Effective Credit Card Fraud Detection Using Data Mining Techniques

Ooi Jing Xian

Lee Kong Chian Faculty of Engineering and Science (LKCFES), Universiti Tunku Abdul Rahman (UTAR), Rawang, Selangor, Malaysia

DOI: https://dx.doi.org/10.51244/IJRSI.2025.1213CS008

Received: 24 October 2025; Accepted: 30 October 2025; Published: 15 November 2025

ABSTRACT

Businesses and consumers around the world face financial and security problems related to credit card fraud. Fraudulent activities are becoming more sophisticated, and therefore the need for effective and/or efficient fraud detection systems has become essential. This study focuses on how machine learning techniques can be applied to detect credit card fraud specifically, and how to overcome challenges like class imbalance, high dimensionality and complexity of real-world data sets. The IEEE-CIS Fraud Detection dataset, a publicly available and highly complex dataset, was utilized to evaluate the performance of various machine learning models. This study compares five machine learning models which are Logistic Regression, Random Forest, XGBoost, LightGBM, and Deep Neural Networks (DNN), to establish a performance baseline using the full dataset with the k-fold stratified cross validation method. Feature engineering was subsequently performed on the best-performing model (LightGBM), utilizing gain-based importance and cumulative feature importance to identify and retain the most relevant features. The reduced dataset was used to retrain the model, and its performance was evaluated against the full dataset to assess the effectiveness of the feature engineering process. An important finding is that feature engineering helped to reduce dataset dimensionality and improve model predictive performance, especially for fraudulent transaction detection. Consequently, the results showcase ensemble methods and advanced feature selection techniques as a possibility for constructing robust fraud detection systems. This research adds to the literature of machine learning applications in the area of fraud detection and it advances our understanding of how to obtain a balance between computational efficiency, interpretability, and accuracy. This study addressed to limitations of the traditional approaches and used state of the art machine learning methodologies in order to provide practical and theoretical contributions to the fight against credit card fraud and for future research and to real world implementations.

INTRODUCTION

General Introduction

Credit card fraud significantly impacts individuals, businesses, and financial institutions in today's digital economy. With an increasing prevalence of online transactions comes an increase in sophisticated fraud schemes, having a cost of doing business, and whittling away at consumer trust. Detection of these advanced fraud patterns is beyond the capabilities of the traditional rule-based detection systems, thus underlining the need for data-driven solutions.

In this study we employ machine learning techniques to determine how models, ranging from the traditional to the more advanced, can improve fraud detection accuracy. Through the study of methods that enable handling at volumes of transactions and limiting of false positives, this research attempts to make a contribution to the field of fraud prevention and building safe digital transactions.

Importance of the Study

Financial institutions need effective fraud detection methods to protect financial assets, especially in digital payment systems, where consumers' confidence plays a major role in banking systems existence. It is

particularly valuable as the topic of machine learning models that can learn with the evolving of fraud tactics with a better speed and accuracy in fraud detection. This is focused on solving the industry's need for a better fraud detection solution by focusing on model performance and feature selection.

This study provides the means of identifying critical transaction features and building guidelines for building proactive and cost-effective fraud detection systems through the insights on the optimal model. Finally, the results help financial institutions to prevent fraud, lower financial losses, and create a secure online payment environment.

Problem Statement

This rapid growth of digital payment systems has brought with it rapid increases of credit card fraud, which has become a tremendous source of financial and operational problems for individuals, businesses and financial institutions around the world. (Patel, K., 2023). Each year a great deal of money is lost due to credit card fraud which undermines consumer confidence in digital transactions, erroneously perceived as safer than cash transactions. As complex and evolving fraud patterns become more and more common, traditional fraud detection systems that are highly dependent on rule-based or statistical methods become less and less adequate. These methods are finding it hard to keep up with new fraud methods, leading to very high undetected fraudulent transaction rates and increasing false positive rates that hurt legitimate customers. (Raj and Portia, 2011).

As the fraudsters continue to refine the schemes to stay under the radar of the detection systems, the need for more accurate fraud detection approaches for finding known and new fraud patterns is very urgent. There are promising alternatives in terms of machine learning and data mining techniques, but there is also the open question of exactly which models perform best, under which conditions. This constitutes a problem for the financial sector where currently there are no adequate fraud detection solutions that achieve good accuracy, adaptability and scalability to meet the needs of financial organisations. (Ghosh and Reilly, 1994). Filling this gap is crucial to improving fraud detection systems, boosting financial security and restoring trust in digital payment methods.

Aims and Objectives

This study is primarily intended to identify the type of datasets, develop data mining techniques and feature selection techniques that can lead to improving detection accuracy and efficiency in identifying the most effective one in credit card fraud detection. Through systematic evaluation of publicly available credit card fraud datasets, different machine learning models, and critical data features, the study contribute to the evolution of robust fraud detection systems that will be able to accommodate dynamic fraud patterns in the future. To achieve this aim, the study focuses on the following objectives:

(i) To select appropriate dataset for the study by reviewing the existing datasets.
(ii) To evaluate traditional and advance machine learning models based on the selected dataset and pick the best performer.
(iii) To select appropriate features from the selected dataset and evaluate the best performer using the reduced dataset.

Scope and Limitation of the Study

We conduct an experiment to compare and evaluate some data mining techniques around credit card fraud detection and to discriminate between frauds and legitimate transactions. Once a literature review is conducted, the specific machine learning algorithms that will be used will be determined, allowing for justifying a selection based on current research and the effectiveness of particular algorithms. By taking this approach, we guarantee that only relevant, well-supported techniques will be used in the analysis, establishing a strong base upon which model evaluation is performed.

This study has been chosen to be implemented in the Python programming language because of its support in data mining and machine learning libraries. Due to Python’s libraries (such as Scikit Learn and TensorFlow), it

is easier to implement and test several machine learning algorithms, making it favourable even for those who are new to data mining. Furthermore, Python is popular for both educational research and studies of the industry for fraud detection, which makes the results in this study more accessible and repeatable. Thus, the intent for selecting our language (as opposed to R) lies in shortening the research process without shortening analytical depth.

One limitation of this study is that we are limited to using publicly available datasets, which might not include all of the real-world fraud patterns possible. To keep things tractable and understandable, the study will utilize widely studied algorithms and the most suitable feature selection methods. Since the purpose of this study is theoretical evaluation and comparison, this approach foregoes highly experimental methods or real-time testing.

Besides, due to the limitation of the computing power and resources, the hyperparameter tuning used in this research on the selected models will be default, with the optimal settings based on the recommendation and suggestion from previous relevant research papers from literature review. Fine tuning is not involved in this research due to the computing power and cost limitation.

This study aims to pragmatically contribute to a practical understanding of how different machine learning models are suited for fraud detection by defining a clearly and manageable scoped problem.

Contribution of the Study

This work enhances the field of credit card fraud detection by examining and testing various data mining techniques in an attempt to discover what techniques are most suitable to identify fraudulent transactions. This research employs a comparative analysis of model performance, examining both traditional and advanced machine learning techniques and shedding light on which of those are most successful in achieving a trade-off between accuracy, interpretability, and computational efficiency. In financial institutions where fraud detection systems continue to be improved in an environment of better fraud, this contribution is particularly relevant.

This study's second contribution is its ability in feature selection and dimensionality reduction. Distinguishing between the most relevant features of a dataset can improve model performance, significantly reducing the computational load, which could make real-time fraud detection practical. This study evaluates the performance of models generated from both full and reduced datasets to provide practical guidelines for feature selection in fraud detection and to provide a streamlined approach for future researchers and practitioners wishing to improve the performance of their models while still maintaining accuracy.

In addition, a structured methodology for evaluating a credit card fraud detection model will be proposed in this research for use as a reference by newcomers from data mining or machine learning. This thesis functions as a practical guideline that details each step for those in the beginning stages and future researchers of learning these techniques. Hence, the study is not only of technical advance to fraud detection but also of accessibility and reproducibility in this field of research.

Our hope is ultimately that this work will enable the development of more robust and adaptable fraud detection systems. This research addresses key gaps in model selection, feature importance, and practical application in order to facilitate a safer digital economy and to further bolster the financial sector’s defence against emerging fraud threats.

LITERATURE REVIEW

Introduction

With the growth of online transactions and the digital payment systems, credit card fraud has become a major global issue. Fraudulent activity does not only cause financial losses; it also erodes the trust in the electronic payment system. However, the traditional approach to fraud detection, rule-based systems, does not keep up with the continuously changing fraudulent behaviour. As a result, there has been a rising interest in machine learning-based approaches that are able to detect intricate patterns and adjust to new fraud techniques.

Identifying fraud is increasingly being recognized as a capability gap that can benefit from the power of machine learning. However, the problem of fraud detection comes with its unique set of challenges, like imbalanced datasets, high-dimensional datasets, missing data/incomplete data, etc. However, these challenges demand the design of robust methodologies that efficiently deal with these complexities but still achieve high predictive performance.

The goal of this literature review is to discuss previous studies and techniques for detecting credit card fraud, to serve as a basis on which the research methodology used in this study is based. It reviews the paper data characteristics, data pre-processing, machine learning algorithm and feature engineering, as well as evaluation metric.

This chapter analyses the current state of research and deduces gaps and opportunities that shaped the development of the method presented in Chapter 3.

Credit Card Fraud Detection Overview

As a result of the popularity of digital payment systems and the evolution of fancy fraudulent activities, credit card fraud detection is a hot research problem in the financial domain. Fraudulent transactions are often highly damaging to both businesses and the individuals they deal with, along with the financial institutions themselves. For this reason, research and practical development of reliable and efficient methods to detect and prevent fraud is a priority.

The goal of fraud detection systems is to detect anomalies or patterns in transactional data that are out of the ordinary and which signal fraudulent behaviour. Conventional approaches, including rule-based systems, require explicit indications of suspicious transactions as pre-defined rules and thresholds. Though these methods are easy to implement and are interpretable, they are often restrictive when it comes to adapting to novel and ever-changing fraud patterns. Take, for instance, an example of the rule-based systems that might fail to detect sophisticated fraud techniques that deviate from a defined pattern in a typical measure of time.

Fraud detection is one of the most interesting areas that machine learning can revolutionize, where systems learn patterns using historical data and make predictions based on emerging patterns. In the meantime, while rule-based systems cannot find these complex and nonlinear relationships in the data, machine learning models can detect these relationships with a higher accuracy. The scope of this project involves techniques through ensemble methods (such as Random Forest, XGBoost, and LightGBM) and deep learning models (such as neural networks) that have XGBoost shown much promise in tackling difficulties in fraud detection. Fraud detection, however, is a hard problem on its own, which makes the modelling process less straightforward. The problem with fraud datasets is that there is a class imbalance problem where the ratio between fraudulent versus legitimate transactions will often lead to a very small proportion for fraud. There is an imbalance in the data, where machine learning models tend to predict the majority class, thus being less sensitive to fraudulent cases. Another issue is that the data is typically very high dimensional; real-world datasets can have hundreds or thousands of features.

Some of these features may be irrelevant or redundant, resulting in the models being more computationally complex and adding noise into the models. Moreover, transactional datasets usually lack data or are incomplete and therefore must be passed through robust pre-processing techniques to ensure model performance is not compromised as well. The context for subsequent sections' methodologies and techniques is set by this overview. The shift from legacy fraud detection systems to machine learning approaches is discussed, and the challenges of the new paradigm are showcased. The challenges for practicing machine learning, along with its opportunities, contribute to the methodological decisions of this study.

Datasets for Fraud Detection

Using the appropriate datasets is important to improve credit card fraud detection models. With the appropriate datasets, researches will be able to analyse and evaluate the effectiveness of their algorithms especially when there is a lot of different types of the transaction data. Researchers have consistently highlighted the importance of utilizing authentic datasets to advance the field of credit card fraud detection.

Borah et al. (2020) and Nishi et al. (2022) utilized datasets from Kaggle and the UCI Machine Learning Repository in their studies. These datasets have gained recognition for offering standards that allow researchers to evaluate fraud detection methods in comparable situations. Borah et al. emphasize in such datasets on how these datasets can be used for developing and evaluating the machine learning models. However, Nishi et al. focus on analyses with these datasets to enhance the performance indicators of the detection techniques such as Random Forests, Artificial Neural Networks, and Logistic Regression.

The role of actual and real-world data obtained from different institutions has been described by Beigi and Amin Naseri (2020) and Koralage (2019). These real-world datasets present the complexities of fraud detection within practical scenarios. When it comes to processing both public and real-world data, Beigi and Amin Naseri pointed out that there are difficulties when there are noises and incomplete information in the data. On the other hand, Koralage also looks at the possibilities of employing data mining methods on actual datasets in order to reveal and discuss the details of fraud detection in contexts.

In addition, Kumain (2020) also look at the problem of imbalance in the real-world datasets and also, give details on how SMOTE can be used to balance these datasets to help in enhancing the efficiency of the fraud detection research. With regard to the augmentation of samples, SMOTE assists in creating a new dataset that makes the enhancing of the fraud detection model possible. However, on the other hand, Alamri and Ykhlef (2024) discussed about the challenges and limitations of SMOTE which it may inadvertently introduce noise, leading to overfitting and reduced model performance.

Therefore, there is a need to use the information from real datasets in constructing and evaluating the models for fraud detection. Real-world datasets can be used to obtain an idea about the functioning of fraud detection in real environment while public datasets can be used to compare the performance of the algorithms. As for the usage of SMOTE in handling imbalances of datasets, we will need to check if the selected dataset is suitable to implement SMOTE or not to make sure that it will really enhance the efficiency of real-life datasets in the fraud detection research. Through employing of such datasets and methodologies, one will be in a position to develop a reliable fraud detection model that can be applicable in the real world.

