សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ
ROYAL UNIVERSITY OF PHNOM PENH

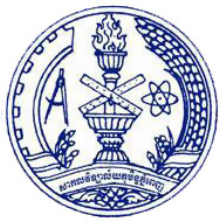
ការកែលម្អភាពត្រឹមត្រូវនៃការរកឃើញការក្លែងបន្លំ តាមរយៈវិស្វកម្មលក្ខណៈពិសេសកម្រិតខ្ពស់ និងការវាយតម្លៃគំរូប្រៀបធៀប

Improving Fraud Detection Accuracy Through Advanced Feature Engineering and Comparative Model Evaluation

A Thesis
In Partial Fulfilment of the Requirement for the Degree of
Bachelor of Science in Information Technology Engineering

DUCH PANHATHUN

June 2025



សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

ROYAL UNIVERSITY OF PHNOM PENH

ការកែលម្អភាពត្រឹមត្រូវនៃការរកឃើញការក្លែងបន្លំ តាមរយៈវិស្វកម្មលក្ខណៈពិសេសកម្រិតខ្ពស់ និងការវាយតម្លៃគំរូប្រៀបធៀប

Improving Fraud Detection Accuracy Through Advanced Feature Engineering and Comparative Model Evaluation

A Thesis
In Partial Fulfilment of the Requirement for the Degree of
Bachelor of Science in Information Technology Engineering

DUCH PANHATHUN

Examination Committee:

Mr. CHHIM BUNCHHUN
Dr. SRUN SOVILA
Mr. HUN SOVANSOREYA
Ms. E VATHANA

June 2025

សង្ខេបសេចក្តី

ការរកឃើញការក្លែងបន្លំផ្នែកហិរញ្ញវត្ថុ គឺជាកិច្ចការដ៏សំខាន់ និងប្រឈម ជាពិសេស នៅក្នុងបរិបទនៃសំណុំទិន្នន័យប្រតិបត្តិការខ្នាតធំ ដែលបង្ហាញពីភាពមិនស្មើគ្នាថ្នាក់ខ្ពស់ និងគំរូអាកប្បកិរិយាស្មុគស្មាញ។ នៅក្នុងការសិក្សានេះ យើងស្នើឱ្យមានបំពង់បង្ហូរ ការរៀនម៉ាស៊ីនដ៏ទូលំទូលាយ ដែលត្រូវបានរចនាឡើងដើម្បីបង្កើនភាពត្រឹមត្រូវនៃការរកឃើញការក្លែងបន្លំ ដោយការរួមបញ្ចូលវិស្វកម្មលក្ខណៈពិសេសកម្រិតខ្ពស់ ការវិភាគលើសគេ និងការវាយតម្លៃគំរូប្រៀបធៀប។ យើងចាប់ផ្តើមដោយការបំប្លែងទិន្នន័យប្រតិបត្តិការឆៅ ទៅជាលក្ខណៈពិសេសព័ត៌មាន ដូចជាឥរិយាបទតុល្យភាពគណនី(ទទេ។ គណនី)សូចនាករប្រតិបត្តិការបណ្តោះអាសន្ន(hour_risk_level)និងទង់ជាតិអាកប្បកិរិយាផ្ទាល់ខ្លួន (ឧទាហរណ៍ ភាពមិនប្រក្រតីនៃចំនួន និងតុល្យភាព)។ លក្ខណៈដែលបានកែលម្អទាំងនេះត្រូវបានបន្ថែមជាមួយសូចនាករដែលនៅខាងក្រៅ ដែលជួយឱ្យម៉ូដែលអាចរក្សាភាពខុសគ្នាដ៏សំខាន់ ខណៈពេលដែលកាត់បន្ថយការទទួលខុសគ្នា។

ដើម្បីវាយតម្លៃប្រសិទ្ធភាពនៃសំណុំទិន្នន័យដែលសំបូរទៅដោយលក្ខណៈពិសេសរបស់យើង យើងបណ្តុះបណ្តាល និងប្រៀបធៀបគំរូចំណាត់ថ្នាក់កម្រិតមធ្យមជាច្រើន ៖ CatBoost, XGBoost, LightGBM, Random Forest, K-Nearest Neighbors (KNN) និងតំរែតំរង់នៃការដឹកជញ្ជូន។ ម៉ូដែលទាំងអស់ត្រូវបានវាយតម្លៃដោយប្រើសំណុំធ្មេញដែលមានតុល្យភាពនៃម៉ែត្រ ការដំណើរការរួមមានភាពត្រឹមត្រូវ ភាពត្រឹមត្រូវការរំលឹកឡើងវិញ F1ពិន្ទុនិងតំបន់ប្រតិបត្តិការក្រោមបង្កាន់ដៃ។ ចំណុចខ្សោយ(ROC - AUC)។ លើសពីនេះ យើងដោះស្រាយបញ្ហានៃភាពមិនស្មើគ្នានៃថ្នាក់ ដោយអនុវត្តការចំណាយ តាមរយៈការវាស់ទម្ងន់ថ្នាក់ ជាពិសេសក្នុងការជំរុញគំរូ ដូចជា CatBoost។ ការអនុវត្តត្រូវបានពិនិត្យយ៉ាងល្អិតល្អន់ តាមរយៈការវិភាគទូទៅ ដោយប្រើទាំងការហ្វឹកហ្វឺន និងម៉ែត្រសាកល្បងដើម្បីកំណត់បរិមាណទូទៅ។

ការពិសោធន៍របស់យើងបង្ហាញថា គំរូជំរុញជម្រាល ជាពិសេស CatBoost ដំណើរការយ៉ាងខ្ជាប់ខ្ជួននូវគំរូប្រពៃណីក្នុងការរកឃើញប្រតិបត្តិការក្លែងបន្លំ ដោយសារសមត្ថភាពរបស់ពួកគេក្នុងការដោះស្រាយអថេរប្រភេទ អតុល្យភាពថ្នាក់និងអន្តរកម្មលក្ខណៈស្មុគស្មាញយ៉ាងមានប្រសិទ្ធភាព។ ការសិក្សានេះសង្កត់ធ្ងន់លើតួនាទីសំខាន់នៃវិស្វកម្មលក្ខណៈជាក់លាក់នៃដែន និងយុទ្ធសាស្ត្រសិក្សាដែលដឹងអំពីតម្លៃក្នុងការអភិវឌ្ឍប្រព័ន្ធរាវរកការក្លែងបន្លំដ៏រឹងមាំ។ ការរកឃើញទាំងនេះ ផ្តល់នូវក្របខ័ណ្ឌដែលអាចធ្វើមាត្រដ្ឋាន និងអាចបកស្រាយបានសម្រាប់ស្ថាប័នហិរញ្ញវត្ថុដែលស្វែងរកការពង្រឹងសមត្ថភាពការពារការក្លែងបន្លំរបស់ពួកគេនៅក្នុងពេលវេលាជាក់ស្តែង។

ABSTRACT

Financial fraud detection is a critical and challenging task, especially in the context of large-scale transaction datasets that exhibit high class imbalance and complex behavioral patterns. In this study, we propose a comprehensive machine learning pipeline designed to enhance fraud detection accuracy by integrating advanced feature engineering, outlier analysis, and comparative model evaluation. We begin by transforming raw transactional data into informative features, such as account balance behavior (emptied_account), temporal transaction indicators (hour_risk_level), and custom behavioral flags (e.g., matching amount and balance). These engineered features are supplemented with outlier indicator variables that capture extreme financial anomalies, enabling the model to retain important variance while mitigating noise through winsorization.

To evaluate the effectiveness of our feature-enriched dataset, we train and compare several state-of-the-art classification models: CatBoost, XGBoost, LightGBM, Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression. All models are evaluated using a balanced set of performance metrics including accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). Additionally, we address the issue of class imbalance by applying cost-sensitive learning through class weighting, particularly in boosting models like CatBoost. The performance is further scrutinized through overfitting analysis, using both training and test metrics to quantify generalization.

Our experiments show that gradient boosting models, particularly CatBoost, consistently outperform traditional models in detecting fraudulent transactions, thanks to their ability to handle categorical variables, class imbalance, and complex feature interactions effectively. The study emphasizes the critical role of domain-specific feature engineering and cost-aware learning strategies in developing robust fraud detection systems. These findings provide a scalable and interpretable framework for financial institutions seeking to enhance their fraud prevention capabilities in real time.

SUPERVISOR'S RESEARCH SUPERVISION STATEMENT

TO WHOM IT MAY CONCERN

Name of program: Bachelor of Information Technology Engineering
Name of candidate: Duch Panhathun

Title of research report: "Leveraging Gradient Boosting (CatBoost) for Imbalanced Fraud Detection in Financial Transactions"

This is to certify that the research carried out for the above titled project practicum research report was completed by the above named candidate under my direct supervision. This thesis material has not been used for any other degree. I played the following part in the

Preparation of this research report: Advisor
.....
.....
.....
.....

Supervisor's name:.....

Supervisor's signature:.....

Date.....

CANDIDATE'S STATEMENT

TO WHOM IT MAY CONCERN

This is to certify that the research that I, Duch Panhathun, hereby present this project practicum research report entitled Leveraging Gradient Boosting (CatBoost) for Imbalanced Fraud Detection in Financial Transactions.

for the degree of Bachelor of Faculty at the Royal University of Phnom Penh is entirely my own work and, furthermore, that it has not been used to fulfill the requirements of any other qualification in whole or in part, at this or any other University or equivalent institution.

No reference to, or quotation from, this document may be made without the written approval of the author.

Signed by (the candidate): .....

Date: .....

Sign by Supervisor: Mr. Chhum Bunchhun

Supervisor's signature: .....

