Credit Card Fraud Detection Using Novelty Detection Techniques

Author: Abdullah Barayan

Supervisor: Dr Yuhua Li

Moderator: Prof. Alun Preece

A dissertation submitted in partial fulfilment of the requirements for the degree of:

MSc Advanced Computer Science

October 2022

School of Computer Science and Informatics

Cardiff University

Acknowledgements

First and foremost, I would like to thank and praise **Allah** for providing me with strength and encouragement during the difficult moments of completing my dissertation. I am extremely grateful for His endless love, mercy, and grace.

I would like to extend a special thanks to my supportive supervisor, **Dr Yuhua Li**, for providing guidance and feedback during this dissertation. I have no doubt that his unlimited support throughout all phases of this dissertation contributed to its success.

I want to acknowledge my country, the **Kingdom of Saudi Arabia**, and its **Cultural Bureau** in the United Kingdom for sponsoring me and providing me with the opportunity to pursue my master's degree at Cardiff University.

From the bottom of my heart, I would like to express my profound gratitude to my family, especially my **parents**, **spouse**, and **daughter**. Their belief in me has kept my confidence and motivation high during this process.

.

Abstract

Credit card fraud is a growing issue affecting the financial industry and cardholders worldwide, resulting in billions of dollars in losses annually. A considerable amount of research has been done to detect credit card fraud. Most proposed machine and deep learning fraud detection approaches use a supervised learning methodology that requires appropriately labelled and balanced training datasets. Organising these datasets takes a great deal of time and effort. It is difficult for these approaches to deal with imbalanced or unlabeled datasets. However, few studies have employed unsupervised novelty detection methods, focusing on addressing the imbalance and lack of unlabeled data issues that supervised methods suffer from them. On the other hand, high data dimensionality has not received much attention, mainly when using unsupervised techniques, which are challenging, unlike supervised approaches. This study proposes unsupervised novelty detection techniques with the help of dimensionality reduction methods to detect credit card fraud. The European cardholder dataset has been used to evaluate this approach. In this study, One-Class SVM, Isolation Forest, and Autoencoder novelty detection techniques have been utilised with the help of Autoencoder to reduce dimensionality for the detection of credit card frauds, and the performance of these techniques are compared mainly based on the AUC and FNR. The experimental results show that using unsupervised novelty detection techniques and Autoencoder as a feature reduction method has a promising potential to detect credit card fraud. The results also demonstrate that the proposed approach deals effectively with imbalanced and unlabeled datasets and can reduce training and prediction time. AE-AE achieved the highest AUC score and the lowest FNR, with 93% and 10.169%, respectively.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AVS	**Address Verification Systems**
AE	**Autoencoder**
CNP	**Card-Not-Present**
CVM	**Card Verification Method**
DT	**Decision Trees**
FTC	**Federal Trade Commission**
IF	**Isolation Forest**
LOF	**Local Outlier Factor**
LR	**Logistic Regression**
KNN	**K-Nearest Neighbors**
MSE	**Mean Squared Error**
OCSVM	**One-Class Support Vector Machine**
PIN	**Personal Identification Number**
PCA	**Principal Component Analysis**
RF	**Random Forest**
SVM	**Support Vector Machines**
VCC	**Virtual Credit Card Numbers**
UMAP	**Uniform Manifold Approximation and Projection**

1. Introduction

1.1. Research Motivation

The growth of modern technology and the reliance on cyberspace have created an ideal environment for fraud, giving rise to significant increases in fraudulent activities, costing billions of dollars annually [1]. Fraudulent activities have arisen in many areas. Today, bank fraud, specifically credit card fraud, is a pretty common fraud that has received much research attention [2].

In today's economies, credit cards have become the most prevalent means of payment [3]. A credit card offers more than allowing a customer to buy on credit. It includes many benefits for its holder, such as insurance coverage and reward points, and is accepted almost everywhere. In the fourth quarter of 2021, the number of new credit card users increased by 60 % in comparison to 2020 [4]. Credit card popularity and rapid spread, especially with e-commerce overgrown, have created a perfect environment for fraud. In other words, online-commerce growth results in more online purchases. Consequently, fraudsters are now more likely to obtain credit card information via the internet. Credit card fraud has significant consequences for individuals, card providers, corporations, and the government, including financial losses of billions of dollars and reputational damage [1], [5]. In 2018, credit card fraud cost the world $24.26 billion [6].

Despite the extensive research and growth in fraud detection technologies, current figures show that credit card fraud detection remains a significant challenge for the financial industry. In 2018, credit card fraud increased by 18.4% and continues to rise, ranking as the number one form of identity theft fraud [6]. Additionally, according to Federal Trade Commission (FTC), the credit card was the most common payment form identified in all fraud reports in the U.S, with a total of 459,297 reported in 2020 [7]. For these reasons, as well as the severe consequences and losses that strongly impact individuals, businesses, and governments, there is a growing need to detect credit card fraud. The main motive of this study is to eliminate credit card fraud, designing a fraud detector that can detect successful fraudulent transactions from legitimate ones as they occur.

1.2. Research Statement

The design of credit card fraud detection systems is challenging for several reasons, such as lack of labelled data, continuous evolution of fraud strategies and high dimensionality of the data [2]. Several machine learning and deep learning methods [8]–[13] have recently been proposed to detect credit card fraud. Most proposed methods are based on a supervised learning approach, which requires well-labelled and balanced training datasets [14]. However, this is against the nature of real-life transaction data, where frauds are a small share of everyday transactions. Also, collecting, organising and grouping these datasets requires considerable time and effort. Moreover, supervised methods detect known fraud patterns that they trained on and could be ineffective when dealing with fraud events that are new to the system and misclassified them. In contrast to supervised learning, unsupervised novelty detection uses available unlabeled normal data to train a model to detect novel patterns in the future and can address the imbalanced classification and unlabeled data issues [11], [15].

However, we also noticed that most of the studies focused on addressing the class imbalance problem, and less attention was on the issue of the high dimensionality of the data. Mainly when using unsupervised novelty detection. Unlike supervised learning, feature reduction is challenging when using a one-class learning approach [16]. This study will address the following research questions:

- How accurate are novelty detection techniques for building a credit card fraud detection model?

- How efficient is an unsupervised novelty detection approach for addressing imbalanced classification issues?

- How efficient is Autoencoder as a dimensionality reduction method when it is fitted with only one class in reducing computational time and enhancing model performance?

1.3. Research Aim and Objectives

Aim:

Design and develop a data-driven credit card fraud detection model that can differentiate between legitimate and fraudulent transactions using novelty detection techniques.

Objectives:

- Choose the appropriate well-known datasets for credit card fraud detection, then Preprocess and analyse the chosen datasets.

- Reduce the data dimensions by implementing different Autoencoder architectures and choosing the best one to represent the data.

- Find the optimal hyperparameters and architecture for the proposed novelty detection algorithms to detect fraud.

- Compare the effectiveness of Autoencoder as a feature reduction method in reducing computational time and enhancing model performance.

- Compare the results of the three proposed algorithms, and select the algorithm that primarily achieves the best scores in terms of AUC (i.e. the ability of a classifier to distinguish between classes) and lower false negative rate at the same time.

1.4. Target Audience

This study's target audience is people interested in or planning to research the financial security field based on novelty detection techniques, especially in transactional credit card fraud detection.

1.5. Outline

This dissertation consists of eight chapters as follows:

Chapter 1 - Introduction: This chapter presents the study's motivation, statement, aim and objectives, and details of its intended audience.

Chapter 2 - Background and Literature Review: This chapter provides the background knowledge needed to comprehend the subsequent chapters. Further, it shows the existing literature on credit card fraud detection.

Chapter 3 - Methodology: This chapter presents the adopted study methodology for credit card fraud detection.

Chapter 4 - Implementation: This chapter describes the implementation of our models, starting with describing the data and then the preprocessing and training phases.

Chapter 5 - Results and Evaluation: This chapter discusses the results of our experimental implementations. Furthermore, it compares these findings to previous study results.

Chapter 6 - Limitations and Future work: This chapter presents the possible limitations and future work suggestions.

Chapter 7 - Conclusion: This chapter concludes and summarises the study's findings.

Chapter8 - Reflection on Learning: This chapter reflects on the lessons learned from conducting this study.

2. Background and Literature Review

This chapter presents a background of credit card fraud, followed by a literature review. It begins by providing a general knowledge of fraud and credit card fraud and an overview of the measures taken against credit card fraud and the issues and challenges facing the development of credit card fraud detection systems. Following that, a brief explanation of novelty detection and dimensionality reduction methods. Finally, an overview of previous research related to this study is provided.

2.1. Fraud

Throughout history, fraud has taken many forms. There are several definitions of fraud and fraudulent behaviour. The Concise Oxford Dictionary defines *fraud* as "criminal deception, use of false representation to gain unjust advantage". However, Along with today's technology and telecommunication growth, fraudulent activity has grown, resulting in significant losses [1]. Multiple research has been and continues to be undertaken to prevent and detect fraud. However, fraud patterns and techniques are continuously evolving. According to the FTC estimates, total fraud losses in the United States in 2021 were $5.8 billion, up more than 70% from 2020 [7]. Fraudulent activities have occurred in many areas. According to [2], the most common areas of fraud are bank, insurance, telecommunication and internet marketing fraud, see Figure 1. However, bank fraud, particularly credit card fraud, is the most common and received much research attention [2].

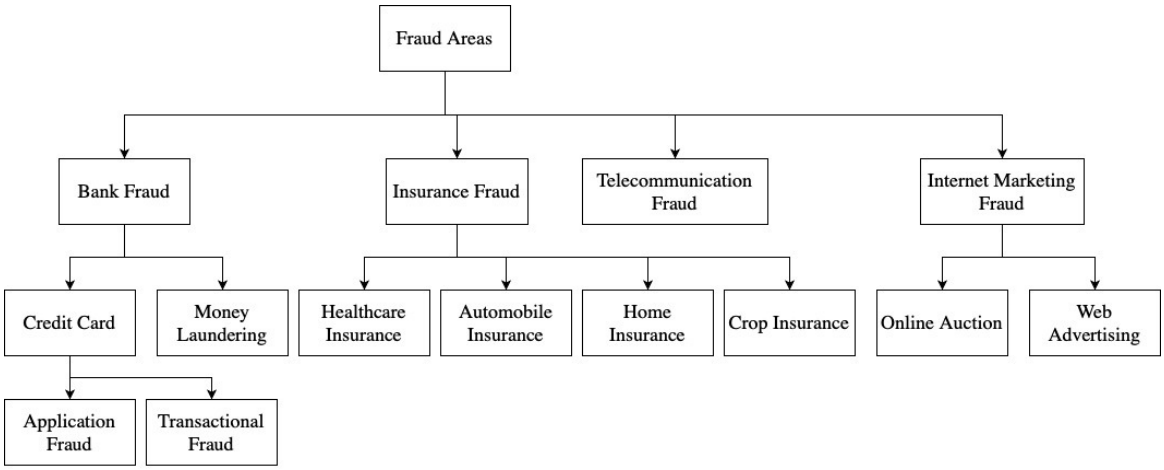

Figure 1 Taxonomy of The Most Common Areas of Fraud [2].

2.2. Credit Card Fraud

A *credit card* is a plastic or metal card used to pay for products and services on credit [2]. Credit cards nowadays have become the most prevalent payment method [3]. The credit card does not only allow an individual to buy on credit. It includes many advantages for its holder, such as insurance coverage, bonus points, discounts, and cashback, as well as being accepted almost everywhere. According to [4], the number of credit card accounts reached 196 million in the fourth quarter of 2021; 20.1 million new accounts were opened, representing over 60% more than in 2020.

Moreover, with the evolution of e-business and the increased use of the internet for online shopping over the last few years, the usage of credit cards has grown significantly [3]. Global credit card transactions increased by around 6% between 2019 and 2020 [17]. As the number of credit card users grows throughout the world, the number of fraudulent activities have been constantly increased [3], [18]. *Credit card fraud* is a term that refers to the unauthorised use of credit card information to access products and services or to obtain money [18]. This fraud impacts individuals, credit card providers, businesses, and governments [5], [18]. In 2018, credit card fraud cost the world **$24.26** billion [6].

2.3. Credit Card Fraud Types

Credit card fraud might take several forms, and they are continuously evolving. [2], [11], [19] have discussed different fraud types. Some common credit card fraud types have been explained below:

- ***Application Fraud***: In this type of fraud, fraudsters submit a new credit card application based on incorrect or stolen personal information.

- ***Card-Not-Present Fraud (CNP)***: This fraud occurs without using a physical card, mostly through e-commerce, email or over the phone, where just credit card information is needed. Most Card-Not-Present frauds acquire card information fraudulently via data breaches or card owners, frequently through phishing.

- ***Lost/Stolen Card Fraud***: This type refers to the unauthorised usage of lost or stolen credit cards without their owner's knowledge. Fraudsters usually attempt to spend as much money as they can before the card gets frozen.

- ***Intercepting Mailed Cards:*** This type of fraud on a card requested by a customer but never delivered. The credit card is intercepted and activated by the fraudster during delivery. For instance, a card is taken from the victim's mailbox and used by the fraudster.

- ***Counterfeit Fraud***: This type occurs when information is fraudulently obtained to construct a fake card containing actual card details. These details are acquired by skimming the owner's credit card without their knowledge. Chip-and-PIN technology has reduced this fraud.