Data Pre-processing Techniques

The key to ensure that data is in proper shape to run in a machine learning model is about the data pre-processing. For credit card fraud detection, features such as missing values, categorical values, feature scaling are reasonably well addressed (if not overlooked) using effective pre-processing techniques to prepare the dataset for both model training and evaluation. According to the best practice found in the literature, the following pre-processing steps were done as applicable to this study.

To understand how to handle missing values is an essential step in data pre-processing, most important because most of the datasets, we use have missing entries, and their absence will hinder the model generation and performance of the model. First, there were studies using arbitrary number imputation, that is, to substitute missing data with a particular value to denote and treat missing entries explicitly. Especially useful when data is not missing at random since models are able to recognize and work with missingness as another category. According to Peng et al. (2023), this way of approach works best for such tree-based models as the Random Forest or XGBoost, because these are the models that natively deal with placeholder values without much bias. For numerical data, statistical imputation methods such as mean, median, or mode imputation are recommended. (Patel et al., 2020). However, most researchers used arbitrary number imputation as it is simple and aligns with the needs of the model chosen. In addition, this approach guarantees that we maintain an up-to-date dataset that is consistent across different types of features, leading to a simple and clean pre-processing pipeline.

Another well-known step in data pre-processing was encoding the categorical variables to numerical representations that would aid their use in machine learning models. For this purpose, one-hot encoding was used, which is the commonly adopted way of mapping categorical variables into columns without creating unintended ordinal relationships. It generates binary columns for each category of a categorical feature, and a value of 1 indicates the presence of the category and 0 otherwise. Since one-hot encoding ensured all

categories receive equal treatment and natural order/ranking is not assumed, it's recommended in most research papers. This is especially critical for algorithms like logistic regression and neural networks, where improperly encoded categorical variables can lead to biased learning. (Khaled et al., 2024). By generating a unique binary column for each category, one-hot encoding maintains the categorical nature of the data while making it compatible with numerical models.

Feature scaling was a crucial pre-processing step in this study, aimed at standardizing the dataset to ensure that all features contributed equally to the model's learning process. Without proper scaling, features with larger numeric ranges could dominate the model's optimization process, leading to skewed or suboptimal outcomes. To address this, the StandardScaler from Scikit-learn was addressed to transform all the numerical features to have a mean of zero and a standard deviation of one. To ensure uniform scaling of features, so that they maintain their relative distributions, StandardScaler was chosen as its properties qualitatively appeared most suitable for this task. (De Amorim, 2023). One advantage of this approach is that it is particularly beneficial for algorithms like logistic regression and neural networks, which are gradient-based and need to optimize over a cost function (that involves multiplication between input features and their weights) that can be sensitive to the scale of input features. Standardizing the data set makes the learning process more stable and efficient, thus making convergence and model performance better.

In cases when the test dataset was not provided, a common technique is to split the dataset into training and validation subsets to evaluate model performance on unseen data. According to Muraina (2022), stratified sampling is crucial in maintaining the distribution of fraud and non-fraud transactions across subsets, ensuring balanced representation, split ratio should be determined based on the dataset size. One of the research papers recommended 70:30 split and another research paper explored an 80:20 split for datasets which have limited number of fraud cases, which tend to have the best performance.

Overall, data pre-processing is a crucial building block to any good credit card fraud detection model, utilizing some strategy to improve data quality and your model accuracy. In each of the cases above, researchers need to systematically address challenges like missing values and will impute arbitrary numbers, convert categorical variables to one-hot encoding, and apply feature scaling using StandardScaler, which will prepare datasets such that more accurate and reliable machine learning models can subsequently be built on them. These pre-processing steps, when carefully applied, guarantee that categorical or numerical variables are properly transformed and thus obey the data integrity, thereby improving the potential of cheating transaction detection as well as improving the ability of the algorithms to learn.

Model Evaluation Techniques

Model evaluation is a key step in the machine learning process to evaluate how well a model generalizes to data that they have never seen. There are many ways to evaluate various tasks or properties of the data. This section discusses a few widely used methods of model evaluation including holdout, k fold cross validation and leave one out cross validation.

The hold-out method is one of the simplest methods to evaluate the model and is done by splitting the dataset into two subsets, namely the training set and the test set. The training set is used to train the model and evaluation is done on rest set. This approach is quick and easy and the main limitation to this method is that the model's performance can be highly sensitive to how the data is split which, in consequence, could indicate variance in results of model performance.

Since k-fold cross validation (CV) is able to overcome the shortcomings of the hold out method is one of the most widely used method by which the models are evaluated. In k-fold CV, the dataset is divided into the k-fold folds in a way so that each fold is roughly of the same size. Finally, it used the model for k−1 folds and evaluate on the one-fold not used in the training set. It repeats this process k times and each fold is used as a validation set just once when training the model on the other k-1 folds. Average over all of k iterations and such gives a more reliable estimate of the model's capacity to generalize to unseen data. One significant benefit of k fold CV is that each one of the instances inside the dataset is monotonously used for training and validation, which allows every data point to be included in the performance evaluation.

The number k (how many folds) used is dependent on the dataset and the model being evaluated. 10-fold cross validation is one of the most common options which is commonly used in all the fields like machine learning competitions and academic research. For this variant, the dataset has 10 folds, and it is used as validation set just once for a given fold. On all 9 folds, except the train fold, the model is trained, and then the evaluation score is averaged over 10 iterations. The 10-fold cross validation method suits well for the numerical efficiency while providing a reliable performance estimate. Because it is effective and relatively computationally demanding, it is widely regarded as the standard in many applications.

There is commonly k=10, but values of k such as 5 or 20 are also used based on the dataset size and the complexity of the model. If one chooses a smaller value of k (say 5), it is because that dataset is large and it is computationally expensive to train a model, in this case one would want a larger value (say 20) because even more accurate performance estimates are what is required. Stratified k-fold cross validation is generally used for classification tasks, especially when there are imbalanced classes. In this variant, the data is split such as that all fold contains approximately the same proportion of samples of each class as in the original dataset. By doing this it make sure that there is no bias in the performance evaluation due to uneven class distribution for which the result may be misleading, especially when there is an imbalanced dataset.

The next evaluation method is leave-one-out cross validation (LOOCV), that can be written as a special case of k fold CV with k equal to the number of samples in the data set. In LOOCV, every data point is a validation set once for exactly, and all other data points are used for training. However, LOOCV will generate almost unbiased estimation of model performance, but it is computationally expensive for large datasets as the model is trained and evaluated n times where n is number of samples.

Because its compromise between computational cost and estimation reliability for performance is a standard, 10-fold cross validation has been chosen. Machine learning frameworks usually define it as the default method, and it has also been proven to work well in both small and large datasets. Finally, it can offer a good generalization estimate on the model without being computationally expensive, especially with modern computational resource.

To wrap up, techniques like k-fold cross validation (for example 10-fold cross validation) are necessary to ensure that the models don't over fit the training data, rather can generalize well to the unseen data. Although 10-fold cross validation is a commonly used practice, there are alternatives such as stratified k fold or leave one out which can be used based on the task and data characteristics. The pros and cons of each technique should be considered, and the method of evaluation should be selected based on particular goals and constraints of the process by which the model is developed.

Machine Learning Models for Fraud Detection

Machine learning has become essential in credit card fraud detection due to its ability to analyse large scale transactional data to find complex patterns associated with fraudulent activity. Traditional algorithm, ensemble models and deep learning models have been developed and tested for fraud detection. According to Kalid et al. (2024), deep learning, neural networks and ensemble learning were the most common of the top performing effective techniques identified in a systematic review. In this section we explore these models, their characteristics, and how they are suited to take care of the tasks of fraud detection.

Deep Learning and Neural Networks

In recent years, deep learning models such as deep neural networks (DNNs) have delivered a lot of success in fraud detection. They are great at catching relationships in very high-dimensional datasets using many layers of interconnected neurons. It has been shown in the studies that DNNs are capable of learning complex feature representations and detecting subtle patterns associated with fraud. (Alkhatib et al., 2021). For example, after examining dropping layers or batch normalization, we note that they are necessary to tackle problems, such as avoiding overfitting, in DNNs trained on imbalanced datasets.

Deep learning architecture variants have also been applied in certain fraud detection contexts of deep learning, like Long Short-Term Memory (LSTM) networks or Convolutional Neural Networks (CNNs). LSTMs are

especially handy in sequence-based transactional data, whereas CNNs have been useful in extracting spatial features in structured datasets. While deep learning models are powerful, they come with a significant computational burden and demand access to large datasets for optimal performance, which may not be practical in certain settings.

Ensemble Learning Models

Random Forest, Gradient Boosting Machines (e.g., XGBoost and LightGBM) and bagging techniques on ensemble learning models are widely regarded as one of the best performing techniques for fraud detection. These models are quite effective for noisy and imbalanced datasets by combining multiple base learners leading to robust and higher predictive accuracy.

A bagging-based ensemble method called Random Forest construct multiple decision trees and average their prediction to avoid overfitting. It has been well praised for its interpretability and scalability in high dimensional feature space. The gradient boosting algorithms, namely XGBoost and LightGBM build decision tree in a sequential manner, each tree corrects the errors of the previous tree. Due to their ability to handle sparse data, regularization and optimization of computational cost, these models are popular in fraud detection research.

It has been shown in almost all studies that ensemble methods are better than their individual counterparts in fraud detection tasks. For example, XGBoost has been spotted having the flexibility to tune parameters to deal with class imbalance and LightGBM's leaf wise tree building strategy allows it to be more accurate and faster than training. (Taha and Malebary, 2020). Although ensemble models have these strengths, they can be computationally expensive and often require some hyperparameter tuning to perform optimally.

Traditional Models

Although traditional models, Logistic Regression and Decision Trees, are simple and effective, they have been widely studied for fraud detection. Open as a benchmark to evaluate the performance of more advanced techniques and provide a simple means of understanding the fundamental characteristics of fraud detection datasets.

Among the traditional models, Logistic Regression is one of the most used for fraud detection. A linear algorithm that evaluates a logistic function to estimate the probability of an event (say fraud) occurring. Despite its simple and interpretable model, this makes it a popular choice for researchers and practitioners as it reveals the contribution of features to prediction. If the linear relation between features and the target variable is approximately linear, then Logistic Regression provides a really good fit. (Wang and Zhao, 2022). However, its performance decays when dealing with data that presents interdependent non-linear relationships or high-dimensional complexity, as that present in fraud detection datasets. Furthermore, Logistic Regression is not efficient on imbalanced datasets and needs class weighting or oversampling to be sensitive to minority classes. Another traditional one is Decision Trees, which have been used for classifying fraud detection problems. Splitting the dataset into hierarchical structures based on feature values produces a visual and interpretable representation of decision-making. Moreover, Decision Trees are able to capture non-linear relationships and interactions between features, which makes their flexibility greater than that of logistic regression. However, Decision Trees tend to overfit, above all in high-dimensional datasets or when training on imbalanced datasets. Due to this limitation, their integration has been done into ensemble methods, such as Random Forest and Gradient Boosting Machine, which prevent overfitting via aggregation and regularization.

Traditional techniques, such as Logistic Regression and Decision Trees, are simple to understand and interpret but less performant than more complex ones (such as ensemble and deep learning). Nevertheless, these models still play a role as benchmarks and a baseline for other research on fraud detection. In particular, in several situations, Logistic Regression has significant advantages over decision trees. Its linear nature is mentioned by Wang and Zhao (2022), which makes it stable and robust in datasets with small size or when you face the risk of overfitting. Furthermore, Logistic Regression has a simple implementation and also consistently performs well, so it can be baselined for clarity in feature importance reading.

Comparative Insights from Existing Research

There are hundreds of different machine learning models for fraud detection at different levels of complexity, interpretability, and usefulness in addressing the specific difficulty that arises with having transactional data. This enables existing research on these models to offer useful insights into what these models are good at, what they are poor at, and where they should be applied.

Because of their simplicity and their ease of interpretation, traditional models like logistic regression are often used as benchmarks. Logistic regression has a clear interpretation of feature importance and works reliably in datasets where there are linear relationships and they have reasonable dimensionality. It is unable to discover non-linear correlations, which hinders its use in the advanced fraud detection tasks. More flexible in capturing non-linear relationships, decision trees are also more likely to overfit to a dataset as the dimensionality of a dataset increases, which makes them unreliable in and of themselves but more reliable when combined together using ensembling techniques.

Random Forest, XGBoost, and LightGBM are ensemble learning methods that are proven to be the best for fraud detection. The predictions of these models are derived by combining the predictions of several base learners to enhance robustness and accuracy. Random Forest's ability to work with high-dimensional data as well as interpret the results, plus XGBoost and LightGBM's performance on imbalanced data sets because they're gradient-boosting trees with regularization mechanisms, further makes them important forest models. The ensemble methods we have seen so far have thus far proven to outperform traditional models in accuracy of predictions as well as in being able to handle complex data sets.

One model attracting recent attention is Deep Neural Networks (DNNs), which are able to learn hierarchical representations of data with deep learning. It has been demonstrated in various studies that DNNs are especially good at learning complex, non-linear patterns, which makes them particularly good at spotting small fraud activities. However, they come up short on the complications of great class imbalance and because of the high computational requirements and sensitivity to overfitting, particularly in large datasets. Although deep learning methods have proven successful, their deployment in practice requires significant investment along with fine-tuning to reach a functional level.