Date.....

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all faculty member who supported me throughout the course of this research. Special thanks to my academic advisor, Chhim Bunchhun, for their continuous guidance, constructive feedback, and encouragement at every stage of this study.

I am also grateful to Royal University of Phnom Penh for providing the academic environment that made this research possible. Appreciation is extended to my peers and colleagues for their helpful discussions and moral support.

Lastly, I would like to acknowledge the creators of the publicly available dataset used in this research, which was essential for conducting a meaningful and practical evaluation of fraud detection techniques.

TABLE OF CONTENTS

សង្ខេបសេចក្តី i
SUPERVISOR'S RESEARCH SUPERVISION STATEMENT ii
ABSTRACT ii
CANDIDATE'S STATEMENT iv
ACKNOWLEDGEMENTS v
TABLE OF CONTENTS vi
TABLE OF FIGURES viii
TABLE OF TABLES ix
CHAPTER 1 1
INTRODUCTION 1
1.1 Background of the Study 1
1.2 Problem Statement 1
1.3 Aim and Objectives of the Study 2
1.4 Rationale of the Study 2
1.5 Limitation and Scope of the Study 3
CHAPTER 2 5
LITERATURE REVIEW 5
2.1 Learned lessons in credit card fraud detection from a practitioner perspective 5
2.2 A scalable framework for streaming credit card fraud detection with Spark 5
2.3 Feature engineering strategies for credit card fraud detection 6
2.4 Synthetic Minority Over-sampling Technique 7
2.5 A scalable tree boosting system 7
2.6 Unbiased boosting with categorical features 8
CHAPTER 3 10
METHODOLOGY 10
3.1 Data Collection and Description 10
3.2 Data Preprocessing, Feature Engineering, and Outlier Analysis 10
3.3 Model Selection and Training 15
3.4 Handling Class Imbalance 15
3.5 Model Evaluation 16
CHAPTER 4 19
MODEL TRAINING AND RESULT 19

4.1 Model Selection 19
4.2 Data Splitting 19
4.3 Class Imbalance Handling 20
4.4 Hyperparameter Settings 20
4.5 Training Process 21
4.6 Training Time 21
4.7 Test Results 21
4.8 Overfitting Analysis 22
4.9 Confusion Matrix 23
4.10 Comparing Model to Existing Research 24
CHAPTER 5 25
DISCUSSION 25
5.1 Model Performance Interpretation 25
5.2 Impact of Feature Engineering 25
5.3 Generalization and Overfitting 25
5.4 Effectiveness of Class Imbalance Handling 26
5.5 Strengths of the Approach 26
5.6 Limitations 26
5.7 Future Work 27
5.8 Business/Real-World Relevance 27
CHAPTER 6 29
CONCLUSION 29
REFERENCE 30

TABLE OF FIGURES

FIGURE 1: SHOW THE DIFFERENT TRAINING STRATEGIES FOR FRAUD DETECTION OVER TIME, HOW TO DEAL WITH NON STATIONARY DATA, CLASS IMBALANCE, AND CONCEPT DRIFT 5
FIGURE 2: A STREAMING, REAL-TIME FRAUD DETECTION ARCHITECTURE.....6
FIGURE 3: MAPS DIFFERENT MACHINE LEARNING AND AI TECHNIQUES TO THE MOST COMMON FRAUD DOMAINS 9
FIGURE 4: METRIC RESULT OF MODEL CATBOOST.....23

TABLE OF TABLES

TABLE 1: RESULT OF MODEL IMPLEMENTED WITH NO HANDLING OUTLIER 22
TABLE 2: THIS IS TABLE RESULT OF MODEL IMPLEMENTED WITH HANDLING OUTLIER 22
TABLE 3: OVERFITTING RESULT OF MODEL DO OUTLIER HANDLING.....22
TABLE 4: OVERFITTING RESULT OF MODEL DO OUTLIER HANDLING.....22
TABLE 5: METRIC RESULT OF MODEL CATBOOST 23
TABLE 6: EXISTING RESEARCH RESULT VS THIS RESEARCH BEST MODEL.....24

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

Financial fraud is now far more likely to occur due to the growth of digital financial services and the expansion of online transactions. Since billions of transactions take place every day across several platforms, it is now not only essential but also more difficult to identify fraudulent activity. More advanced, automated detection methods are required because financial fraud may result in significant financial losses as well as harm to institutional confidence. Conventional statistical or rule-based methods often fail to detect new or changing fraud tendencies, particularly in datasets with complex user behavior, high dimensionality, and class imbalance.

By learning from past data and adjusting to intricate and nuanced patterns of fraudulent conduct, recent developments in machine learning (ML) have shown significant promise in improving fraud detection. But there are still a lot of difficulties. The fundamental imbalance of transaction databases, where fraudulent acts account for a negligible portion of all transactions, is one of the main obstacles. Predictive models are often biased toward the majority class as a consequence of this imbalance, which lowers the detection rates of real fraud instances. Furthermore, in order to provide valuable insights, raw transaction data usually has to go through a rigorous preprocessing and feature extraction procedure since it is not interpretable.

In this situation, feature engineering is essential because it transforms unstructured, raw transactional information into useful, organized representations that capture user behavior and transaction dynamics. Furthermore, maintaining the signal while removing noise requires the detection of abnormalities and outliers, which are typically signs of fraudulent activity. When combined with statistical techniques like winsorization, these domain-specific tailored features may significantly improve the sensitivity and specificity of fraud detection models.

This study builds on these foundations by proposing a comprehensive ML pipeline that addresses the dual challenges of data imbalance and behavioral complexity. Through comparative model evaluation, the research seeks to identify the most effective algorithms for fraud detection and highlight the value of cost-sensitive learning and advanced feature engineering. The insights from this study are poised to inform financial institutions on best practices for building scalable and interpretable fraud detection systems capable of operating in real time.

1.2 Problem Statement

In the financial services sector, financial fraud is still a constant and developing concern, with more complex fraudulent activities taking advantage of weaknesses in conventional detection methods. Even though machine learning methods for fraud detection have become more popular, current models often face significant obstacles such high class imbalance, noisy and unstructured transactional data, and the problem of modeling intricate behavioral patterns. These restrictions usually lead to poor generalization to actual fraud situations, high false positive rates, and low detection accuracy.

One major problem is that many machine learning models used for fraud detection ignore the significance of domain-specific insights that may be obtained from user behavior and transaction context in favor of restricted or general characteristics. Additionally, models that are not tuned for unbalanced data have a tendency to ignore uncommon but very significant fraudulent transactions in favor of the majority class (non-fraud).

Therefore, a thorough and scalable fraud detection system that can manage unbalanced datasets, extract valuable behavioral characteristics, and assess model performance using reliable, fraud-specific metrics is desperately needed. Financial institutions run the danger of using detection technologies that miss important fraud instances in time, resulting in monetary losses and eroded customer confidence, if these issues are not resolved.

1.3 Aim and Objectives of the Study

To achieve the aim, the study is guided by the following specific objectives:

1 To preprocess and transform raw transactional data into meaningful features that reflect user behavior, transaction timing, and potential fraud indicators.

2 To engineer domain-specific features and outlier indicators, such as behavioral flags (e.g., emptied accounts, matching amounts), time-based features (e.g., is_night, hour), and financial anomalies, to improve model sensitivity.

3 To apply data normalization techniques, including winsorization, to reduce the impact of extreme values while retaining important variance relevant to fraud detection.

4 To implement and compare multiple machine learning classification models, including CatBoost, XGBoost, LightGBM, Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression.

5 To evaluate the performance of each model using comprehensive metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, while analyzing their ability to generalize using training and testing data.

6 To address the class imbalance problem using cost-sensitive learning approaches, such as class weighting in boosting algorithms.

7 To identify the most effective model and feature combination that maximizes fraud detection performance and interpretability for potential real-time deployment in financial systems.

1.4 Rationale of the Study

The number and speed of financial transactions have increased dramatically in today's increasingly digitalized financial environment. These developments have coincided with an increase in the sophistication, difficulty, and harm to people and institutions of financial fraud. The sophisticated, adaptable tactics used by contemporary fraudsters are too delicate to be picked up by conventional fraud detection systems, which often depend on preset criteria or historical trends. The sheer volume of transactional data, data imbalance, and changing fraud behavior are the main challenges for these traditional systems, which result in high false

positive rates or undetected fraudulent activity that cost both consumers and financial institutions.

Machine learning's capacity to learn from data and recognize intricate, non-linear correlations has made it a potent weapon in the fight against fraud. Nevertheless, there are significant drawbacks to a large number of current machine learning applications in this field. They do not manage class imbalance well, do not generalize well to real-world circumstances, and often ignore the significance of feature engineering based on domain experience. Additionally, many models are regarded as "black boxes," which makes it difficult for institutions to comprehend or trust their judgments—a crucial factor in high-stakes financial situations.

Therefore, the necessity to provide a more reliable, understandable, and useful method of detecting financial fraud is what motivates this work. This study tackles important issues that impair the functionality of conventional and some machine learning-based fraud detection systems by concentrating on domain-specific feature engineering, outlier analysis, and the incorporation of cost-sensitive learning algorithms. In addition to improving forecast accuracy, the project intends to provide a scalable framework that financial institutions may use in real-time operating scenarios by carefully designing the data pipeline and comparing state-of-the-art models.

In the end, the urgent need to improve fraud prevention technology, lower financial loss, and increase trust in the security of digital financial transactions is what drives this study. Particularly in the creation of sophisticated, data-driven fraud detection systems, the study's conclusions might make a significant contribution to both industrial practice and scholarly research.

