CART), on two datasets. Among the encoders, James-Stein

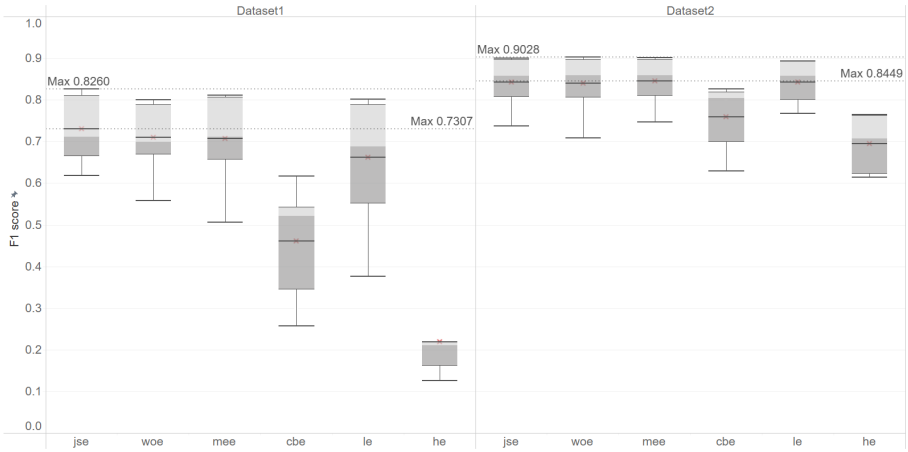


Figure 3.8: F1-score distribution across encoders and datasets (DataSet1 vs. DataSet2)

consistently delivers the highest average F1-score across all three classifier groups, with particularly strong results in the Boosting category (0.8049) and low variability (0.0899). The m-estimate and WOE encoders also show competitive performance, achieving average F1-scores above 0.75 across all classifier types, while maintaining relatively low standard deviations. In contrast, the CatBoost and Hashing encoders yield the lowest average performances and exhibit the highest standard deviations, particularly in the Boosting and Ensemble groups, indicating substantial variability and instability. Notably, the Hashing encoder reaches an average F1-score of only 0.4786 in Boosting and 0.4136 in Non-linear models, with standard deviations exceeding 0.20, which highlights its limited suitability for classification tasks in the context of imbalanced data with high-cardinality features. Overall, the results emphasize that target-based encoders such as James-Stein, WOE, and m-estimate offer more robust and reliable performance across various classifier types, with Boosting algorithms benefiting the most from high-quality encoding strategies.

3.4. Conclusions of the Chapter

This chapter aims to determine the most appropriate encoding technique for handling highly imbalanced datasets. We conducted an extensive

Encoder	Boosting			Ensemble			Non-linear		
	Max	Avg.	Std. Dev.	Max	Avg.	Std. Dev.	Max	Avg.	Std. Dev.
jse	0.9107	**0.8049**	0.0899	0.8708	0.7595	0.1004	0.8257	**0.7604**	0.0498
woe	0.9119	0.7861	0.1076	0.8674	**0.7651**	0.0955	0.8246	0.7529	0.0544
mee	0.9118	0.7875	0.1253	0.8685	0.7584	0.1016	0.8235	0.7607	0.0503
cbe	0.8830	0.6221	0.2145	0.8518	0.6619	0.1577	0.7909	0.5244	0.2377
oe	0.9092	0.7706	0.1487	0.8699	0.7074	0.1528	0.8197	0.7449	0.0578
he	0.7800	0.4786	0.2541	0.7185	0.4356	0.2750	0.6340	0.4136	0.2054

Table 3.3: Maximum, average, and standard deviation of F1-scores for each encoder across Boosting, Ensemble, and Non-linear classifiers

experiment to evaluate six encoding methods, categorized into target-agnostic and target-based approaches. The study incorporated various ML methods, including ensemble learning techniques, as well as linear and non-linear models. The primary focus was on transaction datasets, where the target variable distinguished between regular and fraudulent transactions. These datasets posed significant challenges due to their complexity and the presence of several high-cardinality features. The complete list of features for datasets are available in Appendix 4.4.

While a number of studies have explored encoding techniques, most of the studies have relied on publicly accessible balanced datasets, which may not reflect the challenges posed by highly imbalanced real world scenarios. This chapter contributes to filling this gap by specifically investigating the performance of encoding techniques in the context of imbalanced datasets.

The findings, presented in Figure 3.8 and Table 3.2, underscore the critical role of selecting an appropriate encoding method when working with imbalanced datasets and ML models. Our analysis demonstrates the following key insights:

- Importance of Target-Based Encoding Methods. The results reveal that target-based encoders, particularly James-Stein and Weight of Evidence (WOE) encoders, often outperform their counterparts in improving model performance. These methods effectively leverage the relationship between features and the target variable.

- Challenges with High-Cardinality Features. Encoding high-cardinality features, a common characteristic of transaction datasets, remains a significant challenge. Techniques such as hash-

ing and One-Hot encoding can intensify the curse of dimensionality, negatively impacting the performance of ML models. Hash Encoding has the lowest performance, with F1 score of 0.4786 for Boosting, 0.4356 for Ensemble, and 0.4136 for Non-linear, supporting the notion that encoders lacking target information are generally less efficient in predictive modeling.

- Limitations of the CatBoost Encoder. Contrary to expectations, the CatBoost encoder, which is tailored for categorical data, exhibited suboptimal performance in imbalanced datasets. This highlights the need for careful evaluation of encoding techniques in the context of specific data distributions. Further investigation is needed to understand better the CatBoost encoder's poor performance in scenarios involving both class imbalance and high-cardinality categorical features.

- Diverse Impact Across Algorithms. The results revealed that certain encoding techniques consistently outperformed others when paired with specific ML algorithms. For example, target-based encoders such as James-Stein and WOE achieved the highest F1-scores when used with gradient boosting models, while tree-based algorithms like CART and RF were more sensitive to encoding choices. These findings underscore that the interaction between the encoding method, algorithm architecture, and data characteristics can significantly influence model performance, highlighting the importance of selecting encoding strategies that are well aligned with both the learning algorithm and the nature of the dataset.

SOM [78], often called Kohonen map, is a powerful unsupervised ML technique that falls under the category of artificial neural networks. Developed by Finnish professor Teuvo Kohonen in the 1980s, SOMs are used for dimensionality reduction, data visualization, clustering, and pattern recognition tasks. Despite the method being created at the end of the 20th century, it is still widely used for many actual applications. For example, its combination with multidimensional scaling [41], [40] enlarged its possibilities to understand patterns in data. The fundamental concept behind SOM is to map high-dimensional input data onto a lower-dimensional grid while preserving the topological relationships between data points.

Several parts of this Chapter has been published in [A.2]. The results were presented in [D.1] and [E.1] conferences. This chapter introduces a new method called Feature Selection for Imbalanced Data Using SOM (FID-SOM), which leverages the weights abilities of a SOM for feature selection.

4.1. FID-SOM: Feature Selection for Imbalanced Data

This section introduces the proposed FID-SOM method, a feature selection method specifically designed for imbalanced financial fraud detection. The core idea of the method is to perform feature selection through a process of feature conversion using SOM. Unlike traditional feature selection approaches that operate directly on the original feature space, FID-SOM transforms the data into a topological representation using SOMs, enabling the identification of features that contribute most to distinguishing between classes. As illustrated in Figure 4.1, the algorithm begins by encoding (see Chapter 3) and standardizing the input data, followed by training a SOM on the transformed feature space. Once SOM is trained, each data instance is mapped to its corresponding BMU, and the associated neuron's weight vector is used as a new representation of the instance. These BMU weight vectors form the basis for evaluating the relevance of each feature. Feature importance scores are then computed based on its variance.

In the FID-SOM framework, the Self-Organizing Map consists of a two-dimensional grid of nodes (neurons) arranged in a rectangular

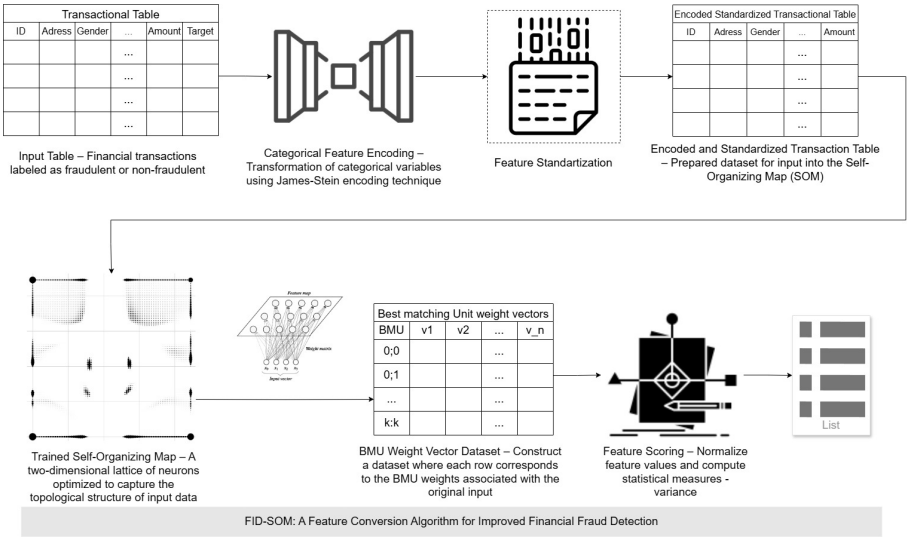


Figure 4.1: Novel method FID-SOM of feature selection for imbalanced data using SOM

layout. Each neuron is associated with a weight vector that has the same dimensionality as the input data. Through an unsupervised learning process, SOM adjusts these weight vectors to reflect the structure of the input space, thereby enabling the projection of high-dimensional data into a two-dimensional topological map that preserves the relationships between data points.

The dimensions of the map are evaluated by calculating the quantity of neurons based on the number of observations present in the training data, employing a formula [131]:

$$M \cong 5\sqrt{n}, \tag{4.1}$$

where M represents the number of neurons, approximating an integer value near the outcome derived from the right side of the equation, while n stands for the number of observations in the training set of SOM. The number of rows and columns of SOM is $\cong \sqrt{M}$.

The network learning is introduced briefly below. The SOM network weights of vectors are initially assigned random values. During each iteration, an input vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, where m is the length of the SOM neuron weight vector. To determine the neuron that best represents the input, the Euclidean distance between the in-

put vector and each neuron's weight vector $W_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$ is computed as follows:

$$d_{ij} = \sqrt{\sum_{k=1}^{m}(x_{ik} - w_{jk})^2}, \tag{4.2}$$

where d_{ij} denotes the Euclidean distance between input vector X_i and the weight vector W_j of the j-th neuron, and $k \in \{1, 2, \ldots, m\}$ indexes the individual feature dimensions.

The neuron that exhibits the smallest distance for a given input data point is identified as the BMU for that data point. After identifying the best matching unit, the training process involves selecting the neighboring neurons of the BMU. These neighboring neurons are determined by a specific criterion, often based on their spatial proximity to the BMU within the neural network. Once the neighbors are established, the weight vectors associated with these neighboring neurons are updated using a neighborhood function.

Classical manner to update weights of neuron is as follows [77]:

$$w_{jk}(t+1) = w_{jk}(t) + \eta(t)T_{j^*j}(t)(x_{ik} - w_{jk}(t)), \tag{4.3}$$

where

- t is a number of iteration,
- is the k-th component of the weight vector of the j-th neuron at iteration t.
- $\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda_\eta}\right)$ is the learning rate at the iteration t. It decreases over time to gradually reduce the influence of new input data on the weights.
- $T_{j^*j}(t) = \exp\left(-\frac{\|W_{j^*} - W_j\|^2}{2\sigma(t)^2}\right)$ is the neighborhood function value between j^*-th BMU and the j-th neuron at iteration t.
- $\|W_{j^*} - W_j\|$ is the lateral distance between neurons j^* and j, where W_{j^*} is a BMU.
- x_{ik} is the k-th component of the input point X_i.
- $\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_\sigma}\right)$ is neighborhood size.

Hyperparameters for SOM training are such:

- η_0 is learning rate,
- λ_η is a constant that determines the rate of decay,
- σ_0 is neighborhood size,
- λ_σ is a constant that determines the rate of decay for the neighborhood width.

The neighborhood function defines how much influence each neighbor should have on the BMU and its surrounding neurons. Typically, the influence decreases with distance from the BMU, effectively creating a decaying effect on the updates. In essence, the process of identifying the BMU and updating the weights of its neighbors through the neighborhood function forms the basis of a self-organizing map algorithm.

The example below could be employed to explain SOM. Let us say we have a two-dimensional dataset which is visualized on the left side of Figure 4.2. In the middle, we have SOM visualized on the grid/coordinate space. The graph on the right shows data points (grey dots) and neurons in weight space. The red dots are BMU, and the grey cross are neurons that never became BMU.

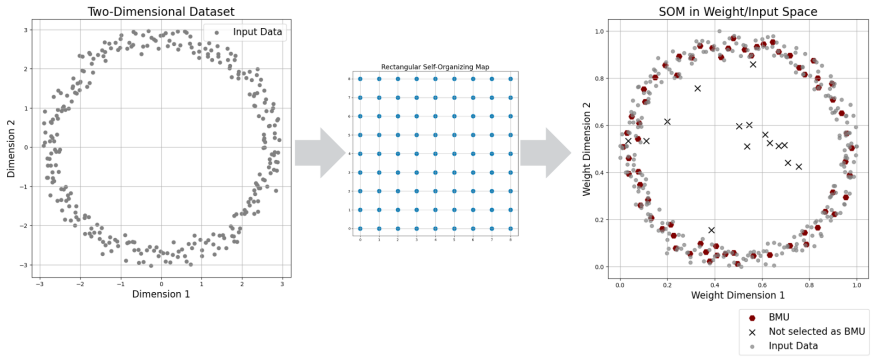


Figure 4.2: The process of SOM training and BMU assignment in a 2D feature space: input data (left), initialized SOM grid (middle), and trained SOM with BMUs aligned to the input distribution (right)

After collecting the weight vectors corresponding to BMU, we obtain a data frame with dimensions of $n_{BMU} \times m$, where n_{BMU} represents

the number of BMUs and m signifies the number of features or the length of each weight vector. Notably, n_{BMU} remains equal to or less than the total number of neurons, denoted as M since not every individual neuron is selected for the role of a BMU. Subsequently, this data frame serves as a foundation for the feature selection process, a pivotal step in refining the most relevant features from the original dataset, i.e., we try to decrease the number of features m significantly.

We propose to select a subset of features based on SOM weight variation. By normalizing the BMU data and calculating the variance of each attribute, we determine the importance of each feature in capturing the data's variability. The attributes are arranged in descending order according to their variance. This results in a list of features sorted by their significance. Subsequently, we can choose the desired number of features from the top of this ordered list.

SOM is used for clustering tasks [39]. However, we employ SOM's generalization capabilities to solve the dimensionality reduction problems for sharply imbalanced datasets. These ideas make the core of a novel method, FID-SOM, for feature selection for imbalanced data using SOM. The algorithm of the proposed method is presented by pseudo-code in Algorithm 2.

Algorithm 2 FID-SOM (Feature selection for Imbalanced Data Using SOM)

Require: $X \in \mathbb{R}^{n \times m}$: training dataset with n samples and m features
Require: $params = \{map\_size, iterations, learning\_rate, neighb\_radius\}$
Require: d: number of desired features
Ensure: F: set of d selected feature indices

1: $SOM \leftarrow \text{Train\_SOM}(X, \text{params})$
2: $W_{BMU} \leftarrow \text{ExtractBMUWeights}(SOM)$
3: $W_{BMU}^{norm} \leftarrow \text{MinMaxScale}(W_{BMU}, [0, 1])$
4: $V \leftarrow \text{Variance}(W_{BMU}^{norm})$ ▷ a vector of length m with variances per attribute
5: $idx \leftarrow \text{ArgsortDescending}(V)$
6: $F \leftarrow idx[1{:}d]$ ▷ Select top d features
7: **return** F

The method performs feature selection by automatically adapting to the inherent characteristics of the data. It identifies and retains features

that contain the most relevant information for subsequent analysis or visualization. This design allows the method to operate consistently across datasets with varying dimensionality and complexity.

The weight vector is critical for mapping high-dimensional transactional data into a lower-dimensional space, preserving the topological relationships of the input data. In the context of fraud detection, this enables SOM to cluster similar transactions together while highlighting outliers, which often correspond to fraudulent behavior. SOM provides a structured framework for analyzing complex transactional patterns and identifying anomalous activities by associating each node with a weight vector.

4.2. Quantitative Assessment of FID-SOM

In this section, we present the results obtained from our experimental study. Our experiments were designed to test the proposed method FID-SOM described by Algorithm 2. We provide a detailed description of used datasets and decisions made in data preparation and data splitting, supported by quantitative and qualitative assessments, along with visual aids such as tables and figures.

4.2.1. Classifiers and Metrics for FID-SOM Evaluation

To evaluate the performance of the proposed FID-SOM, a comparative analysis was conducted against a diverse set of established feature selection techniques. These methods were selected to represent three major categories of feature selection approaches: filter methods, wrapper methods, and model-based methods. Such a comparison ensures that FID-SOM is benchmarked against widely used strategies that differ in their underlying assumptions and mechanisms.

Univariate feature selection methods [104] were applied as a representative class of filter approaches. These methods evaluate each feature independently of others, based solely on its statistical relationship with the target variable.

The first is the Univariate feature selection method based on F-test (UniF), which measures linear dependency between numerical features and the target class using the ratio of between-group variance to within-

group variance [111]:

$$F = \frac{\text{MS}_{\text{between}}}{\text{MS}_{\text{within}}} = \frac{\frac{\sum_{k=1}^{K} n_k (\bar{x}_k - \bar{x})^2}{K-1}}{\frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_{ik} - \bar{x}_k)^2}{N-K}},$$

where K is the number of classes, n_k is the number of samples in class k, N is the total number of observations across all K classes and $\bar{x}_k$ and $\bar{x}$ denote class-wise and overall means.

The second is the univariate feature selection method based on χ^2-test (UniChi2), which evaluates the dependence between non-negative feature values and class labels by comparing observed and expected frequencies [111]:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

where O_{ij} and E_{ij} are the observed and expected frequencies under the assumption of independence.

Finally, Univariate feature selection method based on Mutual Information (UniMI) was used to capture both linear and non-linear dependencies between features and the target. The measure is based on information theory concepts [111] and implemented following the *scikit-learn* framework [104]:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)\, p(y)},$$

where $p(x, y)$ is the joint probability of feature x and class labels y.

Recursive Feature Elimination (RFE) [53] represents a wrapper-based method. RFE operates by recursively training a predictive model, ranking features according to their importance, and eliminating the least significant ones at each iteration. This process continues until the desired number of features is reached. Unlike univariate filter methods, RFE considers the effect of feature subsets on model performance, which can lead to higher accuracy. However, this advantage comes at the expense of computational cost, particularly for large datasets or high-dimensional feature spaces.

Finally, a model-based feature selection method was employed using the XGB Importance method (XGBImp) [24]. This technique is based on the XGBoost algorithm, a powerful gradient boosting framework.

Feature importance scores are derived from the contribution of each feature to the model's splits and predictions, providing a ranking that reflects their predictive relevance. Model-based approaches such as this are capable of capturing complex feature interactions and non-linear relationships. Nevertheless, they are sensitive to model hyperparameters and can exhibit bias towards features that dominate early splits in tree-based models.