2.4. Impacts of Credit Card Fraud

Credit card fraud has significant consequences, including financial losses and reputational damage for businesses and card issuers. Indeed, the impact of fraud is difficult to determine as corporations and banks do not prefer to declare the number of losses caused by fraud. Additionally, we can only evaluate the loss of frauds that have been discovered; we cannot estimate the size of not reported or not detected frauds [13]. However, according to the Nilson report [20], global card fraud losses were $28.85 billion in 2020. Only in the United States, losses totalled $10.24 billion in 2020, up from $9.62 billion in 2019. According to the UK Finance report [21], unauthorised financial fraud losses were £824.8 million over payment cards, remote banking, and cheques in 2019. Card payments represented 48 % of all frauds. UK Finance has estimated that the total fraud losses through card payments issued in the United Kingdom were £620.6 million in 2019. In particular, 76% of these frauds came from card-not-present (CNP) payments, 15% from lost and stolen cards, 6% from card identity theft, 2% from counterfeit cards and 1% from cards not received. The UK Finance report also shows that the introduction of Chip and PIN standards reduced Counterfeit card fraud losses to £12.8 million in 2019 compared to £169.8 million in 2008.

2.5. Measures Against Credit Card Fraud

Credit card fraud is a challenging problem and dramatically impacts the economy and people. A considerable effort has been placed against credit card fraud. Measures taken against credit card fraud can be divided into two types: *fraud prevention* and *fraud detection* [22]. *Fraud prevention* is the first defence against fraud, aimed at restricting fraudulent activities from taking place at all. [23], [24] discussed some technologies used to prevent fraud which are explained below.

- ***Address Verification Systems (AVS)*:** This is a technique used to verify whether a billing address matches the address of a credit card holder. AVS is used widely for CNP fraud prevention. By matching the billing address details, it can be determined whether the customer is the same as the cardholder. However, for international transactions, AVS is not that practical.

- ***Card Verification Method (CVM)*:** A three- or four-digit security number on a card front or back is not included in the magnetic stripe. During a CNP transaction, these verification codes are used to verify that the individual making a payment has their card with them, which helps to protect payment cards against CNP fraud.

- ***Personal Identification Number (PIN)*:** A security number authorising financial transactions. The difference between the PIN and CVM codes is that the PIN is generated by the cardholder, and the CVM is generated by the card provider.

- ***Virtual Credit Card Numbers (VCC):*** These are uniquely generated credit card information that allows the processing of CNP transactions and secures actual credit card account information.

However, fraudsters frequently develop methods for evading prevention techniques; thus, when fraud cannot be prevented, it should be detected as it starts. *Fraud detection* aims to identify and report successful fraudulent activities before they are complete.

2.6. Credit Card Fraud Detection

Credit card fraud detection involves determining whether an incoming transaction is legitimate or not. This identification is derived from recorded transaction data. Typically, transaction data includes attributes such as transaction date, customer address, and transaction amount [2]. The large scale and dimensionality of the transaction data make it difficult for human investigators to examine them for abnormalities in real time. Automated fraud detection systems are utilised to counteract this situation [25]. Fraud detection systems can be developed based on *Expert-driven* or *Data-driven* approaches. However, typical fraud detection systems are developed based on a parallel combination of both approaches to benefit both techniques' advantages [25]–[27]. The *Expert-driven* approach identifies whether the incoming transaction is normal or fraudulent based on rules made by the domain investigators. On the other hand, *Data-driven* approaches are based on data mining tools such as machine learning algorithms, mainly novelty detection methods [2]. They learn to differentiate between fraudulent and legitimate transactions based on the recorded data and then identify the incoming transactions as either normal or fraudulent. This dissertation will focus on automated data-driven approaches based on novelty detection techniques.

2.7. Credit Card Fraud Detection Issues and Challenges

The development of fraud detection systems is challenging for various reasons [2]. One key reason is the strong class imbalance in the data, in which fraudulent transactions are less represented than legitimate ones [14]. Most machine learning algorithms underperform with imbalanced class distributions [28]. Another challenge is the large amount of data and its high dimensionality, which complicates data mining and detection, slowing the detection process [2]. Additionally, fraud activities evolve continuously, so the fraud detection system must be adequate adaptability enough to detect new, unknown cases [2], [15]. Finally, the problem of real-time detection, selecting the appropriate methods to cope with limited resources and provide real-time detection increases the ability to process many transactions [2].

2.8. Novelty Detection

Novelty detection refers to identifying novel or abnormal behaviours in data that do not conform to typical behaviour. It is also known as anomaly and outlier detection [29], [30]. Novelty detection has drawn significant scientific research interest across different application domains. These include faults and failure detection in complex industrial systems [31], network intrusion detection [32], credit card fraud detection [33] and medical diagnostic issues [34], [35]. A reviews about novelty detection techniques is described in [29], [30], [36]. They discuss several novelty technique categories, such as classification-based, Nearest neighbour-based, clustering-based, and statistical methods. Moreover, novelty detection methods can be divided into three categories according to the availability of labels and the learning approach.

2.8.1. Supervised Learning

The supervised learning approach assumes a training dataset with labelled samples for normal and abnormal events is available [36]. The labelled data is used to train the model to differentiate between normal and abnormal classes. Then unseen data points are compared against the model to identify their category. In supervised novelty detection, there are two significant challenges [29]. First, the distribution of datasets used in novelty detection is typically highly imbalanced, where abnormal events are less represented than normal ones. Most supervised learning algorithms cannot deal with the significant imbalance in the dataset, which negatively impacts classifier performance [37]. Different methods are developed and implemented to obtain relatively balanced data [37], [38]. However, selecting the appropriate technique is essential, as some sampling techniques could cause overfitting, slow the learning process, or reduce algorithm performance due to the absence of potentially valuable data [38], [39]. Second, collecting, organising and grouping well-labelled and balanced training datasets requires considerable time and effort.

2.8.2. Semi-Supervised Learning

A semi-supervised learning novelty detection technique assumes the availability of unlabeled data and a small portion of labelled data, including normal and abnormal classes, during the training [30].

2.8.3. Unsupervised Learning

The unsupervised learning approach does not rely on the availability of tagged data. It needs only available normal data. It is the most practised approach since obtaining sufficient data from normal events is usually possible, but that is seldom the case with abnormal events [36]. This dissertation will use unsupervised novelty detection as the training process needs normal data samples only. The employed novelty detection methods are explained in section 2.10.

2.9. Dimensionality Reduction

Data dimensionality reduction involves reducing the number of features in a dataset. Reducing the data dimensionality decreases the computational cost and could enhance the model's performance. Dimensionality reduction approaches could be typed into *feature selection* and *feature extraction* [40]. *Feature selection* is the process of choosing relevant variables from a dataset's actual variables. In contrast, *Feature extraction* transforms the features into a lower dimension by extracting new features from the actual variables. Some common feature selection and extraction methods are listed below. Autoencoder was employed as a feature reduction method in this dissertation and explained in section 2.10.3.

A. Feature Selection methods:
   - Filter methods
   - Wrapper methods
   - Embedded methods

B. Feature Extraction methods:
   - Autoencoders (AE)
   - Principal Component Analysis (PCA)
   - Uniform Manifold Approximation and Projection (UMAP)

2.10. Proposed Novelty Detection and Feature Reduction Methods

2.10.1.One-Class Support Vector Machine (OCSVM)

The OCSVM is one of the most commonly used unsupervised techniques for novelty detection. The OCSVM require no target labels for training. OCSVM learns the border for the normal samples and determines the data points beyond the boundary as outliers [41]. In our implementation, we employed A Radial Basic Function (RBF) kernel, defined as follows [42].

$$\mathrm{K}(X1, X2) = \exp\left(-\frac{||X1 - X2||^2}{2\sigma^2}\right)$$

Moreover, in this study, we have searched various *nu* values. *nu* is a crucial parameter of OCSVM that allows error acceptance. A larger nu allows for a larger error on the normal class but probably causes a smaller error on the abnormal class and vice versa. Generally, it is necessary to maintain a balance between normal and novel class errors, depending on the application domain.

2.10.2.Isolation Forest (IF)

The isolation forest technique is one of the most popular unsupervised techniques for novelty detection. Unlike other novelty detection methods, it identifies novel data points by isolating outliers in the data rather than profiling normal samples and determining distances between points [43]. This technique identifies abnormalities based on the principle that they are few and different. An ensemble of isolation trees is used in an IF to isolate anomalies in data points. Initially, it selects a feature from a set of features and then randomly selects a split value between the max and min values for that feature. As a result of this random partitioning of features, abnormal data points will have shorter paths in trees, thus different from the rest [44]. However, after an IF model is trained, decision-making requires an outlier score, calculated as follows [43]:

$$S(x, n) = 2^{-\frac{\mathrm{E}(h(x))}{c(x)}}$$

In this equation, *h(x)* is the path length of *x*, *c(n)* is the average path length of an unsuccessful search in a binary search tree, and *n* is the number of dataset instances. $S(x, n)$ is a score ranging from 0 to 1, and high scores near 1 are more likely to be outliers [43].

2.10.3. Autoencoders (AE)

Autoencoder is an unsupervised artificial neural network that attempts to compress input data into a lower dimensional representation and then decodes the low-dimensional representation to reconstruct the data to its original form [41]. Autoencoder consists of three parts: encoder, code and decoder. The encoder compresses the input data and forms the code, which is then used by the decoder to reconstruct the input. Autoencoder learns to compress the data while minimising the reconstruction error; in this study, Mean Squared Error (MSE) was used as a loss function. Figure 2 shows the general structure of an AE. Autoencoder can be used for different applications; In this study, it will be used for novelty detection and dimensionality reduction.

- **Dimensionality Reduction**

When AE is used as a features reduction method, the low-dimensional representation (code) is extracted and used as input to the detector algorithm.

- **Novelty Detection**

When AE is used as a novelty detection method, it is trained only on normal data points and learned to reconstruct them well. Thus, when an abnormal data point that different from normal samples, the AE will have difficulty reconstructing it, and the reconstruction error will be high. In this case, the error that exceeds a specific threshold will be considered abnormal.

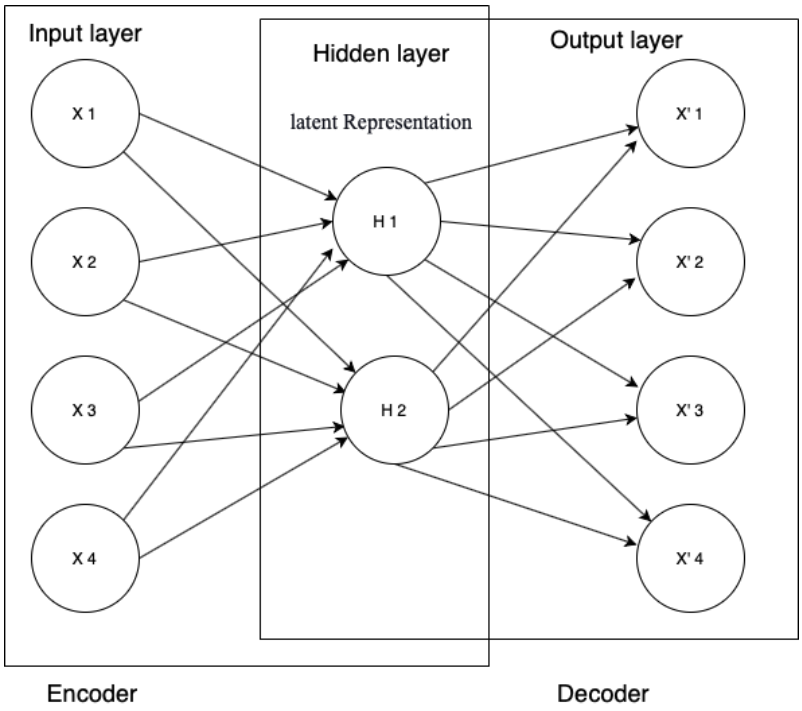

Figure 2 Autoencoder General Structure.

2.11. Literature Review

Over the past few years, credit card fraud has become an increasingly severe issue worldwide. Considerable research on credit card fraud detection has been conducted to reduce losses. The researchers employed various novelty detection methods. A brief review of the most recent related work is outlined below. Different methods are classified into two main categories based on their learning mode: supervised and unsupervised learning approaches.

2.11.1.Supervised Learning

For the implementation of supervised learning, several algorithms and resampling methods have been applied to detect fraudulent credit card transactions [9], [10], [45]–[49]. [9] have applied under-sampling to handle the highly skewed dataset and researched various supervised learning techniques for credit cards fraud detection, such as Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), Ensemble Learning, K-Nearest Neighbors (KNN) and Decision Trees (DT). The results show that all the proposed classifiers performed well overall. Moreover, a comparative study was conducted on different supervised learning algorithms and sampling methods on a dataset containing credit card transactions [10]. Five machine learning techniques were applied to the balanced dataset with different sampling techniques, including Oversampling, Under-sampling, Both sampling, Random Oversampling Examples (ROSE) and Synthetic Minority Oversampling Technique (SMOTE). The results revealed that SMOTE sampling and logistic regression performed well, with an AUC of 97.04 % and a precision of 99.99 %. In contrast, other classifiers performed well in terms of AUC and Accuracy but not in terms of precision. In addition, [46] used five supervised machine learning and feature selection methods to detect application fraud. This paper uses the German credit dataset to measure the efficiency of these machine learning methods and the filter and wrapper features selection methods. According to the study results, using filter and wrapper methods improved the prediction accuracy of J48 and PART.