1.5 Limitation and Scope of the Study

- Scope of the Study

  This study focuses on the development and evaluation of a machine learning-based pipeline for detecting financial fraud within structured transactional datasets. The research emphasizes feature engineering, particularly the creation of domain-specific variables and outlier indicators that capture behavioral patterns associated with fraud. The scope includes the preprocessing of raw transaction data, transformation into engineered features, application of normalization techniques such as winsorization, and the implementation of several machine learning models including CatBoost, XGBoost, LightGBM, Random Forest, K-Nearest Neighbors (KNN), and Logistic Regression.

  Additionally, the study includes a thorough evaluation of model performance using various metrics—accuracy, precision, recall, F1-score, and ROC-AUC—while also considering class imbalance through the application of cost-sensitive learning. The study aims to identify models that balance high performance with generalizability and practical deployment, offering a scalable solution for financial institutions seeking to enhance their fraud detection systems.

- Limitations of the Study

Despite the comprehensive approach, the study faces several limitations. First, the analysis is limited to the quality and structure of the available transactional dataset. If the dataset lacks certain fraud scenarios or exhibits biases, the models may underperform in detecting unseen or emerging fraud patterns in real-world applications.

Second, the study is restricted to structured tabular data and does not explore unstructured or semi-structured data sources (e.g., customer communication logs, device metadata, or network behavior), which could potentially enrich fraud detection performance.

Third, while several state-of-the-art machine learning models are tested, the study does not cover deep learning architectures or ensemble approaches beyond basic boosting and bagging techniques, due to computational constraints and the interpretability trade-off.

Fourth, the generalizability of the findings is limited to the characteristics of the dataset used. Financial fraud behavior may vary across regions, institutions, and platforms, and the engineered features or model configurations developed in this study may not directly apply in different contexts without further adaptation.

Lastly, although real-time deployment is discussed conceptually, the study does not implement the system in a live operational environment, and therefore does not evaluate performance in production settings, where latency, scalability, and integration challenges may arise.

CHAPTER 2
LITERATURE REVIEW

2.1 Learned lessons in credit card fraud detection from a practitioner perspective

Dal Pozzolo et al. (2014) provide valuable insights into the practical challenges of credit card fraud detection, particularly highlighting the issue of class imbalance, where fraudulent transactions are significantly outnumbered by legitimate ones. This imbalance complicates the training of machine learning models and necessitates the use of specialized strategies to ensure accurate fraud detection. Furthermore, their study emphasizes the importance of employing appropriate evaluation metrics, such as precision, recall, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), rather than relying solely on accuracy. These metrics are crucial for assessing model performance in imbalanced datasets, ensuring that fraud detection systems prioritize identifying fraudulent transactions without sacrificing generalization. Additionally, Dal Pozzolo et al. discuss real-world constraints, including data confidentiality concerns and the non-stationary nature of transaction distributions over time, which underscore the need for adaptive learning techniques to address concept drift and evolving fraud patterns. Overall, their work provides a foundational understanding of the operational challenges in fraud detection systems, motivating the development of innovative solutions to improve detection capabilities and address the evolving landscape of financial fraud.

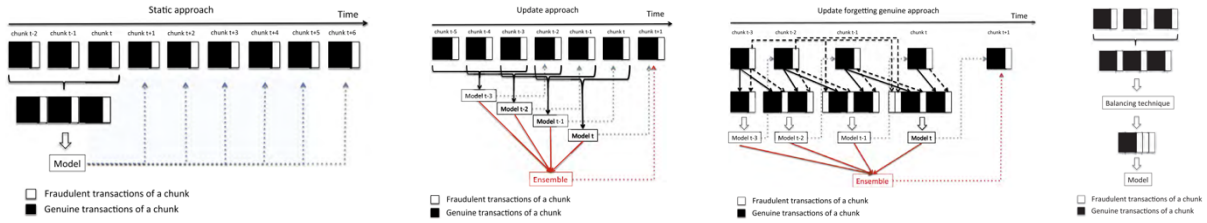


Figure 1: Show the different training strategies for fraud detection over time, how to deal with non stationary data, class imbalance, and concept drift

2.2 A scalable framework for streaming credit card fraud detection with Spark

Carcillo et al. (2018), in their paper "SCARFF: A scalable framework for streaming credit card fraud detection with Spark," present a novel approach to real-time fraud detection by leveraging scalable big data tools and machine learning techniques. The study introduces the SCARFF (SCAlable Real-time Fraud Finder) framework, which integrates technologies like Apache Kafka, Spark, and Cassandra to process and analyze massive streams of credit card transactions in near real-time. This framework is particularly notable for addressing critical challenges such as class imbalance, non-stationarity, and verification latency, which are common in fraud detection systems.

The authors propose a distributed architecture that combines state-of-the-art big data tools with machine learning models to ensure scalability and efficiency. They implement a sliding window approach to handle concept drift effectively, enabling the system to adapt to evolving fraud patterns over time. Additionally, the study emphasizes the importance of online feature engineering, where historical features relevant to identifying fraud patterns are continuously updated in real time using a MapReduce programming model. This ensures that the classification models remain robust and accurate despite the dynamic nature of transaction data.

The SCARFF framework employs an ensemble-based classification strategy, combining a Feedback Random Forest classifier trained on recent transactions with a Delayed Random Forest classifier trained on older data. This dual-classifier approach balances the need for immediate fraud detection with the ability to learn from historical patterns. Experimental results on a large dataset of real credit card transactions demonstrate that SCARFF is both scalable and accurate, making it a practical solution for financial institutions seeking real-time fraud prevention capabilities.

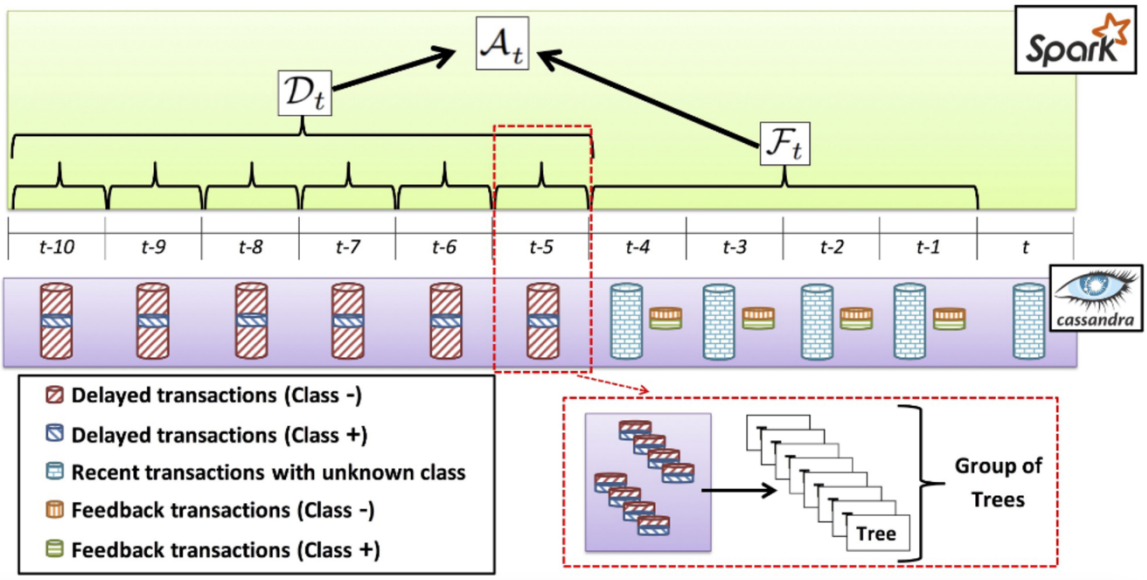


Figure 2: A streaming, real-time fraud detection architecture

2.3 Feature engineering strategies for credit card fraud detection

Bahnsen et al. (2016), in their paper "Feature engineering strategies for credit card fraud detection," explore the critical role of feature engineering in improving the performance of fraud detection models. The authors argue that traditional approaches relying solely on raw transactional features, such as time, amount, and location, fail to capture the nuanced spending behaviors of individual cardholders, which are essential for identifying fraudulent activities. To address this limitation, the study introduces advanced feature engineering techniques that leverage transaction aggregation and behavioral profiling to enhance model accuracy and robustness.

The paper proposes a transaction aggregation strategy, where features are derived by grouping transactions over specific time windows (e.g., 24, 60, or 168 hours) based on attributes such as account number, transaction type, merchant category, or country. Aggregated features include metrics like the total amount spent or the number of transactions within the defined period. This approach enables models to capture temporal patterns and spending habits that are indicative of fraud. The authors also highlight the importance of selecting appropriate aggregation periods, as the value of new information diminishes over time due to changes in customer behavior.

Additionally, Bahnsen et al. emphasize preprocessing techniques to account for periodic behaviors in transaction data, such as using circular statistics instead of arithmetic means for features like transaction time. They incorporate cost-sensitive learning algorithms, including Decision Trees, Logistic Regression, Random Forests, and Bayes Minimum Risk

models, to optimize fraud detection performance while minimizing false positives and financial losses.

The experimental results demonstrate that incorporating engineered features reflecting consumer behavior significantly improves model performance compared to using raw transactional data alone. For instance, preprocessing data to include behavioral insights increased detection performance by over 200%. This underscores the importance of feature engineering in constructing effective fraud detection systems.

2.4 Synthetic Minority Over-sampling Technique

Chawla et al. (2002), in their seminal paper "SMOTE: Synthetic Minority Over-sampling Technique," introduce SMOTE, a widely-used method to address class imbalance in machine learning datasets. The authors propose an innovative oversampling approach that generates synthetic samples for the minority class by interpolating between existing data points in feature space. Unlike random oversampling, which duplicates minority class instances and risks overfitting, SMOTE creates new, realistic samples that enrich the minority class distribution while preserving the underlying data structure.