The purpose of feature selection is to increase the performance of ML algorithms. The efficacy of the feature selection methods was evaluated using the XGBoost, CatBoost, and RF machine learning algorithms. The main reason for choosing RF is its good performance on data related to financial fraud detection [34], [8]. Meanwhile, XGBoost and CatBoost usage is gaining popularity and demonstrating strong performance [56], [26]. We are aware that LR is often employed to solve fraud detection tasks. However, our decision was not to use this algorithm for further experiments as it showed weak performance in scenarios where hyperparameters were not being optimized, or data was not balanced [A.1]. We did not use parameter hypertuning or data sampling in order to find the pure effect of feature selection methods.

To evaluate the goodness of the method, we use five metrics suitable for imbalanced datasets, namely F1 score, MCC, G-Mean, AUC-PR, and AUC-ROC. A description of each metric can be fount in Section 1.3.4.

4.2.2. Data Used for Experiments

For our experimental analysis, we employed three datasets. Among these, two datasets were derived from synthetic transactional payments data, while the third dataset represents a real transactional dataset.

Synthetic datasets are the same as in the Section 3. The third dataset contains credit card transactions conducted by European cardholders in September 2013 available at Kaggle. It encapsulates two-day transactions, revealing 492 instances of fraud out of 284 807 transactions. Notably, the dataset exhibits a substantial imbalance, with the positive class (frauds) constituting a mere 0.172% of all transactions. The dataset exclusively comprises numerical input variables resulting from a Principal Component Analysis (PCA) transformation. Regrettably, the disclosure of the original features and additional contextual information is hidden due to confidentiality constraints. Principal components V1

through V28 are derived from PCA, while "Time" and "Amount" are the only features unaffected by the transformation. "Time" denotes the seconds elapsed between each transaction and the initial transaction in the dataset, while "Amount" represents the transaction amount. The "Time" feature is used for splitting purposes. In the rest of the text, this dataset will be called DataSet3.

Table 4.1 represents the distribution between fraudulent and legitimate instances in each dataset. For the experiment, we used sharply imbalanced datasets. In size, those datasets are very different. In DataSet1, which contains 3 445 553 cases and 25 attributes, the class distribution is 99.86% "non-fraud" and 0.14% "fraud". In DataSet2, which contains 1 852 394 cases and 11 attributes, the level of fraud is slightly higher at 0.52%. DataSet3, while maintaining a high majority of 99.83% "non-fraud," differs with a fraud rate of 0.17%, covering 284 807 cases and 29 attributes. These statistics reveal the imbalances and differences in attributes between each dataset, providing valuable insights for designing and evaluating robust fraud detection models.

Table 4.1: Summary statistics of the datasets used in the experiments

Category	DataSet1	DataSet2	DataSet3
Not Fraud (Percentage)	99.86%	99.48%	99.83%
Fraud (Percentage)	0.14%	0.52%	0.17%
# of instances	3 277 610	1,852,394	284,807
# of features	25	11	30

4.2.3. Data Preprocessing

In our comparison of feature selection methods, it is essential to train ML models. To facilitate this, we employ both training and testing datasets. Subsequently, we outline an appropriate approach for splitting the data into training and testing sets specifically tailored for fraud detection tasks.

Fraudulent data is inherently temporal, as mentioned in Section 1, meaning that observations are dependent on previous observations in a sequential manner. This temporal dependence leads to correlations between data points that are close in time. The model might risk intro-

ducing temporal leaks if a standard train-test split is used on time series data, where data is randomly shuffled and partitioned. This can result in unrealistic correlations between the training and testing sets, leading to overly optimistic estimates of the model's performance [139].

A credit card, investment, or any other type of fraud data has a concept drift property [17], [35], [89]. Concept drift refers to the phenomenon where the underlying statistical properties of the data distribution change over time. This can happen due to various reasons, such as changes in user behavior, fraud patterns, or market conditions.

The classical train-test split assumption of independent and identically distributed samples does not hold well for time series data, especially when dealing with concept drift in domains like fraud detection [54] [104]. When concept drift occurs, the assumption that the training and testing data are drawn from the same distribution is violated. To address this issue and create a more realistic evaluation setup for fraud detection, we suggest using $TimeSeriesSplit$. Instead of random shuffling, we suggest splitting the data chronologically, where the training data X_{train} comes from earlier periods, and the testing data X_{test} comes from later periods. This simulates a real-world scenario where the model is trained on historical data and tested on more recent data. FID-SOM and other feature selection methods used X_{train} to define the proper set of features.

Each dataset is split using a time-based approach. The dataset is divided so that the earliest 80% of instances would be for training, and the rest of the data, which has timestamps later than the training set, is left for testing (see Figure 4.3).

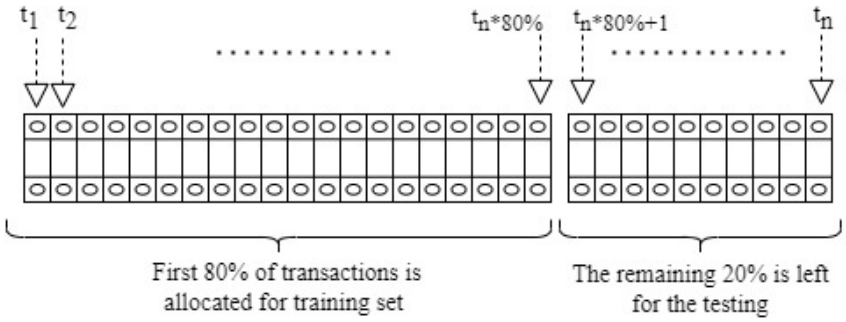


Figure 4.3: Dataset split based on transactions' time

So, we are not setting up the split date, but instead, we are dynamically determining the split based on the chronological order of the data entries. This time-based approach ensures that the model is trained on historical data and then evaluated on more recent data, simulating a real-world scenario where the model makes predictions on new, unseen observations.

We have used the categorical feature encoding method, James-Stein encoder, discovered as comparatively best for imbalanced data in the paper [A.1], where six feature encoders were compared.

Encoded data are scaled before training a SOM. This is essential because it ensures that all features contribute equally to the training process, regardless of their original units or magnitude. SOMs rely on the calculation of distances to map high-dimensional data onto a lower-dimensional grid. If features have vastly different ranges or units, for example, "Transaction amount" and "Age", those with larger magnitudes can dominate the distance calculation, effectively overshadowing features with smaller scales. This imbalance can lead to a biased SOM, where the map primarily reflects variations in high-magnitude features, neglecting meaningful patterns in others. By scaling the data, usually through standardization (Z-score scaling) or normalization (MinMax scaling), we ensure that all features are on a comparable scale, allowing SOM to identify and represent the intrinsic structure of the data more accurately. In our case, we use normalization which brings all features into the same range [0, 1], ensuring equal importance during training.

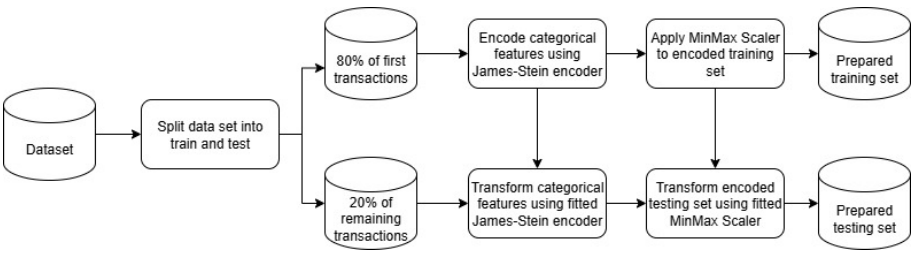


Figure 4.4: Data preprocessing steps which include data splitting, encoding, and normalization.

4.2.4. Observed Outcomes and Performance Metrics

A comprehensive overview of SOM configurations used in this study is provided in Table 4.2. SOM dimensions were determined using the heuristic formula $M \approx 5\sqrt{n}$, where n is the number of training instances and M is the total number of neurons (see Equation 4.1). For DataSet1, consisting of 2 622 088 training instances (80% of 3 277 610), the estimated number of neurons is $M \approx 5 \times \sqrt{2622088} \approx 5 \times 1,619 \approx 8095$, which corresponds to a 90×90 grid (8,100 neurons in total). The training process for this SOM involved 4 048 216 iterations. The number of iterations was set as:

$$\text{max\_steps} = \text{round}(5 \cdot \sqrt{N} \cdot 500)$$

This follows two established heuristics: Kohonen's recommendation to train for 500–1000 steps per SOM unit [78] so, this results in a total of approximately $2500 \cdot \sqrt{N}$ training steps. For DataSet2, a slightly smaller SOM with a 78×78 grid (6,084 neurons) was used, based on 1,481,916 training instances, and trained over 3,043,349 iterations. DataSet3, the smallest of the three, employed a compact 49×49 grid (2,401 neurons) with 1 193 330 training iterations.

Table 4.2: Summary of self-organizing maps configuration

Property	DataSet1	DataSet2	DataSet3
Size	90x90	78x78	49x49
Iterations	4 048 216	3 043 349	1 193 330

The upper part of Figure 4.5 presents SOM in the grid space, where the dots represent BMUs and the size of the dots represents how many instances each BMU has. The lower part of Figure 4.5 shows the dependency of a number of instances covered by a minimal number of BMU. The dashed horizontal line marks 95% of instances, and the dashed vertical line shows how many BMUs are required to cover these 95% of instances.

We observe that SOM for each dataset is quite different. DataSet1 has many clusters, while DataSet2 is separated into two parts. The BMUs of SOM of DataSet3 have an almost uniform distribution with one very massive neuron.

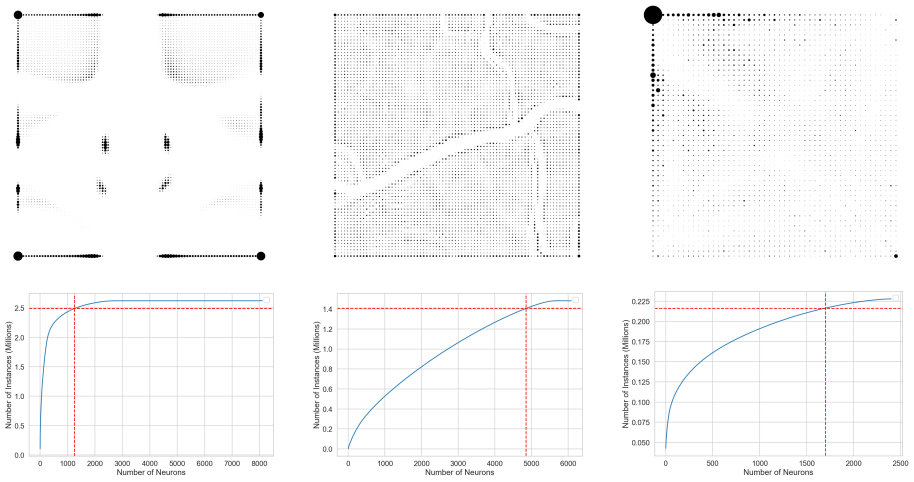


Figure 4.5: Visualisation of the trained Self-Organized Map for each dataset: DataSet1, DataSet2, DataSet3. The curves show the dependency of the number of instances covered by the number of BMUs. The dashed horizontal line marks 95% of instances, and the dashed vertical line shows how many BMUs are required to cover these 95% of instances

For each dataset, a set of number of selected features was chosen relative to the total number of available features. This design reflects a strategy to evaluate model performance under slight-to-moderate feature reduction, helping reduce redundancy, avoid overfitting, and maintain interpretability and predictive power.

Due to the large size and computational cost associated with DataSet1, only three different feature sets were tested: [20, 22, 24]. For DataSet2, which contains fewer features, a broader range of four values was explored: [10, 9, 8, 7]. Similarly, DataSet3 was evaluated using four values: [21, 23, 25, 27]. Feature subsets were evaluated with three ML algorithms across five metrics, totaling $(3+4+4)\times 3\times 5 = 11\times 3\times 5 = 165$.

Each time, one or several feature selection methods were marked as the winning methods if they had the highest score of a particular metric. An example of how to identify the winning (best performing) feature selection methods is shown in Table 4.3. In this table, a snapshot of our experiments is shown. Here, the evaluation is performed by selecting the winning method for DataSet1 using the XGB classifier with 20 selected features for F1-score, MCC, and G-Mean. We mark that our proposed method, FID-SOM, became the best five times, and

other methods became the best one time. The complete set of results is presented in Table 4.5-Table 4.7.

Table 4.3: Example of the evaluation performed by selecting the winning method for DataSet1 using the XGB classifier with 20 selected features.

Method	F1	ROC-AUC	PR-AUC	MCC	G-MEAN
Baseline	0.82	**1.00**	0.95	0.83	0.85
FID-SOM	**0.95**	**1.00**	**0.98**	**0.95**	**1.00**
UniChi2	0.82	**1.00**	0.94	0.83	0.85
UniF	0.83	**1.00**	0.95	0.84	0.85
UniMI	0.79	**1.00**	0.95	0.80	0.81
RFE	0.82	**1.00**	0.94	0.83	0.85
XGBImp	0.82	**1.00**	0.94	0.82	0.85

Table 4.4 presents a comparative analysis of various feature selection methods. The "No. of winnings" column shows how often the particular method was selected as the best-performing method.

For a comprehensive evaluation, we computed the mean performance outcomes for each method across five random seed values. In this case, the results of FID-SOM are still outstanding (Table 4.4). On some occasions, various methods achieved identical outcomes and are thus all designated as winners, preventing the percentage values in Table 4.4 from totaling 100%.

Table 4.4: Comparison of feature selection methods across all tested feature calibrations

Method	No. of winnings	Total	Percentage
Baseline	33	165	20%
FID-SOM	**73**	**165**	**44.24**%
UniChi2	33	165	20.00%
UniF	18	165	10.91%
UniMI	44	165	26.67%
RFE	16	165	9.7%
XGBImp	19	165	11.52%

The proposed method FID-SOM is distinctive in its efficiency due to the utilization of a novel feature selection technique introduced in

this thesis. It achieved a success rate of 44.24%, the highest among all methods considered. It outperforms the second-best method, UniMI, which achieved a success rate of 26.67%.

Different methods can have different optimal number of features. Considering this, we selected the best result for each metric/model/method from all compared feature sets. Results are shown in the Table 4.5 - Table 4.7. Each result represents the mean of five experiments conducted with different random seeds, along with the number of features that yielded the highest performance and the standard deviation across these five experiments.

Table 4.5: Feature selection methods' comparison with different machine learning models on DataSet1

	F1-Score			AUC-ROC			PR-AUC			MCC			G-MEAN		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.400	-	-	0.999	-	-	0.490	-	-	0.409	-	-	0.575	-	-
FID-SOM	**0.591**	**24**	**0.021**	**1.000**	**24**	**0.000**	**0.657**	**24**	**0.020**	**0.591**	**24**	**0.020**	**0.756**	**24**	**0.020**
RFE	0.403	24	0.030	0.999	22	0.007	0.489	24	0.070	0.412	24	0.019	0.575	24	0.042
UniChi2	0.415	20	0.015	0.999	20	0.001	0.502	20	0.019	0.424	20	0.015	0.585	20	0.011
UniF	0.413	20	0.015	0.999	20	0.001	0.509	20	0.019	0.423	20	0.015	0.582	20	0.011
UniMI	0.403	24	0.016	0.999	24	0.000	0.489	24	0.025	0.412	24	0.017	0.575	24	0.015
XGBImp	0.415	22	0.019	0.999	24	0.000	0.492	22	0.017	0.424	22	0.018	0.586	22	0.019
RandomForestClassifier															
Baseline	0.844	-	-	**1.000**	-	-	0.967	-	-	0.851	-	-	0.864	-	-
FID-SOM	**0.957**	**24**	**0.003**	**1.000**	**24**	**0.000**	**0.980**	**24**	**0.006**	**0.958**	**24**	**0.003**	**0.998**	**24**	**0.001**
RFE	0.852	22	0.010	**1.000**	**22**	**0.000**	0.970	22	0.004	0.858	22	0.009	0.872	22	0.010
UniChi2	0.851	22	0.019	**1.000**	**24**	**0.000**	0.973	24	0.004	0.857	22	0.017	0.873	22	0.018
UniF	0.851	22	0.013	**1.000**	**24**	**0.000**	0.973	24	0.004	0.857	22	0.012	0.873	22	0.012
UniMI	0.848	24	0.013	**1.000**	**24**	**0.000**	0.969	24	0.004	0.855	24	0.011	0.870	24	0.014
XGBImp	0.853	24	0.019	**1.000**	**24**	**0.000**	0.972	24	0.004	0.859	24	0.017	0.873	24	0.017
XGBClassifier															
Baseline	0.820	-	-	**1.000**	-	-	0.949	-	-	0.827	-	-	0.849	-	-
FID-SOM	**0.962**	**24**	**0.000**	**1.000**	**22**	**0.000**	**0.987**	**22**	**0.000**	**0.962**	**24**	**0.000**	**1.000**	**24**	**0.000**
RFE	0.825	22	0.000	**1.000**	**24**	**0.000**	0.949	24	0.000	0.832	22	0.000	0.853	22	0.000
UniChi2	0.824	20	0.000	**1.000**	**24**	**0.000**	0.948	24	0.000	0.831	20	0.000	0.853	20	0.000
UniF	0.828	20	0.000	**1.000**	**20**	**0.000**	0.948	24	0.000	0.836	20	0.000	0.853	20	0.000
UniMI	0.820	24	0.007	**1.000**	**20**	**0.000**	0.949	24	0.002	0.827	24	0.006	0.849	24	0.006
XGBImp	0.820	24	0.000	**1.000**	**20**	**0.000**	0.945	24	0.000	0.827	24	0.000	0.849	24	0.000

Table 4.6: Feature selection methods' comparison with different machine learning models on DataSet2

	F1-Score			AUC-ROC			PR-AUC			MCC			G-MEAN		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.750	-	-	0.991	-	-	0.769	-	-	0.765	-	-	0.791	-	-
FID-SOM	**0.828**	**7**	**0.014**	**0.997**	**7**	**0.002**	0.866	7	0.011	**0.831**	**7**	**0.013**	**0.870**	**7**	**0.011**
RFE	0.751	7	0.014	0.989	9	0.008	0.765	9	0.008	0.763	9	0.012	0.803	7	0.011
UniChi2	0.711	8	0.022	0.987	9	0.002	0.722	8	0.014	0.727	8	0.019	0.765	8	0.018
UniF	0.705	9	0.014	0.987	9	0.002	0.722	9	0.010	0.723	9	0.012	0.759	9	0.011
UniMI	0.825	8	0.014	**0.997**	**8**	**0.001**	**0.867**	**8**	**0.013**	0.829	8	0.013	0.864	8	0.011
XGBImp	0.752	9	0.008	0.990	10	0.001	0.768	10	0.009	0.761	8	0.008	0.806	9	0.006
RandomForestClassifier															
Baseline	0.622	-	-	0.967	-	-	0.767	-	-	0.660	-	-	0.682	-	-
FID-SOM	**0.835**	**7**	**0.017**	0.978	7	0.002	**0.868**	**7**	**0.004**	**0.839**	**7**	**0.014**	**0.866**	**7**	**0.013**
RFE	0.707	7	0.025	0.966	10	0.002	0.774	7	0.006	0.725	7	0.021	0.760	7	0.021
UniChi2	0.626	10	0.028	0.964	10	0.003	0.763	10	0.009	0.662	10	0.024	0.686	10	0.023
UniF	0.626	10	0.033	0.964	10	0.002	0.763	10	0.009	0.662	10	0.029	0.686	10	0.026
UniMI	0.826	8	0.020	**0.979**	**8**	**0.002**	0.863	8	0.006	0.832	8	0.017	0.856	8	0.016
XGBImp	0.706	7	0.030	0.963	10	0.003	0.776	7	0.008	0.724	7	0.025	0.759	7	0.025
XGBClassifier															
Baseline	0.543	-	-	0.973	-	-	0.641	-	-	0.593	-	-	0.622	-	-
FID-SOM	**0.855**	**7**	**0.000**	**0.998**	**7**	**0.000**	**0.895**	**7**	**0.000**	**0.857**	**7**	**0.000**	**0.889**	**7**	**0.000**
RFE	0.573	7	0.000	0.979	7	0.000	0.639	7	0.000	0.613	7	0.000	0.650	7	0.000
UniChi2	0.606	8	0.000	0.983	8	0.000	0.646	8	0.000	0.640	8	0.000	0.676	8	0.000
UniF	0.537	9	0.000	0.978	10	0.000	0.603	9	0.000	0.582	9	0.000	0.621	9	0.000
UniMI	0.847	8	0.000	0.997	8	0.000	0.891	8	0.000	0.851	8	0.000	0.877	8	0.000
XGBImp	0.577	9	0.000	0.979	7	0.000	0.639	7	0.000	0.613	9	0.000	0.656	9	0.000