2.11.2.Unsupervised Learning

For the implementation of unsupervised learning, several machine learning and deep learning approaches have been applied, such as OCSVM [50], IF [50] and AE [51]. [50], proposed the grid search approach to estimate the hyperparameters of an OCSVM for fraud detection. The datasets used in this investigation include German and European credit card datasets. The results show that the GS-OCSVM identifies fraud more than the isolation forest in the German credit card dataset; the True negative rate (TNR) of GS-OCSVM and isolation forest was similar in the European cardholders dataset. In [52], a 10-layer deep Variational Autoencoder (VAE) was used and compared to DT, SVM, and AdaBoost. The results show that the AdaBoost outperformed the other models in precision. In contrast, the recall for VAE was the highest. [11] proposed Sparse Autoencoder (SAE) and Generative Adversarial Network (GAN) to detect fraud transactions. SAE was used to perform feature extraction by obtaining the representations of genuine transactions. Then the GAN was trained using the latent representation. Next, the SAE and the GAN discriminator were used to determine if a transaction was legitimate or fraudulent. The results show that the SAE-GAN performance was better than using GAN alone. In addition, the authors have compared the proposed model with other state-of-art used One-Class GP and SVDD. The SAE-GAN perform better in terms of precision and F1 score.

The main conclusion drawn from previous studies is that machine learning and deep learning have shown promising results for credit card fraud detection. It is noticeable that most proposed methods are based on a supervised learning approach, which requires appropriately labelled and balanced training datasets [14]. However, this is against the nature of real-life credit card transaction data, where fraud accounts for a small proportion of daily transactions. Additionally, collecting and labelling these datasets requires considerable time and effort. Although different resampling techniques are proposed to handle the imbalance in the data, these methods could lead to overfitting or removal of relevant data points from the dataset. Furthermore, supervised methods may not be effective when detecting new fraud patterns and misclassifying them since they detect known ones [15].

Unlike supervised learning, unsupervised novelty detection relies on available unlabeled normal data to train a model to detect novel patterns in the future. So can address the imbalanced classification and unlabeled data issues [11]. However, it also noticed that most of

the studies focused on addressing the class imbalance problem, and less attention was on the issue of the high dimensionality of the data. Mainly when using unsupervised novelty detection. In contrast to supervised learning, feature reduction is challenging when employing a one-class learning approach [16]. This study's main contribution is using Autoencoder as a feature reduction method in an unsupervised credit card fraud detection approach, addressing the imbalance classification problem and the high dimensionality in the data. To the best of our knowledge, this is the first time it has been used for credit card fraud detection.

3. Methodology

This chapter discusses the methodology adopted in this study for credit card fraud detection. We first introduce the overall credit card fraud detection pipeline and then provide an overview of each pipeline component. Finally, we explain all the metrics used to evaluate the performance of our proposed approach.

3.1. Overall Methodology

This study formulated credit card fraud detection as an unsupervised novelty detection problem, where the novelty detector builds on the available unlabeled normal transaction to detect abnormal events in the future. The proposed credit card fraud detection model operates at the transactional fraud level. It aims to differentiate between fraudulent and legitimate transactions, thus detecting successful fraudulent transactions as they start.

Figure 3 shows the flow of the adopted methodology. In our proposed approach, the dataset with legitimate and fraudulent transactions was first collected, then pre-processed and split into training and testing sets. The training set contains only legitimate transactions, and the testing set includes normal and fraudulent transactions. The legitimate transactions are then trained on a dimension reduction algorithm to extract the low-dimensional representation. The low-dimensional representation is then trained on a novelty detection model to detect credit card fraud. Finally, the trained models will be tested and evaluated based on the testing set. The purpose of choosing this approach was to address three major challenges discussed in Chapter 2: the class imbalance problem, unlabeled data, and the high dimensionality of the data.

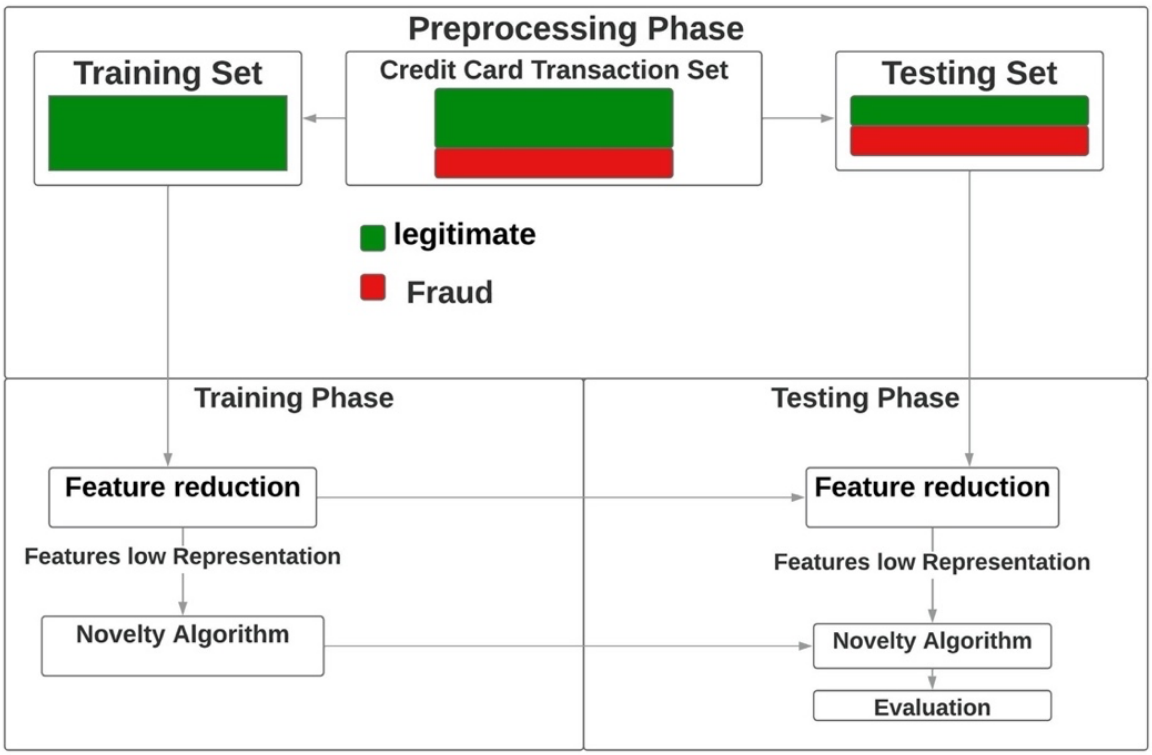

Figure 3 The Adopted Methodology For Credit Card Fraud Detection Model Development.

3.2. Dataset Collection

As a basis for training and evaluating our transactional fraud detection model, we needed both legitimate and fraudulent credit card transactions. For the dataset, we aimed to use not simulated transaction data, reflecting real-world transaction characteristics. However, finding real-world transaction datasets is challenging due to the sensitivity of the data. Consequently, we examined previous studies to obtain knowledge of the datasets used to make the proper selection decision. As a result, three datasets were found [53],[54] and [55]. [53] it is a simulated credit card transactions dataset where transactions are classified into fraud and non-fraud. [54] German credit data is a dataset for credit applications classified customers into good and bad credit. [55] it is an actual European cardholders' transactions dataset where transactions are classified as fraudulent and legitimate.

The dataset selected for this study is the European cardholders' transactions dataset, as it is the relatively recent available, not simulated dataset for transactional fraud. Additionally, the fraud to legitimate transactions ratio and the data characteristics reflect reality. The dataset was originally collected for research collaboration between Wordline and the Machine Learning

Group of ULB (Université Libre de Bruxelles) on big-data mining and fraud detection [56]. The dataset contains no confidential client information disclosure, so it could be shared and utilised to develop fraud detection methods. A public dataset release is available for download from Kaggle [55]. The dataset is described in detail in Chapter 4.

3.3. Preprocessing Phase

A dataset's quality directly affects model performance. Preprocessing is, therefore, essential in the detector development. In this phase, the dataset is cleaned to eliminate the noise in the data. As part of the implementation of the model, the Pre-processing phase was used to remove duplicate samples and missing or null values. Then, splitting the dataset into training and testing sets, where the training set contains just legitimate transactions, and the testing set includes both legitimate and fraudulent transactions. Panda library [57] was used to check and remove missing or null values and duplicated rows. The dataset was split into testing and training sets using the Scikit-learn library [58]. Additionally, the dataset variables' values were scaled due to the difference in the distribution of the variables. The scaling was done using the Standard and Min-Max scalers using the Scikit-learn library. The mathematical equation of the Min-Max scaler is given below [58].

$$Xscaled = \frac{x - min(x)}{max(x) - min(x)}$$

The mathematical equation of the Standard scaler is given below, where u is the mean and s is the standard deviation of the training data points [58].

$$Xscaled = \frac{x - u}{s}$$

3.4. Training Phase

3.4.1. Feature Reduction

The high dimensionality of the data is one of the challenges in developing and designing credit card fraud detection system. Reducing features helps compress data and reduce storage space and computation time. Also, it could enhance the model prediction performance. In this study, Autoencoder was employed as a dimensionality reduction technique. The primary idea behind Autoencoder is to compress the data to maintain most of the relevant information so that decoder can decode the low representation to its original shape. This compressed version can help the models characterise the data better. In addition, it has proven effective in reducing computation time and improving performance when trained only on one class in other fields [59], [60]. In this study, different AE architectures were constructed, and the architecture with the best reconstruction accuracy was chosen for dimensionality reduction and other architectures were employed as novelty detection models.

3.4.2. Novelty Detection Models

Once the hidden low representation of legitimate transactions is obtained, it is used as input to train the novelty detection models. Our study uses the most common unsupervised novelty detection methods presented in the credit card fraud detection literature, including OCSVM, IF and AE. The algorithms were implemented using Python's Scikit-learn [58] and Keras [61] libraries since these libraries offer a wide range of functionalities and high-level frameworks.

3.5. Testing Phase

After the training phase, the trained AE, OCSVM and IF were obtained. They are tested and evaluated using different metrics based on the testing set. The evaluation aims to determine our model's likelihood of detecting unseen data and confirm its suitability for the intended use. Firstly, the original test set representation is fed into the trained AE encoder, and the low-dimensional representation is extracted. Then, novelty detection models take the compressed representations as input to determine whether they are fraudulent or legitimate transactions and evaluate the model's performance. A more detailed explanation of the evaluation metrics is provided in the next section 3.6.

3.6. Evaluation Metrics

The main goal of a credit card fraud detector is to differentiate between fraud and legitimate transactions, detecting successful fraud transactions as they begin. We used the common evaluation metrics in the credit card fraud detection literature to evaluate our models. We evaluate the results in terms of the confusion matrix and its related metrics. Additionally, the ROC curve (receiver operating characteristics curve) and AUC (area under the ROC curve) will be used for evaluation purposes.

3.6.1. Confusion Matrix

A *confusion matrix* is a method used to summarise a model's performance, providing a better understanding of actual and predicted classifications and errors made by the model [62]. Table 1 illustrates the general confusion matrix structure for binary classification. Fraudulent transaction (Positive class) was labelled -1, while legitimate transaction (Negative class) was labelled a 1. The terminologies for the confusion matrix values are explained as follows:

Table 1 General Confusion Matrix Structure

Class	Predicted (-1)	Predicted (1)
Actual (-1)	TP	FN
Actual (1)	FP	TN

- **True Positive (TP):** The number of fraudulent transactions that are correctly classified as fraudulent.
- **True Negative (TN):** The number of legitimate transactions that are correctly classified as legitimate.
- **False Positive (FP):** The number of legitimate transactions that are incorrectly classified as fraudulent.
- **False Negative (FN):** The number of fraudulent transactions that are incorrectly classified as legitimate.

The calculated confusion matrix values are also used to calculate several performance metrics. These metrics are explained below [30], [62].

1. **False Negative Rate (FNR):** The rate of fraudulent events is considered as normal. It is measured as:

$$FNR = \frac{FN}{(TP + FN)}$$

2. **False Positive Rate (FPR):** The rate of normal events considered as fraudulent. It is measured as:

$$FPR = \frac{FP}{(TN + FP)}$$

3. **Precision:** Shows the ability of the detector to not label a sample that is legitimate as fraudulent. It is measured as:

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

4. **Recall:** Shows the ability of the detector to detect fraudulent events. It is measured as

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

5. **F-measure:** Is a weighted harmonic mean of the precision and recall. It is measured as

$$F - measure = 2\frac{Precision \times Recall}{Precision + Recall} \times 100$$

3.6.2. Receiving Operating Characteristic (ROC) curve

The Receiving Operating Characteristic (ROC) curve is a graph presenting the performance of a classification model over a variety of thresholds. It is created by plotting the TPR versus the FPR, where each point on the curve represents a particular classification threshold [63]. ROC curve is insensitive to the class distribution, making it proper when dealing with imbalanced class distribution [30], [63].

3.6.3. AUC (area under the ROC curve)

AUC is the measure of the ability of a classifier to distinguish between classes and is used as a numerical summary of the ROC curve [30]. The higher the AUC, the better the performance of the detector at distinguishing between the genuine and fraudulent classes.

The prime focus of fraud detection is to increase the detection of actual fraud events and decrease misclassified fraud events; this is assuming that misclassified fraud (false negative) could cause far more losses than false alarms (false positive), both false alarms and misclassified fraud are expensive and could cost the same [14]. Where misclassified fraud could result in financial loss, false alarms could restrict client financial access or financial loss. Thus, determining a cost-based indicator in credit card fraud detection is a complicated issue [14]. This study determined that the most important thing is that the model makes correct predictions, considering reducing misclassified fraud. Therefore, we used the computed AUC value (i.e. the ability of a classifier to distinguish between classes) as the primary evaluation metric and the FNR (i.e. the rate of misclassified fraud transactions) as the second metric when comparing different algorithms' results in the experiments of this study.

4. Implementation

This chapter discusses the implementation of our fraud detection model. We first introduce the environmental setup, followed by the dataset description. The following sections discuss preprocessing the dataset, feature reduction, and novelty detection models.