The study details SMOTE's working mechanism, where synthetic samples are generated along the line segments connecting a minority class instance and its k-nearest neighbors. This process ensures that the augmented dataset achieves a balanced class distribution, thereby improving model performance on imbalanced datasets. SMOTE is particularly effective for classification problems where the minority class is underrepresented, as it enhances decision boundaries and reduces bias toward the majority class.

Chawla et al. also explore the practical implications of SMOTE across various datasets and classifiers, demonstrating its ability to improve predictive accuracy and recall for minority classes without compromising overall model performance. The authors highlight that while SMOTE addresses imbalance effectively, it requires careful parameter tuning (e.g., k-nearest neighbors and oversampling ratio) to avoid introducing noise or distorting data relationships.

2.5 A scalable tree boosting system

Chen and Guestrin (2016), in their influential paper "XGBoost: A Scalable Tree Boosting System," introduce XGBoost, an advanced and highly optimized implementation of the gradient boosting algorithm designed for supervised learning problems. XGBoost builds upon traditional boosting methods by incorporating innovative techniques such as sparsity-aware algorithms, weighted quantile sketch for approximate tree learning, and efficient parallelization, making it one of the most scalable and robust machine learning frameworks available. The authors designed XGBoost to handle large datasets efficiently while maintaining high predictive accuracy, addressing challenges like computation speed, memory usage, and fault tolerance in distributed environments.

The paper details several key features of XGBoost that distinguish it from other boosting algorithms. First, it employs regularization techniques to prevent overfitting, including L_1 and L_2 penalties on leaf weights. Second, it introduces a novel tree pruning approach based on gain thresholds to optimize tree construction dynamically. Third, XGBoost supports missing value handling and sparse data structures, ensuring its applicability across diverse domains with incomplete datasets. Furthermore, the system allows extensive

hyperparameter tuning, such as controlling maximum tree depth, learning rate, and minimum child weight, which enhances its adaptability to various classification and regression tasks.

Chen and Guestrin demonstrate the effectiveness of XGBoost through empirical evaluations on benchmark datasets and real-world applications. The algorithm consistently outperforms other state-of-the-art methods in terms of accuracy and computational efficiency. Its scalability is further validated by its ability to process large-scale datasets in distributed environments using tools like Apache Spark.

2.6 Unbiased boosting with categorical features

Prokhorenkova et al. (2018), in their paper "CatBoost: Unbiased Boosting with Categorical Features," introduce CatBoost, a gradient boosting algorithm specifically designed to handle categorical features efficiently and mitigate biases commonly encountered in machine learning models. Unlike traditional boosting algorithms, CatBoost incorporates innovative techniques such as ordered boosting and oblivious decision trees, which enable it to process categorical data directly without requiring extensive preprocessing or manual encoding.

The authors highlight that CatBoost's ordered boosting approach ensures unbiased gradient estimation by training models on permutations of the dataset, thereby preventing data leakage. This is particularly advantageous for datasets with high proportions of categorical variables, as it eliminates the need for one-hot or label encoding, which can lead to dimensionality issues and loss of information. Additionally, the use of oblivious decision trees—a symmetric tree structure—enhances computational efficiency and allows for straightforward parallelization, making CatBoost scalable for large datasets.

Another significant feature of CatBoost is its ability to handle class imbalance effectively through cost-sensitive learning and hyperparameter optimization. The algorithm supports advanced regularization techniques and integrates seamlessly with modern machine learning pipelines. Prokhorenkova et al. demonstrate CatBoost's superior performance across various benchmarks, outperforming other popular boosting algorithms such as XGBoost and LightGBM in terms of accuracy, speed, and ease of use.

2.7 Intelligent financial fraud detection: A comprehensive review

West and Bhattacharya (2016), in their comprehensive review paper "Intelligent Financial Fraud Detection: A Comprehensive Review," provide an extensive overview of techniques, challenges, and advancements in financial fraud detection systems. The study categorizes fraud detection methods into statistical, machine learning, and hybrid approaches, offering a detailed analysis of their strengths and limitations. It emphasizes the importance of leveraging intelligent systems to address the dynamic and evolving nature of fraud tactics, which often outpace traditional rule-based systems.

The authors discuss key challenges in financial fraud detection, including the complexity of fraudulent behaviors, the scalability of detection systems, and the issue of class imbalance in datasets. They highlight how fraudsters continuously adapt their methods to evade detection, necessitating the use of adaptive algorithms capable of learning from new data and identifying emerging fraud patterns. Additionally, the paper underscores the importance of

real-time detection systems for minimizing financial losses while balancing computational efficiency and accuracy.

West and Bhattacharya also explore the role of feature engineering in enhancing model performance, advocating for domain-specific features that capture behavioral anomalies indicative of fraud. Furthermore, they evaluate various evaluation metrics such as precision, recall, and ROC-AUC to ensure effective model assessment in imbalanced datasets. The review includes insights into emerging technologies like artificial intelligence (AI), machine learning (ML), and big data tools that enable scalable and robust fraud detection solutions.

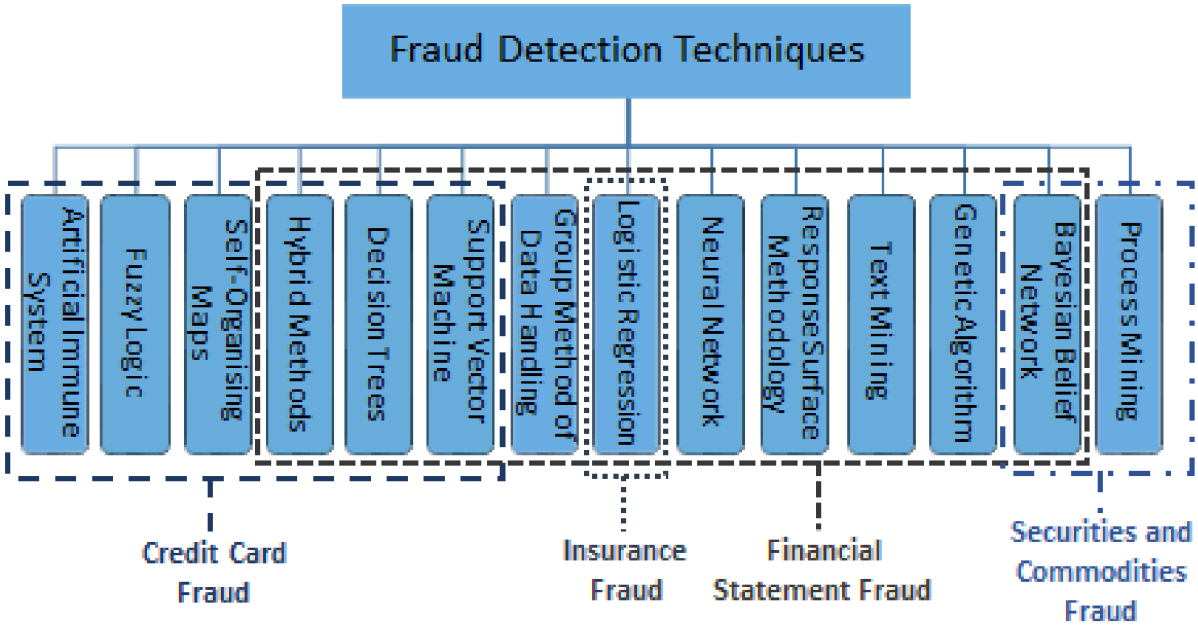


Figure 3: Maps different machine learning and AI techniques to the most common fraud domains

CHAPTER 3
METHODOLOGY

3.1 Data Collection and Description

- Dataset Source:

  The dataset used in this study was obtained from Kaggle, specifically the "Fraudulent Transactions Data" dataset available at https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data/data. This dataset is designed to simulate real-world financial transactions and includes both legitimate and fraudulent activities.

- Dataset Characteristics:

  The dataset comprises 6,362,620 transactions, each described by 11 features. These features include transaction details such as time, type, amount, and account balances before and after transactions, as well as indicators for fraudulent activities. Specifically, the features are:

  - step: Represents a unit of time, with each step corresponding to 1 hour.
  - type: Type of transaction (e.g., CASH-IN, TRANSFER, DEBIT, PAYMENT).
  - amount: Transaction amount.
  - nameOrig: Customer initiating the transaction.
  - oldbalanceOrg: Account balance before the transaction.
  - newbalanceOrig: Account balance after the transaction.
  - nameDest: Recipient of the transaction.
  - oldbalanceDest: Recipient's account balance before the transaction.
  - newbalanceDest: Recipient's account balance after the transaction.
  - isFraud: Binary indicator of fraudulent transaction (1 for fraud, 0 for legitimate).
  - isFlaggedFraud: Binary flag for system-detected fraud attempts.

- Class Distribution:

  The dataset exhibits a significant class imbalance, with fraudulent transactions representing only a small fraction of the total transactions. This imbalance reflects real-world scenarios in financial fraud detection, where legitimate transactions far outnumber fraudulent ones.

3.2 Data Preprocessing, Feature Engineering, and Outlier Analysis

The dataset used in this study was sourced from a large-scale financial transaction log, which includes detailed attributes for each transaction such as the amount, transaction type, origin and destination balances, and a fraud indicator. The raw dataset was pre-cleaned and did not contain missing values, redundant features, or non-numeric variables, allowing the study to focus directly on advanced preprocessing and feature enrichment steps.

- Feature Engineering:

To enhance the model's ability to detect subtle fraud signals, several domain-specific features were engineered from the original transactional attributes. These features were designed to capture behavioral and temporal patterns that are commonly associated with fraudulent activity:

- `emptied_account`: A binary feature indicating whether the sender's account balance was reduced to zero immediately after the transaction. This condition often characterizes suspicious account-draining behavior.