The trade-offs between interpretability and predictive power in different models are consistent across the literature. Ensemble methods and deep learning techniques perform far better on the complexities involved in this unusual problem of fraud detection but lack the simplicity and transparency as compared to traditional models such as logistic regression. Based on the specific requirements of the task (e.g., interpretability, computational efficiency, high predictive accuracy), the model choice varies.

From the existing research, it becomes evident that the models need to be chosen depending on the properties of the dataset and the goals of the study. Appreciating the strengths and weaknesses of different machine learning methods, researchers can build methods to draw from the best-suited methods in the context of effective fraud detection.

Feature Engineering in Fraud Detection

Credit card fraud detection is a domain which requires feature engineering, where we can engineer features that machine learning models can focus on, remove noise from, spend less time and effort on to find relevant features, and ultimately improve computational efficiency. Feature engineering is important to optimize model performance on the huge, highly complex fraud detection datasets. Several techniques have been investigated in existing research that ranges from manual feature selection to automated methods with their pros and cons.

Manual Feature Selection

Manual feature selection is necessary to select possible features that will help perform the model better by using domain expertise. However, this approach is easy to interpret, but the model is overly dependent on the researcher's expertise. When you have features in the several hundred range, manual selection may indeed reveal some actual insights, but it's just not practical for high-dimensional datasets.

It has been shown by research that manual feature selection suffers from biases when features are chosen based on subjective judgments. (García et al., 2015). This method is, however, less adaptive to datasets with complex feature interactions and thus only effective in static environments such as fraud detection.

Statistical Feature Selection

In this research, statistical feature selection methods were investigated, such as correlation analysis and univariate selection, which innovate to reduce dimensionality by only retaining features with significant statistical correlation with the target variable. These techniques are computationally efficient and easy to implement, that's why they have become a popular pre-processing method for fraud detection.

Moreover, the interaction between features and non-linear relationships are commonly encountered in transactional datasets but statistical methods often don't incorporate them in the analysis. For instance, a feature with low individual importance might still contribute significantly in combination with others. (Bolón et al., 2013). Due to this limitation, model-based feature selection techniques that utilize machine learning algorithm to measure feature importance have been gaining popularity.

Model-Based Feature Selection

Model-based feature selection involves using machine learning models to rank features based on their contribution to predictive accuracy. Techniques such as gain-based importance, cumulative feature importance, recursive feature elimination (RFE), and SHAP (SHapley Additive exPlanations) values have gained prominence in fraud detection research.

Table **Error! No text of specified style in document.**.1: Model-Based Feature Selection Techniques

Techniques	**Description**
Gain-Based Importance	Gain refers to how much more accurate decision trees become by using a feature for splitting. Gain based feature importance is provided by XGBoost and LightGBM algorithms as part of their built-in functionality, and is a great method for high dimensional datasets. Gain-based importance captures non-linear relationships between features and is computationally efficient. (Shi et al., 2019). However, it only pays attention to feature contributions to decision tree splits and meanwhile neglects feature contributions that are not directly but indirectly made through interactions.
Cumulative Feature Importance	Using this approach, features are ranked by the importance scores (e.g., gain) and top features are selected which cumulatively cover a percentage say 90% of total. This approach smoothly trades off dimensionality reduction with the retention of important predictive information. (Zheng and Casari, 2018). As an example, in studies it is widely used for ensuring that the selected feature set contains most of the dataset predictive power while discarding less relevant or redundant features. Strong empirical results coupled with its intuitive implementation make it a robust technique for feature engineering in fraud detection.
Recursive Feature Elimination (RFE)	RFE recursively removes the features with the lowest importance (and retraining the model at each step to check for feature contributions). RFE, however, demands a large runtime and may not scale well for larger datasets, despite providing an optimal subset of features.
SHAP Values	SHAP values represent how much each feature makes a contribution to a model's predictions and it's a very interpretable space to select features. SHAP is effective to understand feature interactions, however, it is computationally expensive specially for a big model and dataset. (Lundberg and Lee, 2017).

Dimensionality Reduction Techniques

Dimensionality reduction techniques, Principal Component Analysis (PCA) and autoencoders, reduce the original features to a lower dimension, without losing much information. A widely used linear method for

dimensionality reduction is PCA, that projects the features on the principal components. A non-linear alternative to PCA is given by autoencoders, a sort of neural network.

Dimensionality reduction methods are effective to simplify datasets but in doing so they can reduce interpretability of the transformed features, which are some combinations of the original features. Although Zheng and Casari (2018) did mention this trade off, they make them less suitable for tasks such as fraud detection where interpretability is often critical to understanding what factors are leading to model decisions.

Applicability to Fraud Detection

Feature engineering techniques must balance interpretability, computational efficiency, and predictive performance. Robust and interpretable methods for selecting relevant features in high-dimensional datasets, gain-based importance and cumulative feature importance as employed in this study, are presented. Importance gain relies on the power of machine learning models to predict outputs and picks features that directly correspond to the problem at hand. This is complemented with cumulative feature importance that can be used to retain only the most important features and is used to reduce dataset dimensionality to improve computational efficiency.

In comparison to other model-based techniques, these methods are fast to compute and fit well with ensemble learning models like XGBoost and LightGBM, which are popular with fraud detection research. Techniques such as SHAP and RFE offer a more thorough understanding of feature interactions, but at a cost of significant computational effort, which makes them impractical for high-dimensional datasets. Although dimensionality reduction techniques reduce feature count, they can do so at the expense of interpretability and thus conflict with the needs for fraud detection tasks.

Evaluation Metrics for Fraud Detection

There is a necessity to evaluate credit card fraud detection models and their performance with the help of accurate metrics. These cases are vital in making sure the models identify the transactions correctly while at the same time, avoiding false positives and negatives. The various categories of metrics include precision, recall, accuracy, F1 score and AUC ROC (Area Under the Receiver Operating Characteristic Curve) these are useful in measuring the performance of the fraud detection systems, we shall look at the details of the various metrics as proposed by the different authors.

First, precision defines the ratio of correctly identified transactions from all transactions which are considered suspicious. There is a need to minimize false alarms because they create problems to legitimate users as noted by Abdulaziz (2021). Nishi et al. (2022) also laid emphasis on the aspect of accuracy in fraud detection systems to maintain users' trust along with the concern. High precision when developing a model of flagged transactions means that all flagged transactions are fake, and real customers are not delayed or inconvenienced in any way.

On the other hand, recall is the measures associated with the sensitivity of the model in identifying all the transactions. Recall is concerning the positive cases (fraud cases identified), among all the frauds. This metric is necessary to avoid leaving activities which may result in loss of money. Abdulaziz (2021) and Kumain (2020) also stressed the need to attain recall to enable the identification of fraud, especially in such critical application where the ability to detect fraud should be the priority over other metrics.

Concerning the F1 score, the F1 score is the average of the precision and the recall, giving a score that takes into account both the positives and the false negatives. F1 score is the mean of precision and recall so as to measure the performance of the given model. Goyal and Manjhvar (2020) and Suresh and Raj (2018) noted that the F1 score is a suitable measure to use when evaluating the performance of fraud detection models because it presents a balanced evaluation of the results. In other words, having a high F1 score means that the model has high accuracy in the identification of fraud while at the same time having the ability to capture all the activity that is existing.

In other words, accuracy in the model is about defining the exact ratio of the number of positive and true negative results to the total number of cases. While this feature is rather useful, it may lead to deception in the case of fraud detection when working with imbalanced datasets, where the number of non-fraud transactions is usually much higher than the number of fraudulent ones. Beigi & Amin Naseri (2020) and Koralage (2019) have stated that this is insufficient since an accurate model may fail to detect fraud because it could merely classify the transactions as genuine.

AUC ROC (Area Under the Receiver Operating Characteristic Curve) is the measure that evaluates the performance at thresholds to a model. It is an indirect measure of the models' capacity to distinguish nonfraudulent transactions as assessed by scores between zero and one. The above AUC ROC diagram shows that the higher the value of AUC, the better and well-trained model, and close to 1. Abdulaziz (2021) and Nishi et al. (2022) also stress the importance of AUC ROC in providing an assessment of the models' performance. It makes it easier to define the position of a threshold that optimizes the sensitivity and specificity of the performance in equal measure.

If we look into the research, we can find that assessment measures have a significant role in the analysis of the effectiveness of the fraud detection models and the same fact is also marked by Abdulaziz (2021). In the paper by Nishi et al. (2022), the authors expand on the significance of precision, recall, F1 score, and AUC ROC through elaboration of how they all encompass a thorough evaluation of the model. Goyal & Manjhvar (2020) and Suresh & Raj (2018) explained that for considering the F1 score which is a measure of Precision and Recall that balance positives and negatives. On the other side, as a result of the imbalance nature of the fraud detection datasets, Beigi and Amin Naseri (2020) as well as Koralage (2019) have pointed out that assessment methods that focus on the accuracy only should be avoided.

Therefore, for the model of the fraud detection to meet accuracy and reliability standards, performance indicators such as precision, recall, F1 score, and AUC ROC are employed together. However, there is a need to evaluate the performance of a model with a priority list. After going through the papers, we had concluded that we should focus and prioritise on the recall metric specifically on fraud detection models then only followed by other evaluation metrics.

Summary

In this chapter we gave an overview of the existing research, methodologies and techniques in credit card fraud detection that will pave the way to the methodological decisions presented in Chapter 3. The literature review first talked about the significance and challenges of fraud detection in the financial industry, and the difficulty of using the traditional rule-based systems to address this, and the rise in the use of machine learning methodologies to tackle this problem.

Fraud detection research is heavily dependent on datasets; the review examined the features, strengths, and weaknesses of publicly available datasets especially the real-world dataset. Also included was discussion of the complexities of data pre-processing, such as handling missing values, encoding, categorical features, and class imbalances, given the importance of reliable techniques in preparing data for machine learning models.

Besides, we also introduced various machine learning models used for fraud detection, including: traditional algorithms such as Logistic Regression; advanced ensemble methods such as Random Forest, XGBoost, LightGBM; and deep learning approaches such as DNNs. The strengths and limitations of each model were critically evaluated, from which the applicability of the models in solving fraud detection challenges was noticed. The capability of ensemble methods and deep learning models in dealing with high dimensional and imbalanced data was highlighted and traditional models were appreciated for their interpretability and simplicity.

Importantly, it was found that feature engineering was a critical step in maximizing model performance, and gain based importance and cumulative feature importance were shown to intelligently reduce dimensionality

while preserving predictive power. It also reviewed trade–offs between computational efficiency, interpretability, and performance as it pertained to the selection of feature engineering methods.

Drawing insight from this literature review, the methodological choices in Chapter 3 are informed by best practices and employ the most up to date techniques. This chapter critically analyzes existing research to provide the theoretical and contextual basis required for developing an effective and efficient fraud detection system.

METHODOLOGY AND WORK PLAN

Introduction

The methodology chapter describes the structured approach employed in this research to attain the stated objectives. In this chapter we give a thorough account of the processes followed, beginning with the dataset selection and pre-processing to the training and evaluation of, and further refinement to machine learning models for credit card fraud detection. As the primary data source for modelling, the IEEE-CIS Fraud Detection dataset is used which provides a complete base.

The methodology is divided into two key phases: the analysis of the full dataset and the exploration of reduced datasets through feature engineering. In phase 1, multiple machine learning models have been trained and evaluated on the entire dataset using validation performance metrics. For our further optimization, we choose the best performing model based on some evaluation metrics. After that, feature engineering techniques are applied to find the most important features and make reduced data sets. After choosing one such model, we retrain and re-evaluate it on these reduced datasets to compare model performance with and without feature reduction.

Requirement/ Specification/ Standards

This section outlines the computational resources, programming tools, and standards adopted during the research process. These specifications were chosen to efficiently handle the large dataset, perform computationally intensive tasks such as feature engineering and model training, and ensure reproducibility.

Computational Platform

This research was run with computational experiments on a Huawei Cloud Elastic Cloud Server (ECS) with GPU acceleration. Due to resource limitations and to enable scalability we chose a cloud deployment instead of a local deployment. The dataset was big and high dimensional, large enough and high dimensional that it would have been impractical to process on a local machine. (Truong and Dustdar, 2011). The flexibility to dynamically allocate enough memory and enough computation power for training and evaluation was provided by cloud resources. (Al-Janabi et al., 2014).

Since machine learning tasks are computationally intensive tasks, like feature engineering, model training and hyperparameter tuning, we opted for a GPU instance rather than a CPU only instance. Gradient boosting algorithms such as XGBoost and neural network models are optimized for parallel processing on GPUs, which cuts down the training time significantly.

As Huawei Cloud has solid infrastructure, competitive GPU providers, and easy integration with machine learning tools, we decided to deploy on Huawei Cloud. Moreover, Huawei Cloud is supported with robust data intensive workload and cost performance optimization, rendering it suitable for the research.

Operating system was Linux based; the server was running on Ubuntu 24.04. It was decided to use Ubuntu for the compatibility of machine learning libraries, frameworks and tools. As it is relatively stable, secure and supports GPU drivers and software like CUDA, it has widely adopted in data science community.

Table **Error! No text of specified style in document.**.2: System Specification for Research Experiments

Specification	**Details**
Deployment Method	Cloud Computing
Cloud Vendor	Huawei Cloud
Cloud Service	Elastic Cloud Server (ECS) with GPU acceleration
Instance Type	p2s.8xlarge.8
vCPU	32
GPU	4 * NVIDIA Tesla V100 / 4 * 32 GiB
Memory	256 GB
Storage	Extreme SSD 100GB
Operating System	Ubuntu 24.04 server 64bit
NVIDIA Driver Version	550.120
CUDA Version	12.4

Programming Tools

For this research, we carefully selected the tools for programming to be compatible with machine learning workflows, easy to use, and reproducible. It was chosen to use Python as their primary programming language due to its large ecosystem of libraries to allow data mining, machine learning, and data visualization. (Larose et al., 2019). For this study, the development environment was Jupyter Notebook, which enabled exploratory analysis through an interactive platform, iterative coding, and seamless connection with Python libraries.