4.1. Environmental Setup

The proposed models were implemented in Python 3 using the Scikit-learn [58] and Keras modules [61]. Experiments were carried out in the Google Colaboratory environment with 12 GB of RAM and 106 GB of disk space. Using Google's environment gives free access to GPUs and requires only a few setups. The laptop used in the research was equipped with a 2.7 GHz Dual-Core Intel Core i5 CPU and 8 GB RAM and ran mac OS Catalina.

4.2. Dataset Description

The dataset used in this study comes from credit card transactions made by European cardholders over two days in September 2013. The dataset was originally collected for research collaboration between Wordline and the Machine Learning Group of ULB (Université Libre de Bruxelles) on big-data mining and fraud detection. There is a total of 284,807 transactions, 492 of which are fraudulent. The dataset is significantly imbalanced, with fraudulent transactions representing 0.17 % of all samples. Figure 4 shows the highly skewed distribution of the data towards the genuine class.

Additionally, we used UMAP to reduce the dimension to two-component and visualise the data in Figure 5, which shows 3000 normal samples and 492 fraudulent samples. We also plot the distribution of the features in Figure 6. Figures 5 and 6 show that the distributions of non-fraud and fraud samples are not just imbalanced; they also overlap. Further, feature distribution is similar in features such as V6, V13, V20 and V25, making distinguishing between the two classes challenging. Table 2 describes all features in the dataset. Features V1 to V28 are numerical values obtained from Principal Component Analysis (PCA) for the original features. This transformation hides the original features to keep the information confidential. The only features not transformed by PCA are time and amount.

Table 2 The Definition Of Features In The Dataset.

Features	Type	Description
Time	Float	In seconds, the time elapsed between each transaction and the first transaction in the dataset.
Amount	Float	The transaction amount.
V1..V28	Float	The transformed features with PCA.
Class	Int	Response variable 0 = Legitimate, 1 = Fraud.

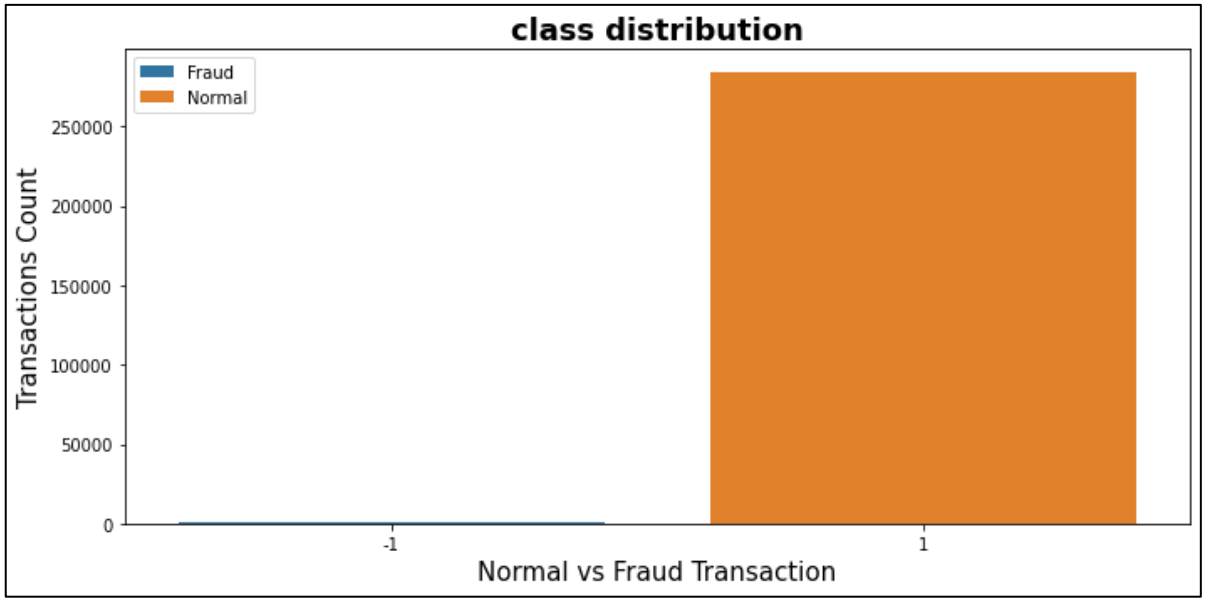

Figure 4 The Class Distribution Of Transactions Made By European Cardholders.

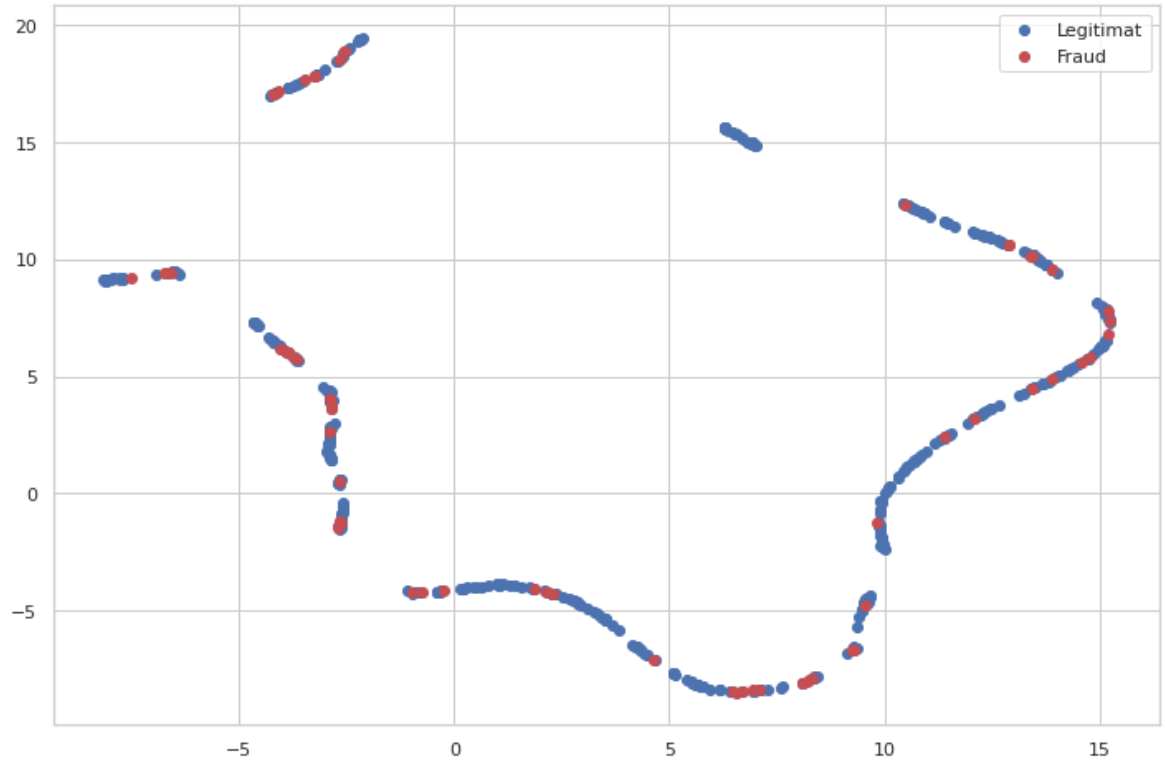

Figure 5 UMAP representation for 3000 legitimate and 492 fraudulent samples

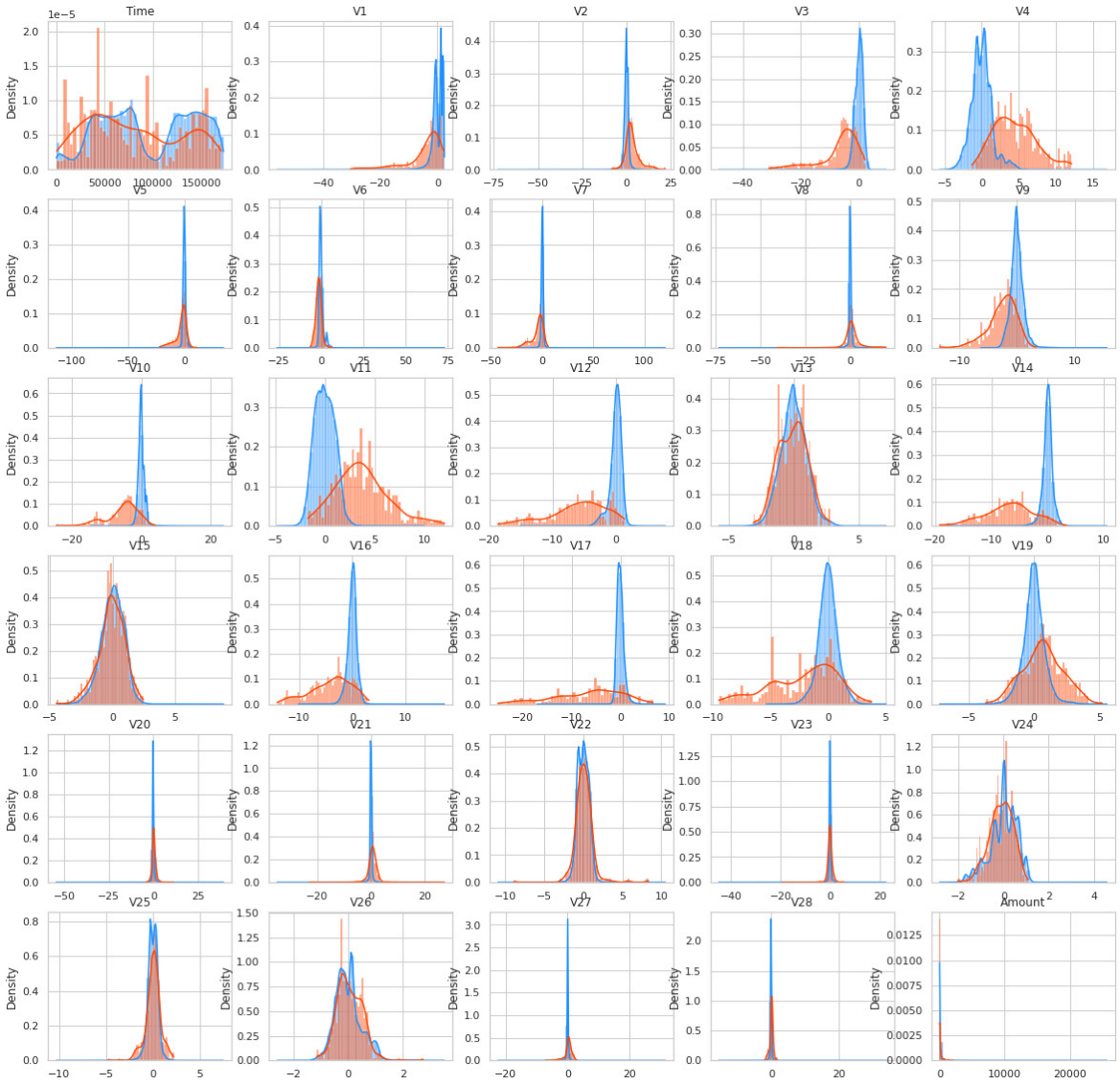

Figure 6 The Features Distribution

4.3. Pre-processing

4.3.1. Data Cleaning

The dataset is loaded and pre-processed at this phase. read_csv() function in the Panda library was used to read the data in CSV format and save it as a Dataframe to facilitate cleaning and further analysis. Then, we used the isna(), isnull() and duplicated() functions in created Dataframe object to check the missing, Nan and duplicated values. The dataset has no missing or Nan values. However, the dataset has 1854 duplicated rows, where 1822 are normal, and 32 are fraud. The duplicated samples were removed to avoid bias in the evaluation as they are the

exact data point. Only the first duplicated sample was kept, with 1081 total removed samples. Table 3 shows the number of samples before and after the duplicates are removed.

Table 3 Dataset Structure Before and After Removing Duplicates.

Transactions	Original Dataset	Dataset After Cleaning
Legitimate	284315	283253
Fraudulent	492	473

4.3.2. Splitting the Dataset

Once the dataset is cleaned, the dataset is then split into training and testing sets. The training set included all normal transactions, excluding 473 random samples, which were added to the 473 fraud samples to form the test set. Table 4 shows the number of transactions in the training and testing sets. To do this, we used the train_test_split() function in the Scikit-learn library to exclude the random 473 normal samples from the normal transactions. We then used concatenate() in the NumPy library to add them to the fraud samples.

Table 4 Training and Testing sets Segmentation.

Dataset	Legitimate	Fraudulent	Total
Training set	282780	_____	**282780**
Testing set	473	473	**946**
Total	**283253**	**473**	**283726**

4.3.3. Scaling the Data

The statistical analysis revealed that the features' means, maximums, and minimums differ. So, the training and testing sets were scaled before they were fed into the feature reduction and novelty detection algorithms. We used the StandardScaler and MinMaxScaler classes in the Scikit-learn preprocessing submodule. MinMaxScaler was used with the AE algorithm, and StanderScaler was used with the OCSVM and IF methods.

4.4. Feature Reduction

Once the pre-processing phase is completed, the training set will be used to train the Autoencoder to reduce the features. Four different Autoencoder architectures were constructed. The four AE architectures implemented with the hyper-parameters are given in Table 5. The details of each network architecture are shown in Table 6. The four networks were trained on 70% of the training set and then evaluated on the remaining 30%. Reconstruction accuracy was used to evaluate the architectures. The architecture with the best reconstruction accuracy will be chosen to extract the features. Table 7 shows the reconstruction accuracy of the AE architectures. As we can see, the AE 2 gets the highest reconstruction accuracy at 99.34%, which means it learned good feature representations. Therefore, AE 2 was chosen for the feature reduction. After that, we extract the AE2 encoder part to get the low-dimensional representation of the training and testing sets to train and evaluate the OCSVM, AEs and IF. Figure 7 shows the low-dimensional representation extracted from AE2 using UMAP for 3000 legitimate and 492 fraudulent samples. We can observe from the graph that fraud and legitimate transactions are more detachable in the extracted hidden representation compared to the original representation in Figure 5.