Table 4.7: Feature selection methods' comparison with different machine learning models on DataSet3

	F1-Score			AUC-ROC			PR-AUC			MCC			G-MEAN		
	Mean	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**	**Mean**	**#**	**Std**
CatBoostClassifier															
Baseline	0.818	-	-	0.982	-	-	0.805	-	-	0.826	-	-	0.849	-	-
FID-SOM	0.813	21	0.018	**0.986**	**25**	**0.004**	**0.811**	**27**	**0.007**	0.822	21	0.018	0.842	21	0.011
RFE	0.818	23	0.017	0.982	25	0.004	0.803	27	0.007	0.824	23	0.016	**0.855**	**23**	**0.015**
UniChi2	0.809	21	0.009	0.984	27	0.003	0.806	27	0.007	0.819	21	0.009	0.836	21	0.009
UniF	0.813	21	0.015	0.982	27	0.003	0.802	27	0.006	0.820	21	0.014	0.847	21	0.012
UniMI	0.814	23	0.012	0.982	27	0.004	0.794	27	0.008	0.821	23	0.012	0.848	23	0.010
XGBImp	**0.821**	**21**	**0.021**	0.982	25	0.004	0.805	27	0.006	**0.828**	**21**	**0.019**	**0.855**	**21**	**0.019**
RandomForestClassifier															
Baseline	0.815	-	-	0.938	-	-	0.792	-	-	0.825	-	-	0.841	-	-
FID-SOM	0.820	27	0.006	0.935	23	0.011	0.791	23	0.006	0.831	27	0.006	0.839	27	0.010
RFE	0.814	27	0.008	**0.942**	**27**	**0.014**	0.791	27	0.005	0.825	27	0.007	0.837	25	0.009
UniChi2	**0.825**	**25**	**0.010**	**0.942**	**27**	**0.011**	0.789	23	0.006	**0.836**	**25**	**0.009**	**0.844**	**25**	**0.009**
UniF	0.819	21	0.009	0.939	23	0.013	**0.794**	**21**	**0.006**	0.794	21	0.006	0.839	21	0.009
UniMI	0.822	27	0.011	0.939	21	0.010	**0.794**	**21**	**0.006**	0.833	27	0.011	**0.844**	**25**	**0.009**
XGBImp	0.819	25	0.011	0.939	27	0.011	0.789	23	0.007	0.829	25	0.011	0.839	25	0.009
XGBClassifier															
Baseline	0.796	-	-	0.974	-	-	0.803	-	-	0.810	-	-	0.841	-	-
FID-SOM	0.801	21	0.000	**0.987**	**23**	**0.000**	0.824	23	0.000	0.833	23	0.000	**0.856**	**27**	**0.000**
RFE	0.800	27	0.000	0.984	21	0.000	0.827	27	0.000	**0.834**	**27**	**0.000**	**0.856**	**27**	**0.000**
UniChi2	**0.806**	**23**	**0.000**	0.985	23	0.000	0.827	21	0.000	**0.834**	**21**	**0.000**	**0.856**	**21**	**0.000**
UniF	0.800	27	0.000	0.985	21	0.000	**0.833**	**27**	**0.000**	0.841	27	0.000	**0.856**	**27**	**0.000**
UniMI	0.799	27	0.000	0.980	27	0.000	0.827	25	0.000	**0.834**	**25**	**0.000**	**0.856**	**25**	**0.000**
XGBImp	0.796	27	0.000	0.984	27	0.000	0.821	27	0.000	0.827	27	0.000	**0.856**	**27**	**0.000**

In DataSet1, using the CatBoostClassifier, the FID-SOM approach stands out with an impressive F1 score of 0.591, compared to the baseline's 0.40. Similarly, in DataSet2 with the RF Classifier, FID-SOM excels with an F1 score of 0.835, outperforming the baseline's 0.622. Moving to DataSet3, utilizing the XGBClassifier, FID-SOM maintains its robust performance with a notable F1 score of 0.801, surpassing the baseline's 0.796. Across all three datasets, FID-SOM consistently achieves superior results in various metrics, including ROC, PR, MCC, and G-Mean, demonstrating its effectiveness as a classification method. These findings underscore the potential of FID-SOM in enhancing predictive capabilities and model performance across diverse datasets.

In these detailed tables (Table 4.5 - Table 4.7), our proposed method, FID-SOM, has been marked 32 times as giving the best values for the selected metric.

Table 4.8: Comparison of feature selection methods across feature calibrations yielding the best metric values

Method	**No. of winnings**	**Total**	**Percentage**
Baseline	2	45	4.44%
FID-SOM	**32**	**45**	**71.11%**
UniChi2	8	45	17.78%
UniF	6	45	13.33%
UniMI	8	45	17.78%
RFE	5	45	11.11%
XGBImp	6	45	13.33%

The experimental results demonstrate the effectiveness of our proposed method. Notably, our proposed method works significantly better on the dataset structures when SOM can identify many homogeneous clusters and fewer neurons cover more data points, as can be seen in Figure 4.5. To get better results, one can vary the SOM architecture based on the dataset. In this study, the goal was to set up the same experimental environment rather than aiming for the highest performance metric for each dataset.

4.3. Discussions

Even though DataSet3 is very popular among researchers, it is difficult to compare our work with other papers. The primary challenges arise because some papers do not specify the splitting ratio or the type of split—random or time-based. In many cases, studies addressing the fraud-detection problem in credit card transactions unrealistically evaluate the performance of the proposed method by splitting the dataset into train and test using random split (see [84], [66]). This assumes that the data is independently and identically distributed over time. However, in real-world scenarios, credit card transaction data often exhibits temporal dependencies and non-stationarity, making this assumption flawed. As a result, models trained on one time period may not perform well on data from another due to shifts in transaction patterns, fraudulent activities, or changes in user behavior. This is important because in time-series data, observations are typically dependent on previous observations, and shuffling the data randomly could lead to data leakage.

However, in order to compare FIDSOM performance against other published work, we did a split using stratified random split, selecting 80% of data points for the training set and 20% for testing on DataSet3. For the comparison, we selected only those papers that clearly specified the splitting share. We did not include papers that use data balancing methods like oversampling or undersampling before splitting the data into training and testing datasets, e.g., in this way, technically removing imbalance problems, which is not possible in real-life scenarios. Applying sampling methods before splitting the dataset into train and test sets leads to deceptively high results.

The baseline performance of four widely recognized ensemble learning models, specifically focusing on their F1 scores, is presented in the paper [116]. The models evaluated include RF, XGBoost, LightGBM, and CatBoost. No additional feature engineering or optimization steps were implemented for this baseline assessment, ensuring that the F1 scores reflect the models' classification abilities. The F1 scores are as follows: RF achieved an F1 score of 0.846, XGBoost obtained a slightly lower score of 0.840, LightGBM trailed with a score of 0.749, while CatBoost led the group with an F1 score of 0.853. These results provide an initial benchmark for further model refinement.

Table 4.9: Comparison with other papers splitting data in a time-based manner with a share of 70/30 for training and testing

Paper	Year	F1-Score	Recall	Precision
[48]	2019	0.82	0.73	0.93
[44]	2023	0.84	0.74	**0.97**
FIDSOM*	2024	**0.85**	**0.76**	**0.97**

*FIDSOM with XGB classifier selecting 23 features. Data split is done by selecting 70% of the first data points for training and 30% remaining data points for testing.

Table 4.10: Comparison with other papers splitting data randomly with a share of 80/20 for training and testing

Paper	Year	F1-Score	Recall	Precision
[150]	2024	0.85	**0.84**	0.86
[68]	2023	0.85	0.76	**0.98**
FIDSOM**	2024	**0.88**	0.82	0.95

** FID-SOM with RF selecting 23 features. Data split is done by randomly selecting 80% of the data points for training and 20% of data points for testing.

4.4. Conclusions of the Chapter

In this chapter, we suggest a novel feature selection method called FID-SOM. The uniqueness of the proposed method is in forming a new dataset containing the best matching units of the trained SOM as vectors of attributes corresponding to the initial features. These attributes are sorted based on variance in descending order. By keeping the desired number of attributes holding the highest variability, we select a smaller number of features corresponding to those attributes for further analysis.

FID-SOM was compared with univariate feature selection methods utilizing the F-test, χ^2 test and mutual information, the Recursive Feature Elimination method, and the XGB Importance method. The effectiveness of the feature selection methods was evaluated using F1

score, MCC, G-Mean, AUC-PR, and AUC-ROC metrics when performing XGBoost, CatBoost, and Random algorithms on three datasets.

The success of the method was evaluated by counting how many times the method became the best-performing method. The proposed FID-SOM method has demonstrated noteworthy achievement by reaching a success rate of 71.11% (Table 4.8). This accomplishment is not only meaningful because of its ability to perform on par with, if not surpass, existing methodologies but also shows its innovative potential. Notably, the FID-SOM method is highlighted when compared with the performance of the second-best method, which yielded a success rate of 17.78% (Table 4.8).

Fraud detection is a critical activity aimed at preventing financial losses by accurately identifying fraudulent/illegal transactions. The main objective of this dissertation is to develop a method that rationally reduces the existing feature set in order to improve the classification efficiency of imbalanced data, thereby enhancing the detection of fraud cases. Given the high imbalance of fraud-related datasets, traditional machine learning algorithms often fail to achieve acceptable performance.

This dissertation explores innovative methods designed to address these challenges: clustering-based classification approaches, feature selection solutions, and categorical variable encoding techniques, all aimed at improving fraud detection effectiveness.

- A clustering-based learning strategy is proposed, in which each cluster is balanced and classified separately. This approach statistically significantly improved the recall of fraud classification from 0.845 to 0.867. Experimental results show that integrating clustering as a preprocessing step enhanced the model's ability to correctly identify fraudulent transactions, while the number of fraud cases incorrectly classified as legitimate decreased by 13.9%.

- Experimental findings revealed that categorical feature encoding strategies incorporating information from the target variable – specifically James-Stein and Weight of Evidence (WOE) methods – ensure considerably higher discriminative power when working with highly imbalanced and high-cardinality datasets. With James-Stein encoding, average F1 scores reached 0.8049 with boosting classifiers, 0.7595 with ensemble models, and 0.7604 with non-linear models. Corresponding WOE results were 0.7861, 0.7651, and 0.7529. Meanwhile, target-agnostic methods such as Label Encoding or Hashing performed significantly worse – in some cases, F1 scores fell below 0.5. It should be noted that all model hyperparameters remained at their default settings – the aim of the experiments was not to optimize the final result but to objectively assess the impact of different encoding strategies.

- The Hashing encoding method, as well as the widely used One-Hot encoding (though not included in this experiment), con-

tributes to the "curse of dimensionality," particularly when categorical features have extremely high cardinality – for example, "Merchant City" has 11 391 unique values. This not only hinders model training but also increases computational costs and the risk of overfitting.

- Experimental research demonstrated that the proposed new feature selection method, FID-SOM, which leverages Self-Organizing Maps and identifies the most informative features based on the variance of Best Matching Unit (BMU) weight vector components, proved to be the best-performing method in 71.11% of all tested configurations. In other words, across most feature set and classifier combinations, it achieved the highest classification performance. The method outperformed traditional selection techniques such as the F-test, χ^2 test, mutual information, Recursive Feature Elimination, and XGBoost Importance analysis.

- Experimental results also showed that when applying time-based transaction data splitting, FID-SOM achieved an F1 score of 0.85 and recall of 0.76. In contrast, with random data splitting, the same models returned significantly higher scores (F1 score is 0.88, recall is 0.82). However, such improvements should be considered artificial, arising from data leakage between training and testing sets. This discrepancy highlighted that random splitting may create a misleading impression of model effectiveness, whereas time-based splitting provides a more reliable basis for evaluating a model's ability to remain robust against evolving fraud patterns.

These insights offer valuable practical guidance for researchers and practitioners seeking to build more effective machine learning systems to combat financial fraud. The proposed FID-SOM method paves the way for further innovations, encouraging the development of adaptive and intelligent fraud detection systems capable of effectively addressing the ever-changing challenges of fraud.

[1] A. Agresti. *Statistical Methods for the Social Sciences*. Pearson, Boston, 5 edition, 2018. ISBN 9780134507101.

[2] A. Ali et al. Financial Fraud Detection Based on Machine Learning: A Systematic Literature Review. *Applied Sciences*, 12(19), Jan 2022. doi: 10.3390/app12199637. Art. no. 19.

[3] E. Altman. Synthesizing credit card transactions. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21, New York, NY, USA, 2022. ISBN 9781450391481. doi: 10.1145/3490354.3494378.

[4] E. R. Altman. Synthesizing Credit Card Transactions, 2019. URL `https://arxiv.org/abs/1910.03033`.

[5] M. N. Ashtiani and B. Raahemi. Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review. *IEEE Access*, 10:72504–72525, 2022. doi: 10.1109/ACCESS.2021.3096799.

[6] Association of Certified Fraud Examiners (ACFE). Impact of Fraud at U.S. Public Companies. `https://www.acfe.com/about-the-acfe/newsroom-for-media/press-releases/press-release-detail?s=impact-of-fraud-at-us-public-companies-pr`, 2024. Accessed: 2025-09-20.

[7] Association of Certified Fraud Examiners (ACFE). Occupational Fraud 2024: A Report to the Nations. Technical report, ACFE, 2024. URL `https://legacy.acfe.com/report-to-the-nations/2024/`. Accessed: 2025-09-20.

[8] M. H. Aung, P. T. Seluka, J. T. R. Fuata, M. J. Tikoisuva, M. S. Cabealawa, and R. Nand. Random Forest Classifier for Detecting Credit Card Fraud based on Performance Metrics. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pages 1–6, 2020. doi: 10.1109/CSDE50874.2020.9411563.

[9] V. Bagdonavičius and L. Petkevičius. Multiple Outlier Detection Tests for Parametric Models. *Mathematics*, 8(12):2156, 2020. doi: 10.3390/math8122156.

[10] S. Bashir, I. U. Khattak, A. Khan, F. H. Khan, A. Gani, and M. Shiraz. A Novel Feature Selection Method for Classification of Medical Data Using Filters, Wrappers, and Embedded Approaches. *Complexity*, 2022, 2022. doi: 10.1155/2022/8190814.

[11] M. Bekkar, H. K. Djemaa, and T. A. Alitouche. Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and applications*, 3(10), 2013.

[12] F. Bourdonnaye and F. Daniel. Evaluating categorical encoding methods on a real credit card fraud detection database, 2021. [Online]. Available: http://www.lusisai.com.

[13] J. Braithwaite. 'Authorized Push Payment' Bank Fraud: What Does an Effective Regulatory Response Look Like? *Journal of Financial Regulation*, 10(2):174–193, 07 2024. ISSN 2053-4841. doi: 10.1093/jfr/fjae006.

[14] L. Breiman. *Classification and Regression Trees*. Routledge, 1st edition, 1984. doi: 10.1201/9781315139470.

[15] E. A. L. M. Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran. A systematic review of literature on credit card cyber fraud detection using machine and deep learning. *PeerJ Comput. Sci.*, 9:e1278, Apr 2023. doi: 10.7717/peerj-cs.1278.

[16] V. Bulavas, V. Marcinkevičius, and J. Rumiński. Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets. *Informatica*, 32(3):441–475, 2021. doi: 10.15388/21-INFOR457.

[17] H. Bullock and M. Edwards. Temporal Constraints in Online Dating Fraud Classification. In *Proceedings of the 9th International Conference on Information Systems Security and Privacy*, pages 535–542, Lisbon, Portugal, 2023. SCITEPRESS - Science and Technology Publications. doi: 10.5220/0011689000003405.

[18] S. Cao, X. Yang, C. Chen, J. Zhou, X. Li, and Y. Qi. Titant: Online real-time transaction fraud detection in ant financial. *arXiv preprint arXiv:1906.07407*, 2019.

[19] E. M. Carneiro, C. H. Q. Forster, L. F. S. Mialaret, L. A. V. Dias, and A. M. Cunha. High-Cardinality Categorical Attributes and Credit Card Fraud Detection. *Mathematics*, 10(20), 2022. doi: 10.3390/math10203808.

[20] F. Castaño, E. F. Fernañdez, R. Alaiz-Rodríguez, and E. Alegre. PhiKitA: Phishing Kit Attacks Dataset for Phishing Websites Identification. *IEEE Access*, 11:40779–40789, 2023. doi: 10.1109/ACCESS.2023.3268027.

[21] M. Chalé and N. D. Bastian. Generating realistic cyber data for

training and evaluating machine learning classifiers for network intrusion detection systems. *Expert Systems with Applications*, 207: 117936, 2022. doi: 10.1016/j.eswa.2022.117936.

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 2002. doi: 10.1613/jair.953.

[23] H. Chen, T. Li, X. Fan, and C. Luo. Feature selection for imbalanced data based on neighborhood rough sets. *Information Sciences*, 483: 1–20, 2019. doi: 10.1016/j.ins.2019.01.041.

[24] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[25] W. Chen, K. Yang, Z. Yu, Y. Shi, and C. L. P. Chen. A survey on imbalanced learning: latest research, applications and future directions. *Artificial Intelligence Review*, 57(6):137, May 2024. doi: 10.1007/s10462-024-10759-6.

[26] Y. Chen and X. Han. CatBoost for Fraud Detection in Financial Transactions. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 176–179, 2021. doi: 10.1109/ICCECE51280.2021.9342475.

[27] Y. Chen, L. Ma, D. Yu, H. Zhang, K. Feng, X. Wang, and J. Song. Comparison of feature selection methods for mapping soil organic matter in subtropical restored forests. *Ecological Indicators*, 135: 108545, 2022. doi: 10.1016/j.ecolind.2022.108545.

[28] A. Cherif, A. Badhib, H. Ammar, S. Alshehri, M. Kalkatawi, and A. Imine. Credit card fraud detection in the era of disruptive technologies: A systematic review. *Journal of King Saud University - Computer and Information Sciences*, 35(1):145–174, Jan 2023. doi: 10.1016/j.jksuci.2022.11.008.

[29] D. A. Cieslak and N. V. Chawla. Learning Decision Trees for Unbalanced Data. In *Machine Learning and Knowledge Discovery in Databases*, pages 241–256", isbn= 978–3–540–87479–9, doi = 10.1007/978–3–540–87479–9$_3$4, 2008.