Python 3.10.15 was chosen for its balance of modern features while being compatible with a large number of machine learning frameworks in common use. Moreover, Python comes with a rich set of libraries and tools that make it easy to implement sophisticated algorithms and workflows to accommodate large datasets. Though R is a very powerful tool for statistical analysis, it was used in this research because of Python's versatility and efficiency in machine learning tasks. (Ozgur et al., 2017).

Due to its interactive nature, we chose Jupyter Notebook as the development environment to pre-process data, train models, and visualize results on the fly. It also provided for the iterative refinement of models and facilitated exhaustive documentation of code cells, which was indispensable for keeping track of experimental steps and results. In fact, Jupyter was able to seamlessly connect to cloud environments to run experiments on remote resources.

Table **Error! No text of specified style in document.**.3: Programming Tools and Development Environment

Tool	**Details**
Programming Language	Python (Version 3.10.15)
Development Environment	Jupyter Notebook (Version 7.2.2)

Frameworks and Libraries

In this research, the frameworks and libraries were selected on the basis of their reliability, efficiency and suitability to work with credit card fraud detection computational and analytical requirements. They were used for data pre-processing, model training, as well as result visualization. The choice of frameworks was driven by their compatibility with Python 3.10.15 and their optimization for large-scale datasets and GPU-based computations. PyTorch is selected as it provides a more beginner-friendly experience because of its easier installation procedure, clear dataflow, and general ease of integration, making it more appropriate for learning or tasks where speed is a top priority compared to Tensorflow. (Novac et al., 2022). Moreover, newer version of Tensorflow requires AVX support which is not friendly to older devices with old CPU.

Table **Error! No text of specified style in document.**.4: Frameworks and Libraries Used in the Research

Category	**Frameworks/Libraries**
Data Pre-processing	Pandas, NumPy, Scikit-learn
Model Training	XGBoost, LightGBM, PyTorch, Scikit-learn
Visualization	Matplotlib, Seaborn

Standards

In order to promote the credibility, reproducibility and ethical integrity of this research, a set of standards was followed throughout the study. The evaluation metrics used, how to handle the data, and making experimental workflows reproducible were governed by these standards. Following these principles guaranteed that we did our research rigorously and following the best practices in machine learning and academic research.

Table **Error! No text of specified style in document.**.5: Standards for Evaluation, Data Handling, and Reproducibility

Aspect	**Standards Applied**
Evaluation Metrics	AUC-ROC, Precision, Recall, F1-score
Data Handling	Ethical use of publicly available datasets
Reproducibility	Controlled environment with defined hyperparameters

Methodology Overview

This section offers a high-level view of the methodology used in this research and presents a flowchart to represent the sequence of steps taken to fulfil the objectives. the full dataset analysis and reduced datasets evaluation after feature engineering. Here each step is briefly described and, in more detail, explained in the sections following.

The methodology starts with the selection of the dataset, which is the suitable dataset. The second part covers data pre-processing, from cleaning missing values to encoding categorical variables and scaling numerical features. The processed data is split into two subsets: training and validation for model training and evaluation.

In the first phase, we train and evaluate five different models (logistic regression, random forest, XGBoost, LightGBM, and deep neural networks) on the full data. The best-performing model is determined by using AUC-ROC and recall as validation metrics. In the second phase, the best-performing model was subject to

feature engineering to find the most important features. Thresholds of cumulative feature importance are used to create reduced datasets. Once we get reduced datasets, we retrain the model and re-evaluate its performance.

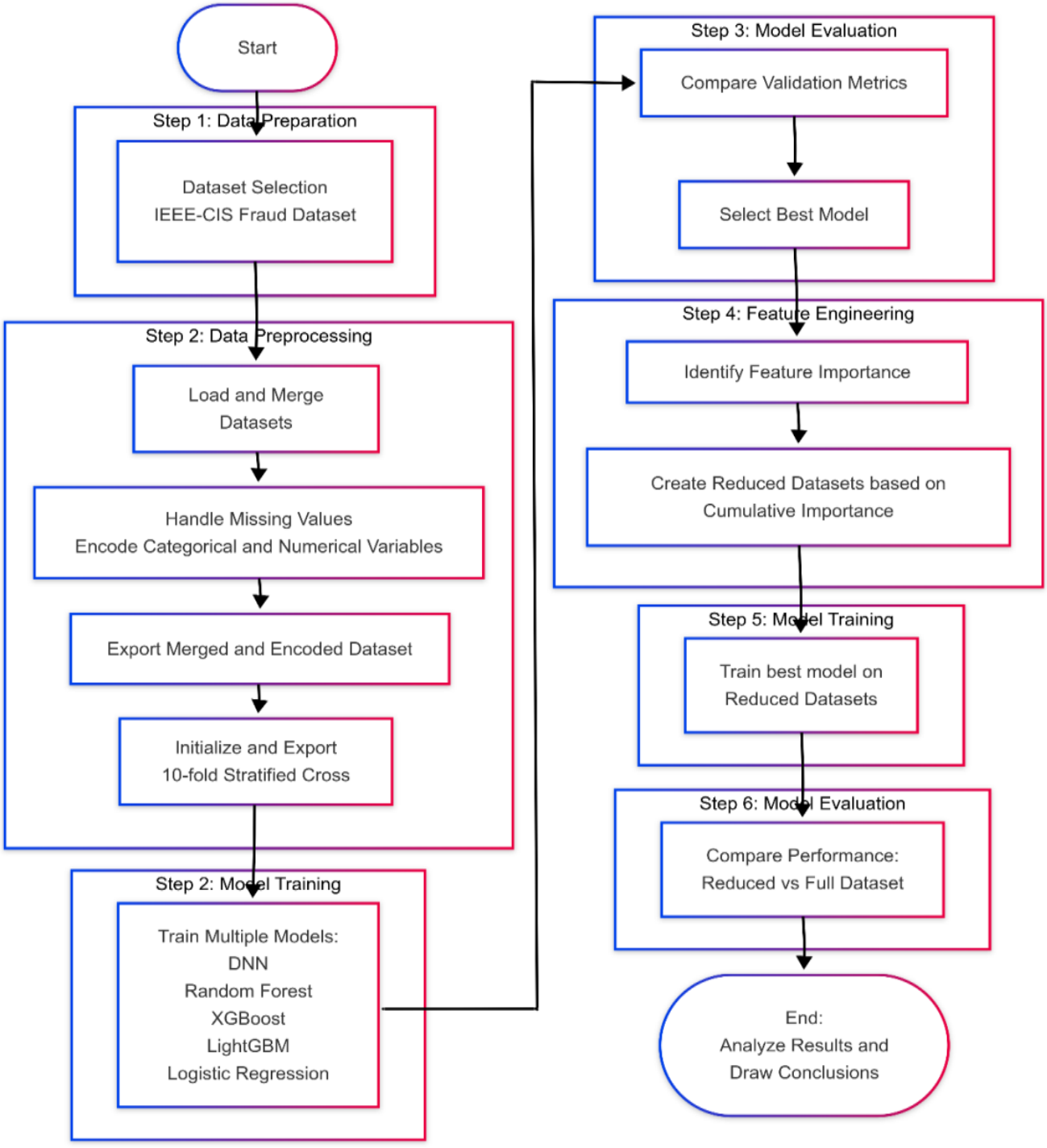

Figure **Error! No text of specified style in document.**.1: Flowchart of the Research Methodology

Dataset Selection

The first step in research is always selecting the right dataset, for it influences the reliability, relevance, and applicability of the end results. The analysed domain for this research is credit card fraud detection, and as with most machine learning tasks, the datasets are either synthetic or derived from real-world data. Several datasets were explored to ensure that the research addresses practical challenges and finding a dataset that can provide realistic fraud detection scenarios. Finally, the IEEE-CIS Fraud Detection Dataset was selected because of its rich features that are consistent with the real-world problems to satisfy the research objectives.

On the other hand, synthetic datasets, commonly used for experimental purposes, were not found to be appropriate for this research because these datasets are typically generated programmatically to create controlled fraud detection scenarios. (Bolón et al., 2013). Although this facilitates easy experimentation, the nuance and complexity of real-world data (missing values, class imbalance, etc.) are not present (Figueira and Vaz, 2022). Because these simplifications may hinder the applicability of results to real-world problems, this research looks to change this.

Extensive exploration of various real-world datasets was also done; this includes Kaggle and the UCI Machine Learning Repository. Although both platforms have a wide variety of datasets for fraud detection, many of them turned out to be unfit for our research goals. For example, the dataset, Credit Card Fraud Detection dataset from Kaggle, has about 280,000 records, which is way less than what we want to use to train models. Besides, it does not have identity-related features, which are highly useful for complete fraud detection. Similarly, datasets available from the UCI repository were often outdated or had a low diversity of features, unable to adequately represent current fraud detection challenges.

Among multiple evaluated options, the IEEE CIS Fraud Detection Dataset has been chosen as the most suitable for the purpose of this research. This dataset was released as part of a Kaggle competition and has gained considerable fame due to its current industry relevance to real fraud detection tasks. Data related to transactions and identity data. There are 394 features that capture details of transactional data for each transaction, containing details such as transaction amounts, card details, and product codes. This is complemented by identity data, which provides additional information about the users' devices and environments through 41 features, e.g., browser types and device specifications. These components together form an extensive dataset, with a total of 434 features to explore different facets of the fraud detection.

Its selection was largely due to the size and diversity of the dataset. Having a dataset that contains more than 590,540 records for records in training data, the dataset is rich enough for training complex machine learning models and testing the performance of the models. The use of 434 features makes it possible to extensively engineer features and to discover and rank the most important features for fraud detection. In addition, the dataset itself is inherently class imbalanced, with only about 3.5% of transactions being fraudulent, which is similar to real-world fraud detection problems. This demand for an imbalance challenge requires a deeper level of metrics, like AUC-ROC and Recall, to guarantee that the models find the fraudulent transactions while ignoring the overwhelming majority non-fraudulent class.

The IEEE-CIS dataset is very well suited to do this research, however, it does have some drawbacks. It’s one of the most notable issues because the test dataset doesn’t contain labels for the target variable “isFraud”. This results in the test dataset being unfit for direct evaluation of models during the research process. However, validation datasets had to be used to assess the performance of the models. Furthermore, the dataset has many features with missing values, for which we need to do careful pre-processing with these gaps while avoiding introducing bias. Beyond this, the dataset is also of high dimensionality, with 434 features, which means that dimensionality reduction techniques are needed to improve computational efficiency and prevent overfitting. The Kaggle competition utilized the IEEE-CIS Fraud Detection Dataset generously provided by Vesta Corporation. Vesta Corporation is a fraud prevention and payment solutions leader, and their contribution of this dataset has been critical in moving the research agenda for fraud detection forward. The dataset is of real-world transactions; hence the research done on this data is practical and impactful.

In conclusion, after evaluating several datasets, the IEEE-CIS Fraud Detection Dataset was chosen as the best dataset for this research. Its flexibility, diversity of features, and ability to mimic real-world fraud detection challenges create a good basis to train and evaluate machine learning models. The dataset tackles practical problems like class imbalance and missing data to ensure that the results in this work are both reliable and practical for the real world. This is a selection made with the intent to fit the dataset to meet the research objectives and to provide useful answers to the domain of fraud detection.

Data Pre-processing

Data pre-processing is a crucial step in the machine learning pipeline, as it ensures the data is clean, consistent, and ready for training and evaluation. (García et al., 2015). This section describes the pre-processing steps undertaken to prepare the IEEE-CIS Fraud Detection Dataset for model development. These steps included merging datasets, handling missing values, splitting data into training and validation sets, encoding categorical variables, and standardizing numerical features.

Data Integration

The IEEE-CIS dataset is composed of two separate files: train_transaction.csv and train_identity.csv. These datasets were merged on the common key "TransactionID" using a left join to ensure all transaction records are retained, even if identity data was unavailable. This integration created a unified dataset containing both transactional and identity features.

Handling Missing Values

There were several features with missing values in the merged dataset. As missing values can degrade model performance, all the missing values were replaced with a specific placeholder based on the type of the column. For example, the word "Missing" is used to replace missing values in a categorical column and the median value is used to replace the missing values in a numerical column. This approach is much better than a previous approach I used which is replacing all missing values with a single arbitrary number "-999" regardless of the feature. By doing so, we can ensure that the models will not run into errors processing the data and will also keep track of the missingness as a potentially useful signal in the detection of fraud without distort relationships or introducing outliers. A considerable percentage of the dataset has missing values in identity features before filling the missing values. Missing values were assessed so that they were not lacking in any class and that the imputation strategy didn't lead to any bias. (Peng et al., 2023).

Data Splitting

Here, in the revised approach, the performance of the model was tested using 10-fold CV instead of a fixed 80:20 train validation split. The method of this is we divide our dataset into ten subsets of the same size (folds). Each iteration uses a particular fold as validation set and training on other nine. We repeat this ten times, and on each fold, we use it exactly once as the validation data. Finally, these performance metrics are averaged across all folds to gain a better and more credible estimate for model generalization.