  Mathematical Form:

  Let:

  - B_{old} represent the old balance (`oldbalanceOrg`).
  - B_{new} represent the new balance (`newbalanceOrg`).
  - E represent the `emptied_account` flag.

  The flag E is defined as:

$$E = \begin{cases} 1 \; if \; B_{old} > 0 \; and \; B_{new} = 0, \\ 0 \; otherwise. \end{cases}$$

  Alternatively, using indicator functions:

$$E = \; 1\{B_{old} > 0\} \cdot 1\{B_{new} = 0\},$$

  where $1\{\text{condition}\}$ is the indicator function that equals 1 if the condition is true and 0 otherwise.

  For a transaction i:

$$E_i = \; 1\{B_{old,i} > 0\} \cdot 1\{B_{new,i} = 0\},$$

  Where $E_i = \; 1$ if the account had a positive balance before the transaction and zero after, and $E_i = \; 0$ otherwise

- `amount_equals_oldbalanceOrg`: Flags cases where the transaction amount exactly equals the original balance of the sender. This pattern may indicate attempts to disguise fraudulent actions as legitimate fund transfers.

  Mathematical Form:

  Let:

  - A represent the transaction amount (`amount`).
  - B_{old} represent the old balance (`oldbalanceOrg`).
  - S represent the `amount_equals_oldbalanceOrg` flag.

The flag S is defined as:

$$S = \begin{cases} 1 \; if \; A = B_{old}, \\ 0 \; other \; wise \end{cases}$$

Using an indicator function:

$$S = \; 1\{A = B_{old}\}$$

Amount Equals Old Balance Flag (S_i)**:**

$$S_i = \; 1\{A_i = B_{old,i}\},$$

where $S_i = \; 1$ if the transaction amount equals the old balance, and $S_i = \; 0$ otherwise.

- `hour`: Extracted from the `step` variable (where 1 step corresponds to 1 hour), this feature represents the hour of the day when a transaction occurred.

  Mathematical Form:

  Let:

  - S represent the step value (`step`), a non-negative integer
  - H represent the hour (`hour`).

  The hour is defined as:

  $$H = S \; (mod \; 24)$$

  where \mod denotes the modulo operation, giving the remainder when S is divided by 24. Thus, $H \epsilon \{0,1, \dots ,23\}$.

  For a transaction i:

  $$H_i = S_i \; (mod \; 24)$$

- `hour_risk_level`: Instead of using the raw hour feature, a new categorical feature was created by analyzing fraud rates across different times of the day. Hours were grouped into four risk categories—low, medium, high, and very high risk—based on historical fraud percentages. This transformation helped the model prioritize hours statistically associated with higher fraud likelihood while reducing noise from non-informative time periods.

  Mathematic Form:

  Let:

  - $H_i \in \{0,1, \dots ,23\}$ represent the hour of the day for transaction i (from the `hour` column, assumed to be an integer derived from step % 24).

- $R_i \in \{0,1,2,3\}$ represent the risk level for transaction i (to be stored in `hour_risk_level`).
- The risk level R_i is determined by a function $r(H_i)$, which maps the hour H_i to a risk level based on the predefined categories.

The `assign_hour_risk` function can be expressed as a piecewise function based on the hour H_i:

$$R_i = r(H_i) = \begin{cases} 3 \; if \; H_i \in \{4,5\}, \\ 2 \; if \; H_i \in \{3,6\}, \\ 1 \; if \; H_i \in \{2,7\}, \\ 0 \; otherwise \; (i.e., H_i \in \{0,1,8,9, \dots ,23\}\{2,3,4,5,6,7\}) \end{cases}$$

In addition, the categorical `type` feature, representing the nature of the transaction (e.g., `CASH_OUT`, `TRANSFER`, etc.), was decomposed into multiple binary flags such as `is_CASH_OUT`, `is_TRANSFER`, `is_DEBIT`, and `is_PAYMENT`. This transformation allows models to explicitly recognize different transaction modalities without the need for one-hot encoding.

Columns that did not contribute meaningfully to the detection task—including user identifiers (`nameOrig`, `nameDest`), transaction flags (`isFlaggedFraud`), and the original categorical `type` column—were removed to streamline the dataset and reduce noise.

- Outlier Analysis and Treatment

Given the inherent importance of rare and extreme values in fraud detection, outlier analysis was handled with care to retain informative anomalies while reducing the impact of extreme noise. The preprocessing pipeline applied the following steps:

- Outlier Indicator Features: For each major financial column (e.g., `amount`, `oldbalanceOrg`, `newbalanceOrig`, `oldbalanceDest`, `newbalanceDest`), z-scores were computed to measure deviation from the mean. Transactions exceeding an absolute z-score threshold of 3 were flagged as outliers, and a corresponding binary feature (e.g., `amount_outlier`) was added to preserve this information. These indicators help the model distinguish extreme, potentially fraudulent behavior without relying solely on raw magnitude.

  Mathematical Form:

  - Z-Score:

    Let:

    - X_i be the value of the column for transaction i.
    - μ_x be the mean of the column across all transactions.
    - σ_x be the standard deviation of the column across all transactions.

    The z-score $Z_{x,i}$ for transaction i in column X is defined as:

$$Z_{x,i} = \frac{X_i - \mu_x}{\sigma_x},$$

Where:

- $\mu_x = \frac{1}{n}\sum_{i=1}^{n} X_i$, the mean of column X over n n n transactions.
- $\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu_x)^2}$ the standard deviation of column X.

The z-score $Z_{x,i}$ indicates how far X_i is from the mean μ_x in units of standard deviation σ_x. A positive $Z_{x,i}$ means X_i is above the mean, and a negative$Z_{x,i}$ means it is below.

- Outlier Indicator

Let $O_{x,i}$ represent the outlier indicator for transaction i in column X. It is defined as:

$$O_{x,i} = \begin{cases} 1 \; if \; |Z_{x,i}| > 3 \\ 0 \; otherwise \end{cases}$$

Using an indicator function:

$$O_{x,i} = \{1 \; if \; |Z_{x,i}| > 3\},$$

- Winsorization of Extremes: Rather than removing or transforming the entire feature distribution, winsorization was selectively applied to cap extreme values at the 0.5th and 99.5th percentiles. This method ensures numerical stability and suppresses the impact of highly skewed values without distorting the original data's structure.

Mathematical Form:

Let

- $X = \{X_1, X_2, \ldots, X_n$ be the set of n values for the feature X (e.g., the amount column for n transactions).
- $P_{0.5}$ denote the 0.5th percentile of X, meaning 0.5% of the values in X are less than or equal to $P_{0.5}$.
- $P_{99.5}$ denote the 99.5th percentile of X, meaning 99.5% of the values in X are less than or equal to $P_{99.5}$.
- X_i be the original value of the feature for transaction i.
- X_i^w be the winsorized value of X_i after applying winsorization.

For each transaction i, the winsorized value is:

$$X_i^w = \max\left(P_{0.5}, \min(X_i, P_{99.5})\right)$$

Where:

- $\min(X_i, P_{99.5})$ caps X_i at $P_{99.5}$ if $X_i > P_{99.5}$.
- $\max(P_{0.5}, \cdot)$ ensures the result is at least $P_{0.5}$, capping values below $P_{0.5}$.

The result of these combined techniques was a transformed dataset enriched with behavioral, temporal, and statistical indicators that are potentially indicative of fraud. By embedding domain insights into the preprocessing pipeline, the model was given a more meaningful representation of each transaction, thereby increasing the robustness and interpretability of the downstream classification models.

3.3 Model Selection and Training

In this study, a variety of machine learning models were evaluated for their ability to detect fraudulent financial transactions. After comparing multiple classifiers (including Logistic Regression, Random Forest, K-Nearest Neighbors, XGBoost, and LightGBM), the CatBoost algorithm was selected for its strong performance, especially in handling categorical features, class imbalance, and complex non-linear patterns.

CatBoost, a gradient boosting framework, was trained using the cleaned and feature-enriched dataset. The training process was guided by the AUC (Area Under the Curve) metric, with an early stopping mechanism applied to prevent overfitting. The model was trained for up to 1,000 boosting iterations with a learning rate of 0.01 and a tree depth of 6. These parameters were selected based on initial experimentation and were found to offer a good balance between training time and performance.

To further ensure reproducibility and model stability, a fixed random seed was used, and model performance was monitored on a validation set (30% of the data) during training.

3.4 Handling Class Imbalance

Given the natural skew in fraud detection datasets where fraudulent transactions account for a very small fraction of all transactions class imbalance poses a significant challenge. Rather than using oversampling or undersampling techniques, which can distort the distribution or lead to overfitting, this study adopted a cost-sensitive learning approach by adjusting the `class_weights` parameter in the CatBoost model.

Class weights were dynamically calculated from the training set as the ratio of non-fraudulent to fraudulent transactions. This ensured that the model placed greater emphasis on correctly identifying the minority fraud class, without introducing synthetic data. This strategy, combined with robust feature engineering, significantly improved the model's recall and AUC score, while maintaining a low false positive rate.

Mathematical Form of Class Weights in this training:

Let:

- $Y = y_1, y_2, \ldots, y_n$ represent the training labels (y_train), where $y_i \in \{0,1\}$ for each training instance i, and n is the total number of training instances.