[30] Common Sense Institute. Crime & Public Safety. `https://www.commonsenseinstituteus.org/colorado/research/crime-and-public-safety`, 2025. Accessed: 2025-09-20.

[31] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

[32] E. G. Dada, T. Mapayi, O. M. Olaifa, and P. A. Owolawi. Credit Card Fraud Detection using k-star Machine Learning Algorithm, 2025.

[33] J. Dai, Q. Liu, X. Zou, and C. Zhang. Feature selection based on fuzzy combination entropy considering global and local feature correlation. *Information Sciences*, 652:119753, Jan 2024. doi: 10.1016/j.ins.2023.119753.

[34] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41(10): 4915–4928, 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2014.02.026.

[35] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi. Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3784–3797, 2018. doi: 10.1109/TNNLS.2017.2736643.

[36] J. Davis and M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[37] A. de la Cruz Huayanay, J. L. Bazán, and C. M. Russo. Performance of evaluation metrics for classification in imbalanced data. *Computational Statistics*, August 2024. ISSN 1613-9658. doi: 10.1007/s00180-024-01539-5.

[38] A. V. Dorogush, V. Ershov, and A. Gulin. CatBoost: Gradient boosting with categorical features support, 2018. URL `http://arxiv.org/abs/1810.11363`.

[39] P. D'Urso, L. D. Giovanni, and R. Massari. Smoothed Self-Organizing Map for Robust Clustering. *Information Sciences*, 512: 381–401, Feb 2020. doi: 10.1016/j.ins.2019.06.038.

[40] G. Dzemyda, O. Kurasova, and J. Žilinskas. *Multidimensional Data Visualization*. Springer Optimization and Its Applications. Springer New York, NY, 2013. ISBN 978-1-4419-0236-8. doi: 10.1007/978-1-4419-0236-8.

[41] G. Dzemyda, M. Sabaliauskas, and V. Medvedev. Geometric MDS Performance for Large Data Dimensionality Reduction and Visu-

alization. *Informatica*, 33(2):299–320, 2022. ISSN 0868-4952. doi: 10.15388/22-INFOR491.

[42] G. Egozi and R. Verma. Phishing Email Detection Using Robust NLP Techniques. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 7–12, 2018. doi: 10.1109/ICDMW.2018.00009.

[43] European Union. Directive (EU) 2017/1371 of the European Parliament and of the Council of 5 July 2017 on the fight against fraud to the Union's financial interests by means of criminal law. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017L1371`, July 2017. Official Journal of the European Union, L 198, pp. 29–41.

[44] H. Fanai and H. Abbasimehr. A novel combined approach based on deep Autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, 217:119562, 2023. ISSN 0957-4174. doi: 10.1016/j.eswa.2023.119562.

[45] T. Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010.

[46] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?, 10 2014.

[47] P. M. Figliola. The EMV chip card transition: Background, status, and issues for congress, 2015. URL `https://sgp.fas.org/crs/misc/R43925.pdf`.

[48] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479:448–455, 2019. ISSN 0020-0255. doi: 10.1016/j.ins.2017.12.030.

[49] M. Gabryel, M. M. Scherer, Ł. Sułkowski, and R. Damaševičius. Decision Making Support System for Managing Advertisers by Ad Fraud Detection. *Journal of Artificial Intelligence and Soft Computing Research (JAISCR)*, 11(4), 2021. doi: 10.2478/jaiscr-2021-0020.

[50] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1.

[51] R. K. Goel. Uncharitable acts in charity: Socioeconomic drivers of charity-related fraud. *Social Science Quarterly*, 101(4):1397–1412,

2020.

[52] I. J. Good. *Probability and the Weighing of Evidence*. Charles Griffin & Company Ltd., 1950.

[53] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[54] R. H. Assaad and S. Fayek. Predicting the Price of Crude Oil and its Fluctuations Using Computational Econometrics: Deep Learning, LSTM, and Convolutional Neural Networks. *Econometric Research in Finance*, 6:119–137, 2021. doi: 10.2478/erfin-2021-0006.

[55] I. Y. Hafez, A. Y. Hafez, A. Saleh, A. A. A. El-Mageed, and A. A. Abohany. A Systematic Review of AI-Enhanced Techniques in Credit cCrd Fraud Detection. *Journal of Big Data*, 12(1):6, 2025. ISSN 2196-1115. doi: 10.1186/s40537-024-01048-8.

[56] P. Hajek, M. Z. Abedin, and U. Sivarajah. Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework. *Information Systems Frontiers*, 25:1985–2003, 2023. doi: 10.1007/s10796-022-10346-6.

[57] J. T. Hancock, T. M. Khoshgoftaar, and J. M. Johnson. Evaluating classifier performance with highly imbalanced Big Data. *Journal of Big Data*, 10(1):42, 2023. doi: 10.1186/s40537-023-00724-5.

[58] B. Harris. Sparkov: Synthetic data generation tool for Apache Spark. `https://github.com/namebrandon/Sparkov`, 2022. Accessed on July 30, 2023.

[59] A. B. Hassanat, A. S. Tarawneh, G. A. Altarawneh, and A. Almuhaimeed. Stop Oversampling for Class Imbalance Learning: A Critical Review, 2022. URL `https://arxiv.org/abs/2202.03579`.

[60] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 2009. doi: 10.1109/TKDE.2008.239.

[61] M. Herland, R. Bauder, and T. Khoshgoftaar. The effects of class rarity on the evaluation of supervised healthcare fraud detection models. *Journal of Big Data*, 6(1):21, 2019. doi: 10.1186/s40537-019-0181-8.

[62] C. Hill, A. Hunter, L. Johnson, and A. Coustasse. Medicare Fraud in the United States: Can it Ever be Stopped? *The Health Care*

Manager, 33(3):254–260, July/September 2014. doi: 10.1097/HCM.0000000000000019.

[63] S. Huang, H. Chen, T. Li, H. Chen, and C. Luo. Feature selection via minimizing global redundancy for imbalanced data. *Applied Intelligence*, 52(8):8685–8707, 2022. doi: 10.1007/s10489-021-02855-9.

[64] E. Ileberi, Y. Sun, and Z. Wang. Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. *IEEE Access*, 9:165286–165294, 2021. doi: 10.1109/ACCESS.2021.3134330.

[65] Y. Irvin-Erickson. Identity fraud victimization: a critical review of the literature of the past two decades. *Crime Science*, 13(1):3, February 2024. ISSN 2193-7680. doi: 10.1186/s40163-024-00202-0.

[66] M. A. Islam, M. A. Uddin, S. Aryal, and G. Stea. An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. *Journal of Information Security and Applications*, 78:103618, 2023. ISSN 2214-2126. doi: 10.1016/j.jisa.2023.103618.

[67] A. Janavičiutė, A. Liutkevičius, G. Dabužinskas, and N. Morkevičius. Experimental Evaluation of Possible Feature Combinations for the Detection of Fraudulent Online Shops. *Applied Sciences*, 14(2):919, 2024. doi: 10.3390/app14020919.

[68] S. Jiang, R. Dong, J. Wang, and M. Xia. Credit Card Fraud Detection Based on Unsupervised Attentional Anomaly Detection Network. *Systems*, 11(6):305, 2023. doi: 10.3390/systems11060305.

[69] J. Johnson and T. Khoshgoftaar. Medicare fraud detection using neural networks. *Journal of Big Data*, 6(1):63, 2019. doi: 10.1186/s40537-019-0225-0.

[70] J. M. Johnson and T. M. Khoshgoftaar. Hcpcs2Vec: Healthcare Procedure Embeddings for Medicare Fraud Prediction. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing*, pages 145–152, 2020. doi: 10.1109/CIC50333.2020.00026.

[71] J. M. Johnson and T. M. Khoshgoftaar. Encoding Techniques for High-Cardinality Features and Ensemble Learners. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science*, pages 355–361, 2021. doi: 10.1109/IRI51335.2021.00055.

[72] J. Jordon, L. Szpruch, F. Houssiau, M. Bottarelli, G. Cherubin, C. Maple, S. N. Cohen, and A. Weller. Synthetic Data: An Overview of Its Benefits and Challenges, 2022. URL https://royalsociety.org/-/media/policy/projects/privacy-enhancing-technologies/Synthetic_Data_Survey-24.pdf. Commissioned by The Royal Society.

[73] R. Kanapickienė and Živilė Grundienė. The Model of Fraud Detection in Financial Statements by Means of Financial Ratios. *Procedia - Social and Behavioral Sciences*, 213:321–327, 2015. ISSN 1877-0428. doi: 10.1016/j.sbspro.2015.11.545.

[74] P. Kaur and A. Gosain. Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. In *Advances in Intelligent Systems and Computing*, volume 653, 2018. doi: 10.1007/978-981-10-6602-3_3.

[75] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[76] C. Kieffer and G. Mottola. Understanding and combating investment fraud. *Financial decision making and retirement security in an aging world*, 185, 2017.

[77] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982. ISSN 1432-0770. doi: 10.1007/BF00337288.

[78] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78 (9):1464–1480, 1990. doi: 10.1109/5.58325.

[79] M. Koziarski. Radial-Based Undersampling for imbalanced data classification. *Pattern Recognition*, 102, 2020. doi: 10.1016/j.patcog.2020.107262.

[80] M. Kubat and S. Matwin. Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, pages 179–186, 1997.

[81] V. Kumar, U. Kumar, and D. de Grosbois. Collaboration in Combating Identity Fraud. *Journal Name Here*, 28, dec 2007.

[82] A. Langousis and A. A. Carsteanu. Undersampling in action and at scale: application to the COVID-19 pandemic. *Stochastic Environmental Research and Risk Assessment*, 34(8), 2020. doi: 10.1007/s00477-020-01821-0.

[83] Legal Information Institute, Cornell Law School. Credit Card Fraud. URL https://www.law.cornell.edu/wex/credit_card_fraud. Accessed: 2025-07-22.

[84] Y.-T. Lei, C.-Q. Ma, Y.-S. Ren, X.-Q. Chen, S. Narayan, and A. N. Q. Huynh. A distributed deep neural network model for credit card fraud detection. *Finance Research Letters*, 58:104547, 2023. ISSN 1544-6123. doi: 10.1016/j.frl.2023.104547.

[85] W. Li, C. shu Wu, and S. mei Ruan. CUS-RF-Based Credit Card Fraud Detection with Imbalanced Data. *Journal of Risk Analysis and Crisis Response (JRACR)*, 12(3):110–123, 2022. doi: 10.54560/jracr.v12i3.332.

[86] Y. Li, T. Li, and H. Liu. Recent advances in feature selection and its applications. *Knowledge and Information Systems*, 53(3):551–577, 2017. doi: 10.1007/s10115-017-1059-8.

[87] E. A. Lopez-Rojas and S. Axelsson. BankSim: A Bank Payment Simulation for Fraud Detection Research. In *The 26th European Modeling and Simulation Symposium*, 2014. [Online]. Available: https://www.researchgate.net/publication/265736405.

[88] Y. Lu, Y.-M. Cheung, and Y. Y. Tang. Bayes Imbalance Impact Index: A Measure of Class Imbalanced Data Set for Classification Problem. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3525–3539, 2020. doi: 10.1109/TNNLS.2019.2944962.

[89] T.-D. Mai, K. Hoang, A. Baigutanova, G. Alina, and S. Kim. Customs Fraud Detection in the Presence of Concept Drift, 2021. URL https://arxiv.org/abs/2109.14155.

[90] J. Mamčenko and B. Šustickienė. Mokėjimo kortelių saugumas ir kredito kortelių sukčiavimo prevencija. *Technologijos ir menas*, (12): 24–30, 2021. URL https://vtdko.lt/wp-content/uploads/2022/10/Technologijos_ir_menas_2021_12.pdf.

[91] A. Maratea, A. Petrosino, and M. Manzo. Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*,

257:331–341, 2014. ISSN 0020-0255. doi: 10.1016/j.ins.2013.04.016.

[92] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947. doi: 10.1007/BF02295996.

[93] D. Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1), 2001. doi: 10.1145/507533.507538.

[94] J. Moeyersoms and D. Martens. Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector. *Decision Support Systems*, 72:72–81, 2015. doi: 10.1016/j.dss.2015.02.007.

[95] R. C. Moore, D. P. W. Ellis, E. Fonseca, S. Hershey, A. Jansen, and M. Plakal. Dataset Balancing Can Hurt Model Performance. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 1–5, June 2023. doi: 10.1109/icassp49357.2023.10095255.

[96] C. Mougan, J. M. Álvarez, S. Ruggieri, and S. Staab. Fairness Implications of Encoding Protected Categorical Attributes. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 454–465, 2023.

[97] M. A. Munson and R. Caruana. On Feature Selection, Bias-Variance, and Bagging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 144–159. Springer, 2009.

[98] Z. Noroozi, A. Orooji, and L. Erfannia. Analyzing the impact of feature selection methods on machine learning algorithms for heart disease prediction. *Scientific Reports*, 13(1):22588, 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-49962-w.

[99] C. Ordonez. Clustering binary data streams with K-means. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, page 12–19, New York, NY, USA, 2003. ISBN 9781450374224. doi: 10.1145/882082.882087.

[100] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, 37

(5):2671–2692, 2022. doi: 10.1007/s00180-022-01207-6.

[101] S. H. Park and Y. G. Ha. Large Imbalance Data Classification Based on MapReduce for Traffic Accident Prediction. In *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 45–49, 2014. doi: 10.1109/IMIS.2014.6.

[102] M. S. A. Patwary, D. V. Gokhale, and M. Rahman. On Testing Equality of Two Independent Proportions: To Pool or Not To Pool. *Far East Journal of Theoretical Statistics*, 52(3):193–213, 2016. doi: 10.17654/TS052030193.

[103] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python.

[104] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[105] J. Perols. Financial Statement Fraud Detection: An Analysis of Statistical and Machine Learning Algorithms. *AUDITING: A Journal of Practice & Theory*, 30(2):19–50, May 2011. doi: 10.2308/ajpt-50009.

[106] B. Pes and G. Lai. Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. *PeerJ Comput. Sci.*, 7:e832, Dec 2021. doi: 10.7717/peerj-cs.832.

[107] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: unbiased boosting with categorical features, 2019. URL `https://arxiv.org/abs/1706.09516`.

[108] F. Provost. Machine Learning from Imbalanced Data Sets 101. In *Proceedings of the AAAI 2000 Workshop on Imbalanced Data Sets*, New York University, 2000. Extended Abstract.

[109] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1): 81–106, 1986. doi: 10.1023/A:1022643204877.

[110] M. Rad, A. Amiri, M. H. Ranjbar, H. Salari, and D. McMillan. Predictability of financial statements fraud-risk using Benford's Law. *Cogent Economics & Finance*, 9(1), 2021. doi: 10.1080/23322039.2021.1889756.

[111] S. M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Academic Press, 5th edition, 2014.

[112] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C), 1987. doi: 10.1016/0377-0427(87)90125-7.

[113] Y. Russac, O. Caelen, and L. He-Guelton. Embeddings of Categorical Variables for Sequential Data in Fraud Context. In *Advances in Intelligent Systems and Computing*, 2018. doi: 10.1007/978-3-319-74690-6_53.

[114] T. Ruzgas, L. Kižauskienė, M. Lukauskas, E. Sinkevicius, M. Frolovaite, and J. Arnastauskaite. Tax Fraud Reduction Using Analytics in an East European Country. *Axioms*, 12(3):288, 2023. doi: 10.3390/axioms12030288.

[115] O. Sagi and L. Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8 (4), 2018. doi: 10.1002/widm.1249.

[116] Z. Salekshahrezaee, J. L. Leevy, and T. M. Khoshgoftaar. The effect of feature extraction and data sampling on credit card fraud detection. *Journal of Big Data*, 10(1):6, 2023. doi: 10.1186/s40537-023-00684-w.

[117] G. Sas and G. Bouman. *Practical Fraud Prevention: Fraud and AML Analytics for Fintech and eCommerce, Using SQL and Python*. O'Reilly Media, 2021.

[118] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011. doi: 10.1109/ICDCSW.2011.20.

[119] J. T. Schaefer. The critical success index as an indicator of warning skill. *Weather and Forecasting*, 5(4):570 – 575, 1990. doi: 10.1175/1520-0434(1990)005<0570:TCSIAA>2.0.CO;2.

[120] S. Schuh and J. Stavins. How Do Speed and Security Influence Consumers' Payment Behavior? *FRB of Boston Public Policy Discussion Paper*, 15(1), February 2015. URL `https://ssrn.com/abstract=2675307`.

[121] C. E. Shannon. A Mathematical Theory of Communication. *The*

Bell System Technical Journal, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[122] A. Singh and A. Jain. An efficient credit card fraud detection approach using cost-sensitive weak learner with imbalanced dataset. *Computational Intelligence*, 38(6):2035–2055, 2022. doi: 10.1111/coin.12555.

[123] A. Slakey, D. Salas, and Y. Schamroth. Encoding Categorical Variables with Conjugate Bayesian Models for WeWork Lead Scoring Engine, 2019. [Online]. Available: http://arxiv.org/abs/1904.13001.

[124] R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. W.H. Freeman and Company, San Francisco, 1963.

[125] M. Soltani, A. Kythreotis, and A. Roshanpoor. Two decades of financial statement fraud detection literature review: combination of bibliometric analysis and topic modeling approach. *Journal of Financial Crime*, 30(5), 2022. ISSN 1359-0790. doi: 10.1108/JFC-09-2022-0227.

[126] E. Stankevicius and L. Leonas. Hybrid Approach Model for Prevention of Tax Evasion and Fraud. *Procedia - Social and Behavioral Sciences*, 213:383–389, 2015. ISSN 1877-0428. doi: 10.1016/j.sbspro.2015.11.555.

[127] J. Surowiecki. *The Wisdom of Crowds*. Anchor, 2004.

[128] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour. Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7(1):70, 2020. doi: 10.1186/s40537-020-00349-y.

[129] F. Teichmann. Ransomware attacks in the context of generative artificial intelligence—an experimental study. *International Cybersecurity Law Review*, 4(4):399–414, December 2023. ISSN 2662-9739. doi: 10.1365/s43439-023-00094-x.

[130] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves. Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513:429–441, 2020. ISSN 0020-0255. doi: 10.1016/j.ins.2019.11.004.

[131] J. Tian, M. H. Azarian, and M. Pecht. Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm. *PHM Society European Conference*, 2(1), 2014. doi: 10.36001/phme.

2014.v2i1.1554.

[132] A. Uyar, A. Bener, H. N. Ciray, and M. Bahceci. A frequency based encoding technique for transformation of categorical variables in mixed IVF dataset. In *31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine*, pages 6214–6217, 2009. doi: 10.1109/IEMBS.2009.5334548.

[133] S. M. V and T. Mahalakshmi. An Empirical Study on the Effect of Resampling Techniques in Imbalanced Datasets for Improving Consistency of Classifiers. *International Journal of Applied Engineering Research*, 14(7):1516–1525, 2019. ISSN 0973-4562. URL https://www.ripublication.com/ijaer19/ijaerv14n7_12.pdf.

[134] I. Valentim, N. Lourenço, and N. Antunes. The Impact of Data Preparation on the Fairness of Software Systems. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, pages 391–401. IEEE, 2019.

[135] S. Varshney. The Limits of SCA and Why Fraud Protection Is More Important than Ever, 2021. URL https://www.finextra.com/blogposting/20416/the-limits-of-sca-and-why-fraud-protection-is-more-important-than-ever. Accessed: 2024-05-16.

[136] A. Verikas, M. Bacauskienė, D. Valincius, and A. Gelzinis. Predictor Output Sensitivity and Feature Similarity-Based Feature Selection. *Fuzzy Sets and Systems*, 159(4):422–434, 2008. doi: 10.1016/j.fss.2007.05.020.