Table 5 The AE Hyper-Parameters

Variable	Parameters
Hidden Layers Activation Function	Tanh
Loss Function	Mean Squared Error
Batch Size	32
Epochs	100
Optimizer	Adam

Table 6 Network Architecture Details

Architecture	Input Layer	Hidden Layers	Output Layer
AE 1	30	20,10,**5,**10,20	30
AE 2	30	24,22,**20**,22,24	30
AE 3	30	28,26,24,22,20,**18**,20,22,24,26,28	30
AE4	30	29,27,25,23,21,19,17,15,13,11,9,7,**5**,7,9,11,13,15,17,19,21,23, 25,27,29	30

Table 7 AEs Reconstruction Accuracy

Architecture	Reconstruction Accuracy (%)
AE 1	96.59
AE 2	99.34
AE 3	98.28
AE 4	96.70

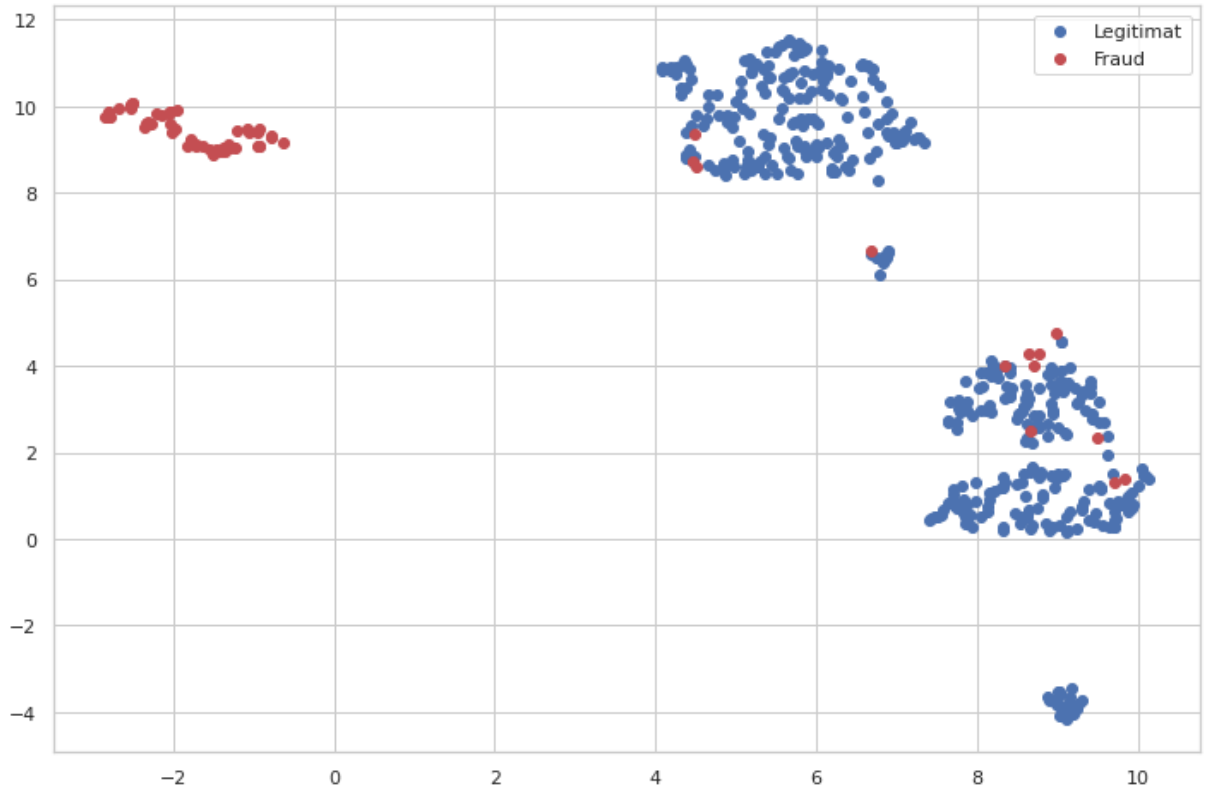

Figure 7 UMAP representation of the extracted low dimension for 3000 legitimate and 492 fraudulent samples

4.5. Novelty Detection Models

Once the low-dimensional representation of the training set is extracted, it is used to train the proposed novelty algorithms. The detailed implementation of the algorithms is explained below for each algorithm.

4.5.1. One-Class SVM (OCSVM)

In training the OCSVM, we employed the Radial Basic Function (RBF) kernel to achieve the non-linear and multiple decision boundaries for inliers. The gamma was set to the Scale, and nu was searched in a different values range. Table 8 shows the Hyper-Parameters values used in training the OCSVM.

Table 8 The OCSVM Hyper-Parameters

Variable	Parameters
kernel	RPF
gamma	Scale
nu	[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08]

4.5.2. Isolation Forest (IF)

Two parameters searched in the training of IF are the contamination and max features. Table 9 shows the Hyper-Parameters values used in training the IF.

Table 9 The IF Hyper-Parameters

Variable	Parameters
Contamination	[0.09, 0.1, 0.17]
Max Features	[20, 10, 5]

4.5.3. Autoencoder (AE)

The AE1 and AE3 architectures were used as novelty detection models based on a specific threshold. To calculate the threshold, we used the training transactions. First, we fed it to the trained AEs architectures. Then we got the reconstructed outputs from the models and compared them to the original ones to calculate the reconstruction errors for each transaction. After that, we investigate various thresholds based on the percentage covering training transactions reconstruction errors. The percentage range from 90 to 99.

However, the AE1 and AE3 architectures were adjusted when trained with the extracted feature, so their input and output layers changed. Table 10 show their architectures when trained with the extracted features from AE2.

Table 10 Adjusted Network Architecture Details

Architecture	Input Layer	Hidden Layers	Output Layer
AE 1	20	10,**5,**10	20
AE 3	20	**18**	20

5. Results and Evaluation

Following the pre-processing of the dataset and feature reduction, the novelty detection algorithms discussed in the previous chapter are trained, and the results are illustrated and discussed in this chapter. Moreover, the findings of this study have been compared with the state-of-the-art unsupervised novelty detection methods on credit card fraud.

5.1. Experiments Workflow

Two main experiments are set up to select the effective model for detecting credit card fraud. The first experiment aims to train novelty detection models on the extracted feature and various parameters, and the parameters and architecture of the proposed algorithms were chosen. Then, in the second experiment, we will compare the novelty detection algorithms with the selected best hyperparameters with and without the proposed dimension reduction approach.

5.2. Experiments 1 Results:

This experiment aims to find the appropriate parameters for OCSVM and IF and architectures and thresholds for AEs. We trained the OCSVM, IF, AE1 and AE3 with the extracted features representation of the training set from AE 2, with various hyperparameters. The comparisons were based primarily on the AUC score and FNR. The best-chosen parameters for each algorithm are shown in Table 11. Figure 8 compares novelty detection algorithms with the selected best parameters. As shown in Figure 8, AE-AE outperforms AE-OCSVM and AE-IF in terms of AUC, F1 score, Recall and FNR. More detailed results are given in Sections 5.2.1, 5.2.2, and 5.2.3 for AE-OCSM, AE-IF and AE-AE with various parameters, respectively.

Table 11 The Final Hyper-Parameters for each Algorithm

AE-OCSVM	
kernel	RPF
gamma	Scale
nu	0.07
AE-IF	
Contamination	0.09
Max Features	5
AE-AE	
Architecture	AE1
Threshold	0.003742

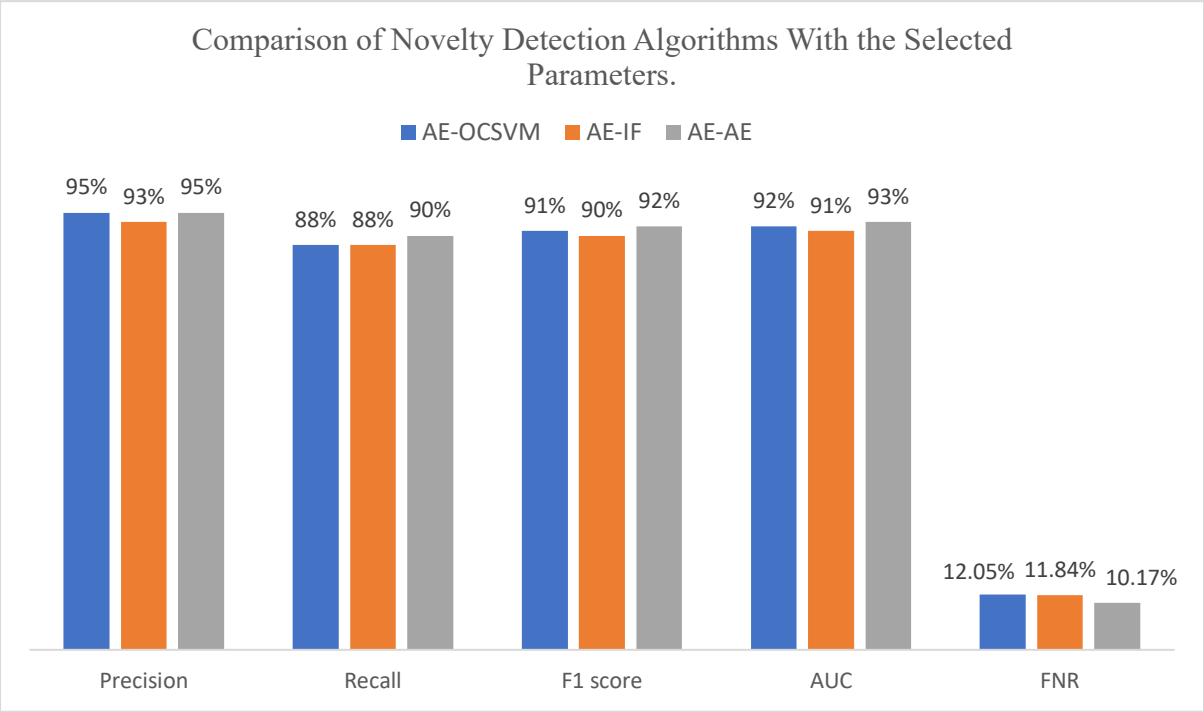

Figure 8 Comparison of Novelty Detection Algorithms With the Selected Final Parameters.

5.2.1. AE-OCSVM

The investigated perparameter when AE-OCSVM was trained is the nu. As discussed previously, nu is a key parameter that, in real-life applications, should be set to a value acceptable to stakeholders and the application domain. Table 12 shows the results of the trained AE-OCSVM with different nu values. Results show that as the nu value increases, the FNR decreases and the recall increases where the lowest FNR was when nu is 0.08. However, the best AUC score, along with the lowest FNR, was found when nu was 0.07. thus, it was selected as the best value for the AE-OCSVM in this study.

Table 12 The AE-OCSVM Evaluation with Different nu Values

kernel	***gamma***	***nu***	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**AUC (%)**	**FNR (%)**	**Training Time (s)**	**Testing Time (s)**
RPF	***Scale***	***0.01***	**99.0**	83.0	90.0	91.0	16.913	192.96	0.22
		0.02	98.0	85.0	91.0	**92.0**	14.799	1737.56	0.41
		0.03	97.0	86.0	91.0	**92.0**	13.742	1291.81	0.62
		0.04	96.0	87.0	91.0	**92.0**	12.896	899.03	0.86
		0.05	96.0	87.0	91.0	**92.0**	12.685	1458.87	1.03
		0.06	95.0	88.0	91.0	**92.0**	12.262	1355.37	1.24
		0.07	95.0	88.0	91.0	**92.0**	12.051	1584.09	1.46
		0.08	93.0	88.0	91.0	91.0	**11.628**	2196.03	1.68

5.2.2. AE-IF

The investigated hyperparameters when AE-IF was trained are the Contamination and Max Features. Table 13 shows the results of the trained AE-IF with different Contamination and Max Features values. Results show that the lowest FNRs were found when Contamination is 0.17, the same as the percentage of outliers in the dataset. In contrast, the AUC and F1 scores were too low. However, the best AUC score, along with the lowest FNR, was found when the Contamination was 0.09 and the max features were 5. thus, they were selected as the best values for the AE-IF in this study.

Table 13 The AE-IF Evaluation with Different Contamination and Max Features Values

Contamination	***Max Features***	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**AUC (%)**	**FNR (%)**	**Training Time (s)**	**Testing Time (s)**
0.09	***20***	92.0	89.0	90.0	90.0	11.416	13.54	0.07
	10	91.0	89.0	90.0	90.0	11.205	12.58	0.06
	5	93.0	88.0	90.0	**91.0**	11.839	9.71	0.07
0.1	***20***	91.0	89.0	90.0	90.0	10.994	13.37	0.07
	10	90.0	89.0	90.0	90.0	11.205	12.55	0.07
	5	92.0	89.0	90.0	90.0	11.416	9.76	0.06
0.17	***20***	86.0	90.0	88.0	90.0	9.514	13.39	0.07
	10	84.0	91.0	88.0	88.0	9.091	12.60	0.07
	5	86.0	86.0	89.0	87.0	**8.668**	9.79	0.06

5.2.3. AE-AE

The trained AE-AE1 and AE-AE3 were used to detect fraud based on different reconstruction errors (Thresholds). Percentiles of reconstruction error of training data were used to specify Thresholds. Tables 14 and 15 show the results of AE-AE1 and AE-AE3 with different Thresholds. Results show that as percentiles increased, the precision and FNR increased because the Threshold value increased, covering more normal and fraud reconstruction error values. The lowest FNRs were found when the Threshold was 0.002785 and 0.000050 for AE-AE1 and AE-AE3, respectively. However, the AE-AE1 shows better performance than AE-AE3. The best AUC score, along with the lowest FNR, was found using AE-AE1 with 0.003742 as the Threshold. Thus, the AE-AE1 and 0.003742 as Threshold were chosen for this study.