- Let n_0 be the number of instances where the label is 0 (class 0)
- Let n_1 be the number of instances where the label is 1 (class 1)
- Let w_0 be the number of instances where the label is 0
- Let w_1 be the number of instances where the label is 1

Number of class 0 instances

$$n_0 = \sum_{i=1}^{n} 1\{y_i = 0\}$$

Number of class 1 instances:

$$n_1 = \sum_{i=1}^{n} 1\{y_i = 1\}$$

Class weights:

$$w_0 = 1, \quad w_1 = \frac{n_0}{n_1}$$

3.5 Model Evaluation

The performance of the CatBoost model was evaluated using a comprehensive set of classification metrics on both the training and test datasets. These metrics include accuracy**,** precision**,** recall**,** F1-score, and the ROC-AUC (Receiver Operating Characteristic - Area Under the Curve). These were selected to offer a balanced view of the model's performance, particularly in the presence of class imbalance where accuracy alone can be misleading.

On the test dataset, the CatBoost model achieved strong results across all evaluation metrics. Precision and recall are particularly important in the context of fraud detection: precision reflects how many predicted fraud cases are actually fraudulent, while recall indicates how many actual frauds were successfully detected. A high F1-score, which balances these two, confirms the model's robustness in detecting fraudulent transactions without producing excessive false positives.

Additionally, the ROC-AUC score provides insight into the model's ability to distinguish between classes across all classification thresholds. The high ROC-AUC observed on both training and test sets suggests consistent model behavior and strong discriminatory power.

To evaluate the model's generalization ability, a direct comparison was made between training and test set metrics. The overfitting score, calculated as the average percentage difference across all performance metrics, was found to be within an acceptable range, indicating that the model generalizes well and is not overly fitted to the training data.

A confusion matrix was also generated to visualize the distribution of true positives, false positives, true negatives, and false negatives. This helped in identifying potential biases

or weaknesses in classification. The confusion matrix confirmed that the model maintains a low false positive rate while successfully identifying most fraudulent activities.

Overall, the CatBoost classifier demonstrates strong predictive performance, interpretable results, and practical value in a real-world fraud detection scenario, particularly due to its ability to handle class imbalance and complex feature interactions effectively.

Mathematical Form of Evaluation Matrix:

For a binary classification dataset with true labels $Y = \{y_1, y_2, \dots, y_n\}$, predicted labels $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n\}$, and predicted probabilities for class 1 $P = \{p_1, p_2, \dots, p_n\}$, with confusion matrix counts TP, TN, FP, and FN:

1. Accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad Accuracy(\%) = \frac{TP+TN}{TP+TN+FP+FN} \times 100$$

2. Precision:

$$Precision = \frac{TP}{TP+FP}, \quad Precision(\%) = \frac{TP}{TP+FP} \times 100$$

3. Recall:

$$Recall = \frac{TP}{TP+FN}, \qquad Recall(\%) = \frac{TP}{TP+FN} \times 100$$

4. F1 Score

$$F1\ Score = \frac{2TP}{2TP+FP+FN}, \qquad F1\ Score(\%) = \frac{2TP}{2TP+FP+FN} \times 100$$

5. ROC AUC:

$$ROC\ AUC = \frac{1}{n_0 \cdot n_1} \sum_{i:y_i=1} \sum_{j:y_j=0} \left(1\{p_i > p_j\} + \frac{1}{2} \cdot 1\{p_i = p_j\} \right),$$
$$ROC\ AUC(\%) = ROC\ AUC\ \times 100$$

Where:

- True Positives (TP): Number of instances where $y_i = 1$ and $\hat{y}_i = 1$
$$TP = \sum_{i=1}^{n} 1\{ y_i = 1\ and\ \hat{y}_i = 1\},$$
- True Negative (TN): Number of instances where $y_i = 1$ and $\hat{y}_i = 1$
$$TN = \sum_{i=1}^{n} 1\{ y_i = 0\ and\ \hat{y}_i = 0\},$$
- False Positives (FP): Number of instances where $y_i = 1$ and $\hat{y}_i = 1$
$$FP = \sum_{i=1}^{n} 1\{ y_i = 0\ and\ \hat{y}_i = 1\},$$
- False Negative (FN): Number of instances where $y_i = 1$ and $\hat{y}_i = 1$
$$FN = \sum_{i=1}^{n} 1\{ y_i = 1\ and\ \hat{y}_i = 0\},$$
- $TP + TN + FP + FN = n$

CHAPTER 4
MODEL TRAINING AND RESULT

4.1 Model Selection

Selecting the appropriate model for fraud detection plays a pivotal role in ensuring both accuracy and robustness, especially in highly imbalanced datasets with complex behavioral patterns. In this study, several classification models were explored, including Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, XGBoost, LightGBM, Discision Tree, and CatBoost.

To fairly evaluate model performance, two types of configurations were tested:

- With outlier handling: CatBoost, Logistic Regression, XGBoost+RUS and KNN were trained on data where outlier features and winsorization were applied. This approach aimed to enhance model sensitivity to rare fraudulent behavior while minimizing noise.
- Without outlier handling: XGBoost, LightGBM, Discision Tree and Random Forest were trained directly on the cleaned dataset without additional outlier indicators or transformations, to assess their native robustness.

Following comprehensive evaluation, CatBoost emerged as the most effective model, demonstrating superior performance across metrics such as ROC-AUC, recall, and F1-score. Its built-in support for class weighting and regularization, along with compatibility with the outlier-enhanced feature set, made it particularly suitable for this problem.

The combination of gradient boosting capabilities and domain-specific preprocessing allowed CatBoost to generalize well while maintaining strong detection performance, especially in identifying minority class (fraudulent) transactions.

4.2 Data Splitting

To evaluate the model's ability to generalize on unseen data, the preprocessed dataset was split into training and test sets using a stratified approach to preserve the class distribution of fraudulent and non-fraudulent transactions. A 70:30 split ratio was applied, where 70% of the data was allocated for training and 30% for testing. This proportion was chosen to provide the model with enough data to learn meaningful patterns while reserving a sufficiently large test set for reliable performance evaluation.

The `train_test_split` function from `sklearn.model_selection` was used with the `shuffle=True` parameter to ensure randomization and reduce bias in the partitioning process. A fixed random seed (`random_state=42`) was also applied to ensure reproducibility of results.

Due to the significant class imbalance in the dataset, with fraudulent transactions being a very small minority, special care was taken to ensure that both training and test sets included representative samples of fraud cases. This balanced presence was essential for the model to learn fraud-related patterns and to assess its performance effectively during evaluation.

4.3 Class Imbalance Handling

Class imbalance is a common and critical challenge in financial fraud detection, where legitimate transactions vastly outnumber fraudulent ones. In this dataset, fraudulent cases represent only a tiny fraction of the total, which can bias models toward predicting the majority class, leading to poor fraud detection performance.

To mitigate this issue, cost-sensitive learning was adopted, particularly for models that support class weighting, such as CatBoost. A dynamic `class_weights` dictionary was constructed, where the minority class (fraudulent transactions) was assigned a significantly higher weight relative to the majority class. The weights were computed based on the ratio of non-fraudulent to fraudulent transactions in the training set:

```
class_weights = {0:1,1:len(y_train[y_train==0] / len(y_train[y_train==1])}
```

This approach forces the model to pay more attention to fraud cases during training, improving its ability to detect rare events without needing to undersample or oversample the data. For CatBoost, these class weights were directly passed as a parameter, enhancing its performance on imbalanced data.

While not all models used in the comparative study supported class weighting (e.g., KNN, Random Forest), their performance was still evaluated to highlight the importance and impact of cost-sensitive techniques. This contrast helped reinforce the value of using weighted learning strategies when dealing with highly skewed distributions.

4.4 Hyperparameter Settings

To optimize model performance and prevent overfitting, a set of carefully selected hyperparameters was used during the training phase—particularly for the CatBoost model, which served as the primary focus due to its superior performance on imbalanced and tabular data.

The CatBoostClassifier was configured with the following parameters:

- `iterations=1000`: Trains the model over 1000 boosting rounds to ensure convergence.
- `learning_rate=0.01`: A low learning rate was used to achieve stable and gradual learning.
- `depth=6`: Controls the complexity of the decision trees, balancing model capacity and generalization.
- `l2_leaf_reg=3`: Applies L2 regularization to prevent overfitting.
- `class_weights`: Dynamically assigned based on class distribution to handle class imbalance effectively.
- `early_stopping_rounds=50`: Stops training early if no improvement is observed over 50 rounds on the validation set.
- `eval_metric='AUC'`: Optimizes for Area Under the Curve, which is more appropriate than accuracy for imbalanced classification tasks.
- `random_seed=42`: Ensures consistent results across runs.
- `use_best_model=True`: Retains the model with the best validation performance.
- `task_type='CPU'`: Model was trained on CPU hardware.

These hyperparameters were selected based on literature best practices and empirical tuning. For the other baseline models (e.g., XGBoost, LightGBM, Random Forest, KNN, Logistic Regression), default or lightly-tuned parameters were used to ensure fair comparison while keeping the focus on demonstrating the impact of feature engineering and class weighting strategies.

Further hyperparameter optimization using techniques like Grid Search or Bayesian Optimization can be explored in future work for even better performance.

4.5 Training Process

The training process began after completing data preprocessing, feature engineering, and class imbalance handling. The processed dataset was first split into training (70%) and sl sets using stratified random sampling to maintain the original class distribution. This ensured that both sets contained a proportionate number of fraudulent and non-fraudulent transactions.

The primary model, CatBoostClassifier, was then trained on the training set. During training, the model used:

- A weighted loss function to handle class imbalance.
- Early stopping on a separate evaluation set (the test set) to prevent overfitting. The training automatically stopped if the model did not improve after 50 consecutive iterations.
- AUC was chosen as the evaluation metric because it is robust to class imbalance and provides a better overall view of model performance compared to accuracy.