This stratified cross validation was adopted to tackle the imbalance in target variable "isFraud", i.e., roughly 3.5% of the transaction records suspected fraudulent. It guarantees that each fold has the same ratio of fraudulent and non-fraudulent instances as that of the original sample and therefore keeps the distribution of classes constant during the course of the validation. An approach that avoids the pitfall of performance bias can be resulted from such an approach in case of uneven class distribution in each fold. Its wide use in model evaluation literature to evaluate model for imbalanced classification problems which has a single random train-test split may not represent a model's generalizability well, support its use.

Additionally, as Muraina (2022) agrees, cross validation is a very handy method when the dataset size is big enough to allow multiple training without overfitting. Details of data splitting for training and validation datasets for the case where the size of the dataset is determinant of the split ratios as mentioned by Muraina (2022) are shown in Table 3.5.

Table **Error! No text of specified style in document.**.6: Data Splitting Details

Components	**Description**
Training Folds	• 90% of the data, used for model training. • Approximately 3.5% of fraudulent transaction records (stratified).

Validation Fold	• 10% of the data, used for evaluating the model's generalization ability. • Approximately 3.5% of fraudulent transaction records (stratified).
Cross Validation Setup	• 10-fold stratified cross-validation with shuffle and fixed random seed (42).
	• Each fold is used once as validation and nine times as part of training.

After the data splitting, the 10-fold indices were saved to disk for reuse to facilitate efficient model development with the naming of "10_fold_indices.pkl".

Model Training Data Preparation

Before each model was trained on a separate Jupyter Notebook, we loaded the saved datasets from the disk and proceeded with further data pre-processing steps such as target encoding, label encoding, embedding encoding and one-hot encoding. The dataset contained categorical variables that needed to be converted into a numerical format for some of the machine learning algorithms.

One-hot encoding was applied to transform categorical variables into binary indicator variables. This method is chosen as it preserved all unique categories while ensuring compatibility with numerical models and having very good performance in handling high-missing-rate problems. (Yu et al., 2020). As for the Logistic Regression and LightGBM, target encoding was used as the dataset has high-cardinality categorical features and one-hot encoding would create too many sparse features leading to high memory usage and overfitting (Zeng, G., 2023). Meanwhile, label encoding was used for XGBoost. Since XGBoost was tree-based, label encoding is much more efficient than one-hot encoding with no high-dimensional sparse matrices (Gupta, H. and Asha, V., 2020). Lastly, embedding encoding was used for deep neural network as it was often the recommended approach for high-cardinality categorical features and allowing the model to learn semantic relationships between categories (Dahouda et. al., 2021).

For the 10-fold cross validation set up, its training and validation folds had also been processed independently to avoid data leakage. In encoding, the columns of the validation folds were reindexed to match the columns of the respective training folds in each iteration. All folds have the same feature dimensions across the training and the validation fold and thus in case columns are present in the training fold but missing in the validation fold, those were filled with zeros.

Model Training on Full Dataset

In the training phase, we developed machine learning models using the pre-processed full dataset. For this research, five machine learning models were selected, each of which has a strength of handling structured data. Deep Neural Networks (DNN), Random Forest, XGBoost, LightGBM, and Logistic Regression were these models. We outline the training process, the hyperparameters used, and the considerations we made geared toward maximizing the model performance. Importing the pre-processed full dataset and followed by data preparation steps like encoding and standardization, each training was done separately on a new Jupyter Notebook.

Selected Models

To balance interpretability, scalability, and performance, we selected our machine learning models. To investigate a number of approaches for fraud detection, a range of choices of models from simple linear models to more advanced ensemble methods and deep learning architectures was chosen. This diversity enabled a thorough study of how various algorithms deal with the specific difficulties of the dataset: high dimensionality, class imbalance, and missing values.

The models selected are a mix of traditional statistical techniques, tree-based ensemble methods, and neural networks, which were the top 3 most common effective machine learning techniques. (Kalid et al., 2024). We chose each model based on its strengths in tackling certain aspects of the problem so that we could ensure that the research covers identifying which approach is the best way to detect credit card fraud. However, not solely

based on individual research papers, the selection of these five models is because literature review shows that there are many research studies for different fraud detection methods that always claim superior performance for their own approach. Such observation adds to the analysis of the context-dependent model performance and the difficulty in determining the absolute superiority of the models.

To this end, our model selection was motivated by multiple criteria, like widespread acceptance and the proven track record of these models in both industry and academic applications, their suitability to deal with imbalanced financial transaction data, and their interpretability, which is important in the financial domain where model decisions ought to be explained. (Delamaire et al., 2009). As such, this multi-faceted approach to model selection provides a pragmatic and well-rounded evaluation without being reliant on reported performance metrics from individual studies.

Table **Error! No text of specified style in document.**.7: Selected Models for Model Training

Models	**Descriptions**
Deep Neural Network	Leveraging on the flexibility of neural networks to explore if the model is able to detect fraud effectively. Although DNNs are able to capture complex patterns, it is a work of careful tuning to prevent overfitting. On recent research done in 2021, DNN has achieved the highest result compared to previous works on the IEEE-CIS Dataset. (Alkhatib et al., 2021).
Random Forest	Reducing overfitting by averaging predictions from multiple decision trees in bagging, an ensemble method. This made it suitable for the dataset as its high dimensional data and missing values could be handled. Random Forest, a better model, chooses the best feature rather than the most significant feature amongst a random subset of data. (Jonnalagadda et al., 2019).
LightGBM	A speed and efficiency optimized gradient boosting algorithm. It comes with a leaf-wise growth strategy for handling large (and high dimensional) datasets. Results from a recent study suggest that LightGBM was more performant than other machine learning algorithms and produced the best results. (Taha and Malebary, 2020).
XGBoost	Another gradient boosting algorithm, as efficient as it is scalable. An extension of logistic regression, it includes regularization terms, which make it robust against overfitting, and is suitable for handling imbalanced datasets. In many situations, XGBoost outperforms and is efficient in dealing with big dataset. (Abdulghani et al., 2021).
Logistic Regression	Due to its simplicity and interpretability, this model was chosen as a baseline. Although it supposes linear relationships between features and the target variable, its performance establishes a benchmark for comparing more complex models in one of the research, Logistic Regression becomes the one with the best performance when there is no further tuning compared to other techniques. (Wang and Zhao, 2022).

Training Process

The models we chose were trained in a manner to set a performance level for each algorithm involved in the evaluation process. This initial assessment allowed us to compare the models based on their abilities without any adjustments to hyperparameters or intricate setups. By sticking to widely accepted configurations during training, the process aimed at fairness and straightforwardness while evaluating how well each model performed under the circumstances.

In utilizing logistic regression in this scenario, Logistic Regression was trained using a manual early stopping mechanism across 5000 iterations with a patience of 50 rounds. The model was configured using max_iter=1, warm_start=True, and trained iteratively instead of with default parameters. For categorical features, Target Encoding was used, and for numerical features, standard scaling was performed. The setup lowers the possibility of fair performance evaluation in an uncontrolled manner.

For Random Forest, n_estimators=100 was used here (random forest is trained with 100 trees) and importantly max_depth=None (unlimited tree depth). For categorical features, OneHot Encoding was used, then standard scaling was applied. Metrics such as AUC-ROC, classification metrics were evaluated on performance. The input given in the setup did give us a good base to test how an ensemble method was going to perform.

The core parameters for the XGBoost and LightGBM models were configured identically to allow for comparison. We selected a learning rate of 0.05 to ensure learning does not occur too quickly while at the same time not fitting too well. The models were trained for a maximum of 5000 boosting rounds, and early stopping was triggered if validation performance (measured by AUC-ROC) did not improve over the last 50 boosting rounds. Most importantly, neither of the 2 models had the max_depth parameter explicitly defined, hence they defaulted to their pre-set values. For XGBoost, categorical variables were label-encoded and numerical features were scaled. LightGBM used target encoding for categorical features without scaling. This setting corresponded to a typical setup of gradient boosting algorithms and served as a good enough baseline to evaluate their performance on the data under realistic conditions.

A Deep Neural Network (DNN) with embedding layers for categorical variables and standardized numerical inputs was used. The architecture included four hidden layers of sizes 256, 128, 64, and 32, each followed by ReLU activations and dropout layers (rate 0.3). The output layer used a single neuron for binary classification with sigmoid activation via “BCEWithLogitsLoss”. Early stopping (patience = 50) was applied during training with a maximum of 5000 epochs. Evaluation included AUC-ROC and other classification metrics.

A standard configuration was defined across models as 5000 training iterations with early stopping at 50 rounds to stop early enough to not overfit and to spare valuable computational resources. Although each model followed the same evaluation protocol and cross-validation strategy, pre-processing pipelines were adjusted appropriately for the algorithm used. This included different encoding strategies such as One-Hot, Label, Target, and Embedding, paired with scaling where relevant. This approach made performance evaluation of each model comparable and fair under realistic conditions.

The problem of class imbalance in the IEEE-CIS dataset, with only ~3.5% of transactions labelled as fraudulent, was addressed using tailored techniques for each model. For Logistic Regression and the Deep Neural Network (DNN), class weights were set to "balanced" and the binary cross-entropy loss (BCEWithLogitsLoss) was used in DNN to give higher importance to the minority class. For XGBoost and LightGBM, the “scale_pos_weight” parameter was calculated and applied based on the training data to emphasize the fraud cases during training. The Random Forest classifier also utilized the “balanced” class weight strategy, automatically adjusting weights inversely proportional to class frequencies. These mechanisms helped mitigate the bias toward non-fraudulent transactions and improve model sensitivity to fraudulent patterns.

For efficient training, GPU acceleration on the Huawei Cloud Elastic Cloud Server (ECS) for models such as XGBoost, LightGBM, and DNN was used. Compared to the previous software, this approach resulted in a significant reduction in training time, particularly for computationally expensive data sets and high-dimensional feature tasks. Less resource-intensive algorithms, logistic regression and random forest, were run on the CPU.

This training process used standardized configurations across all models without any hyperparameter tuning, establishing a robust baseline for performance comparison. The aim was to evaluate each model under consistent and fair conditions, revealing the raw potential of each algorithm when applied to the IEEE-CIS dataset which characterized by high dimensionality and severe class imbalance. Each model leveraged appropriate pre-processing, class imbalance handling, and early stopping techniques to ensure reliable evaluation. In the subsequent section, the performance of these baseline models on the validation sets is analysed.

Model Evaluation on Full Dataset

The trained models were then evaluated on the validation set using a set of comprehensive metrics in the evaluation phase. These metrics was chosen to give insight into how well each model handled the unique

challenges of the dataset (being imbalanced and needing to be able to identify fraudulent transactions very well). Metrics such as AUC-ROC, Recall (Recall_1), Precision (Precision_1), and F1-Score (F1-Score_1) were examined in evaluating each of the models to predict their capabilities.

Evaluation Metrics

Given the imbalanced nature of the dataset, with fraudulent transactions comprising only ~3.5% of the data, specific metrics were prioritized to ensure robust evaluation:

Table **Error! No text of specified style in document.**.8: Performance Metrics used for Model Evaluation

Metrics	**Descriptions**
AUC-ROC	The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the model's ability to distinguish between the two classes (fraudulent and non-fraudulent transactions) across all classification thresholds. A higher AUC-ROC value indicates better discriminatory power. It is especially useful in evaluating performance on imbalanced datasets, as it is insensitive to class proportions.
Recall	Recall measures the proportion of true positives correctly identified out of all actual instances of a class. It is critical for understanding how well the model captures specific classes. Recall_0 represents the proportion of non-fraudulent transactions (Class 0) correctly identified as non-fraudulent while Recall_1 represents the proportion of fraudulent transactions (Class 1) correctly identified as fraudulent. For fraud detection, Recall_1 is particularly important as it indicates the model's ability to capture fraudulent transactions effectively, minimizing false negatives.
Precision	Precision measures the proportion of true positives out of all predicted positives. Precision_0 indicates the proportion of non-fraudulent predictions that are actually non-fraudulent while Precision_1 indicates the proportion of fraudulent predictions that are actually fraudulent. Precision is crucial for ensuring that the model does not generate too many false positives, which can disrupt business processes.
F1-Score	The F1-score is the harmonic mean of Precision and Recall. It provides a single measure to evaluate the balance between false positives and false negatives. F1-Score_0 reflects the balance between Precision_0 and Recall_0 while F1-Score_1 reflects the balance between Precision_1 and Recall_1. For fraud detection, F1-Score_1 is critical as it considers both the model's ability to detect fraud (Recall_1) and its reliability in labelling predictions as fraudulent (Precision_1).

For fraud detection tasks, the primary focus is on metrics related to fraudulent transactions (Class 1), such as Recall_1, Precision_1, and F1-Score_1. (Leevy et al., 2022). These metrics reflect the model's effectiveness in identifying fraud while minimizing the rate of false positives

Selection of the Best Model

By evaluating the metrics, the best-performing model will be selected. The model selection will be based on the balance between recall, precision and an AUC-ROC (Mittal and Tyagi, 2019). With this selection criteria, we will be able to distinguish the model that able to find fraudulent transactions with low rate of false positives, making it a good choice for the task of fraud detection.

Feature Engineering on Best Performing Model

Improving the performance of machine learning models on high-dimensional data, such as the IEEE-CIS dataset, is essential to apply machine learning to this problem, and feature engineering plays a key role. After running models trained on the full dataset and find out the most successful model. Feature engineering will then perform using the feature importance values yielded from the model to obtain a reduced dataset consisting of the most important features. The purpose of this process was to increase model performance by dropping

out features that are irrelevant or redundant and so as to simplify computational complexity. (Kondo et al., 2019).