[137] S. Viaene and G. Dedene. Insurance Fraud: Issues and Challenges. *The Geneva Papers on Risk and Insurance - Issues and Practice*, 29(2):313–333, April 2004. ISSN 1468-0440. doi: 10.1111/j.1468-0440.2004.00290.x.

[138] T. Vogl, C. Schmied, L. S. Sloth, M. Perslev, M. Nielsen, and C. Igel. Impact of Preprocessing in Radiomics: Beware of Data Leakage. *Scientific Reports*, 14(1):9470, 2024. doi: 10.1038/s41598-024-62585-z.

[139] M. Vorndran, A. Schütz, J. Bendix, and B. Thies. Current Training and Validation Weaknesses in Classification-Based Radiation Fog Nowcast Using Machine Learning Algorithms. *Artificial Intelli-*

gence for the Earth Systems, 1(2), 2022. doi: 10.1175/AIES-D-21-0006.1.

[140] I. Vosyliūtė and N. Maknickienė. Investigation of Financial Fraud Detection by Using Computational Intelligence. In *12th International Scientific Conference Business and Management 2022*, 2022. doi: 10.3846/bm.2022.787. Article Number: bm.2022.787.

[141] T. Vyšniūnas, D. Čeponis, N. Goranin, and A. Čenys. Risk-Based System-Call Sequence Grouping Method for Malware Intrusion Detection. *Electronics*, 13(1):206, 2024. doi: 10.3390/electronics13010206.

[142] M. Ward. EMV card payments – An update. *Information Security Technical Report*, 11(2):89–92, 2006. ISSN 1363-4127. doi: 10.1016/j.istr.2006.03.001.

[143] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature Hashing for Large Scale Multitask Learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 1113–1120, 2009.

[144] C. G. Weng and J. Poon. A new evaluation measure for imbalanced datasets. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*, pages 27–32, 2008.

[145] Q. Xiao, H. Li, J. Tian, and Z. Wang. Group-Wise Feature Selection for Supervised Learning. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3149–3153, 2022. doi: 10.1109/ICASSP43922.2022.9746666.

[146] X. Xie, H. Liu, S. Zeng, L. Lin, and W. Li. A novel progressively undersampling method based on the density peaks sequence for imbalanced data. *Knowledge-Based Systems*, 213, 2021. doi: 10.1016/j.knosys.2020.106689.

[147] Z. Xie and X. Huang. A Credit Card Fraud Detection Method Based on Mahalanobis Distance Hybrid Sampling and Random Forest Algorithm. *IEEE Access*, 12:162788–162798, 2024. doi: 10.1109/ACCESS.2024.3421316.

[148] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan. Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3–11, 2013. doi: 10.1016/j.neucom.2012.04.039.

[149] J. Zhai, J. Qi, and C. Shen. Binary imbalanced data classification based on diversity oversampling by generative models. *Informa-*

tion Sciences, 585, 2022. doi: 10.1016/j.ins.2021.11.058.

[150] C. Zhao, X. Sun, M. Wu, and L. Kang. Advancing financial fraud detection: Self-attention generative adversarial networks for precise and effective identification. *Finance Research Letters*, 60:104843, 2024. doi: 10.1016/j.frl.2023.104843.

[151] X. Zhou. Shrinkage Estimation of Log-odds Ratios for Comparing Mobility Tables. *Sociol Methodology*, 45(1):320–356, 2015. doi: 10.1177/0081175015570097.

[152] Q. Zhu. On the Performance of Matthews Correlation Coefficient (MCC) for Imbalanced Dataset. *Pattern Recognition Letters*, 136: 71–80, 2020. ISSN 0167-8655. doi: 10.1016/j.patrec.2020.03.030.

[153] R. Zuech, J. Hancock, and T. M. Khoshgoftaar. Detecting web attacks using random undersampling and ensemble learners. *Journal of Big Data*, 8(1), 2021. doi: 10.1186/s40537-021-00460-8.

List of features employed for model training in DataSet1:

- **Current Age** - current age of the card owner.
- **Retirement Age** - retirement age of the card owner.
- **Zipcode** - zipcode of the card owner.
- **Per Capita Income - Zipcode** - per capita income grouped by zipcode of the card owner.
- **Yearly Income - Person** - yearly income of the card owner.
- **Total Debt** - card owner' total debt.
- **FICO Score** - is used by lenders to help make accurate, reliable, and fast credit risk decisions across the customer life cycle. The credit risk score rank-orders consumers by how likely they are to pay their credit obligations as agreed. Even though score intervals vary depending on the credit scoring model, credit scores from 580 to 669 are generally treated as fair; 670 to 739 are treated as a good; 740 to 799 are treated as very good, and 800 and up are treated as an excellent.
- **Num Credit Cards** - number of cards owned by the same person.
- **Credit Limit** - credit limit of the card.
- **Gender_Male** - gender of the card owner.
- **CardBrand_Discover** - binary feature representing if the card is "Discover".
- **CardBrand_Visa** - binary feature representing if the card is "Visa" (Otherwise, card is "MasterCard").
- **CardType_Debit** - binary feature representing if the card is Debit.
- **CardType_Debit (Prepaid)** - binary feature representing if the card is Debit Prepaid (Otherwise, the card is Credit).

- **HasChip_YES** - binary feature representing if the card has a chip. Chips are the small, square computer chips that appear on debit, credit and prepaid cards to help safeguard them against fraud.
- **Month** - the number of the month when transaction was made.
- **Day** - the number of the day when transaction was made.
- **MCC** - id of the merchant. For instance, Apple (MCC=5045) or McDonalds (MCC=5814).
- **City_cat** - city of the card owner.
- **Merchant_City_cat** - city of the merchant.
- **State_cat** - State of the card owner.
- **Use Chip_Online Transaction** - binary feature representing if the the transaction was made online.
- **Use_Chip_Swipe_Transaction** - binary feature representing if the the transaction was made by swiping through the card reader.
- **Valid_in_Days** - number of days until card will be expired.
- **hour_bin** - hour bin, for instance 12:00-13:00, when transaction was made.
- **Amount** - transferred amount.
- **Error_cat1** and **Error_cat2** - error that happen during the transaction.
- **Is_Fraud_Yes** - target feature. It is binary feature representing if the transaction is labeled as fraudulent or regular.

List of features employed for model training in DataSet2:

- **Category** - category of the transaction.
- **Amount** - transferred amount.
- **Gender** - gender of the card owner.
- **City** - city of the card owner.

- **State** - state of the card owner.
- **City Population** - population of the city where the card owner resides.
- **Job** - occupation of the card owner.
- **Month** - the number of the month when the transaction was made.
- **Day** - the number of the day when the transaction was made.
- **Hour Bin** - hour bin, for instance 12:00-13:00, when the transaction was made.
- **Birth Year** - birth year of the card owner.

List of features employed for model training in DataSet3:

- **V1, V2, . . . , V28**: Principal components of real features obtained with PCA.
- **Time**: Seconds elapsed between each transaction and the first transaction in the dataset.
- **Amount**: Transaction amount.
- **Class**: target feature.

Articles in Clarivate Web of Science journals:

[A.1] Breskuvienė, Dalia; Dzemyda, Gintautas. Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions // *International Journal of Computers Communications & Control*. Oradea: Agora University. ISSN 1841-9836. eISSN 1841-9844. 2023, vol. 18, iss. 3, art. no. 5433, p. [1-17]. DOI: 10.15837/ijccc.2023.3.5433.

[A.2] Breskuvienė, Dalia; Dzemyda, Gintautas. Enhancing credit card fraud detection: highly imbalanced data case // *Journal of Big Data*. ISSN 2196-1115. 2024, vol. 11, iss. 1, sp. [182]. DOI: 10.1186/s40537-024-01059-5.

Chapter in the peer-reviewed scientific book

[B.1] Breskuvienė, Dalia; Dzemyda, Gintautas. Imbalanced data classification approach based on clustered training set // *Data Science in Applications* / Editors: Dzemyda, G., Bernatavičienė, J., Kacprzyk, J. Cham: Springer, 2023. ISBN 9783031244520. eISBN 9783031244537. p. 43-62. (*Studies in Computational Intelligence*, ISSN 1860-949X, eISSN 1860-9503; vol. 1084). DOI: 10.1007/978-3-031-24453-7_3.

Posters in international conferences with published abstracts

[C.1] Breskuvienė, Dalia; Dzemyda, Gintautas. Clustering-based optimization in fraud detection classifier training // *EURO 2022: [32nd European Conference on Operational Research (EURO XXXII)]*, Espoo, Finland, July 3-6, 2022: Abstract Book. Espoo: Aalto University, 2022. ISBN 9789519525419. p. 152. Available online: `https://www.euro-online.org/conf/admin/tmp/program-euro32.pdf`.

Posters in international conferences

[D.1] Breskuvienė, Dalia; Dzemyda, Gintautas. Adapt or fall behind: A deep dive into machine learning techniques for detection of the evolving fraud in the financial realm // *13th Annual Counter Fraud, Cybercrime and Forensic Accounting Conference*, June 12–13, 2024, Portsmouth, UK.

Posters in national conferences with published abstracts

[E.1] Breskuvienė, Dalia; Dzemyda, Gintautas. Highly imbalanced data case: pattern-guided feature selection to detect financial fraud // *DAMSS: 15th Conference on Data Analysis Methods for Software Systems*, Druskininkai, Lithuania, November 28-30, 2024. Vilnius: Vilniaus universiteto leidykla, 2024. eISBN 9786090711125. p. 12-13. (*Vilnius University Proceedings*, eISSN 2669-0233; vol. 52). DOI: 10.15388/DAMSS.15.2024.

[E.2] Breskuvienė, Dalia; Dzemyda, Gintautas. What is a concept drift, and does it affect machine learning performance? // *DAMSS: 14th Conference on Data Analysis Methods for Software Systems*, Druskininkai, Lithuania, November 30 - December 2, 2023. Vilnius: Vilniaus universiteto leidykla, 2023. eISBN 9786090709856. p. 14. (*Vilnius University Proceedings*, eISSN 2669-0233; vol. 39). DOI: 10.15388/DAMSS.14.2023.

[E.3] Breskuvienė, Dalia; Dzemyda, Gintautas. Autoencoder for fraudulent transactions data feature engineering // *DAMSS: 13th Conference on Data Analysis Methods for Software Systems*, Druskininkai, Lithuania, December 1–3, 2022. Vilnius: Vilniaus universiteto leidykla, 2022. ISBN 9786090707944. eISBN 9786090707951. p. 11. (*Vilnius University Proceedings*, eISSN 2669-0233; vol. 31). DOI: 10.15388/DAMSS.13.2022.

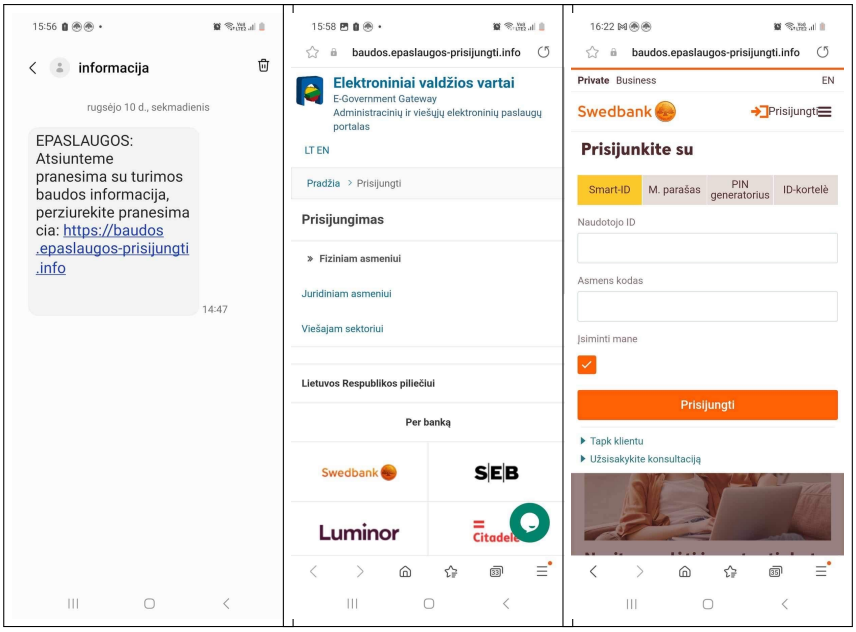
CURRICULUM VITAE

Dalia Breskuvienė is a doctoral candidate at the Institute of Data Science and Digital Technologies at Vilnius University. She holds a Bachelor's degree in Statistics and a Master's degree in Mathematics from Vilnius University. Her academic interests lie at the intersection of data science, machine learning, and financial crime detection, with particular emphasis on optimizing imbalanced training datasets for improved classification performance. In parallel to her doctoral studies, Dalia has accumulated extensive professional experience in the financial sector, working as a Senior Data Scientist and Interim Product Owner at Danske Bank and currently as a Data Scientist at If P&C Insurance. Her professional expertise includes predictive modeling, customer segmentation, fraud analytics, and data visualization, employing advanced tools and methodologies including Python, SQL, R, and Tableau. In addition to her industry engagements, she is actively involved in academic teaching at Vilnius University and through Erasmus+ teaching initiatives abroad. The commitment to methodological rigor, practical relevance, and the pursuit of innovative solutions to complex data-driven challenges characterize her work.

Finansinis sukčiavimas (angl. *Financial fraud*) yra plačiai paplitusi problema, turinti reišmingų ekonominių pasekmių: blogėja privačių įmonių finansinė būklė ir stabilumas, prastėja viešųjų paslaugų kokybė, mažėja disponuojamų pajamų lygis bei mažėja išteklių kiekis labdaros organizacijoms. Sukčiavimas reikšmingai ir neigiamai veikia gyvenimo kokybę visose srityse ir visose šalyse. Įvairūs sukčiavimo scenarijai skiriasi patiriamų nuostolių dydžiu ir naudojamų technikų sudėtingumu.

Šioje disertacijoje nagrinėjamas kredito kortelių sukčiavimo aptikimas iš finansinių institucijų perspektyvos. Vartojamas terminas „kredito kortelių sukčiavimas“ atitinka nusistovėjusią vartoseną Lietuvos finansinių institucijų dokumentuose bei yra vartojamas akademiniuose šaltiniuose tokiuose kaip [90]. Kredito kortelių sukčiavimas – tai tapatybės vagystės forma, kai kito asmens kredito kortelės informacija neteisėtai pasisavinama ir naudojama pirkimams arba pinigų išėmimui iš sąskaitos be savininko žinios ar leidimo [83]. Yra daugybė būdų, kaip pasisavinti kreditinių kortelių duomenis. Vienas iš palyginti nesudėtingų, tačiau veiksmingų metodų yra siųsti nuorodas asmenims, siekiant paskatinti juos atlikti pirkimus ar pavedimus. Paveikslėlyje S.1 iliustruojamas sukčiavimo atvejis, kuris atitinka tipinį duomenų išviliojimo (angl. *phishing*) scenarijų, skirtą pavogti banko prisijungimo duomenis ir atlikti neteisėtus sandorius. Toks procesas prasideda nuo SMS žinutės, imituojančios patikimą instituciją, pavyzdžiui, valdžios instituciją ar banką, kuri raginama gavėją skubiai patikrinti pateiktą pranešimą per atsiųstą nuorodą. Paspaudus nuorodą, vartotojas nukreipiamas į apgaulingą svetainę, kuri vizualiai primena tikrą prisijungimo puslapį, siekiant išvilioti jo prisijungimo duomenis. Įvedus duomenis, jie patenka į sukčių rankas, suteikdami prieigą prie tikrosios banko sąskaitos. Tai leidžia sukčiams inicijuoti neteisėtus sandorius, sukeliančius finansinius nuostolius. Šis pavyzdys parodo, kaip lengvai gali būti pavogti ir panaudoti kreditinių kortelių duomenys, dažnai aukai to net nesuvokiant. Kai sukčiai įgyja šiuos duomenis, finansinėms institucijoms tenka užduotis per milisekundes juos aptikti ir sustabdyti.

Šioje disertacijoje kreditinių kortelių sukčiavimas apibrėžiamas kaip stadija, kai nusikaltėlis jau turi kortelės duomenis (kortelės numerį, galiojimo datą, savininko vardą, CVV kodą) ir siekia juos panaudoti pirkiniams ar paslaugų įsigijimams, siekdamas pasisavinti aukos lėšas.



S.1 pav. Trys žingsniai iki pinigų praradimo: SMS su nuoroda; apgaulinga svetainė; pinigų pervedimas

Sparčiai tobulėjant skaitmeninėms technologijoms ir daugėjant internetinių finansinių operacijų, sukčiavimo būdai tampa vis sudėtingesni bei sunkiau aptinkami tradicinėmis priemonėmis, todėl finansinės institucijos vis dažniau pasitelkia dirbtinio intelekto (DI) ir mašininio mokymosi algoritmus efektyvesniam sukčiavimo aptikimui. Pasitelkusios pažangius DI gebėjimus, institucijos gali proaktyviai aptikti ir užkirsti kelią sukčiavimo atvejams ir apsaugoti klientus, stiprinti savo reputaciją. Mašininio mokymosi algoritmai vis plačiau taikomi sukčiavimo prevencijai [68], [85]. Vis dėlto taikant mašininį mokymąsi duomenims, sudarytiems iš kreditinių kortelių transakcijų, kyla ne vienas iššūkis, iš kurių svarbiausias yra duomenų disbalansas – sukčiavimo atvejų paprastai būna mažiau nei 1 % visų transakcijų.

Kreditinių kortelių sukčiavimo aptikimą, taikant mašininio mokymosi metodus, apsunkina keli veiksniai: didelis kategorinių kintamųjų kiekis duomenyse [12], viešai prieinamų duomenų bazių trūkumas moksliniams tyrimams, netinkamas kai kurių vertinimo rodiklių taikymas nesubalansuotų duomenų atveju [35] bei koncepcijos poslinkis

(angl. *concept drift*) – reiškinys, kai laikui bėgant keičiasi duomenų paskirstymas arba ryšys tarp požymių ir tikslo kintamojo [89]. Sukčiavimo aptikimo algoritmai turi veikti ypač greitai – daugelis finansinių institucijų nustato milisekundėmis matuojamus laiko limitus sandorio įvertinimui [18].

Tyrimų sritis

Šiame tyrime daugiausia dėmesio skiriama nesubalansuotų duomenų optimizavimui sprendžiant klasifikavimo uždavinius, tokius kaip, finansinio sukčiavimo aptikimas taikant kategorinių duomenų kodavimo ir požymių atrankos metodus siekiant padidinti aptikimo tikslumą ir interpretuojamumą. Taip pat tiriama požymių konvertavimo į naują skaitinę erdvę galimybė kaip tarpinė požymių atrinkimo proceso dalis.

Tyrimo problema

Aptikti finansinį sukčiavimą sudėtinga, nes reikia spręsti itin didelę klasių nesubalansuotumo problemą, atrinkti informatyviausius požymius ir efektyviai koduoti kategorinius kintamuosius. Tradiciniai požymių atrankos ir kategorinio kodavimo metodai neužtikrina tikslaus ir kokybiško mašininiu mokymusi grįsto sukčiavimo aptikimo, todėl gaunama pernelyg daug klaidingai teigiamų rezultatų arba praleidžiama sukčiavimo atvejų.