Table 14 The AE-AE1 Evaluation with Different Threshold Values

Per%	*Threshold*	Precision (%)	Recall (%)	F1 score (%)	AUC (%)	FNR (%)	Training Time (s)	Testing Time (s)
90	*0.002785*	89.0	92.0	90.0	90.0	**8.051**	2302.83	0.10
91	*0.002934*	90.0	91.0	91.0	90.0	8.898		0.10
92	*0.003097*	91.0	91.0	91.0	91.0	8.898		0.10
93	*0.003272*	93.0	91.0	92.0	92.0	9.322		0.09
94	*0.003482*	94.0	90.0	92.0	92.0	9.746		0.10
95	*0.003742*	95.0	90.0	92.0	**93.0**	10.169		0.09
96	*0.004079*	96.0	89.0	93.0	**93.0**	10.593		0.09
97	*0.004561*	97.0	89.0	93.0	**93.0**	11.017		0.12
98	*0.005294*	97.0	88.0	92.0	**93.0**	12.288		0.09
99	*0.006585*	99.0	84.0	91.0	91.0	16.102		0.09

Table 15 The AE-AE3 Evaluation with Different Threshold Values

Per%	*Threshold*	Precision (%)	Recall (%)	F1 score (%)	AUC (%)	FNR (%)	Training Time (s)	Testing Time (s)
90	*0.000050*	89.0	85.0	87.0	87.0	**15.222**	1437.34	0.45
91	*0.000052*	89.0	84.0	87.0	87.0	15.645		0.44
92	*0.000056*	90.0	84.0	87.0	87.0	16.279		0.36
93	*0.000060*	92.0	84.0	87.0	88.0	16.490		0.33
94	*0.000066*	93.0	83.0	88.0	88.0	17.125		0.36
95	*0.000073*	94.0	82.0	88.0	88.0	17.759		0.18
96	*0.000083*	95.0	82.0	88.0	**89.0**	17.759		0.19
97	*0.000101*	96.0	82.0	89.0	**89.0**	17.759		0.26
98	*0.000133*	97.0	81.0	88.0	**89.0**	19.239		0.19
99	*0.000224*	98.0	80.0	88.0	**89.0**	20.085		0.19

5.3. Experiments 2 Results:

This experiment examines the proposed feature reduction approach's effectiveness in prediction performance and computational time. In this experiment, the OCSVM, IF, AE1 and AE3 were trained without the extracted features representation from AE 2, then compared with the previous experimental results of the trained models with the extracted features. The comparisons were based on the evaluation metrics and the training and testing time. Figures 9, 10, 11, 12 and 13 compare the evaluation metrics and computational time of the chosen models with the best parameters from the previous experiment with and without using the AE as feature reduction. Results show that using AE as feature reduction enhanced the evaluation metrics, particularly in terms of AUC, F1 and precision, and reduced computational time. Tables 16, 17, 18 and 19 below provide more detailed results for each algorithm trained without the extracted features with various parameter values. These results compared with Tables 12, 13,

14 and 15; results also showed a significant improvement in the performance in all searched parameters when using AE as a feature reduction.

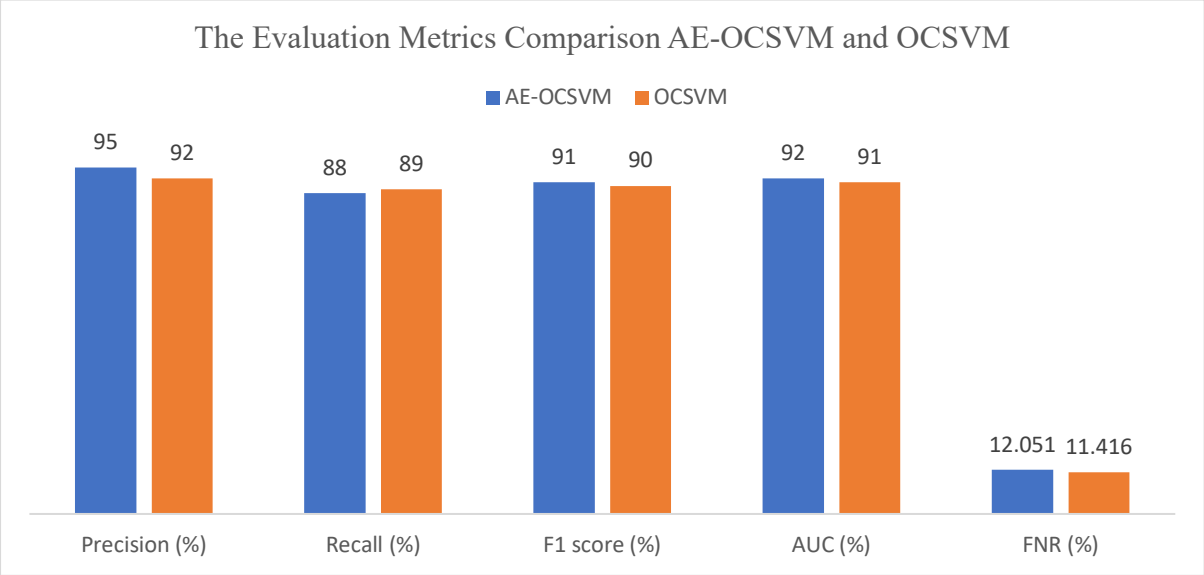

Figure 9 The Evaluation Metrics Comparison AE-OCSVM and OCSVM

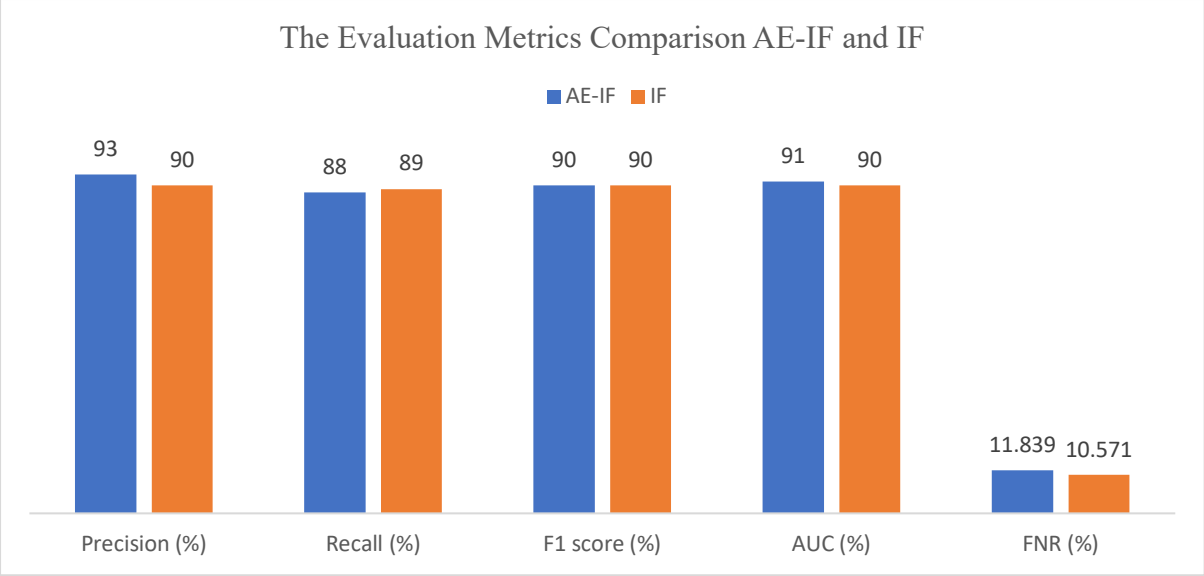

Figure 10 The Evaluation Metrics Comparison AE-IF and IF

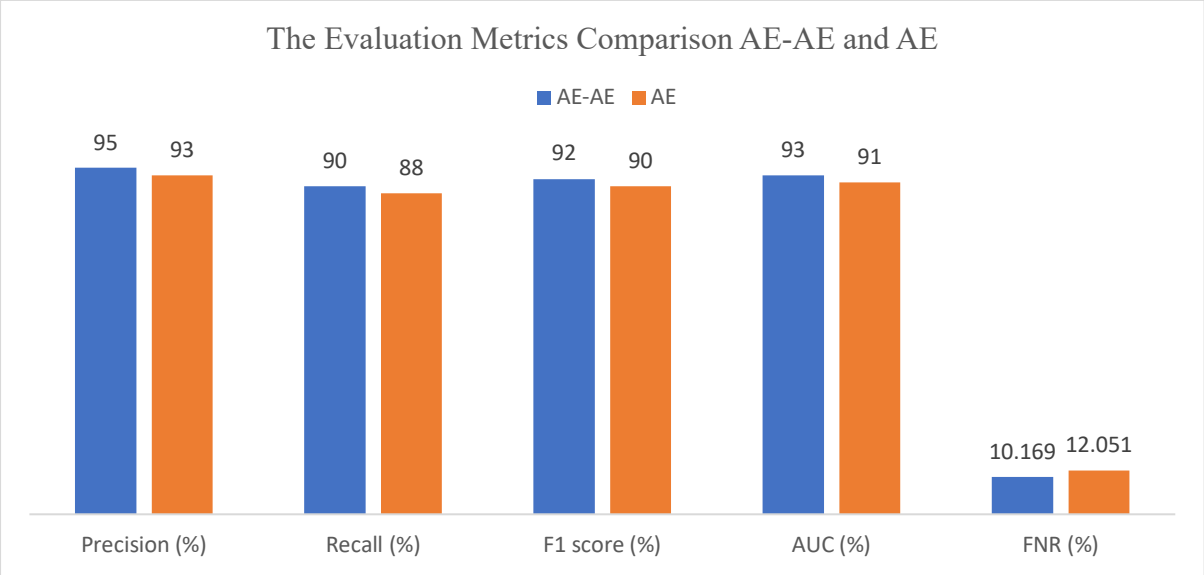

Figure 11 The Evaluation Metrics Comparison AE-AE and AE

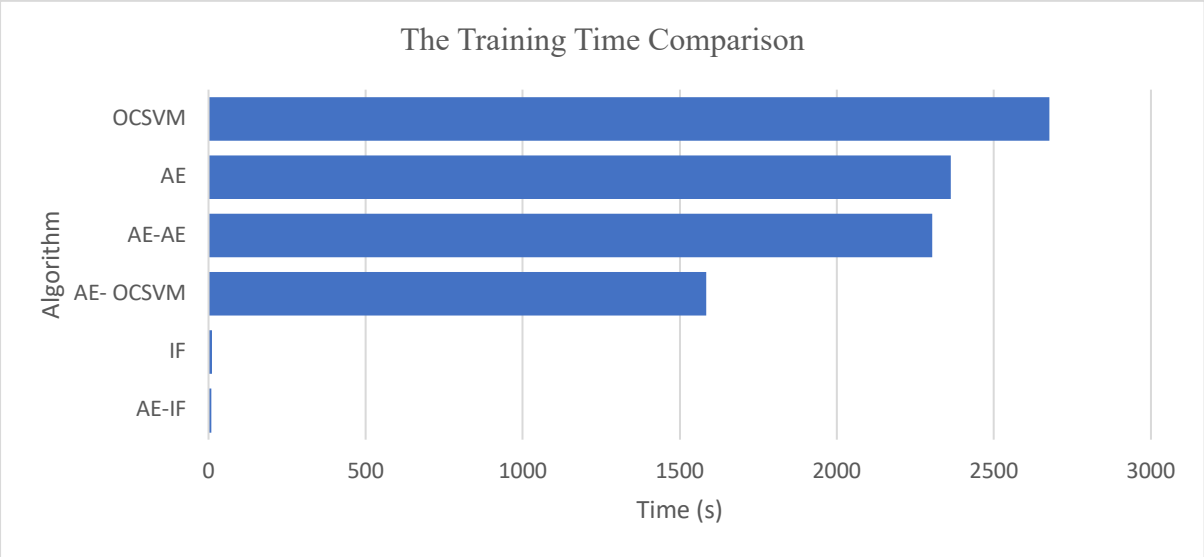

Figure 12 The Training Time Comparison.

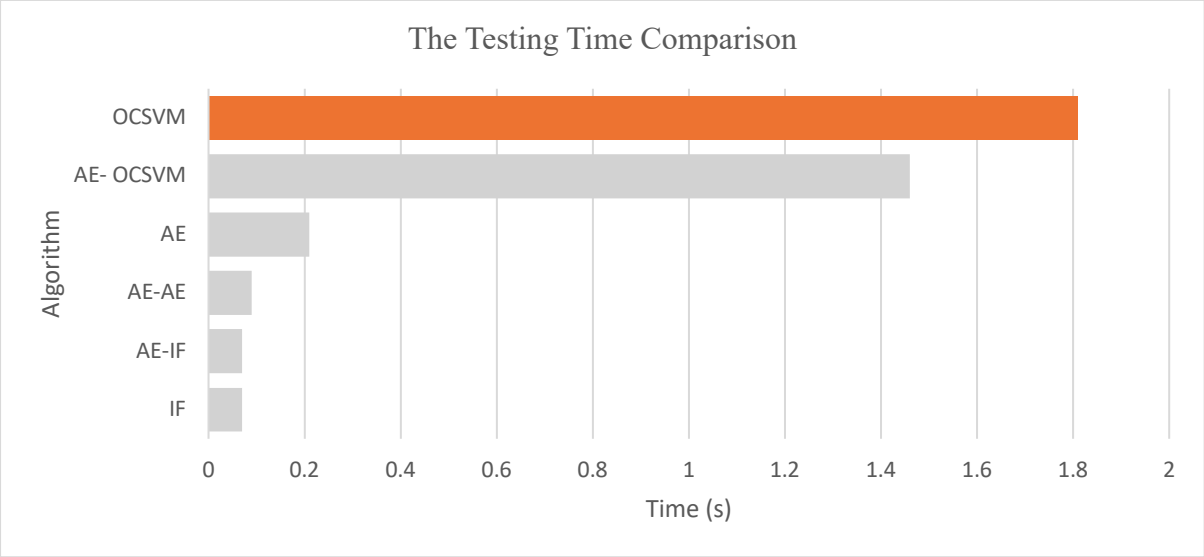

Figure 13 The Testing Time Comparison.