Feature Importance Analysis

Feature importance scores generated by the model explain how much each feature is contributing to the model performance. The gain-based importance metric will be used for this research. Gain is a measure of how much accuracy a feature helps to improve when that feature is used as a split in the decision tree. The decision-making with the model is influenced more by the features with higher gain values. Using a gain-based importance metric is more suitable for fraud detection since the metric can capture the actual impact on model performance, which will better reflect the rare event detection capability and consider the class imbalance, which is an inherent property of fraud cases. (Shi et al., 2019). When using gain importance, it calculates the loss function before and after a split, for example, it can be the log loss (also known as binary cross-entropy loss) for classification and mean squared error for regression.

Feature importance scores will be taken and ranked based on their gain values from the model trained on the full dataset. A cumulative importance curve will be drawn for the full dataset to find what percentage of total importance was covered by the top-ranked features. This analysis provided a clear visualization of the distribution of feature importance and provided the selection of the most relevant features.

Threshold-Based Feature Selection

To create a reduced dataset, a threshold-based approach was applied to select the top features. Two thresholds will be experimented with different percentage of the cumulative importance. At the first threshold, the threshold percentage will be determined by referring to the cumulative importance curve where the curve starts to flatten. While for the second threshold, we will be using the percentage where the total number of features of first threshold was halved. These thresholds were chosen to strike a balance between retaining predictive power and reducing the dimensionality of the dataset where for high-risk domain such as fraud detection, a higher threshold was used as it has a high cost of errors. (Zheng and Casari, 2018). Features were included in the reduced dataset until their cumulative importance reached the specified threshold.

Reduced Dataset Creation

This stage involves refinement of the original dataset by keeping only those features that revealed the most useful ones by comparing cumulative feature importance using feature importance analysis. Cumulative importance plot was used to choose the feature importance that were considered to be the most important amongst all 10 folds of the cross-validation process, and then aggregated and averaged feature importance values from all 10 folds. This approach guaranteed to ensure feature selection was based on a full and representative view of model behaviour across the entire dataset than just one split.

It identified the top features and filtered the features (along with the target variable) to include only the ones mentioned above. For all samples that were being reduced, this reduction was applied consistently. Due to integrity with respect to the present cross validation work flow, the same “10_fold_indices.pkl” file was reused to sustain the same folds as used in the complete feature training phase. Such consistency enabled a sensible comparison of the full and reduced models. At this stage, no additional imputation or standardization will be performed, every step will be the same as the previous to make sure it’s standardised so that we are able to identify the impact of the feature engineering.

As categorical variables were never encoded on the fly, they were encoded using the same encoding scheme as in the model development stage and encoding was used consistently on all folds to avoid potential data leakage. It legacy of both training fairness and evaluation integrity while lowered feature dimensionality for efficiency and interpretability.

Model Training on Reduced Dataset

After performing feature engineering, the reduced datasets were used to retrain the best-performing model to evaluate the impact of dimensionality reduction on model performance. The training process for the reduced datasets followed the same standardized approach as the full dataset training, ensuring consistency in methodology and comparability of results.

Training Setup for Reduced Dataset

For the reduced datasets we filtered the original dataset to just the selected features using the first and second cumulative importance thresholds mentioned earlier. These reduced datasets were used as training dataset on these configurations (shown in Table 3.8) same as the training did on the full dataset.

Table **Error! No text of specified style in document.**.9: Training Setup for Reduced Datasets

Configuration	**Details**
Learning Rate	0.05
Number of Boosting Rounds	5000
Early Stopping Rounds	50
Objective Function	Binary Classification

Training Process on Reduced Dataset

The training process was run again with only this subset of features chosen as important during the cumulative feature importance analysis across all folds and the effect the feature reduction has from a model performance perspective evaluated. The reduced dataset has same fraud ratio and class structure as original dataset but it preserves same target variable “isFraud”.

To stay consistent in experiments, we reused the same 10-fold stratified cross-validation split already used in the training phase of the whole dataset. Both train and validation set in each fold were filtered to only keep the selected features so that fair comparison and match to the original data structure are possible.

Reduced feature set was used for training each fold on the best performing type model. A new model instance was then initialized and trained with early stopping until max of 5000 boosting rounds with patience of 50 rounds for each fold. Training was stopped early when any score from AUC ROC on the validation set did not improve after 50 consecutive iterations.

Model Evaluation on Reduced Dataset

Finally, the chosen model trained on the reduced datasets was evaluated to test the effect of feature engineering on model performance. To compare the reduced datasets with full dataset model, the two different cumulative importance thresholds were used to create reduced datasets and then evaluated using the same validation set. For the same purposes as the full dataset, the same metrics as AUC-ROC, Recall, Precision and F1-Score were used to analyse how dimensionality reduction affected the predictive power of the model.

Summary

In this chapter, a methodology to perform credit card fraud detection in its use of the IEEE-CIS dataset was explained. It started by introducing its approach as a whole and ended with a flowchart of the whole experimental pipeline. The chapter described the core parts of the methodology for which it has divided into four, these are the dataset selection, data pre-processing, complete dataset training and evaluation, feature engineering, and retraining using the reduced dataset.

All methodological choices were grounded in best practices and in order to ensure reproducibility and robustness. This allowed the research to survive by being relevant and valid in the use or a real world, publicly available dataset. Missing values were taken very carefully into account during data pre-processing; both numerical features were median imputed, and categorical ones were assigned the "Missing" category. Depending on which algorithm we are using, different techniques of encoding categorical features are used, like One-Hot Encoding, Target Encoding, or Embedding Encoding. When possible, features were selectively standardised on a numeric basis.

To apply a consistent 10-fold stratified cross validation, all five selected models LightGBM, XGBoost, Random Forest, Deep Neural Network (DNN), and Logistic Regression were trained and evaluated with 5000 maximum iterations and early stopping patience of 50 rounds. AUC-ROC, precision, recall, accuracy and F1-score were captured to be able to make a comprehensive comparison.

The five were evaluated and the best performing model will then be chosen to proceed with feature engineering. A feature importance measure based on a gain was used to aggregate across all 10 folds and construct a reduced dataset based on a feature set selected to account cumulatively for two different thresholds of the total importance. In order to prevent discrepancies from occurring among evaluation splits, the same 10-fold splits were reused for training the chosen model on the reduced dataset.

The methodology is one of a structured, fair and repeatable evaluation of models with high baseline performance, interpretability, and efficiency. Indeed, the chapter is designed such that not only are the findings credible, but the findings are also applicable in the real-world deployment of fraud detection systems.

RESULTS AND DISCUSSIONS

Introduction

In this chapter, we will describe the results of the experiments of five different machine learning and deep learning techniques applied to the IEEE-CIS dataset for the purpose of fraud detection using credit cards. The goal of this evaluation is to determine which model is the best one to be used under realistic data conditions such as high dimensionality and large class imbalance to accurately identify the fraudulent transaction.

The experiments were done by training and validating the same consistent 10-fold stratified cross validation setup based on every model, Logistic Regression, Random Forest, XGBoost, LightGBM, and Deep Neural Network (DNN). Thus, each fold contained the same amount of fraudulent and non-fraudulent cases, which means that we would make a reliable fair comparison between models. All models were also run with the same early stop criteria (no more than 5000 iterations, patience of 50 rounds) and the same pre-processed dataset that dealt with appropriately encoded missing values.

The evaluation focused on key classification metrics including Area Under the Receiver Operating Characteristic Curve (AUC-ROC), recall, and precision, which are particularly important in the context of imbalanced datasets like fraud detection. These metrics provide a more balanced view of model performance, reflecting both the model's ability to detect fraud and its accuracy in minimizing false positives. Additional metrics such as accuracy and F1-score were also recorded to support the comparative analysis.

Next, each model is presented with results in these following sections from the final evaluation metrics averages to the detailed evaluation metrics on each fold in the cross validation of each model, and a comparative discussion is made among the most optimal model and what we were able to learn from the experimental results.

Results of Model Evaluation

The performance of each model on the final averages after 10-fold were summarized as below:

Table **Error! No text of specified style in document.**.1: Model Evaluation Result on Full Dataset (Final Average)

	DNN	Random Forest	XGBoost	LightGBM	Logistic Regression
AUC_ROC	0.9450	0.9340	0.9739	0.9728	0.8602
Recall_0	0.9976	0.9993	0.9991	0.9992	0.8482
Recall_1	0.6311	0.4063	0.6670	0.6677	0.7139
Precision_0	0.9868	0.9789	0.9881	0.9881	0.9879
Precision_1	0.9054	0.9530	0.9644	0.9669	0.1457
F1-Score_0	0.9922	0.9890	0.9936	0.9936	0.9127
F1-Score_1	0.7437	0.5695	0.7885	0.7898	0.2420

Detailed results of 10-Fold Cross Validation on Each Model

The performance of each model on each fold were summarized as below:

Table **Error! No text of specified style in document.**.2: Model Evaluation Result on Each Fold (Full Dataset)

Deep Neural Network (DNN)							
Fold	**AUC ROC**	**Recall_0**	**Recall_1**	**Precision_0**	**Precision_1**	**F1-Score_0**	**F1-Score_1**
1	0.9461	0.9978	0.6288	0.9871	0.9252	0.9924	0.7487
2	0.9425	0.9978	0.6210	0.9860	0.8991	0.9919	0.7346
3	0.9473	0.972	0.6307	0.9876	0.9157	0.9924	0.7469
4	0.9415	0.9977	0.6171	0.9864	0.9101	0.9920	0.7355
5	0.9411	0.9969	0.6302	0.9864	0.8893	0.9916	0.7377
6	0.9491	0.9980	0.6433	0.9873	0.9115	0.9926	0.7543
7	0.9439	0.9975	0.6288	0.9868	0.9167	0.9921	0.7459
8	0.9434	0.9977	0.6304	0.9869	0.9241	0.9923	0.7495
9	0.9406	0.9976	0.6086	0.9860	0.8986	0.9918	0.7257
10	0.9478	0.9978	0.6323	0.9872	0.9178	0.9925	0.7488
Random Forest							
Fold	**AUC ROC**	**Recall_0**	**Recall_1**	**Precision_0**	**Precision_1**	**F1-Score_0**	**F1-Score_1**
1	0.9332	0.9992	0.3998	0.9878	0.9505	0.9889	0.5629
2	0.9329	0.9991	0.4076	0.9790	0.9439	0.9889	0.5693
3	0.9447	0.9993	0.4313	0.9798	0.9550	0.9894	0.5942
4	0.9284	0.9995	0.3906	0.9784	0.9642	0.9888	0.5560
5	0.9315	0.9994	0.4105	0.9791	0.9582	0.9891	0.5747
6	0.9301	0.9991	0.4216	0.9794	0.9457	0.9892	0.5832
7	0.9335	0.9991	0.4022	0.9788	0.9443	0.9889	0.5642

8	0.9346	0.9994	0.4156	0.9792	0.9630	0.9892	0.5806
9	0.9324	0.9994	0.3851	0.9782	0.9556	0.9886	0.5490
10	0.9390	0.9992	0.3986	0.9786	0.9493	0.9888	0.5615
XGBoost							
Fold	**AUC ROC**	**Recall_0**	**Recall_1**	**Precision_0**	**Precision_1**	**F1-Score_0**	**F1-Score_1**
1	0.9738	0.9992	0.6893	0.9889	0.9700	0.9940	0.8059
2	0.9711	0.9989	0.9467	0.9873	0.9550	0.9931	0.7711
3	0.9740	0.9990	0.6433	0.9872	0.9603	0.9931	0.7704
4	0.9697	0.9992	0.6447	0.9873	0.9666	0.9932	0.7735
5	0.9716	0.9991	0.6733	0.9883	0.9626	0.9936	0.7924
6	0.9747	0.9989	0.6873	0.9888	0.9588	0.9938	0.8007
7	0.9743	0.9991	0.6718	0.9882	0.9659	0.9937	0.7925
8	0.9771	0.9993	0.6715	0.9882	0.9720	0.9937	0.7943
9	0.9749	0.9990	0.6642	0.9880	0.9615	0.9935	0.7857
10	0.9777	0.9993	0.678	0.9884	0.9709	0.9938	0.7983
LightGBM							
Fold	**AUC ROC**	**Recall_0**	**Recall_1**	**Precision_0**	**Precision_1**	**F1-Score_0**	**F1-Score_1**
1	0.9696	0.9991	0.6704	0.9882	0.9625	0.9936	0.7912
2	0.9694	0.9989	0.6457	0.9873	0.9570	0.9931	0.7711
3	0.9753	0.9993	0.6772	0.9884	0.9729	0.9938	0.7985
4	0.9698	0.9993	0.6476	0.9874	0.9717	0.9933	0.7772
5	0.9702	0.9992	0.6762	0.9884	0.9681	0.9938	0.7962
6	0.9733	0.9990	0.6897	0.9889	0.9602	0.9939	0.8028
7	0.9756	0.9992	0.6738	0.9883	0.9687	0.9937	0.7947
8	0.9757	0.9993	0.6797	0.9885	0.9723	0.9939	0.8001
9	0.9721	0.9991	0.6454	0.9873	0.9611	0.9931	0.7722
10	0.9773	0.9993	0.6715	0.9882	0.9720	0.9937	0.7943
Logistic Regression							
Fold	**AUC ROC**	**Recall_0**	**Recall_1**	**Precision_0**	**Precision_1**	**F1-Score_0**	**F1-Score_1**
1	0.8580	0.8437	0.7197	0.9881	0.1430	0.9102	0.2386
2	0.8579	0.8476	0.7110	0.9878	0.1447	0.9124	0.2404
3	0.8710	0.8487	0.7425	0.9891	0.1511	0.9136	0.2511

4	0.8571	0.8462	0.7110	0.9878	0.1436	0.9115	0.2389
5	0.8581	0.8528	0.7173	0.9881	0.1501	0.9155	0.2483
6	0.8595	0.8471	0.7101	0.9877	0.1441	0.9120	0.2396
7	0.8613	0.8492	0.7062	0.9876	0.1452	0.9132	0.2408
8	0.8614	0.8468	0.7146	0.9879	0.1447	0.9119	0.2406
9	0.8529	0.8500	0.6962	0.9872	0.1441	0.9135	0.2387
10	0.8646	0.8498	0.7102	0.9878	0.1464	0.9136	0.2428

Selection of the Best Model

The final decision for the best performer was among the two top performing models, i.e., LightGBM and XGBoost were made after a holistic evaluation of several critical performance metrics. Although XGBoost had a slightly higher AUC-ROC score (0.9739), LightGBM performed better by all the metrics, especially recall_1, precision_1, and f1-score_1 (0.6677 vs. 0.6670; 0.9669 vs. 0.9644; 0.7898 vs. 0.7885). While the differences are marginal, they indicate that LightGBM is indeed more capable of accurately tagging fraudulent transactions (recall) while retaining high level of prediction accuracy (precision) due to the highly imbalanced classification problem of credit card fraud detection which is the main metric for fraud detection because we want to detect the maximum number of fraudulent transactions (Mittal and Tyagi, 2019).