Darbo aktualumas

Finansinis sukčiavimas – tai tyčinis apgaulės veiksmas, padarytas siekiant neteisėtai pasipelnyti arba padaryti nuostolių kitai šaliai. Šis apibrėžimas atitinka Direktyvos (ES) 2017/1371 dėl Europos Sąjungos finansinių interesų apsaugos baudžiamosios teisės priemonėmis 3 straipsnio 2 dalyje pateiktą teisinį pagrindą [43]. Finansinio sukčiavimo nustatymas yra vienas iš aktualiausių šiuolaikinio finansų sektoriaus iššūkių, nes net vienas nenustatytas sukčiavimo atvejis gali sukelti didelių finansinių nuostolių ir pakenkti įstaigos reputacijai. Nors egzistuoja įvairūs dirbtiniu intelektu pagrįsti metodai nesubalansuotų duomenų klasifikavimui, tačiau jie dažnai nesugeba efektyviai aptikti sukčiavimo atvejų, kai duomenų disbalansas yra itin didelis, t.y. mažas

sukčiavimo atvejų kiekis bendrame transakcijų sraute. Todėl vis dar išlieka būtinybė kurti pažangesnius metodus, gebančius efektyviai veikti esant ryškiam duomenų nesubalansuotumui. Šiame darbe siūlomas metodas yra labai aktualus, nes didina sukčiavimo aptikimo tikslumą ir gali būti plačiai taikomas ne tik finansų sektoriuje, bet ir medicinos duomenų analizėje bei kibernetiniame saugume. Tyrimų rezultatai turi tiesioginę praktinę reikšmę, nes metodas buvo išbandytas su realiais finansinių sandorių duomenimis ir gali būti integruotas į esamas finansų įstaigų rizikos valdymo sistemas.

Tyrimo objektas

Šio tyrimo objektas – finansinio sukčiavimo aptikimo procesas, ypatingą dėmesį skiriant požymių atrankos metodams ir konvertavimui į kitą skaitinę erdvę. Tyrime analizuojama, kaip įvairūs požymių atrankos metodai gali padėti efektyviau parinkti informatyvius požymius, siekiant pagerinti nesubalansuotų duomenų klasifikavimo tikslumą taikant mašininio mokymosi algoritmus kreditinių kortelių sukčiavimui aptikiti.

Tyrimo tikslas ir uždaviniai

Šio tyrimo tikslas – sukurti metodą, racionaliai sumažinti esamą požymių rinkinį, leidžiantį pagerintį nesubalansuotų duomenų klasifikavimo tikslumą, siekiant geriau aptikti finansinio sukčiavimo atvejus.

Disertacijos uždaviniai:

- Pasiūlyti nuoseklią mašininio mokymosi strategiją, grindžiamą duomenų klasterizavimu į prasmingas grupes, šių grupių balansavimu bei individualiai pritaikytų klasifikavimo metodų taikymu, siekiant pagerinti sukčiavimo atvejų atpažinimą.
- Įvertinti tikslinių (angl. *target-based*) ir į tikslinį kintamąjį neatsižvelgiančių (angl. *target-agnostic*) kodavimo metodikų įtaką nesubalansuotiems duomenims klasifikuoti.
- Įvertinti saviorganizuojančio žemėlapio galimybes ir panaudojimą kaip tarpinę požymių atrinkimo proceso dalį: transakcijoms klasterizuoti ir požymiams konvertuoti į naują skaitinę erdvę siekiant geriausio požymių parinkimo tiriamame duomenų rinkinyje.

- Sukurti į nesubalansuotus duomenis orientuotą metodą esamam požymių rinkiniui pagerinti ir sumažinti, siekiant geresnių klasifikavimo rezultatų.
- Sukurti eksperimentinę aplinką, kuri užtikrina modelių gebėjimą prisitaikyti prie besikeičiančių sukčiavimo elgsenos tendencijų taikant laiku grindžiamus transakcijų duomenų aibės padalijimus (mokoma ankstesniais duomenimis nei testuojama) ir atlikti eksperimentus su viešai prieinamais anotuotais duomenimis, siekiant įrodyti siūlomo sprendimo efektyvumą.

Tyrimo metodai

Šioje disertacijoje taikomas sisteminis metodologinis požiūris į kredito kortelių sukčiavimo aptikimo tyrimą. Taikyti šie metodai:

- Buvo atlikta išsami literatūros apžvalga, apimanti sukčiavimo aprašymo tipus, nesubalansuotų duomenų iššūkius, kreditinių kortelių sukčiavimo aptikimo metodus ir kitas susijusias temas.
- Buvo lyginami įvairūs kategorinių duomenų kodavimo metodai. Siekiant nustatyti optimalias išankstinio apdorojimo strategijas, įvertintas jų poveikis sukčiavimo aptikimo rezultatams.
- Tyrime teorinės įžvalgos sujungtos su empiriniu eksperimentu, siekiant įvertinti įvairių požymių atrankos metodų efektyvumą sukčiavimui aptikti.
- Pasiūlytas požymių konvertavimo metodas buvo išbandytas naudojant etaloninius ir realaus pasaulio duomenų rinkinius. Eksperimentų metu buvo atliekamas modelių mokymas, validavimas ir efektyvumo vertinimas naudojant įvairias vertinimo metrikas, siekiant nustatyti jų veiksmingumą klasifikuojant transakcijas.

Mokslinis darbo naujumas

Šioje disertacijoje pristatomas naujas dirbtiniu intelektu pagrįstas finansinių transakcijų požymių sistemos optimizavimo metodas, specialiai sukurtas labai nesubalansuotiems duomenims. Pagrindinis mokslinis indėlis – sukurta nauja požymių konversijos sistema, leidžianti

efektyviau modeliuoti duomenis ir padidinti sukčiavimo aptikimo tikslumą. Metodas naudoja netiesines transformacijas, kad būtų galima geriau atskirti su sukčiavimu susijusias charakteristikas ir taip pagerinti klasifikavimo modelių efektyvumą. Išsamūs eksperimentai su realiais ir sintetiniais finansiniais duomenimis rodo, kad siūlomas metodas stipriai pagerina klasifikavimo rodiklius, ypač tais atvejais, kai sukčiavimo atvejai yra labai reti. Be to, pasiūlytas metodas išplečia saviorganizuojančių žemėlapių taikymo galimybes, suteikdamas naujų galimybių spręsti problemas, susijusias su nesubalansuotų duomenų klasifikavimu.

Praktinė darbo vertė

Nesubalansuoti duomenys tebėra didelis mašininio mokymosi iššūkis, ypač realiose gyvenimiškose situacijose, kuriose svarbiausia yra mažumos klasė. Ši savybė būdinga daugeliui sričių, pavyzdžiui, sukčiavimo aptikimui, klientų skaičiaus mažėjimo prognozavimui, medicininei diagnostikai ir anomalijų aptikimui kibernetinio saugumo srityje. Finansinio sukčiavimo aptikimo srityje gebėjimas tiksliai nustatyti retas sukčiavimo operacijas, kartu sumažinant klaidingai teigiamus rezultatus, yra labai svarbus siekiant sumažinti finansinius nuostolius ir veiklos sąnaudas. Tyrimu prisidedama prie šios srities plėtojant ir vertinant požymių atrankos metodą, kuris pagerina klasifikavimo rezultatus esant dideliam klasių disbalansui. Siūlomu metodu siekiama pagerinti sukčiavimo aptikimo tikslumą, optimizuoti skaičiavimo efektyvumą ir pateikti praktinių įžvalgų finansų įstaigoms, diegiančioms sukčiavimo aptikimo modelius didelės rizikos aplinkoje.

Šis tyrimas pabrėžia veiksmingo duomenų paruošimo ir matavimų parinkimo svarbą, kad būtų išvengta perteklinių ar klaidinančių rezultatų su kredito kortelių sukčiavimo aptikimu susijusiuose tyrimuose. Eksperimentai buvo sudaryti atkartojant realaus pasaulio scenarijus, remiantis darbo patirtimi finansų įstaigose ir literatūros apžvalga. Buvo pasiūlytas požymių atrankos metodas, kuriuo siekiama pagerinti kitų požymių atrankos metodų rezultatus, padidinti klasifikatorių tikslumą.

Ginamieji teiginiai

Sukčiavimo aptikimas finansinėse operacijose išlieka sudėtinga užduotis, iš esmės dėl itin didelio klasių nesubalansuotumo ir dinamiškai

kintančio sukčiavimo elgsenos pobūdžio. Šios disertacijos ginamieji teiginiai:

- Siūloma klasterizavimu grįsta mašininio mokymo strategija, apimanti individualų klasterių balansavimą ir klasifikavimą, leidžia reikšmingai pagerinti sukčiavimo atvejų atpažinimo tikslumą (angl. *Recall*) bei sumažinti klaidingai priskirtų sukčiavimo atvejų skaičių teisėtoms transakcijoms.

- Tikslinės (angl. *target-based*) kodavimo metodikos, lyginant su tradicinėmis, į tikslinį kintamąjį neatsižvelgiančiomis (angl. *target-agnostic*) kodavimo schemomis, pasižymi pranašumu klasifikavimo užduotyse, kuriose egzistuoja didelis klasių disbalansas ir aukštas kategorinių požymių kardinalumas, nes tikslinės metodikos geba atskleisti reikšmingus statistinius ryšius, kuriuos dažnai praranda tradiciniai kodavimo būdai.

- Siūlomas FID-SOM (Feature Selection for Imbalanced Data Using SOM) metodas, pagrįstas konkurencinio mokymosi principais, taikant SOM kaip požymių konversijos mechanizmą, kuriame BMU svorio vektorių dispersija naudojama požymių svarbai įvertinti, pagerina požymių atrankos procesą užtikrindamas tikslesnį sukčiavimo aptikimą.

- Siekiant užtikrinti modelių gebėjimą prisitaikyti prie besikeičiančių sukčiavimo elgsenos tendencijų, reikia taikyti laiku grindžiamus transakcijų duomenų aibės padalijimus (mokoma ankstesniais duomenimis nei testuojama), nes tai leidžia vertinti modelio adaptyvumą prognozuojant ateities įvykius aptinkant finansinį sukčiavimą.

Tyrimo aprobavimas ir publikavimas

Šių tyrimų rezultatai pateikti ir patvirtinti publikacijomis dviejuose recenzuojamuose tarptautiniuose žurnaluose, esančiuose Q1 ir Q3 kvartilėse, taip pat recenzuojamuose knygos skyriuose ir konferencijų medžiagoje. Rezultatai pristatyti mokslinėse bendruomenėse dalyvaujant dviejose tarptautinėse ir trijose nacionalinėse konferencijose. Toliau pateiktame sąraše apžvelgiamas mokslinių tyrimų indėlis – straipsniai

žurnaluose, knygų skyriai ir pranešimai konferencijose. Tarp jų verta išskirti straipsnį *Enhancing credit card fraud detection: highly imbalanced data case*, paskelbtą WoS Q1 kvartilės žurnale *Journal of Big Data*, kuris buvo nominuotas Vilniaus universiteto rektoriaus premijai laimėti.

Publikacijos

Straipsniai tarptautiniuose mokslo žurnaluose, kurių citavimo indeksas nurodytas Clarivate Web of Science duomenų bazėje.

1. Breskuvienė, Dalia; Dzemyda, Gintautas. Enhancing credit card fraud detection: highly imbalanced data case // Journal of Big Data. ISSN 2196-1115. 2024, vol. 11, iss. 1, sp. [182]. DOI: 10.1186/s40537-024-01059-5.

2. Breskuvienė, Dalia; Dzemyda, Gintautas. Categorical feature encoding techniques for improved classifier performance when dealing with imbalanced data of fraudulent transactions // International Journal of Computers Communications & Control. Oradea: Agora University. ISSN 1841-9836. eISSN 1841-9844. 2023, vol. 18, iss. 3, art. no. 5433, p. [1-17]. DOI: 10.15837/ijccc.2023.3.5433.

Skyrius recenzuojamoje mokslinėje knygoje

1. Breskuvienė, Dalia; Dzemyda, Gintautas. Imbalanced data classification approach based on clustered training set // Data Science in Applications / Editors: Dzemyda, G., Bernatavičienė, J., Kacprzyk, J. Cham: Springer, 2023. ISBN 9783031244520. eISBN 9783031244537. p. 43-62. (Studies in Computational Intelligence, ISSN 1860-949X, eISSN 1860-9503; vol. 1084). DOI: 10.1007/978-3-031-24453-7_3.

Tarptautinės konferencijos

1. Breskuvienė, Dalia. Adapt or fall behind: A deep dive into machine learning techniques for detection of the evolving fraud in the financial realm // 13th Annual Counter Fraud, Cybercrime and Forensic Accounting Conference, June 12–13, 2024, Portsmouth, UK.

2. Breskuvienė, Dalia. Clustering-based optimization in fraud detection classifier training // EURO 2022: [32nd European Conference

on Operational Research (EURO XXXII)], Espoo, Finland, July 3-6, 2022:

Nacionalinės konferencijos

1. Breskuvienė, Dalia. Highly imbalanced data case: pattern-guided feature selection to detect financial fraud // DAMSS: 15th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 28-30, 2024.

2. Breskuvienė, Dalia. What is a concept drift, and does it affect machine learning performance? // DAMSS: 14th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, November 30 - December 2, 2023.

3. Breskuvienė, Dalia. Autoencoder for fraudulent transactions data feature engineering // DAMSS: 13th Conference on Data Analysis Methods for Software Systems, Druskininkai, Lithuania, December 1–3, 2022.

Tyrime taikyti Python kodai

1. Breskuvienė, Dalia. Kredito kortelių sukčiavimo aptikimo gerinimas. Mokslo Duomenų Archyvas (MIDAS). DOI: 10.18279/MIDAS.261094.

2. Breskuvienė, Dalia. Kategorinių požymių kodavimo metodai, taikant juos nesubalansuotiems duomenims. Mokslo Duomenų Archyvas (MIDAS). DOI: 10.18279/MIDAS.261073.

3. Breskuvienė, Dalia. Klasteriais pritaikyta mokymo strategija kredito kortelių sukčiavimo aptikimui. Mokslo Duomenų Archyvas (MIDAS). DOI: 10.18279/MIDAS.259980.

Disertacijos strukūra

Šią daktaro disertaciją sudaro įvadas, keturi skyriai, išvados ir santrauka lietuvių kalba. Įvado skyriuje pateikiamas įvadas į tyrimą ir disertacijos apžvalga. Pirmame skyriuje pateikiama literatūros apžvalga, kurioje aprašomi finansinio sukčiavimo tipų skirtumai, aptariami

nesubalansuotų duomenų tvarkymo metodai, apžvelgiamas požymių kodavimas ir požymių atrankos metodai. Skyriuje Nr. 2 pristatomas nesubalansuotų duomenų klasifikavimo metodas, pagrįstas klasterizuotu mokymo rinkiniu, įskaitant transakcijų atsitiktinį atrinkimą. Skyriuje Nr. 3 nagrinėjami kategorinių požymių kodavimo būdai, skirti klasifikatoriaus veikimui pagerinti, kai susiduriama su nusikalstamu būdu atliekamomis transakcijomis. Skyriuje Nr. 4 aprašomas pasiūlytas FID-SOM metodas, skirtas sukčiavimo požymiams atrinkti, ir aptariami eksperimentiniai rezultatai. Bendrosios išvados apibendrinamos skyriuje "Bendrosios išvados".

Darbo pabaigoje pateiktos 153 bibliografinės nuorodos. Disertaciją sudaro 168 puslapiai, 30 paveikslų ir 20 lentelių.

S.1. Nesubalansuotų duomenų valdymas aptinkant sukčiavimą: literatūros apžvalga ir tyrimų spragos

Mašininio mokymosi srityje nesubalansuotų duomenų (angl. *imbalanced data*) keliami iššūkiai yra reikšmingi ir plačiai paplitę įvairiose taikymo srityse, tokiose kaip finansinis sukčiavimas, medicina, kibernetinis saugumas. Literatūros apžvalga išryškina keletą veiksmingų metodų šiems iššūkiams spręsti, pabrėžiant duomenų atsitiktinio atrinkimo (angl. *resampling*) svarbą, ypač taikant mažumos klasės papildomų pavyzdžių generavimą (angl. *oversampling*) bei daugumos klasės sumažinimą (angl. *undersampling*). Pažangūs metodai, tokie kaip SMOTE ir jo modifikacijos, pasirodė esą perspektyvūs generuojant sintetinius pavyzdžius ir taip efektyviau subalansuojant duomenų rinkinius.

Ansamblinių metodai (angl. *ensemble methods*) reikšmingai prisidėjo prie klasifikatorių našumo didinimo dirbant su nesubalansuotais duomenų rinkiniais. Šie metodai kuria kelis modelius ir agreguoja jų prognozes, taip sumažindami polinkį teikti pirmenybę daugumos klasei, kuri dažnai pastebima naudojant vieną klasifikatorių. Taip pat veiksmingumu pasižymi specialiai nesubalansuotiems duomenims pritaikyti klasifikatoriai, tokie kaip kaštams jautrūs (angl. *cost-sensitive*) mokymosi algoritmai. Šie algoritmai koreguoja mokymosi procesą, priskirdami didesnes klaidingo priskyrimo sąnaudas mažumos klasei, taip pagerindami retų, bet svarbių atvejų prognozavimo kokybę.

Nors padaryta reikšminga pažanga sprendžiant nesubalansuotų duomenų problemą, literatūroje vis dar išlieka keletas spragų. Reikalingi papildomi empiriniai tyrimai, siekiant sistemingai palyginti skirtingų kategorinių kintamųjų kodavimo (angl. *encoding*) technikų veiksmingumą įvairių tipų sukčiavimo aptikimo duomenų rinkiniuose ir mašininio mokymosi modeliuose. Tokie tyrimai galėtų suteikti tikslingesnes rekomendacijas praktikams dėl tinkamiausių kodavimo strategijų pasirinkimo. Pažangių mašininio mokymosi modelių, tokių kaip ansambliniai metodai ir neuroniniai tinklai, bei požymių kodavimo sąveika taip pat reikalauja tolesnių tyrimų. Atsižvelgiant į tai, kad mašininio mokymosi modelių veiksmingumas glaudžiai susijęs su taikomomis požymių kodavimo strategijomis, išsamesnis šių sąveikų supratimas galėtų padėti kurti efektyvesnes ir tikslesnes duomenų paruošimo procedūras, skirtas sukčiavimo atvejų aptikimui.

Požymių kodavimo pasirinkimai gali turėti reikšmingos įtakos modelio sąžiningumui (angl. *fairness*), ypač jautrioje finansų srityse, todėl būtina inicijuoti tyrimus, kurie ne tik spręstų su etika susijusius klausimus, bet ir siūlytų aiškias gaires sąžiningam bei nešališkam kategorinių kintamųjų kodavimui. Etiniai aspektai ir su šališkumu susijusios rizikos, kurias gali lemti skirtingi kodavimo metodai, iki šiol nėra išsamiai išanalizuoti, ypač kredito kortelių sukčiavimo aptikimo srityje.

Dėl didelio finansinių institucijų turimo požymių kiekio, požymių atranka tampa esmine kredito kortelių sukčiavimo aptikimo dalimi. Tinkamai atrinkus svarbiausius požymius ne tik sumažinamas skaičiavimo sudėtingumas, bet ir užtikrinamas greitas modelio veikimas. Spartus prognozavimas yra būtinas realiu laiku aptinkant sukčiavimą, siekiant išvengti klientų mokėjimų trikdžių. Atliktoje literatūros apžvalgoje atsispindi, jog kredito kortelių sukčiavimo aptikimas turi vykti per milisekundes.