Table 16 The OCSVM Evaluation with Different nu Values

kernel	***nu***	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**AUC (%)**	**FNR (%)**	**Training Time (s)**	**Testing Time (s)**
RPF	***0.01***	97.0	82.0	89.0	90.0	17.759	604.68	0.38
	0.02	97.0	83.0	90.0	90.0	16.702	870.78	0.58
	0.03	96.0	84.0	90.0	90.0	16.068	1575.42	0.82
	0.04	96.0	87.0	91.0	**91.0**	13.108	1503.33	1.07
	0.05	95.0	87.0	91.0	**91.0**	12.685	1814.87	1.36
	0.06	93.0	88.0	91.0	**91.0**	12.051	2177.92	1.60
	0.07	92.0	89.0	90.0	**91.0**	11.416	2676.15	1.81
	0.08	90.0	89.0	90.0	90.0	**10.571**	3312.92	2.27

Table 17 The IF Evaluation with Different Contamination and Max Features Values

Contamination	***Max Features***	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**AUC (%)**	**FNR (%)**	**Training Time (s)**	**Testing Time (s)**
0.09	***20***	89.0	88.0	89.0	89.0	11.628	27.32	0.08
	10	90.0	89.0	89.0	89.0	10.994	15.24	0.07
	5	90.0	89.0	90.0	**90.0**	10.571	11.57	0.07
0.1	***20***	88.0	89.0	88.0	88.0	11.416	22.96	0.08
	10	88.0	89.0	89.0	89.0	10.782	15.70	0.07
	5	89.0	90.0	89.0	89.0	10.359	11.51	0.08
0.17	***20***	83.0	93.0	88.0	87.0	**7.400**	22.97	0.07
	10	84.0	92.0	88.0	87.0	7.611	15.22	0.08
	5	84.0	92.0	87.0	87.0	8.457	11.58	0.07

Table 18 The AE1 Evaluation with Different Threshold Values

Per%	*Threshold*	Precision (%)	Recall (%)	F1 score (%)	AUC (%)	FNR (%)	Training Time (s)	Testing Time (s)
90	*0.001829*	89.0	90.0	90.0	90.0	**10.148**		0.18
91	*0.001914*	90.0	89.0	89.0	90.0	10.994		0.22
92	*0.002003*	91.0	89.0	90.0	90.0	10.994		0.24
93	*0.002103*	92.0	89.0	90.0	90.0	11.416		0.73
94	*0.002226*	93.0	88.0	90.0	**91.0**	12.051	2362.47	0.21
95	*0.002374*	94.0	88.0	91.0	**91.0**	12.474		0.18
96	*0.002544*	96.0	86.0	91.0	**91.0**	13.742		0.20
97	*0.002773*	96.0	85.0	91.0	**91.0**	14.588		0.21
98	*0.003131*	97.0	84.0	90.0	**91.0**	15.645		0.19
99	*0.004055*	99.0	81.0	89.0	90.0	19.027		0.19

Table 19 The AE3 Evaluation with Different Threshold Values.

Per%	*Threshold*	Precision (%)	Recall (%)	F1 score (%)	AUC (%)	FNR (%)	Training Time (s)	Testing Time (s)
90	*0.000039*	88.0	85.0	86.0	87.0	**15.254**		0.10
91	*0.000042*	89.0	85.0	87.0	87.0	**15.254**		0.09
92	*0.000045*	90.0	84.0	87.0	87.0	15.678		0.10
93	*0.000050*	93.0	84.0	88.0	89.0	16.102		0.10
94	*0.000055*	93.0	84.0	88.0	89.0	16.102	2261.79	0.09
95	*0.000063*	93.0	83.0	88.0	88.0	17.373		0.10
96	*0.000074*	94.0	81.0	87.0	88.0	19.068		0.09
97	*0.000091*	96.0	81.0	88.0	**89.0**	19.068		0.10
98	*0.000120*	96.0	80.0	88.0	**89.0**	19.915		0.09
99	*0.000210*	98.0	78.0	87.0	88.0	22.458		0.10

5.4. Discussion

The results of this study provide supporting evidence that using unsupervised novelty detection techniques and Autoencoder as a feature reduction method has a promising potential to detect credit card fraud. The results also demonstrate that the proposed approach deals effectively with imbalanced and unlabeled datasets and can reduce training and prediction time.

Defining and determining performance indicator for credit card fraud detection is an open issue and rely on the business and stakeholder. As discussed in section 3.6, in this study, we chose the most critical factor to be that the model makes correct predictions, considering the reduction of misclassified fraud. Therefore, we used the AUC score as the primary evaluation metric and the FNR as the second metric when comparing different algorithms' results and choosing the parameters and threshold in the experiments of this study. With 93% AUC and 10.17% FNR, the AE-AE1 model achieved the best results, as shown in Figure 8. AE-OCSVM and AE-IF

closely followed these results with 92%, 91% AUC and 12.05%, and 11.84% FNR, respectively.

Figures 12 and 13 compare the training and testing times of the novelty detection algorithms. OCSVM and AE have high computational time, particularly during training, whereas IF has the lowest training and testing time. However, this could be improved by fine-tuning AE's epochs and batch size and employing sample reduction methods to reduce the training set for OCSVM [64], [65].

Using Autoencoder as a feature reduction method with only one class shows efficiency in reducing features. It has decreased the features from 30 in the original dataset to 20, enhanced the model's performance, and reduced the computational time. the results show that the AUC of the AE, OCSVM and IF classifiers have increased from 91.0% to 93.0%, 91.0% to 92.0% and 90.0% to 91.0% and the precision are also improved from 93.0% to 95.0%, 92.0% to 95.0% and 90.0% to 93.0% when trained with the extracted features. The main reason for this high performance is that the novelty detection algorithms are trained with low-dimensional representation. The low-dimensional representation helps the models characterise the transaction better, so the AUC score improved significantly, particularly with AE3. Additionally, we have tried UMAP to reduce the dimension. We fed it by the training set that only contained normal samples and searched various components (3,5,10,20). However, it shows a drop in performance, unlike when it was fitted by normal and fraud data points in [49].

5.5. Comparative Analysis

In a topic such as credit card fraud detection, Performing a precise comparison with prior research is challenging due to the variety of study environments. Studies in credit card fraud have used different learning approaches and feature extraction and selection methods. As a result, their evaluation metrics and results differed. Therefore, we compared only similar previous studies in our study. Our algorithms are compared with prior studies using an unsupervised learning novelty detection approach to solve transactional credit card fraud problems, and Table 20 shows relevant studies. We consider several factors in our comparison: algorithms, datasets, number of features, and the highest achieved evaluation scores.

As shown in Table 20, our models perform as good as previous studies' algorithms. Our models outperform the VAE in all the evaluation metrics. It also achieves better results than the SAE-GAN algorithm in F1 and Recall. Most studies focused on detecting fraud and handling the imbalanced classification problem. However, reducing the computational time is also very important in the detection problem. Comparing our approach to relevant studies, we used 20 features rather than 30 in the original dataset, reduced training and testing time and enhanced performance.

Table 20 Comparison of Current Study Findings with Previous Studies Results

Author(s)	***Algorithm(s)***	***Datasets***	***Number of Features***	***Evaluation Metrics***				
				Precision	**Recall**	**F1 score**	**AUC**	**FNR**
[52]	VAE	European cardholder	30	74.2	81.5	77.6	___	___
[11]	SAE-GAN	European cardholder	50	97.59	79.37	87.36	___	___
[50]	GS-OCSVM	European cardholder	30	___	97.06	90.32	92.28	___
	IF			___	98.53	91.80	93.01	___
Our	AE-AE	European cardholder	20	95.0	90.0	92.0	93.0	10.169
	AE-OCSVM			95.0	88.0	91.0	92.0	12.051
	AE-IF			93.0	88.0	90.0	91.0	11.839

6. Limitations and Future Work

Although the proposed approach offers great promise for detecting credit card fraud, reducing training and testing time, and dealing with imbalanced datasets, the results may have a high FNR. Moreover, due to the scarcity of data on credit card fraud, using only one dataset could limit the generalizability of the experiment's findings. However, due to the limited time allocated for this study, multiple aspects could be improved in the future. A few suggestions are provided below.

- Analyse and assess the performance of the proposed models using other fraud datasets as they become available.

- Expand the investigation by applying other machine and deep learning novelty detection methods, such as Local Outlier Factor (LOF) and Subspace Outlier Detection (SOD).

- Combine two or more novelty detection methods, then analyse and evaluate their performance.

- Apply a hybrid approach by combining unsupervised and supervised methods and evaluating their performance.

- Perform different feature reduction methods or hybrid reduction methods combining different techniques and evaluating their effectiveness.

- Investigate methods that could minimise the effect of class overlapping and improve fraud detection.

7. Conclusion

This study proposes unsupervised novelty detection techniques with the help of dimensionality reduction methods to detect credit card fraud. This approach is proposed to address three challenges facing the design of a fraud detection system: class imbalance, lack of labelled dataset, and high dimensionality of the data. The European cardholder dataset has been used to evaluate this approach.

In this study, we compare and analyse the performance of three unsupervised novelty detection techniques, OCSVM, IF, and AE, with the help of Autoencoder, as a feature reduction method. First, we constructed four AE architectures and chose the architecture with the highest reconstruction accuracy to extract the low-dimension representation of the transactions. Next, the novelty detection methods were trained on the extracted features with various parameters, and the most appropriate parameter was selected for each algorithm.

The experimental results show that the proposed approach effectively detects credit card fraud, handles imbalanced and unlabeled datasets, and reduces training and prediction times. Moreover, using Autoencoder as a feature reduction method with only one class has proven to be efficient in reducing the number of features. It has reduced the number of features from 30 in the original dataset to 20 and enhanced the model's performance compared to those trained without the extracted features. The highest AUC score and the lowest FNR were achieved by AE-AE1.

8. Reflection on Learning

Topic selection was the first step in carrying out this study. We have done many machine learning projects during the taught modules, especially the applied machine learning module. Most of these projects were based on supervised learning and classification methods. However, a limited section of this module was about unsupervised learning and novelty detection methods. This part caught my attention, and I wanted to learn more about it than we did in class. Interestingly, after the lecture, Dr Yuhua Li mentioned an available project this year to detect credit card fraud using novelty detection techniques. At that point, I knew that this project would introduce me and enable me to expand my knowledge in this new area.

The work in this study was broken up into several phases to ensure the most successful possible outcome, and each of these phases provided practical and academic lessons. These phases include:

- Developing a thorough understanding of the topic and the problem that needs to be solved.
- The exploration of research methodologies and methods of implementation to solve the problem.
- The practical implementation and the analysis of the outcomes.

Initially, I was challenged to formulate the problem at hand. I needed to improve my understanding of the issue and its implications. I had to research multiple solutions and existing approaches. Doing so improved my research abilities and furthered my skills. I conducted an extensive survey of the literature in which I studied my options and possible solutions, and I came to identify gaps and opportunities in the works of others. Additionally, reviewing existing research methodologies and methods helped me create a concrete solution tailored to my scope. Moreover, I gained valuable knowledge in articulating methods and approaches I was unfamiliar with before this study.

Lastly, the practical implementation was all about trial and error. One of the key challenges during the implementation was using an Autoencoder since it was different from other used novelty detection methods, which could be called directly using the Scikit-learn module. And it was my first time dealing with a neural network. To accomplish this, I spent several weeks

reading about autoencoders, their types and applications, and how to select the appropriate activation functions and parameters to construct an Autoencoder. Another challenge was determining the evaluation metrics to compare between the models. Choosing metrics to evaluate fraud detection is an open issue and depends on the business or the study's aims. However, after reading and investigating, I settled on the AUC and FNR as evaluation metrics.

Finally, suppose I am given another opportunity to work on credit card fraud detection or any other novelty detection problem. In that case, I believe I will be able to approach it appropriately, reducing the amount of time it takes using what I have learned. In summary, I would like to conclude that this project is one of the most valuable parts of my master's journey. This is because it provided me with many practical and academic skills that I can use in the future. I believe this experience was beneficial and will allow me to develop my work further or engage in new adventures in my future professional and academic life.

References

[1] R. J. Bolton and D. J. Hand, “Statistical Fraud Detection: A Review,” *Statistical Science*, vol. 17, no. 3, pp. 235–249, 2002, [Online]. Available: http://www.jstor.org/stable/3182781

[2] A. Abdallah, M. A. Maarof, and A. Zainal, “Fraud detection system: A survey,” *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, Jun. 2016, doi: 10.1016/j.jnca.2016.04.007.

[3] A. Srivastava, A. Kundu, S. Sural, and A. K. Majumdar, “Credit Card Fraud Detection Using Hidden Markov Model,” *IEEE Trans Dependable Secure Comput*, vol. 5, no. 1, pp. 37–48, Jan. 2008, doi: 10.1109/TDSC.2007.70228.