In addition, LightGBM had a marginally higher overall accuracy than XGBoost as well. Combining the characteristics of the problem and the significance of finding fraud without too high number of false positives, LightGBM was chosen as the most appropriate model based on the balanced and practical performance given in evaluation metrics compared to other models. Random Forest, Logistic Regression and DNN gave us some useful benchmarks, but they were outperformed by gradient boosting models. These results showed that other models had potential yet needed to be further optimized to compete with LightGBM.

Cumulative Feature Importance Threshold

Cumulative feature importance of LightGBM was used as a basis to select the threshold to keep only those features in the reduced dataset. For each boosting iteration, LightGBM gives gain-based importance scores, that cap the contribution of each feature in reducing loss. The obtained scores were aggregated over all ten folds of cross-validation to make feature ranking more robust and reliable.

A descending order of the cumulative importance was plotted to examine accumulation of feature contributions over ranked list. Two of these thresholds were then selected for further experimentation and comparison. In literature as well as practice this is a commonly recommended cut-off. Cumulative importance curve reaches around 95% and starts to flatten, which indicates that subsequent features have low contributions to the performance. This is done in order to retain the features up to this given threshold to retain most of the model's predictivity power with few redundant or noisy features. In this experiment, the 95% threshold kept about 162 features of the full feature set. A second threshold was explored where the cumulative importance half the number of the first threshold (around 86% with 80 features), providing a chance to see if a smaller, yet highly informative subset could be competitive.

This attempt was motivated by the fact that when the feature set is significantly compressed, the trade-off between model simplicity and predictive accuracy will be explored. These two thresholds correspond to two strategic balances: one conserving most of the influential features, the other eliminating as much as possible as long as a considerable fraction of total importance is maintained.

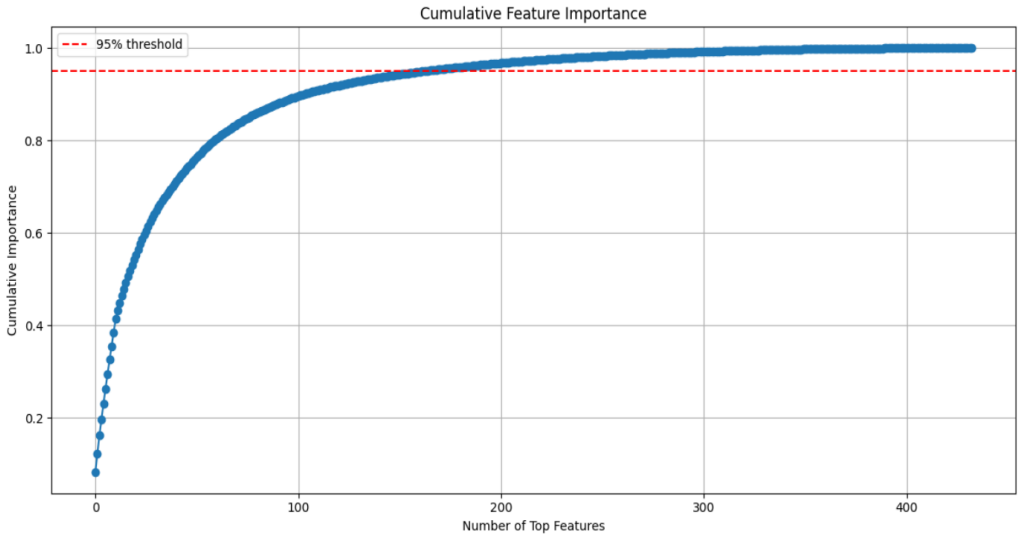

Figure **Error! No text of specified style in document.**.2: Cumulative Feature Importance Curve of LightGBM at 95%

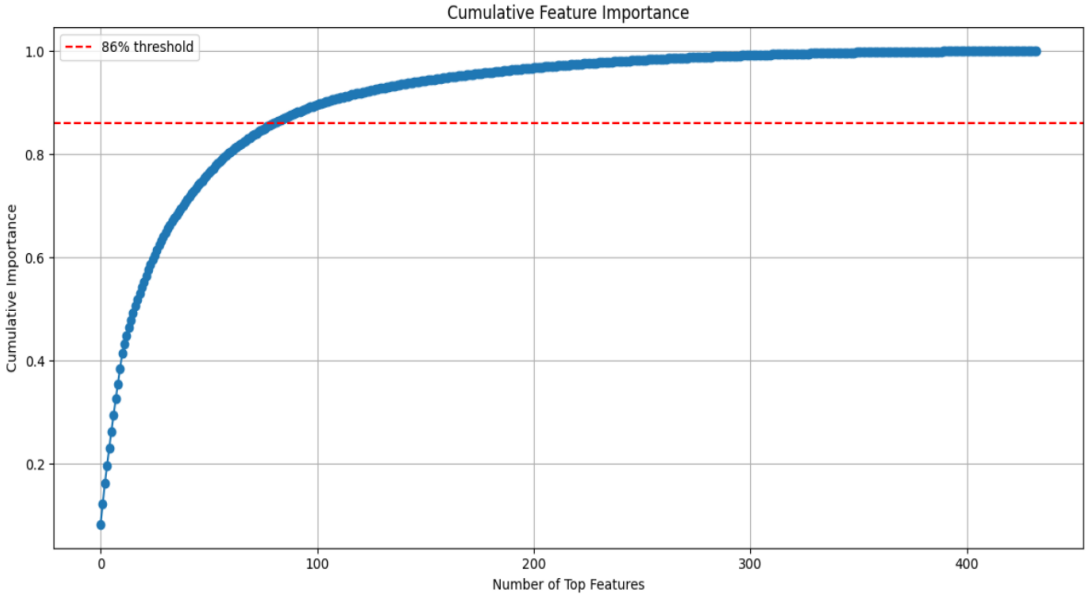

Figure **Error! No text of specified style in document.**.3: Cumulative Feature Importance Curve of LightGBM at 86%

Results and Comparison

The performance of the LightGBM model trained on the reduced datasets was compared against the model trained on the full dataset. The observation is shown in Table 4.3 for the final average values and Table 4.4 for the detailed metric on each fold of the stratified cross validation.

Table **Error! No text of specified style in document.**.3: Model Evaluation Comparison on Reduced Dataset (Average)

	Full Dataset	**95% Threshold**	**86% Threshold**
AUC_ROC	0.9728	0.9731	0.9723
Recall_0	0.9992	0.9992	0.9992
Recall_1	0.6677	0.6689	0.6690
Precision_0	0.9881	0.9881	0.9881

Precision_1	0.9669	0.9677	0.9680
F1-Score_0	0.9936	0.9936	0.9936
F1-Score_1	0.7898	0.7909	0.7910

Table **Error! No text of specified style in document.**.4: Model Evaluation Comparison on Reduced Dataset (Each Fold)

	k-Fold	**Full Dataset**	**95% Threshold**	**86% Threshold**
AUC_ROC	1	0.9696	0.9714	0.9736
	2	0.9694	0.9701	0.9678
	3	0.9753	0.9761	0.9756
	4	0.9698	0.9680	0.9687
	5	0.9702	0.9712	0.9691
	6	0.9733	0.9745	0.9736
	7	0.9756	0.9742	0.9712
	8	0.9757	0.9762	0.9750
	9	0.9721	0.9717	0.9718
	10	0.9773	0.9772	0.9762
Recall_0	1	0.9991	0.9993	0.9993
	2	0.9989	0.9990	0.9991
	3	0.9993	0.9993	0.9992
	4	0.9993	0.9992	0.9993
	5	0.9992	0.9990	0.9990
	6	0.9990	0.9991	0.9991
	7	0.9992	0.9994	0.9993
	8	0.9993	0.9993	0.9994
	9	0.9991	0.9991	0.9992
	10	0.9993	0.9992	0.9991
Recall_1	1	0.6704	0.6767	0.6936
	2	0.6457	0.6597	0.6583
	3	0.6772	0.6917	0.6820
	4	0.6476	0.6409	0.6496
	5	0.6762	0.6704	0.6675
	6	0.6897	0.7023	0.7028
	7	0.6738	0.6617	0.6442
	8	0.6797	0.6759	0.6807
	9	0.6454	0.6468	0.6410

	10	0.6715	0.6633	0.6705
Precision_0	1	0.9882	0.9884	0.9890
	2	0.9873	0.9878	0.9878
	3	0.9884	0.9889	0.9886
	4	0.9874	0.9871	0.9874
	5	0.9884	0.9882	0.9881
	6	0.9889	0.9893	0.9893
	7	0.9883	0.9879	0.9873
	8	0.9885	0.9884	0.9885
	9	0.9873	0.9873	0.9871
	10	0.9882	0.9879	0.9882
Precision_1	1	0.9652	0.9722	0.9742
	2	0.9570	0.9612	0.9645
	3	0.9729	0.9734	0.9697
	4	0.9717	0.9685	0.9704
	5	0.9681	0.9611	0.9603
	6	0.9602	0.9648	0.9661
	7	0.9687	0.9736	0.9708
	8	0.9723	0.9715	0.9744
	9	0.9611	0.9626	0.9657
	10	0.9720	0.9675	0.9638
F1-Score_0	1	0.9936	0.9983	0.9941
	2	0.9931	0.9934	0.9934
	3	0.9938	0.9941	0.9939
	4	0.9933	0.9932	0.9933
	5	0.9938	0.9936	0.9935
	6	0.9939	0.9942	0.9942
	7	0.9937	0.9936	0.9932
	8	0.9939	0.9938	0.9939
	9	0.9931	0.9932	0.9931
	10	0.9937	0.9935	0.9936
F1-Score_1	1	0.7912	0.7979	0.8103
	2	0.7711	0.7824	0.7825
	3	0.7985	0.8087	0.8008

	4	0.7772	0.7713	0.7782
	5	0.7962	0.7898	0.7875
	6	0.8028	0.8129	0.8137
	7	0.7947	0.7879	0.7745
	8	0.8001	0.7971	0.8015
	9	0.7722	0.7737	0.7706
	10	0.7943	0.7870	0.7909

Cumulative feature importance thresholds are used to select the features and this section assesses the effect of feature selection based on these thresholds. We tested two thresholds, i.e., 95% and 86% for 162 and 80 most important features respectively. They were compared against the full dataset containing 434 features to see if they would aid the performance of the model or the model's efficiency.

The model using the 95% threshold obtained a very subtle improvement on AUC-ROC to 0.9731 (as opposed to 0.9728, using the full dataset). It suggests that removing those less informative features led to more noise reduction and improved discrimination power of the model for distinguishing fraudulent and non-fraudulent transactions. Finally, the recall_1 was slightly increased from 0.6677 to 0.6689, indicating a minor increase in model sensitivity. This reduced set was effective in that the F1-score_1, balanced between precision_1 and recall_1 increased to 0.7909.

Moreover, a comparable AUC-ROC of 0.9723 was noted for the 86% threshold using only 80 features, which follows the line of full dataset. Nevertheless, it achieved the best F1-score_1 of 0.7910 and Precision_1 of 0.9680, indicating the model’s success in minimizing false positives. Across all configurations the recall was close but the compact feature set of the 86% threshold was able to perform similarly with 1 out of 5 features, which makes it computationally less expensive. Overall, both reduced datasets essentially kept Avatar comparable or even slightly improved the performance metrics. Of particular note is the 95% threshold because it balances precision and model simplicity very effectively, and the 86% threshold can still hold up precision and F1 performance with fairly aggressive dimensionality reduction.

Upon digging deeper into the performance at the fold level, there are few interesting observations to make. Both reduced datasets had higher AUC-ROC and F1 scores compared to full dataset in Fold 1. For instance, the 86% threshold achieved a recall_1 of 0.6936, the highest among all, as well as a great F1-score_1 of 0.8103. The full and the 86% reduced datasets had lower AUC-ROC (0.9209 and 0.9472 respectively) and F1-Score_1 (0.6531 and 0.7726 respectively) in Fold 3, while the 95% threshold achieved its highest AUC-ROC (0.9761) and F1-Score_1 (0.8087). All models were strong in Fold 6 but while achieving an 86% threshold, the best F1-Score_1 is that of 0.8137 and advocates for aggressive dimensionality reduction in cases of computational constraints. There were some performance trade-offs as well. As an example, recall for Fold 4 decreased slightly for both reduced datasets relative to the full dataset, indicating that in extreme reduction scenarios some important feature interaction may be lost.