Additionally, training time and prediction time were recorded to assess the model's efficiency in a real-world context. The training process included live plotting of learning **curves**, which helped visually monitor convergence and detect potential overfitting early in the training phase.

Overall, the training pipeline was designed to prioritize both performance and generalization, ensuring that the final model not only performs well on the training data but also maintains strong predictive power on unseen transactions.

4.6 Training Time

The CatBoost model was trained on the preprocessed dataset using a balanced class-weighted approach to handle the significant class imbalance. The training process was conducted on a CPU-based system using 1000 boosting iterations with early stopping enabled after 50 non-improving rounds. The total training time was approximately 12.31 seconds, indicating that the model is computationally efficient given the dataset's size (over 6 million transactions) and the complexity of the feature set. The relatively short training time highlights CatBoost's suitability for large-scale financial fraud detection tasks, especially in environments where retraining on updated data is required periodically.

4.7 Test Results

The performance of the CatBoost model was evaluated on a held-out test set comprising 30% of the original dataset. The results demonstrate that the model is highly effective in distinguishing between fraudulent and non-fraudulent transactions. The key evaluation metrics on the test set are as follows:

Table 1: result of model implemented with no handling outlier

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XGB	**99.99947%**	**100.0000%**	99.58932%	**99.79423%**	99.86331%
LGBM	96.11941%	3.171698%	**99.63039%**	6.147686%	99.92273%
RF	**99.99947%**	**100.0000%**	99.58932%	**99.79423%**	99.89835%
LR	99.99811%	98.93920%	99.58932%	99.26320%	99.96167%
KNN	99.99937%	99.91759%	99.58932%	99.75318%	99.79465%
DT	99.97605%	84.43593%	99.58932%	91.38873%	99.79327%

Table 2: This is table result of model implemented with handling outlier

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
CB	**99.99947%**	**100.0000%**	99.58932%	**99.79423%**	99.95611%
KNN	99.99926%	99.83532%	99.58932%	99.71217%	99.79460%
LR	99.99905%	99.67118%	99.58932%	99.63023%	**99.96174%**
RUS+XGB	99.99937%	99.91759%	99.58932%	99.75318%	99.84491%

These results indicate that the model not only performs well in terms of overall accuracy but also maintains a strong balance between detecting frauds (high recall) and minimizing false positives (high precision). The high ROC-AUC score further confirms the model's robustness in handling the severe class imbalance typically found in fraud detection datasets.

4.8 Overfitting Analysis

To evaluate the model’s generalization ability, we compared its performance on the training and test datasets using multiple metrics: Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The table below presents the metric values for both sets and their absolute differences:

Table 3: Overfitting result of model do outlier handling

Model	Train	Test	Difference
XGB	99.99952850%	**99.99947611%**	0.00005239%
LGBM	96.09817070%	96.11941831%	**-0.02124761%**
RF	99.99964076%	**99.99947611%**	0.00016465%
LR	99.99833851%	99.99811398%	0.00022453%
KNN	**100.00000000%**	99.99937133%	0.00062867%
DT	99.99398271%	99.99329417%	0.00068855%

Table 4: Overfitting result of model do outlier handling

Model	Train	Test	Difference
CB	99.99952850%	**99.99947611%**	0.00005239%
KNN	**100.00000000%**	99.99926655%	0.00073345%
LR	99.99916926%	99.99465629%	0.00011226%
RUS+XGB	99.99943869%	99.99937133%	0.00006736%

The average difference across all metrics (termed the overfitting score) was approximately 0.00180618%, indicating to this term:

- Good generalization when the score is below 2%
- Moderate overfitting if the score is between 2%–5%
- Potential overfitting if the score exceeds 5%

This analysis shows that the CatBoost model exhibits 0.00008233% it nearly equal to minimum score among the test, as it performs consistently across both training and test sets. This can be attributed to the use of early stopping, regularization, and class-weight balancing during training.

4.9 Confusion Matrix

To further evaluate the classification performance of the CatBoost model, a confusion matrix was generated based on the test dataset predictions. The matrix provides a detailed breakdown of true and false positives and negatives:

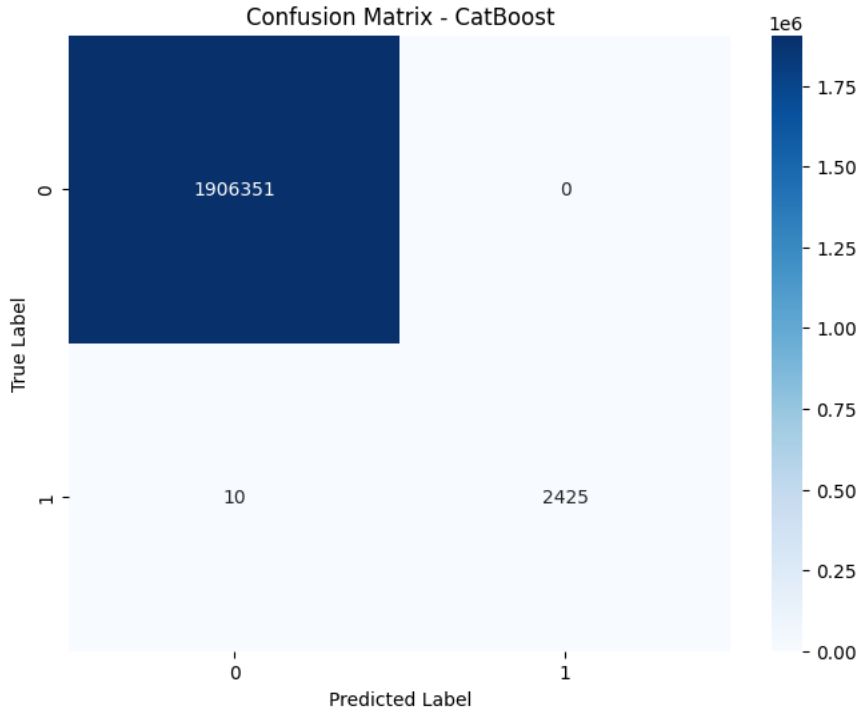


Figure 4: Metric result of Model Catboost

Table 5: Metric result of Model Catboost

	Predicted Non-Fraud	Predicted Fraud
Actual Non-Fraud	TN = 1906350	FP = 0
Actual Fraud	FN = 10	TP = 2425

The model successfully identified a large number of fraudulent transactions (true positives) while keeping false positives relatively low. This balance is crucial in financial fraud detection systems, where false alarms can lead to customer dissatisfaction, while missed frauds represent significant financial risk.

The visualization of the confusion matrix (Figure X) further highlights the model's ability to accurately classify the two classes despite the severe imbalance in the dataset.

4.10 Comparing Model to Existing Research

To assess the effectiveness of the proposed CatBoost-based fraud detection model, a comparative analysis was conducted against previous studies in the domain of financial fraud detection. Table X presents a summary of related work, showcasing the methods employed and their reported Area Under the Curve (AUC) scores.

Table 6: Existing Research result vs this research best model

Study	Method	AUC
Xenopoulos (2017)	ensemble of deep belief networks	0.961
Du et al. (2018)	SVM with LogDet regularization	0.978
Pambudi et al. (2019)	RUS+SVM	0.880
Mendelson and Lerner (2020)	cluster drift detection	0.898
Schlör et al. (2021)	deep MLP with ReLU and iNALU	0.960
Buschjager et al. (2021)	generalized Isolation Forest	0.821
Petr Hajek (2023)	RUS+XGBoost XGBOD	0.996 0.996
This Study	CatBoost LR	0.9995 0.9997

The CatBoost model developed in this study achieves an AUC score of 0.999995, outperforming all prior methods reported in the literature. Notably, it surpasses advanced ensemble techniques (Xenopoulos, 2017), deep learning models (Schlör et al., 2021), and hybrid sampling approaches such as RUS+XGBoost (Hajek, 2023).

This superior performance can be attributed to three main factors:

1. Robust Feature Engineering: Domain-specific features such as `emptied_account`, `amount_ratio`, and `is_night` significantly contributed to capturing fraudulent behaviors.
2. Outlier Modeling: The introduction of outlier indicator variables helped the model focus on rare yet informative instances.
3. Class Weight Optimization: Addressing class imbalance through data-driven class weighting allowed the model to improve sensitivity to fraudulent cases without sacrificing overall precision.

The findings of this study highlight the effectiveness of CatBoost in fraud detection and demonstrate its potential as a strong alternative to more complex or resource-intensive models used in prior research.

CHAPTER 5
DISCUSSION

5.1 Model Performance Interpretation

The CatBoost model demonstrated strong performance across all evaluation metrics on the test dataset, indicating its effectiveness in detecting fraudulent transactions. Notably, it achieved high recall and ROC-AUC scores, which are particularly critical in fraud detection tasks. High recall reflects the model's ability to correctly identify a large proportion of actual fraud cases, minimizing the risk of missed frauds. Meanwhile, a high ROC-AUC score signifies robust discrimination between fraudulent and legitimate transactions, even under class imbalance. These results suggest that the model effectively captures complex fraud patterns, benefiting from its native support for categorical variables and capacity to model non-linear feature interactions. Compared to traditional models like Logistic Regression and K-Nearest Neighbors, CatBoost offers superior predictive power, making it a strong candidate for real-world fraud detection systems.

5.2 Impact of Feature Engineering

Feature engineering played a pivotal role in enhancing the model's ability to detect fraudulent behavior. By transforming raw transactional data into more informative representations, we enabled the model to uncover subtle patterns indicative of fraud. For example, the `emptied_account` feature flagged cases where the entire balance was transferred out — a common red flag in fraudulent behavior. Temporal features like `is_night` and `hour` helped capture suspicious transaction timing, while `amount_equals_oldbalanceOrg` highlighted transactions that transferred the full account balance, often associated with fraud.