Dabartiniai akademiniai kredito kortelių sukčiavimo aptikimo vertinimai dažnai nėra pakankamai reprezentatyvūs. Mokymo ir testavimo aibės yra sudaromos naudojant standartinį atsitiktinio skirstymo (angl. *random sampling*) metodą. Tokie metodai neatspindi realaus pasaulio sąlygų, kuriomis modeliai turi būti mokomi su istoriniais duomenimis ir testuojami su vėlesniais/ateities atvejais.

Sintetinių duomenų naudojimas sukčiavimo aptikimo tyrimams

Duomenų kokybė, pilnumas ir reprezentatyvumas tiesiogiai lemia kuriamų modelių našumą ir patikimumą. Šiame skyriuje aptariami įvairūs iššūkiai, susiję su duomenų prieinamumu finansinio sukčiavimo aptikimo tyrimų srityje. Finansinio sukčiavimo sritis yra viena iš tų, kur prieiga prie duomenų yra itin ribota. Daug didelės apimties duomenų rinkinių yra jautrūs ir jų naudojimas apribotas teisės aktais, tokiais kaip GDPR ar CCPA. Šie reglamentai riboja tyrimų galimybes ir sudaro kliūtis mašininio mokymosi plėtrai.

Šiuo atveju sintetiniai duomenys tampa perspektyvia alternatyva, padedančia spręsti privatumo, sąžiningumo ir daugelį kitų problemų. Norint spartinti mokslinius tyrimus, būtina turėti duomenų rinkinius, kurie pasižymėtų dideliu kiekiu (angl. *volume*), srautu (angl. *velocity*) ir įvairove (angl. *variety*). Pagal [72] pateiktą apibrėžimą, sintetiniai duomenys yra „Duomenys, sugeneruoti naudojant specialiai tam su-

kurtą matematinį modelį ar algoritmą, siekiant spręsti vieną ar kelias duomenų mokslo užduotis.“

Sintetinami gali būti įvairių tipų duomenys, įskaitant lentelinius, vaizdinius ir garso duomenis. Šiame tyrime daugiausia dėmesio skiriama lentelinių duomenų generavimui ir jų sąžiningumo aspektams. Lenteliniai duomenys sudaryti iš eilučių ir stulpelių. Tokiems duomenims generuoti būtina vienu metu modeliuoti kiekvieno stulpelio skirstinį ir užtikrinti eilutėms bei visai lentelei taikomų apribojimų laikymąsi.

Sintetiniai duomenys atlieka svarbų vaidmenį tyrimuose, kai duomenų prieinamumą riboja ne tik teisiniai reikalavimai, bet ir natūralus jų retumas. Puikus sintetinių duomenų taikymo pavyzdys pateiktas šaltinyje [21]. Šiame tyrime sintetiniai duomenys buvo naudojami kuriant realistiškus kibernetinius duomenis mašininio mokymosi klasifikatoriams, skirtus tinklo įsibrovimų aptikimo sistemoms [21]. Tyrimo autoriai padarė išvadą, kad jų pasirinkti generavimo metodai – CTGAN ir TVAE – pakankamai gerai sugeneravo sintetinius kibernetinius duomenis. Tačiau modeliai, mokyti tik su sintetiniais duomenimis, pasižymėjo žemu klasifikavimo atsako rodikliu (angl. *Recall*). Be to, autoriai rekomenduoja modelio mokymo duomenyse turėti bent 15 % realių duomenų.

Duomenų rinkiniai naudojami šiame tyrime

Tyrime naudoti trys duomenų rinkiniai, atspindintys įvairius sukčiavimo kreditinėmis kortelėmis scenarijus, įskaitant sintezuotus ir realius duomenis.

- Sintetinis duomenų rinkinys Nr. 1 [3] – Erik Altman sukurtas sintezuotas duomenų rinkinys, kuris imituoja JAV gyventojų pirkimo įpročius virtualiame pasaulyje. Šiame rinkinyje yra klientai, prekybininkai ir finansiniu sukčiavimu užsiimantys asmenys. Šiame duomenų rinkinyje esantys požymiai buvo sukurti taip, kad jų pagrindinės statistinės charakteristikos (vidurkis, standartinis nuokrypis) atitiktų realius duomenis. Svarbi šio rinkinio ypatybė – susietas veikėjų elgesys: pavyzdžiui, kelionių metu ar savaitgaliais keičiasi išlaidavimo modeliai. Rinkinys taip pat atspindi realius bankinius veiksmus, pvz., perėjimą prie lustinių kortelių 2014 m.

- Sintetinis duomenų rinkinys Nr. 2 [58] – Sparkov įrankiu sugeneruotas sintezuotas kreditinių kortelių transakcijų rinkinys. Šis rinkinys imituoja realaus laiko pirkimo elgseną: atsižvelgiama į išlaidų dažnumą, sumas, kategorijas ir laikinius aspektus (pvz., periodinius mokėjimus ar anomalijas). Jame pateikiami tiek normalūs, tiek apgaulingi veiksmai, taip sukuriant imituotą, bet realybę atspindintį, nesubalansuotą rinkinį.

- Duomenų rinkinys Nr. 3 – viešai prieinamas realių transakcijų rinkinys, apimantis Europos kortelių naudotojų mokėjimus per dvi dienas 2013 m. rugsėjį. Jame užfiksuota 492 sukčiavimo atvejų iš 284 807 transakcijų, t. y. sukčiavimo klasė sudaro tik 0,172% visų duomenų. Dėl konfidencialumo originalūs požymiai nepateikiami – duomenys transformuoti pasitelkus pagrindinių komponentų analizę (PCA). Galutiniai požymiai: V1–V28 (PCA komponentai), *Time* (sekundės nuo pirmos transakcijos) ir *Amount* (transakcijos suma).

S.1 lentelė: Tyrime naudotų duomenų rinkinių suvestinė

Kategorija	Sintetinis rinkinys 1 [3]	Sintetinis rinkinys 2 [58]	Realus rinkinys
Ne sukčiavimas (%)	99,86%	99,48%	99,83%
Sukčiavimas (%)	0,14%	0,52%	0,172%
Transakcijų skaičius	24 386 900	1 852 394	284 807
Požymių skaičius	25	11	30

Visame disertaciniame darbe duomenų skirstymas į mokymosi ir testavimo aibes buvo atliekamas atsižvelgiant į laiko dimensiją – senesni duomenys buvo skirti modelio mokymui, o vėlesni – testavimui. Toks chronologinis skirstymas atspindi realias prognozavimo sąlygas, kai modelis mokomas naudojant istorinę informaciją ir taikomas būsimiems atvejams vertinti.

Tais atvejais, kai tyrimams reikalinga papildoma validavimo aibė (pvz., hiperparametrų parinkimui), ji buvo formuojama iš mokymosi aibės, ją atsitiktinai padalijant į dvi dalis, išlaikant tikslinės klasės proporcijas (stratifikuotas skirstymas).

S.2. Nesubalansuotų duomenų klasifikavimas naudojant klasterizuotą duomenų rinkinį

Literatūros apžvalga išryškino esamas tyrimų spragas nesubalansuotų duomenų klasifikavimo srityje, ypač kredito kortelių sukčiavimo aptikimo kontekste. Pradedant šiuo skyriumi, pateikiamos siūlomos strategijos ir metodai, skirti šioms tyrimų spragoms spręsti.

Šiame skyriuje nagrinėjamos strategijos, skirtos pagerinti modelių tikslumą dirbant su nesubalansuotais duomenų rinkiniais. Siūloma strategija apima kelių klasifikatorių mokymą naudojant klasterizuotus mokymo duomenis. Kiekvienam klasteriui sukuriamas individualus mašininio mokymosi modelis, suformuojant sub-klasifikatorius (angl. *sub-classifiers*). Sprendimas, kurį sub-klasifikatorių naudoti klasifikuojant naują duomenų tašką, yra priimamas remiantis Euklido atstumu tarp naujo duomenų taško ir atitinkamo mokymo rinkinio klasterio centro. Be to, siekiant optimizuoti šių sub-klasifikatorių našumą, pasitelkiamas validacinis (angl. *validation*) duomenų rinkinys, naudojamas klasterių balanso parametrui rasti. Sprendimų priėmimo schema pateikta paveiksle 2.1, esančiame pagrindiniame disertacijos tekste.

Siūlomos strategijos efektyvumas vertinamas pagal klasifikavimo atsako rodiklį. Taikant šią strategiją, lyginamas klasifikavimo atsako rodiklis, gautas nenaudojant klasterizacijos ir pavyzdžių mažinimo, su klasifikavimo atsako rodikliu, pasiektu pritaikius siūlomą metodą testavimo rinkinyje. Klasifikavimo atsako rodiklis yra ypač svarbus aptinkant kredito kortelių sukčiavimo atvejus, nes jis parodo, kokią dalį visų tikrųjų sukčiavimo atvejų modelis sugebėjo aptikti. Aukštas atsako rodiklis rodo, kad modelis sėkmingai identifikuoja didžiąją dalį realių sukčiavimo atvejų, o tai padeda sumažinti finansinius nuostolius ir apriboti nusikalstamą veiklą.

Klasterizavimui buvo naudojamas k-vidurkių (angl. *k-means*) algoritmas, kuriam būtina nurodyti klasterių skaičių. Nors nėra vieno universalaus būdo optimaliam klasterių skaičiui nustatyti, tai galima atlikti vizualiai arba naudojant Silueto (angl. *Silhouette*) rodiklį. Reikia paminėti, jog klasterizavimui buvo naudojami ne visi kintamieji, o tik tie, su kuriais Silueto rodiklis buvo pakankamai aukštas.

Eksperimentams atlikti buvo naudojamas sintetinis duomenų rinkinys Nr. 1. Kadangi šis duomenų rinkinys labai didelis ir turi labai ilgą

transakcijų istoriją, tik dalis jo buvo panaudota eksperimentams. Mokymui, validavimui, ir testavimui buvo parinktas laikotarpis nuo 2014 metų. Buvo išbandyta daugiau nei 280 skirtingų požymių ir klasterių skaičiaus kombinacijų, su kurias pasiektas geriausias rezultatas buvo Silueto rodiklis 0,862248. Toks rezultatas buvo pasiektas naudojant tris kintamuosius, kurie suskirstė mokymų aibę į keturis klasterius. Klasteriai buvo gauti skirtingo dydžio su skirtingais sukčiavimo transakcijų kiekiais. Kombinacijų atrankos algoritmas pateiktas pagrindiniame disertacijos teste (žr. Algoritmas 1). Mokymų aibės klasterių charakteristikos pateiktos lentelėje S.2

Kiekvienam klasteriui atskirai buvo pritaikytas daugumos klasės sumažinimas (angl. *undersampling*). Kiekvieno klasterio klasifikavimo rezultatai buvo įvertinti naudojant validavimo aibę. Ji buvo naudojama ne tik metrikoms apskaičiuoti, bet ir tam, kad būtų galima parinkti individualų kiekvieno klasterio daugumos klasės sumažinimo procentą. Iš lentelės S.2 matyti, kad nėra tiesioginio ar linijinio ryšio tarp klasės mažinimo (angl. *undersampling*) procento, sukčiavimo atvejų dalies ar klasterio dydžio. Visgi pastebime, kad prasčiausi resultatai matomi 4-tame klasteryje, kuris turėjo mažiausią sukčiavimo atvejų dalį, ir norint pasiekti geresnių rezultatų, jam reikėjo mažesnio klasės mažinimo procento.

S.2 lentelė: Klasės mažinimo poveikis įvairiems klasteriams: mokymo ir validavimo aibių metrikos

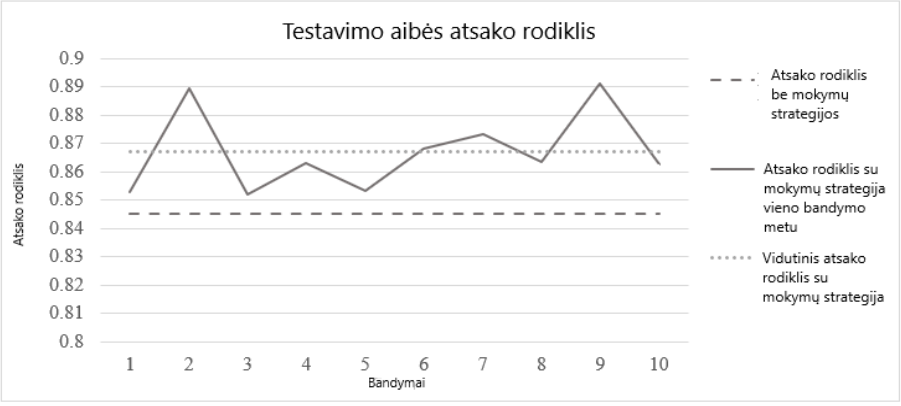
Metrikos	C1	C2	C3	C4
Mokymo aibės dydis	1 522 231	596 897	2 533 953	1 316 248
Validavimo aibės dydis	653 420	255 489	1 084 892	564 483
Sukčiavimo atvejų %	0,19	0,11	0,15	0,02
Klasės mažinimo %	87	91	49	7
Sukčiavimo atvejų % po mažinimo	0,22	0,12	0,30	0,27
Validavimo aibės F1 rodiklis	0,85	0,77	0,82	0,40
Validavimo aibės atsako rodklis	0,75	0,63	0,72	0,31

Klasifikavimo strategija grindžiama klasterizavimu – mokymo duomenims buvo pritaikytas k-means algoritmas, suformuojantis keturis klasterius (kaip aptarta ankstesniame skyriuje). Kiekvienam iš šių klasterių buvo atskirai apmokytas klasifikatorius, naudotas XGBoost algoritmas, tačiau iš principo galima naudoti bet kurį kitą tinkamą

klasifikatorių.

Svarbiausia šio tyrimo dalis yra sukurtos strategijos veikimo įvertinimas nepriklausomame testiniame duomenų rinkinyje, kuris atspindi būsimas, anksčiau nematytas sukčiavimo transakcijas. Testinių duomenų apdorojimo procedūra buvo identiška taikytam validacijos rinkiniui. Visų pirma, duomenys buvo standartizuoti – tai reiškia, kad kiekvieno požymio reikšmės buvo transformuotos taip, kad jų vidurkis būtų lygus nuliui, o standartinis nuokrypis – vienetas. Standartizacijos parametrai buvo apskaičiuoti tik iš mokymo duomenų, siekiant išvengti informacijos nutekėjimo (angl. *data leakage*). Testavimo metu kiekvienas testinio rinkinio taškas pirmiausia buvo priskiriamas vienam iš klasterių pagal mažiausią Euklido atstumą iki klasterio centro. Tuomet klasifikavimas buvo atliekamas naudojant to konkretaus klasterio klasifikatorių.

Siekiant įvertinti siūlomos strategijos stabilumą ir sumažinti atsitiktinumo įtaką, testavimas buvo kartojamas dešimt kartų, naudojant skirtingą atsitiktinių skaičių sėklą (angl. *random seed*) mokymų aibės mažinimo procesui.



S.2 pav. F1 rodiklis testiniame duomenų aibėje

Kaip parodyta S.2 pav., eksperimentiniai rezultatai leidžia daryti išvadą, kad klasifikavimo metodas, pagrįstas klasterizacija ir optimaliu didesnės klasės mažinimu, reikšmingai pagerino modelio veikimo rezultatus. Ši išvada grindžiama pagrindiniame disertacijos tekste atliktu hipotezės testu, kuriuo statistiškai patvirtintas reikšmingas skirtumas tarp lyginamų metodų rezultatų. Klasifikatorius be papildomos mokymo strategijos pasiekė 0,845 atsako rodiklį, o tuo tarpu taikant siūlomą

strategiją, jis padidėjo iki 0,867. Lyginant absoliučius skaičius, pastebima, kad neteisingai suklasifikuotų sukčiavimo atvejų skaičius sumažėjo nuo 323 iki 278 (vidutiniškai), t. y. 13,9 % sumažėjimas, patvirtinantis geresnę klasifikavimo kokybę.

Šiame skyriuje analizuojama nesubalansuotų duomenų problema, pasireiškianti tuo, kad sukčiavimo atvejų klasė ženkliai mažesnė nei teisėtų operacijų, dėl ko tradiciniai klasifikavimo metodai dažnai neaptinka retesnių atvejų. Problemą sprendžia siūlomas klasterizavimu grįstas klasifikavimo metodas, kuris skirtas pagerinti atsako rodiklį aptinkant kredito kortelių sukčiavimą. Metodas pagrįstas duomenų skirstymu į klasterius, kiekvieno jų subalansavimu ir atskirų subklasifikatorių mokymu. Gauti rezultatai patvirtina, kad klasterizacija gali būti efektyvus išankstinio apdorojimo žingsnis sukčiavimo aptikimo sistemose. Be to, literatūroje pažymima, kad požymių atranka ir kategorinių kintamųjų kodavimo metodai gali turėti teigiamos įtakos modelių našumui. Todėl tolesniuose skyriuose šios prielaidos bus nagrinėjamos tais atvejais kai duomenys yra labai smarkiai nesubalansuoti.

S.3. FID-SOM: naujas požymių atrankos metodas nesubalansuotų duomenų požymių atrankai, grindžiamas saviorganizuojančių žemėlapių taikymu

Finansinio sukčiavimo transakcijų duomenys dažnai pasižymi daugybe kategorinių kintamųjų, kurių kardinalumas yra labai didelis. Duomenų paruošimas analizei ir mašininio mokymo klasifikavimui/prognozavimui tampa sudėtingas, jei tokiuose požymiuose esančios kategorijos neturi aiškios tvarkos ar prasmingo atvaizdavimo į skaitines reikšmes. Nors egzistuoja įvairios kodavimo technikos, jų poveikis dideliems, itin nesubalansuotiems duomenų rinkiniams nėra nuosekliai įvertintas. Todėl būtina tirti skirtingų kodavimo metodų įtaką modelio našumui sukčiavimo aptikimo kontekste. Ypač svarbu tai atlikti dirbant su itin nesubalansuotais duomenimis, kur sukčiavimo atvejų dalis sudaro tik nedidelę visų duomenų dalį – netinkamas kodavimas gali įvesti triukšmą arba sukelti modelių persimokymą.

Šiame tyrime siekiame sistemingai įvertinti kodavimo strategijas, pritaikytas didelio kardinalumo kategoriniams požymiams didelės apimties nesubalansuotuose duomenų rinkiniuose, naudojant sukčiavimo aptikimą kaip reprezentatyvų pavyzdį. Tikslas – identifikuoti patikimas

duomenų paruošimo technikas, kurios mažintų šališkumą (angl. *bias*) ir padėtų pagerinti retų sukčiavimo atvejų aptikimą. Tikslui pasiekti buvo išnagrinėti šeši skirtingi kodavimo metodai.

Nors nemažai ankstesnių tyrimų nagrinėjo kodavimo technikas, dauguma jų rėmėsi viešai prieinamais subalansuotais duomenų rinkiniais, kurie ne visada atspindi su realiomis itin nesubalansuotomis situacijomis susijusius iššūkius. Šis skyrius prisideda prie šios spragos mažinimo, konkrečiai analizuodamas kodavimo metodų veikimą nesubalansuotuose duomenyse. Analizė, paremta kelių kodavimo metodų palyginimu eksperimetais skirtingų klasifikatorių kontekste, atskleidė šias pagrindines įžvalgas:

- Į tikslą orientuotų kodavimo metodų svarba. Rezultatai rodo, kad į tikslą orientuoti kategorinių duomenų kodavimai, ypač James-Stein ir Weight of Evidence (WOE), nuosekliai lenkia kitas technikas gerinant modelių našumą. Šie metodai efektyviai išnaudoja ryšį tarp požymių ir tikslo kintamojo.