The use of feature importance-based reduction in fraud detection tasks is confirmed by these findings with 95% being the preferred choice because of consistency and general performance gains.

Summary

In this chapter, IEEE-CIS credit card fraud detection dataset is used to present and discuss the result of five machine learning techniques such as LightGBM, XGBoost, Random Forest, Deep Neural Network (DNN), and Logistic Regression. In terms of evaluation metrics AUC-ROC, Recall, Precision and F1-score were worth considering to measure the performance of a model to detect the transactional fraud in such severe class imbalance.

Among all proposed models, LightGBM seems to yield the best performance in terms of AUC-ROC, recall, and precision still are very good. The gradient boosting framework holds that it is more efficient to handle large scale data and it came along with capability of superior detecting with less overfitting. All folds were performed consistently well by XGBoost, followed closely by DNN and Random Forest. Although logistic regression was the simplest, it showed reliable baseline performance and what insights can be provided from model interpretability.

After the model evaluation is done with the baseline model, the LightGBM gain based feature importance is used to feature engineer. Reduced datasets of 162 and 80 features were created based on a cumulative importance threshold of 95%, and 86% of the features respectively. The objective was to check if dimensionality reduction could lead to a reduction in computational cost while retaining or improving on predictive performance.

Experimental results showed that the 95% threshold dataset obtained slightly better AUC-ROC and F1-Score while using fewer features than the full dataset. On the negative, the smaller dataset but equivalent performance shows that the 86% threshold dataset were indeed almost many features from the original were redundant or at best, contributed minimal gain. However, upon deeper analysis at folds level, aggressive reduction may hinder stability of performance in some folds.

Also in this chapter, we can see the benefit in model selection, proper evaluation metric, and well-engineered feature when detecting fraud. The experiments confirmed that a targeted feature selection of LightGBM is robust and scalable in real world fraud detection systems.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The IEEE-CIS Fraud Detection dataset was used for the development of a credit card fraud detection system that is efficient and effective. The study was performed in two major phases. In the first phase, five selected machine learning models were evaluated in terms of their performance. The second phase was dedicated to feature engineering with the aim for better model performance at the cost of dataset dimensionality.

First, a careful dataset selection was used, whereas the IEEE-CIS dataset was chosen with its great dimensionality, real transaction behaviour patterns and being available for public research. Its size and complexity made it a great exercise to measure robustness under the real-world conditions of imbalanced data, where less than 3.5% of transactions were fraud. In the modelling part, several techniques were used to process the dataset, ready for modelling.

Imputation was used to deal with missing values by filling numerical columns with their median values and assigning them a 'Missing' label for the categorical missing values. Depending on the model used for encoding categorical feature, we encoded the features using suitable strategies such as target encoding, embedding encoding, label encoding and one hot encoding. Finally, numerical features were also standardized using StandardScaler, to make sure that training of the model is uniform. To get fair and consistent evaluation across all models, I employed a stratified 10-fold cross validation.

Early stopping (patience of 50 rounds) was used to avoid overfitting and enough learning time was granted through the training of up to 5000 iterations for each model. AUC ROC, precision, recall, F1 score and accuracy ranged to make the evaluation metrics, for each model, comprehensive. The initial phase of testing five models including Logistic Regression, Random Forest, XGBoost, LightGBM and Deep Neural Network (DNN) was tested.

Among them, LightGBM managed to be the top model by performing well in all of the key evaluation metrics. However, in some folds LightGBM provided a better balance between AUC-ROC, recall, precision and F1-score and this was the best suited tool for fraud detection, compared to XGBoost, for the particular context. In the second part, feature engineering was used on the LightGBM model.

All folds were used to aggregate feature importance scores (gain) to determine the most significant features. Two different thresholds (95% threshold, 162 features retained and 86% to threshold, 80 features retained) were tested to see cumulative importance. We also found that both of the reduced datasets performed or better than the full dataset, which had 434 features.

AUC-ROC score of 0.9731 was obtained by 95% threshold model, which meant that 95% threshold model provides more discriminatory power. Nevertheless, the 86% threshold model yielded the best recall_1 (0.6690), precision_1 (0.9680) and F1-score_1 (0.7910) on fraud detection index slightly exceeding the overall and 95% thresholds models. It brought to light that removing redundant or noisy features both decreased computational cost and sometimes improved model performance. In addition, the performance of the 86% threshold model didn't degrade much due to its large feature reduction. Contrary to this, its effect across folds established the viability of aggressive yet strategic feature selection to yield compact, interpretable, highly efficient models that are practical to deploy in resource constrained environments. Overall, this thesis shows that LightGBM is an ideal model to detect credit card fraud especially in high dimensions and imbalanced situation. Additionally, data pre-processing, encoding strategies and evaluation framework are important to illustrate the model effectiveness. In the meantime, feature engineering with the cumulative importance is a powerful way to make prediction without loss of efficiency.

Dimensionality reduction can be done strategically to improve performance whilst also reducing computational overhead. It provides both academic understanding about machine learning in fraud detection and also practical guidance on implementing scalable and reliable fraud detection systems.

Recommendations for future work

Future work to further enhance the reduced models includes exploring other cumulative importance thresholds (e.g., 90% and 93%) to determine the best tradeoff between dimensionality reduction and predictive power. More advanced feature selection techniques that include interaction effects such as recursive feature elimination or SHAP value-based feature selection could further refine the feature set. (Chen and Jeong, 2007). Furthermore, the hyperparameter optimization of the LightGBM and XGBoost model on the reduced datasets could have achieve better performance; meanwhile, ensemble methods that yield the predictions of the models trained on the full dataset and the reduced dataset, respectively, could benefit from the best of both worlds.

Overall, feature engineering was effective in lowering dataset dimensionality, which lowered computational cost while also keeping or only slightly improving model performance metrics. The results validate the feature engineering process and show its value for enhancing fraud detection systems. Yet, further refinements of the selection of features and the model could further improve the outcomes, so that the results are robust and efficient in fraud detection.

REFERENCES

1. Abdulaziz, A. H., 2021. Credit Card Fraud Detection using Data Mining Techniques: Critical Review Study. American Academic & Scholarly Research Journal, 13(2), 71-78.
2. Abdulghani, A.Q., Uçan, O.N. and Alheeti, K.M.A., 2021, December. Credit card fraud detection using XGBoost algorithm. In 2021 14th International Conference on Developments in eSystems Engineering (DeSE) (pp. 487-492). IEEE.
3. Al-Janabi, S., Patel, A., Fatlawi, H., Kalajdzic, K. and Al Shourbaji, I., 2014, November. Empirical rapid and accurate prediction model for data mining tasks in cloud computing environments. In 2014 international congress on technology, communication and knowledge (ICTCK) (pp. 1-8). IEEE.
4. Alamri, M. and Ykhlef, M., 2024. Hybrid undersampling and oversampling for handling imbalanced credit card data. IEEE Access.
5. Alkhatib, K.I., Al-Aiad, A.I., Almahmoud, M.H. and Elayan, O.N., 2021, May. Credit card fraud detection based on deep neural network approach. In 2021 12th International Conference on Information and Communication Systems (ICICS) (pp. 153-156). IEEE.
6. Beigi, S. and Amin Naseri, M.R., 2020. Credit card fraud detection using data mining and statistical methods. Journal of AI and Data Mining, 8(2), pp.149-160.
7. Bolón, C. V., Sánchez, M. N. and Alonso, B. A., 2013. A review of feature selection methods on synthetic data. Knowledge and information systems, 34, pp.483-519.
8. Chen, X.W. and Jeong, J.C., 2007, December. Enhanced recursive feature elimination. In Sixth international conference on machine learning and applications (ICMLA 2007) (pp. 429-435). IEEE.
9. Dahouda, M.K. and Joe, I., 2021. A deep-learned embedding technique for categorical features encoding. IEEE Access, 9, pp.114381-114391.
10. De Amorim, L.B., Cavalcanti, G.D. and Cruz, R.M., 2023. The choice of scaling technique matters for classification performance. Applied Soft Computing, 133, p.109924.
11. Delamaire, L., Abdou, H.A.H. and Pointon, J., 2009. Credit card fraud and detection techniques: a review. Banks and Bank systems, 4(2).
12. Figueira, A. and Vaz, B., 2022. Survey on synthetic data generation, evaluation methods and GANs. Mathematics, 10(15), p.2733.
13. García, S., Luengo, J. and Herrera, F., 2015. Data Preprocessing in Data Mining (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.
14. Ghosh, S. and Reilly, D.L., 1994, January. Credit card fraud detection with a neural-network. In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on (Vol. 3, pp. 621-630). IEEE.
15. Goyal, R., Manjhvar, A. K., 2020. Review on Credit Card Fraud Detection using Data Mining Classification Techniques & Machine Learning Algorithms. International Journal of Research and Analytical Reviews (IJRAR), 7(1), 972-975.
16. Gupta, H. and Asha, V., 2020. Impact of encoding of high cardinality categorical data to solve prediction problems. Journal of Computational and Theoretical Nanoscience, 17(9-10), pp.4197-4201.

17. Jonnalagadda, V., Gupta, P. and Sen, E., 2019. Credit card fraud detection using Random Forest Algorithm. International Journal of Advance Research, Ideas and Innovations in Technology, 5(2), pp.1-5.
18. Kalid, S.N., Khor, K.C., Ng, K.H. and Tong, G.K., 2024. Detecting frauds and payment defaults on credit card data inherited with imbalanced class distribution and overlapping class problems: A systematic review. IEEE Access.
19. Khaled, S., Rohayanti, H., Zeba, T., Manal O., Md, O., Choi, K. (2024). When to Use Standardization and Normalization: Empirical Evidence from Machine Learning Models and XAI. IEEE Access, vol. 12, pp. 135300-135314. doi: 10.1109/ACCESS.2024.3462434.
20. Kondo, M., Bezemer, C.P., Kamei, Y., Hassan, A.E. and Mizuno, O., 2019. The impact of feature reduction techniques on defect prediction models. Empirical Software Engineering, 24, pp.1925-1963.
21. Koralage, R., 2019. Data Mining Techniques for Credit Card Fraud Detection. Sustain. Vital Technol. Eng. Informatics, (2015), pp.1-9.
22. Kumain, K. (2020). Analysis of Fraud Detection on Credit Cards using Data Mining Techniques. Turkish Journal of Computer and Mathematics Education, 11(1), 235-245.
23. Larose, C.D. and Larose, D.T., 2019. Data science using Python and R. John Wiley & Sons.
24. Leevy, J.L., Khoshgoftaar, T.M. and Hancock, J., 2022, October. Evaluating performance metrics for credit card fraud classification. In 2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI) (pp. 1336-1341). IEEE.
25. Lundberg, S.M., and Lee, S.-I., 2017. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30 (NIPS 2017), pp. 4765–4774.
26. Ozgur, C., Colliau, T., Rogers, G. and Hughes, Z., 2017. MatLab vs. Python vs. R. Journal of data Science, 15(3), pp.355-371.
27. Patel, K., 2023. Credit card analytics: a review of fraud detection and risk assessment techniques. International Journal of Computer Trends and Technology, 71(10), pp.69-79.
28. Peng, J., Hahn, J., Huang, K. (2023). Handling Missing Values in Information Systems Research: A Review of Methods and Assumptions. Information Systems Research, 34(1), pp. 5-26. doi: 10.1287/isre.2022.1104
29. Mittal, S. and Tyagi, S., 2019, January. Performance evaluation of machine learning algorithms for credit card fraud detection. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 320-324). IEEE.
30. Muraina, I., 2022. Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts in 7th International Mardin Artuklu Scientific Research Conference, 2022. Mardin, Turkey.
31. Nishi, N.J., Sunny, F.A. and Bakchy, S.C., 2022, December. Fraud Detection of Credit Card using Data Mining Techniques. In 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.
32. Novac, O.C., Chirodea, M.C., Novac, C.M., Bizon, N., Oproescu, M., Stan, O.P. and Gordan, C.E., 2022. Analysis of the application efficiency of TensorFlow and PyTorch in convolutional neural network. Sensors, 22(22), p.8872.
33. Raj, S.B.E. and Portia, A.A., 2011. Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (pp. 152-156). IEEE.
34. Shi, X., Wong, Y.D., Li, M.Z.F., Palanisamy, C. and Chai, C., 2019. A feature learning approach based on XGBoost for driving assessment and risk prediction. Accident Analysis & Prevention, 129, pp.170-179.
35. Suresh, G., Raj, R. J., 2018. A Study on Credit Card Fraud Detection using Data Mining Techniques. International Journal of Data Mining Techniques and Applications, 7(1), 21-24.
36. Taha, A.A. and Malebary, S.J., 2020. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, 8, pp.25579-25587.
37. Truong, H.L. and Dustdar, S., 2011. Cloud computing for small research groups in computational science and engineering: current status and outlook. Computing, 91, pp.75-91.

38. Wang, T. and Zhao, Y., 2022, January. Credit Card Fraud Detection using Logistic Regression. In 2022 International Conference on Big Data, Information and Computer Network (BDICN) (pp. 301-305). IEEE.
39. Yu, L., Zhou, R., Chen, R., and Lai, K. K. (2020). Missing data preprocessing in credit classification: One-hot encoding or imputation. Emerging Markets Finance and Trade, 58(2), pp. 472–482. doi:10.1080/1540496x.2020.1825935.
40. Zeng, G., 2023. On the analytical properties of category encodings in logistic regression. Communications in Statistics-Theory and Methods, 52(6), pp.1870-1887.
41. Zheng, A. and Casari, A., 2018. Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media,

Competing Interests

The author(s) declare(s) that they have no competing interests.