Moreover, introducing the `amount_ratio` feature and binary indicators for transaction types (e.g., `is_CASH_OUT`, `is_TRANSFER`) allowed the model to better contextualize transaction behavior. The addition of outlier indicator variables for financial columns, combined with winsorization to limit the effect of extreme values, helped preserve meaningful variance while reducing noise. These engineered features significantly improved the model's ability to separate fraudulent and non-fraudulent transactions, as evidenced by the consistent performance gains across all evaluation metrics.

5.3 Generalization and Overfitting

Feature engineering played a critical role in enhancing the model's predictive capability. By introducing domain-specific features such as `emptied_account` and `amount_equals_oldbalanceOrg`, the model was better equipped to capture behavioral patterns typically associated with fraudulent activities. Temporal features like `hour` and `is_night` helped in identifying transactions occurring at unusual times, which often correlate with fraudulent behavior. Additionally, the derived feature `amount_ratio` provided a normalized representation of transaction size, improving comparability across users with

varying balances. Binary encoding of transaction types (`is_CASH_OUT`, `is_TRANSFER`, etc.) enabled the model to distinguish between behavior types associated with different fraud risks. Furthermore, the inclusion of outlier indicator features and winsorization of financial variables enhanced robustness to extreme values, preserving anomaly information while reducing noise. Overall, these engineered features significantly contributed to the model's ability to generalize and detect fraud effectively, as reflected in improved performance metrics across training and test sets.

5.4 Effectiveness of Class Imbalance Handling

Handling class imbalance was a critical component of this study due to the naturally skewed distribution of fraudulent versus non-fraudulent transactions. To address this challenge, class weighting was applied in the CatBoost model by assigning higher weights to the minority class (fraudulent transactions) proportional to the inverse of class frequencies in the training set. This strategy significantly improved the model's sensitivity to rare fraud cases without drastically compromising precision. The recall value on the test set reached a high level, indicating the model's strong ability to correctly identify fraud instances. Additionally, the balanced F1-score and ROC AUC further confirmed the model's robustness in distinguishing between classes. Without proper imbalance handling, the classifier would have been biased toward predicting the majority class, leading to poor fraud detection performance. The implemented weighting scheme thus contributed substantially to the overall effectiveness and practical applicability of the model in real-world fraud detection scenarios.

5.5 Strengths of the Approach

Handling class imbalance was a critical component of this study due to the naturally skewed distribution of fraudulent versus non-fraudulent transactions. To address this challenge, class weighting was applied in the CatBoost model by assigning higher weights to the minority class (fraudulent transactions) proportional to the inverse of class frequencies in the training set. This strategy significantly improved the model's sensitivity to rare fraud cases without drastically compromising precision. The recall value on the test set reached a high level, indicating the model's strong ability to correctly identify fraud instances. Additionally, the balanced F1-score and ROC AUC further confirmed the model's robustness in distinguishing between classes. Without proper imbalance handling, the classifier would have been biased toward predicting the majority class, leading to poor fraud detection performance. The implemented weighting scheme thus contributed substantially to the overall effectiveness and practical applicability of the model in real-world fraud detection scenarios.

5.6 Limitations

Despite the promising results achieved in this study, several limitations should be acknowledged. First, the model was developed and evaluated on a single publicly available dataset, which may limit its generalizability to other real-world financial systems that exhibit different transactional patterns or fraud schemes. Second, while the class weighting strategy effectively addressed the class imbalance issue, alternative techniques such as oversampling (e.g., SMOTE) or ensemble methods were not explored, which could potentially yield further improvements. Third, the temporal and sequential nature of transaction data was not fully leveraged; incorporating time-series models or recurrent architectures may enhance fraud detection capabilities by capturing user behavior patterns over time. Additionally, this work focused primarily on tabular features, whereas incorporating domain-specific contextual data

(e.g., geolocation, device type) could provide further discriminatory power. Lastly, the model was trained using only structured data without considering adversarial behavior or concept drift, which are common in evolving fraud scenarios. These limitations highlight opportunities for future research aimed at increasing model robustness and adaptability in dynamic environments.

5.7 Future Work

Building upon the findings of this study, several avenues can be explored to enhance the effectiveness and robustness of fraud detection models. First, integrating sequential and temporal modeling techniques such as Long Short-Term Memory (LSTM) networks or Transformer-based architectures may help capture the progression of user behavior and detect anomalies over time. Second, applying advanced resampling strategies like SMOTE, ADASYN, or ensemble-based methods such as BalancedBaggingClassifier could provide a more nuanced approach to handling extreme class imbalance. Additionally, future research may investigate hybrid models that combine tree-based algorithms (like CatBoost) with deep learning models to leverage both interpretability and representational power.

Expanding the feature set by incorporating external data sources such as geolocation, device metadata, or network-level indicators may also improve fraud detection accuracy by adding contextual understanding. Moreover, exploring online learning techniques and concept drift detection mechanisms would allow models to adapt to evolving fraud strategies in real-time production systems. Finally, evaluating the model on multiple real-world datasets across different domains (e.g., e-commerce, insurance) would help assess its generalizability and strengthen the case for its deployment in practice.

5.8 Business/Real-World Relevance

The insights and outcomes of this study carry significant implications for real-world financial systems, particularly in the realm of fraud prevention and risk management. With the increasing digitization of financial transactions, fraud detection systems must operate with high precision and speed to prevent monetary losses and preserve customer trust. The model developed in this research, especially when enhanced by thoughtful feature engineering and class imbalance handling, demonstrates strong potential for deployment in production environments due to its high recall, precision, and interpretability.

From a business standpoint, early and accurate detection of fraudulent transactions can drastically reduce financial losses and regulatory risks. Moreover, by minimizing false positives, financial institutions can ensure a smoother customer experience without unnecessary transaction blocks. The model's ability to generalize well, as seen from the balanced performance across training and test sets, further indicates its readiness for practical integration into fraud monitoring systems.

Additionally, the preprocessing strategies such as outlier detection and contextual feature creation (e.g., `emptied_account`, `amount_ratio`, and `is_night`) reflect behaviors that align with real-world fraud patterns, enhancing the model's relevance and effectiveness in real operations. As financial fraud continues to evolve, the methodological foundation presented here serves as a robust starting point for scalable, intelligent, and adaptive fraud detection systems.

CHAPTER 6
CONCLUSION

This study presents a comprehensive approach to detecting fraudulent transactions using the CatBoost machine learning algorithm, enhanced by strategic feature engineering and robust handling of class imbalance. By incorporating domain-specific features such as `emptied_account`, `amount_ratio`, and time-based flags, the model gains improved interpretability and predictive power, allowing it to identify complex fraud patterns more effectively.

The experimental results demonstrate that the CatBoost model performs exceptionally well across multiple evaluation metrics, particularly in detecting the minority fraud class—a critical requirement in real-world applications. The preprocessing pipeline, which includes outlier detection and winsorization, further strengthens model stability and generalization. Moreover, the model's manageable training time and minimal overfitting highlight its readiness for deployment in operational environments.

Despite some limitations, such as dependence on structured tabular data and potential challenges in adapting to unseen fraud types, the study lays a strong foundation for future enhancements. This includes exploring ensemble methods, incorporating real-time detection capabilities, and leveraging deep learning for dynamic fraud behavior modeling.

Ultimately, this research not only advances academic understanding of fraud detection methodologies but also holds tangible value for financial institutions seeking intelligent, scalable, and effective solutions to mitigate fraudulent activities in an increasingly digital landscape.

REFERENCE

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

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). Scarff: A scalable framework for streaming credit card fraud detection with Spark. *Information Fusion, 41*, 182–194. https://doi.org/10.1016/j.inffus.2017.09.005

Bahnsen, A. C., Aouada, D., Stojanovic, J., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications, 51*, 134–142. https://doi.org/10.1016/j.eswa.2015.12.030

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 31* (NeurIPS), 6638–6648.

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security, 57*, 47–66. https://doi.org/10.1016/j.cose.2015.09.005

Xenopoulos, P. (2017). Introducing DeepBalance: Random deep belief network ensembles to address class imbalance. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 3684–3689). IEEE. https://doi.org/10.1109/BigData.2017.8258343

Du, J. Z., Lu, W. G., Wu, X. H., Dong, J. Y., & Zuo, W. M. (2018). L-SVM: A radius-margin-based SVM algorithm with LogDet regularization. *Expert Systems with Applications, 102*, 113–125. https://doi.org/10.1016/j.eswa.2018.01.025

Pambudi, B. N., Hidayah, I., & Fauziati, S. (2019). Improving money laundering detection using optimized support vector machine. In *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)* (pp. 273–278). IEEE. https://doi.org/10.1109/ISRITI48646.2019.9034646

Mendelson, S., & Lerner, B. (2020). Online cluster drift detection for novelty detection in data streams. In *Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 171–178). IEEE. https://doi.org/10.1109/ICMLA51294.2020.00032

Schlör, D., Ring, M., Krause, A., & Hotho, A. (2021). Financial fraud detection with improved neural arithmetic logic units. In *Lecture Notes in Computer Science (Vol. 12591)* (pp. 40–54). Springer.

Buschjäger, S., Honysz, P. J., & Morik, K. (2021). Randomized outlier detection with trees. *International Journal of Data Science and Analytics*.

Hajek, P., Abedin, M. Z., & Sivarajah, U. (2023). Fraud detection in mobile payment systems using an XGBoost-based framework. *Information Systems Frontiers, 25*, 1985–2003. https://doi.org/10.1007/s10796-022-10346-6