[4] B. Pokora, “9 Interesting Credit Card Statistics,” Jun. 08, 2022. https://www.forbes.com/advisor/credit-cards/credit-card-statistics/ (accessed Sep. 06, 2022).

[5] K. J. Barker, J. D’Amato, and P. Sheridon, “Credit card fraud: awareness and prevention,” *J Financ Crime*, vol. 15, no. 4, pp. 398–410, Oct. 2008, doi: 10.1108/13590790810907236.

[6] Shift Credit Card Processing, “Credit Card Fraud Statistics,” 2021. https://shiftprocessing.com/credit-card-fraud-statistics/ (accessed Aug. 03, 2022).

[7] Federal Trade Commission, “New Data Shows FTC Received 2.8 Million Fraud Reports from Consumers in 2021,” 2022. Accessed: Aug. 08, 2022. [Online]. Available: https://www.ftc.gov/news-events/news/press-releases/2022/02/new-data-shows-ftc-received-28-million-fraud-reports-consumers-2021-0

[8] S. Dhankhad, E. Mohammed, and B. Far, “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study,” in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Jul. 2018, pp. 122–125. doi: 10.1109/IRI.2018.00025.

[9] V. N. Dornadula and S. Geetha, “Credit Card Fraud Detection using Machine Learning Algorithms,” *Procedia Comput Sci*, vol. 165, pp. 631–641, 2019, doi: 10.1016/j.procs.2020.01.057.

[10] J. V. v. Sriram Sasank, G. R. Sahith, K. Abhinav, and M. Belwal, “Credit Card Fraud Detection Using Various Classification and Sampling Techniques: A Comparative Study,” in *2019 International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2019, pp. 1713–1718. doi: 10.1109/ICCES45898.2019.9002289.

[11] J. Chen, Y. Shen, and R. Ali, “Credit Card Fraud Detection Using Sparse Autoencoder and Generative Adversarial Network,” in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2018, pp. 1054–1059. doi: 10.1109/IEMCON.2018.8614815.

[12] A. Shah and A. Mehta, “Comparative Study of Machine Learning Based Classification Techniques for Credit Card Fraud Detection,” in *2021 International Conference on Data Analytics for Business and Industry (ICDABI)*, Oct. 2021, pp. 53–59. doi: 10.1109/ICDABI53623.2021.9655848.

[13] S. Venkata Suryanarayana, G. N. Balaji, and G. Venkateswara Rao, “Machine Learning Approaches for Credit Card Fraud Detection,” *International Journal of Engineering & Technology*, vol. 7, no. 2, p. 917, Jun. 2018, doi: 10.14419/ijet.v7i2.9356.

[14] A. Dal Pozzolo, O. Caelen, Y.-A. le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst Appl*, vol. 41, no. 10, pp. 4915–4928, Aug. 2014, doi: 10.1016/j.eswa.2014.02.026.
[15] P. H. Tran, K. P. Tran, T. T. Huong, C. Heuchenne, P. HienTran, and T. M. H. Le, "Real Time Data-Driven Approaches for Credit Card Fraud Detection," in *Proceedings of the 2018 International Conference on E-Business and Applications - ICEBA 2018*, 2018, pp. 6–9. doi: 10.1145/3194188.3194196.
[16] L. H. N. Lorena, A. C. P. L. F. Carvalho, and A. C. Lorena, "Filter Feature Selection for One-Class Classification," *J Intell Robot Syst*, vol. 80, no. S1, pp. 227–243, Dec. 2015, doi: 10.1007/s10846-014-0101-2.
[17] R. de Best, "Visa, MasterCard, UnionPay transaction volume worldwide," Mar. 11, 2022. https://www.statista.com/statistics/261327/number-of-per-card-credit-card-transactions-worldwide-by-brand-as-of-2011/ (accessed Aug. 03, 2022).
[18] A. O. Adewumi and A. A. Akinyelu, "A survey of machine-learning and nature-inspired based credit card fraud detection techniques," *International Journal of System Assurance Engineering and Management*, vol. 8, no. S2, pp. 937–953, Nov. 2017, doi: 10.1007/s13198-016-0551-y.
[19] Y.-A. le Borgne, W. Siblini, B. Lebichot, and G. Bontempi, *Reproducible Machine Learning for Credit Card Fraud Detection - Practi cal Handbook*. Université Libre de Bruxelles. [Online]. Available: https://github.com/Fraud-Detection-Handbook/fraud-detection-handbook
[20] "The Nilson Report - ISSUE 1209," Dec. 2021. Accessed: Aug. 06, 2022. [Online]. Available: https://nilsonreport.com/download_a_free_sample.php?1=4
[21] UK Finance, "Fraud - The Facts 2020," 2020. Accessed: Aug. 05, 2022. [Online]. Available: https://www.ukfinance.org.uk/system/files/Fraud-The-Facts-2020-FINAL-ONLINE-11-June.pdf
[22] P. Juszczak, N. M. Adams, D. J. Hand, C. Whitrow, and D. J. Weston, "Off-the-peg and bespoke classifiers for fraud detection," *Comput Stat Data Anal*, vol. 52, no. 9, pp. 4521–4532, May 2008, doi: 10.1016/j.csda.2008.03.014.
[23] T. P. Bhatla, V. Prabhu, and A. Dua, "Understanding credit card frauds," *Cards business review*, vol. 1, no. 6, pp. 1–15.
[24] I. Molloy, J. Li, and N. Li, "Dynamic Virtual Credit Card Numbers," 2007, pp. 208–223. doi: 10.1007/978-3-540-77366-5_19.
[25] G. Gianini, L. Ghemmogne Fossi, C. Mio, O. Caelen, L. Brunie, and E. Damiani, "Managing a pool of rules for credit card fraud detection by a Game Theory based approach," *Future Generation Computer Systems*, vol. 102, pp. 549–561, Jan. 2020, doi: 10.1016/j.future.2019.08.028.
[26] J. Jurgovsky *et al.*, "Sequence classification for credit-card fraud detection," *Expert Syst Appl*, vol. 100, pp. 234–245, Jun. 2018, doi: 10.1016/j.eswa.2018.01.037.
[27] "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy," *IEEE Trans Neural Netw Learn Syst*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018, doi: 10.1109/TNNLS.2017.2736643.
[28] N. A. Ishak, K.-H. Ng, G.-K. Tong, S. N. Kalid, and K.-C. Khor, "Mitigating unbalanced and overlapped classes in credit card fraud data with enhanced stacking classifiers system," *F1000Res*, vol. 11, p. 71, Jan. 2022, doi: 10.12688/f1000research.73359.1.
[29] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput Surv*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[30] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, “An experimental evaluation of novelty detection methods,” *Neurocomputing*, vol. 135, pp. 313–327, Jul. 2014, doi: 10.1016/j.neucom.2013.12.002.
[31] M. Kim, S. Jung, and S. Kim, “Fault Detection Method Using Inverse Distance Weight-based Local Outlier Factor,” in *2021 International Conference on Fuzzy Theory and Its Applications (iFUZZY)*, Oct. 2021, pp. 1–5. doi: 10.1109/iFUZZY53132.2021.9605086.
[32] S. Naseer *et al.*, “Enhanced Network Anomaly Detection Based on Deep Neural Networks,” *IEEE Access*, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.
[33] M. Jeragh and M. AlSulaimi, “Combining Auto Encoders and One Class Support Vectors Machine for Fraudulant Credit Card Transactions Detection,” in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, Oct. 2018, pp. 178–184. doi: 10.1109/WorldS4.2018.8611624.
[34] L. Clifton, D. A. Clifton, P. J. Watkinson, and L. Tarassenko, “Identification of patient deterioration in vital-sign data using one-class support vector machines,” in *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2011, pp. 125–131.
[35] K.-S. Kim, S. J. Oh, H. bin Cho, and M. J. Chung, “One-Class Classifier for Chest X-Ray Anomaly Detection via Contrastive Patch-Based Percentile,” *IEEE Access*, vol. 9, pp. 168496–168510, 2021, doi: 10.1109/ACCESS.2021.3136263.
[36] D. Miljković, “Review of novelty detection methods,” in *The 33rd International Convention MIPRO*, 2010, pp. 593–598.
[37] V. S. Spelmen and R. Porkodi, “A Review on Handling Imbalanced Data,” in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, Mar. 2018, pp. 1–11. doi: 10.1109/ICCTCT.2018.8551020.
[38] A. D. Pozzolo and G. Bontempi, “Adaptive Machine Learning for Credit Card Fraud Detection,” 2015.
[39] Z. Li, M. Huang, G. Liu, and C. Jiang, “A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection,” *Expert Syst Appl*, vol. 175, p. 114750, Aug. 2021, doi: 10.1016/j.eswa.2021.114750.
[40] A. Kumar, “Machine Learning – Feature Selection vs Feature Extraction,” Aug. 08, 2021. https://vitalflux.com/machine-learning-feature-selection-feature-extraction/ (accessed Sep. 30, 2022).
[41] Amy, “One-Class SVM For Anomaly Detection.” https://medium.com/grabngoinfo/one-class-svm-for-anomaly-detection-6c97fdd6d8af (accessed Oct. 01, 2022).
[42] Sushanth Sreenivasa, “Radial Basis Function (RBF) Kernel: The Go-To Kernel,” Aug. 12, 2020. https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a (accessed Oct. 01, 2022).
[43] E. Lewinson, “Outlier Detection with Isolation Forest,” Jul. 02, 2018. https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e (accessed Oct. 01, 2022).
[44] J. Verbu, “Detecting and preventing abuse on LinkedIn using isolation forests,” Jul. 13, 2019. https://engineering.linkedin.com/blog/2019/isolation-forest (accessed Oct. 01, 2022).
[45] S. M. Ghatge, “Machine Learning Approach for Credit Card fraud Detection,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 5, pp. 4799–4802, May 2022, doi: 10.22214/ijraset.2022.43587.

[46] A. Singh and A. Jain, “Adaptive Credit Card Fraud Detection Techniques Based on Feature Selection Method,” 2019, pp. 167–178. doi: 10.1007/978-981-13-6861-5_15.
[47] M. J. Therese, A. Devi, R. Gurulakshmi, R. Sandhya, and P. Dharanyadevi, “Credit Card Assent Using Supervised Learning,” in *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, Mar. 2022, pp. 1–6. doi: 10.1109/ICSTSN53084.2022.9761307.
[48] V. Jain, M. Agrawal, and A. Kumar, “Performance Analysis of Machine Learning Algorithms in Credit Cards Fraud Detection,” in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Jun. 2020, pp. 86–88. doi: 10.1109/ICRITO48877.2020.9197762.
[49] I. Benchaji, S. Douzi, B. el Ouahidi, and J. Jaafari, “Enhanced credit card fraud detection based on attention mechanism and LSTM deep model,” *J Big Data*, vol. 8, no. 1, p. 151, Dec. 2021, doi: 10.1186/s40537-021-00541-8.
[50] K. Kittidachanan, W. Minsan, D. Pornnopparath, and P. Taninpong, “Anomaly Detection based on GS-OCSVM Classification,” in *2020 12th International Conference on Knowledge and Smart Technology (KST)*, Jan. 2020, pp. 64–69. doi: 10.1109/KST48564.2020.9059326.
[51] M. A. Al-Shabi, “Credit Card Fraud Detection Using Autoencoder Model in Unbalanced Datasets,” *Journal of Advances in Mathematics and Computer Science*, pp. 1–16, Aug. 2019, doi: 10.9734/jamcs/2019/v33i530192.
[52] M. Raza and U. Qayyum, “Classical and Deep Learning Classifiers for Anomaly Detection,” in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, Jan. 2019, pp. 614–618. doi: 10.1109/IBCAST.2019.8667245.
[53] E. A. Lopez-Rojas and S. Axelsson, “BankSim: A Bank Payment Simulation for Fraud Detection Research,” Oct. 2014.
[54] D. Dua and C. Graff, *UCI Machine Learning Repository*. University of California, Irvine, School of Information. [Online]. Available: http://archive.ics.uci.edu/ml
[55] MACHINE LEARNING GROUP - ULB, “Credit Card Fraud Detection.” 2019. Accessed: Aug. 25, 2022. [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?sortBy=voteCount&datasetId=310
[56] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, “Calibrating Probability with Undersampling for Unbalanced Classification,” in *2015 IEEE Symposium Series on Computational Intelligence*, Dec. 2015, pp. 159–166. doi: 10.1109/SSCI.2015.33.
[57] Walt, Ed., “Data Structures for Statistical Computing in Python,” pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.
[58] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html
[59] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, Firdaus, and Jasmir, “Automatic Features Extraction Using Autoencoder in Intrusion Detection System,” in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Oct. 2018, pp. 219–224. doi: 10.1109/ICECOS.2018.8605181.
[60] B. Abolhasanzadeh, “Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features,” in *2015 7th Conference on Information and Knowledge Technology (IKT)*, May 2015, pp. 1–5. doi: 10.1109/IKT.2015.7288799.
[61] F. Chollet and others, *Keras*. https://keras.io. [Online]. Available: https://keras.io
[62] P. Sharma, “Decoding the Confusion Matrix,” Jul. 22, 2019. https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb (accessed Sep. 08, 2022).

[63] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit Lett*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
[64] Y. Li, “Selecting training points for one-class support vector machines,” *Pattern Recognit Lett*, vol. 32, no. 11, pp. 1517–1522, 2011, doi: https://doi.org/10.1016/j.patrec.2011.04.013.
[65] S. Alam, S. K. Sonbhadra, S. Agarwal, P. Nagabhushan, and M. Tanveer, “Sample reduction using farthest boundary point estimation (FBPE) for support vector data description (SVDD),” *Pattern Recognit Lett*, vol. 131, pp. 268–276, 2020, doi: https://doi.org/10.1016/j.patrec.2020.01.004.