- Iššūkiai dėl didelio kardinalumo požymių. Didelio kardinalumo požymių kodavimas, dažnai būdingas transakcijų duomenų rinkiniams, išlieka reikšmingu iššūkiu. Tokios technikos kaip maišos (angl. *hashing*) ir „One-Hot“ kodavimas dar labiau didina dimensijų prakeiksmą (angl. *curse of dimensionality*), neigiamai veikdamos modelių našumą. Maišos kodavimo našumas buvo prasčiausias: F1 rodiklis siekė 0.240 stiprinimo tipo (angl. *Boosting*) modeliams, 0.164 ansambliniams (ang. *Ensemble*) modeliams ir 0.211 nelinijiniams modeliams (angl. *Non-linear*), patvirtindamas prielaidą, kad kodavimo technikos, kurios nenaudoja tikslo informacijos, paprastai yra mažiau efektyvūs prognozavimo užduotyse.

- CatBoost kodavimas trūkumai. Priešingai nei tikėtasi, CatBoost kodavimas, skirtas kategoriniams duomenims, pasirodė esantis neoptimalus dirbant su nesubalansuotais duomenimis. Tai pabrėžia būtinybę kruopščiai vertinti kodavimo metodų tinkamumą konkrečioms duomenų pasiskirstymo situacijoms.

- Skirtinga kodavimo metodų įtaka priklausomai nuo algoritmo. Kodavimo technikų poveikis skyrėsi priklausomai nuo taikyto

mašininio mokymosi algoritmo. Šis faktas akcentuoja būtinybę derinti kodavimo strategijas prie konkretaus algoritmo savybių ir pasirinktų duomenų rinkinių.

Apibendrinant, šis skyrius pabrėžia išsamios kodavimo metodų analizės svarbą, ypač srityse, kur dirbama su nesubalansuotais duomenimis. Identifikuojant įvairių metodų stipriąsias ir silpnąsias puses, pateikiamos praktinės gairės duomenų mokslininkams ir tyrėjams, siekiantiems pagerinti klasifikavimo našumą.

S.4. Kredito kortelių sukčiavimo aptikimo tobulinimas: itin nesubalansuotų duomenų atvejis

Saviorganizacinis žemėlapis (SOM), dažnai vadinamas Kohoneno žemėlapiu, yra galinga mašininio mokymosi be mokytojo technika, priskiriama dirbtinių neuroninių tinklų kategorijai. SOM naudojami dimensijų mažinimo, duomenų vizualizavimo, klasterizavimo užduotims spręsti. Pagrindinė SOM idėja – konvertuoti didelės dimensijos įvesties duomenis į mažesnės dimensijos tinklelį, išsaugant duomenų taškų topologinius ryšius.

Tarkime, kad turime daugiamačius duomenis, pateiktus masyvu X, kurį sudaro n duomenų taškai, ir kiekvienas jų X_i ($i = 1, \ldots, n$) yra vektorius $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ erdvėje $\mathbb{R}^m$. Šie duomenų taškai atspindi objektų ar reiškinių stebėjimus, priklausančius nuo m skirtingų požymių ($x_1, x_2, \ldots, x_m$). Kai kurie požymiai yra skaitiniai, kiti – kategoriniai. Pastarieji prieš analizę transformuojami į skaitinę formą, kad būtų galima vaizduoti X_i kaip vektorių $\mathbb{R}^m$ erdvėje. Be to, kiekvienam duomenų taškui priskiriama klasės etiketė $y_i \in \{0, 1\}$, nurodanti, kuriai kategorijai X_i priklauso.

Mūsų atveju SOM sudaro dvimatė neuronų tinklelio struktūra, išdėstyta stačiakampiu raštu. Kiekvienam neuronui priskiriamas svorio vektorius, kurio matmenys atitinka įvesties duomenų dimensiją. Tinklelio matmenys apskaičiuojami įvertinant neuronų skaičių pagal mokymų aibės duomenų stebėjimų skaičių, naudojant formulę [131]:

$$M \cong 5\sqrt{n}, \tag{S.1}$$

kur n - tranzakcijų skaičius.

Žemiau trumpai apžvelgiamas tinklo mokymosi procesas. Mūsų atveju SOM sudaro dvimatė neuronų tinklelio struktūra, išdėstyta stačiakampiu raštu. Kiekvienam neuronui priskiriamas svorio vektorius $W_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$, kurio matmenys atitinka įvesties duomenų dimensiją. Svorio vektorius veikia kaip neurono „prototipas“, apibūdinantis tam tikrą duomenų erdvės regioną. Pradinėje fazėje visi svoriai inicializuojami atsitiktinėmis reikšmėmis (arba pasirenkami iš įvesties duomenų pasiskirstymo), kad neuronai būtų išdėstyti įvairiose duomenų erdvės vietose.

Iš pradžių neuronų svoriams priskiriamos atsitiktinės reikšmės. Kiekviename žingsnyje iš mokymų aibės duomenų parenkamas įvesties taškas $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, ir apskaičiuojami jo Euklido atstumai iki visų neuronų svorių vektorių pagal formulę (S.2):

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - w_{jk})^2}, \tag{S.2}$$

kur d_{ij} yra atstumas tarp taško X_i ir j-ojo neurono svorio vektoriaus $W_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$.

Neuronas, turintis mažiausią atstumą iki įvesties taško, laikomas neuronu laimėtoju (angl. *Best-Matching-Unit* BMU) tam duomenų taškui. Nustačius BMU, mokymo procesas apima šio neurono kaimyninių neuronų pasirinkimą pagal jų erdvinį artumą tinkle. Tada šių kaimyninių neuronų svorio vektoriai atnaujinami naudojant kaimynystės (angl. *neighboring*) funkciją. Klasikinis svorių atnaujinimo būdas [77]:

$$w_{jk}(t+1) = w_{jk}(t) + \eta(t) T_{j^*j}(t)(x_{ik} - w_{jk}(t)), \tag{S.3}$$

kur:

- $w_{jk}(t)$ – k-asis komponentas j-ojo neurono svorio vektoriuje iteracijoje t, t – iteracijos numeris,
- $\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda_\eta}\right)$ – mokymosi greitis, kuris mažėja laikui bėgant,
- $T_{j^*j}(t) = \exp\left(-\frac{\|W_{j^*} - W_j\|^2}{2\sigma(t)^2}\right)$ – kaimynystės funkcijos reikšmė tarp BMU ir j-ojo neurono iteracijoje t,

- $\|W_{j^*} - W_j\|$ – atstumas tarp neuronų j^* ir j,
- $\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_\sigma}\right)$ – kaimynystės spindulys,
- x_{ik} – įvesties taško X_i k-asis komponentas.

Svarbiausi hiperparametrai SOM mokymui:

- η_0 – pradinis mokymosi greitis,
- λ_η – mokymosi greičio mažėjimo konstanta,
- σ_0 – pradinis kaimynystės spindulys,
- λ_σ – kaimynystės spindulio mažėjimo konstanta.

Kaimynystės funkcija nustato, kiek įtakos turi turėti kiekvienas kaimynas BMU atžvilgiu. Paprastai šis poveikis mažėja didėjant atstumui, taip sukuriant natūralų atnaujinimo intensyvumo mažėjimą. Iš esmės, BMU identifikavimas ir kaimyninių neuronų svorių atnaujinimas naudojant kaimynystės funkciją sudaro pagrindinį saviorganizacinių žemėlapių (SOM) veikimo principą.

Informatyviausiems požymiams atrinkti šioje disertacijoje yra pasiūlytas metodas FID-SOM, naudojantis SOM svorio vektorių variaciją. Normalizavę BMU duomenis ir apskaičiavę kiekvieno atributo dispersiją, atributai surūšiuojami mažėjančia dispersijos tvarka, taip sudarant požymių svarbos sąrašą. Požymis, kurio atitinkamo atributo variacija yra didžiausia, laikomas informatyviausiu. Tuomet iš šio sąrašo galima pasirinkti pageidaujamą požymių skaičių.

Saviorganizacijos žemėlapiai (SOM) dažnai naudojami klasterizavimo užduotims spręsti [39]. Tačiau šioje disertacijoje SOM gebėjimai apibendrinti duomenis pritaikomi dimensijų mažinimo uždaviniams itin nesubalansuotuose duomenų rinkiniuose spręsti. Šios idėjos sudaro naujo metodo FID-SOM pagrindą – požymių atranką nesubalansuotiems duomenims naudojant SOM. Siūlomo metodo algoritmas pateikiamas pseudokodu Algoritme 3.

Šis metodas leidžia efektyviai atlikti požymių atranką tolesnei analizei ar vizualizacijai. Metodas dinamiškai prisitaiko prie duomenų vidinių savybių, taip užtikrindamas automatinį, duomenimis pagrįstą požymių atrankos procesą. Ši savybė ženkliai padidina metodo tinkamumą įvairiose mokslinėse taikymo srityse, kur duomenų rinkiniai dažnai skiriasi savo dimensijomis ir sudėtingumu.

Algoritmas 3 FID-SOM: Požymių atranka naudojant saviorganizacinius žemėlapius (SOM) disbalansuotiems duomenims

Duota: $X \in \mathbb{R}^{n \times m}$: mokymo aibė, turinti n stebėjimų ir m požymių
Duota: $params = \{tinklo\_dydis, iter\_sk, mok\_greitis, kaim\_spindulys\}$
Duota: d: pageidaujamas atrinktų požymių skaičius
Rezultatas: F: atrinktų požymių indeksų aibė (dydžio d)

1: $SOM \leftarrow$ ApmokytiSOM$(X, params)$
2: $W_{BMU} \leftarrow$ GautiBMUSvorius(SOM)
3: $W_{BMU}^{norm} \leftarrow$ MinMaxNormalizuoti$(W_{BMU}, [0, 1])$
4: $V \leftarrow$ ApskaičiuotiDispersijas(W_{BMU}^{norm}) ▷ vektorius, kuriame kiekvieno požymio dispersija
5: $idx \leftarrow$ RikiuotiMažėjančiai(V)
6: $F \leftarrow idx[1{:}d]$ ▷ pasirenkami pirmieji d požymiai
7: **Grąžinti** F

Svorio vektoriai yra esminiai norint perkelti aukštos dimensijos transakcijų duomenis į žemesnės dimensijos erdvę, išsaugant įvesties duomenų topologinius ryšius. Sukčiavimo aptikimo kontekste tai leidžia SOM tinklui klasterizuoti panašias transakcijas kartu ir išryškinti anomalijas, kurios dažnai atitinka sukčiavimo atvejus. SOM suteikia struktūrizuotą pagrindą analizuoti sudėtingus transakcijų modelius ir identifikuoti anomalijas, siejant kiekvieną tinklo neuroną su svorio vektoriumi.

FID-SOM unikalumas toks, kad formuojamas naujas duomenų rinkinys, kuriame neuronai laimėtojai (BMU) iš apmokyto SOM pateikiami kaip atributų vektoriai, atitinkantys pradines savybes. Šie atributai yra surūšiuojami mažėjančia dispersijos tvarka. Išlaikant norimą atributų, pasižyminčių didžiausia variacija, skaičių, atrenkamas mažesnis požymių rinkinys tolesnei analizei.

FID-SOM buvo palygintas su požymių atrankos metodais, pagrįstais F-testu, χ^2 testu ir tarpusavio informacija (angl. *mutual information*), Rekursyvinės požymių eliminacijos (angl. *Recursive Feature Elimination*) metodu bei XGB svarbos (angl. *XGB Importance*) metodu. Požymių atrankos metodų veiksmingumas buvo vertinamas naudojant F1 rodiklį, Mathews koreliacijos koeficientą (MCC), geometrinį vidurkį (G-Mean), AUC-PR ir AUC-ROC metrikas, taikant XGBoost, CatBoost ir Random Forest algoritmus trijuose duomenų rinkiniuose. Detalūs

eksperimentų įvertinimai pateikti lentelėse pagrindiniame disertacijos tekste.

Metodo sėkmė buvo vertinama skaičiuojant, kiek kartų metodas tapo geriausiai veikiančiu metodu. Siūlomas FID-SOM metodas parodė reikšmingą pasiekimą – sėkmės rodiklis siekė 71,11 % . Šis rezultatas yra reikšmingas ne tik dėl to, kad metodas gebėjo konkuruoti su esamais metodais ar juos pranokti, bet ir todėl, kad atskleidė inovatyvų potencialą. Ypač svarbu pabrėžti, kad antras pagal sėkmingumą metodas pasiekė tik 17,78 % sėkmės rodiklį.

Atsižvelgiant į nustatytą veiksmingumą, galima tikėtis, kad FID-SOM taps vienu iš dažnai naudojamų požymių atrankos metodų, leidžiančių sukčiavimo aptikimo specialistams sėkmingai spręsti sudėtingas klasifikavimo problemas. Nors šiame tyrime buvo taikomas standartinis konkurencinio mokymosi mechanizmas svorių vektorių adaptacijai, ateityje būtų tikslinga tyrinėti pažangesnes optimizavimo technikas, pavyzdžiui, klasės specifinius mokymosi greičius ar papildomus svorių atnaujinimo apribojimus. Tokie patobulinimai galėtų padidinti SOM našumą dirbant su nesubalansuotais duomenų rinkiniais ir sustiprinti modelio gebėjimą aptikti sukčiavimo atvejus išlaikant skaičiavimų efektyvumą.

Finansinio sukčiavimo aptikimas yra itin svarbi veikla, siekianti užkirsti kelią finansiniams nuostoliams, tiksliai identifikuojant apgaulingas / neteisėtas transakcijas. Pagrindinis šio tyrimo tikslas – sukurti metodą, racionaliai sumažinti esamą požymių rinkinį, kad būtų pagerintas nesubalansuotų duomenų klasifikavimo tikslumas, siekiant geriau aptikti sukčiavimo atvejus. Atsižvelgiant į didelį sukčiavimo duomenų nesubalansuotumą, tradiciniai mašininio mokymosi algoritmai dažnai nesugeba pasiekti priimtino našumo. Šioje disertacijoje nagrinėjami inovatyvūs metodai, skirti šiems iššūkiams spręsti: analizuojami klasterizavimu grįsti klasifikavimo metodai, požymių atrankos (angl. *feature selection*) sprendimai bei kategorinių kintamųjų kodavimo (angl. *encoding*) technikos, siekiant padidinti sukčiavimo aptikimo efektyvumą.

- Siūloma klasterizavimu grįsta mokymosi strategija, kurioje kiekvienas klasteris subalansuojamas ir klasifikuojamas atskirai, leidžia reikšmingai pagerinti sukčiavimo atvejų klasifikavimo atsako rodiklį (angl. Recall) – nuo 0,845 iki 0,867. Eksperimentų rezultatai rodo, kad integravus klasterizaciją kaip išankstinį apdorojimo žingsnį, modelio gebėjimas teisingai identifikuoti apgaulingas transakcijas pagerėjo, o klaidingai legaliomis priskirtų sukčiavimo atvejų sumažėjo 13,9 %.

- Eksperimentiniai tyrimo rezultatai atskleidė, kad kategorinių požymių kodavimo strategijos, įtraukiančios tikslinio kintamojo informaciją – konkrečiai James-Stein ir Weight of Evidence (WOE) metodai – leidžia užtikrinti ženkliai aukštesnę modelių diskriminacinę gebą dirbant su labai nesubalansuotais ir aukštą kardinalumą turinčiais duomenimis. Taikant James-Stein kodavimą, vidutiniai F1 rodikliai siekė 0,8049 su stiprinimo tipo klasifikatoriais, 0,7595 su ansambliniais modeliais ir 0,7604 su nelinijiniais; WOE rezultatai atitinkamai buvo 0,7861, 0,7651 ir 0,7529. Tuo metu target-agnostic metodai, tokie kaip Label Encoding ar Hashing, reikšmingai atsiliko – kai kuriais atvejais F1 reikšmė nesiekė nė 0,5. Pažymėtina, kad visų modelių hiperparametrai liko numatytieji – eksperimentų tikslas buvo ne optimizuoti galutinį rezultatą, bet objektyviai įvertinti skirtingų kodavimo strategijų poveikį.

- Hashing kodavimo metodas, kaip ir dažnai naudojamas One-Hot kodavimas (nors šis eksperimente nebuvo įtrauktas), prisideda prie „daugiamatiškumo prakeiksmo“ (angl. *curse of dimensionality*), ypač kai kategorinių požymių kardinalumas yra labai aukštas – pvz., „Merchant City“ turi 11 391 unikalias reikšmes. Tai ne tik blogina modelio apmokymą, bet ir didina skaičiavimų sąnaudas bei riziką persimokyti.

- Eksperimentiniais tyrimais buvo parodyta, kad pasiūlytas naujas požymių atrankos metodas FID-SOM, pasitelkiantis saviorganizuojančius žemėlapius ir remiantis neuronų laimėtojų (BMU) svorių vektorių komponentų dispersija informatyviausių požymių identifikavimui, pasirodė esąs geriausias net 71,11 % visų testuotų konfigūracijų, t. y. daugumoje požymių rinkinių ir klasifikatorių kombinacijų jis pasiekė aukščiausią klasifikavimo rezultatą. Metodas pranoko tradicinius atrankos sprendimus, tokius kaip F testas, χ^2 testas, tarpusavio informacija (angl. mutual information), rekursyvinė požymių eliminacija (angl. Recursive Feature Elimination) bei XGBoost svarbos analizė (angl. XGB Importance).

- Eksperimentiniai rezultatai parodė, kad, taikant laiku pagrįstą transakcijų duomenų padalijimą, testuojant FID-SOM pasiekė F1 rodiklį 0,85, atsako rodiklį 0,76. Tuo tarpu naudojant atsitiktinį duomenų dalijimą, tie patys modeliai grąžino gerokai aukštesnius rodiklius (F1 rodiklis 0,88, atsako rodiklis 0,82), tačiau toks rezultatų padidėjimas laikytinas dirbtiniu ir siejamas su duomenų nutekėjimu tarp mokymo ir testavimo rinkinių. Šis skirtumas parodė, kad atsitiktinis skirstymas gali sudaryti klaidingą įspūdį apie modelio efektyvumą, o laiku grįstas padalijimas suteikia patikimesnį pagrindą vertinti modelių gebėjimą būti atspariam besikeičiančioms sukčiavimo tendencijoms.

Šios įžvalgos suteikia vertingų praktinių gairių mokslininkams ir specialistams, siekiantiems kurti efektyvesnes mašininio mokymosi sistemas kovai su finansiniu sukčiavimu. Siūlomas FID-SOM metodas atveria kelią tolesnėms inovacijoms, skatinančioms kurti adaptyvias ir intelektualias sukčiavimo aptikimo sistemas, gebančias veiksmingai spręsti nuolat kintančius sukčiavimo iššūkius.

Dalia Breskuvienė
Feature Conversion for a Better Imbalanced Data Classification:
A Financial Fraud Detection Case
Doctoral Dissertation
Technological Sciences
Informatics Engineering (T 007)
Thesis Editor: Zuzana Šiušaitė

Dalia Breskuvienė
Požymių konversija siekiant gerinti nesubalansuotų duomenų
klasifikavimą: finansinio sukčiavimo atvejis
Daktaro disertacija
Technologijos mokslai
Informatikos inžinerija (T 007)
Santraukos redaktorė: Vilija Butkuvienė

Vilnius University Press
9 Saulėtekio Ave., Building III, LT-10222 Vilnius
Email: info@leidykla.vu.lt, www.leidykla.vu.lt
bookshop.vu.lt, journals.vu.lt
Print run of 20 copies