Fraud Detection in Mobile Payment Systems using an XGBoost-based Framework

Petr Hajek

Science and Research Centre, Faculty of Economics and Administration, University of Pardubice, Studentska 84, Pardubice 532 10, Czech Republic

Mohammad Zoynul Abedin*

Department of Finance, Performance Marketing, Teesside University International Business School, Teesside University, Middlesbrough, TS1 3BX Tees Valley, UK

Uthayasankar Sivarajah

Bradford University School of Management, Emm Lane, Heaton, Bradford, UK

Abstract

Mobile payment systems are becoming more popular due to the increase in the number of smartphones, which, in turn, attracts the interest of fraudsters. Extant research has therefore developed various fraud detection methods using supervised machine learning. However, sufficient labeled data are rarely available and their detection performance is negatively affected by the extreme class imbalance in financial fraud data. The purpose of this study is to propose an XGBoost-based fraud detection framework while considering the financial consequences of fraud detection systems. The framework was empirically validated on a large dataset of more than 6 million mobile transactions. To demonstrate the effectiveness of the proposed framework, we conducted a comparative evaluation of existing machine learning methods designed for modeling imbalanced data and outlier detection. The results suggest that in terms of standard classification measures, the proposed semi-supervised ensemble model integrating multiple unsupervised outlier detection algorithms

*corresponding author

Email addresses: petr.hajek@upce.cz (Petr Hajek), m.abedin@tees.ac.uk (Mohammad Zoynul Abedin), u.sivarajah@bradford.ac.uk (Uthayasankar Sivarajah)

and an XGBoost classifier achieves the best results, while the highest cost savings can be achieved by combining random under-sampling and XGBoost methods. This study has therefore financial implications for organizations to make appropriate decisions regarding the implementation of effective fraud detection systems.

Keywords: mobile payment, fraud detection, machine learning, imbalanced data, outlier detection

1. Introduction

Mobile payment transactions are carried out using mobile phone technologies that allow users to deposit, withdraw, spend, transfer and send money. There are nearly three hundred mobile payment services worldwide, which are particularly popular in Sub-Saharan Africa and Asia. In 2020, mobile payment transactions totaled $767 billion, conducted by approximately 1.2 billion registered users according to Statista. In addition, mobile payments have reportedly enormous potential during the COVID-19 pandemic, as it can greatly increase the promptness and efficiency of money transfers while minimising the necessity of face-to-face contact with bank and government staff (Blumenstock, 2020).

Recent mobile payment case studies (Iman, 2018; Jocevski et al., 2020; Verkijika, 2020) suggest that mobile payment systems have been challenged by several types of factors that have emerged in the context of advances in financial technology. Commercial and technical factors have been identified as particularly important to their future growth. As regards the first group of factors, the need to increase cost efficiency is particularly emphasised because most mobile payment transactions in developing countries are low value but high volume (Franque et al., 2022). Technical factors include, in particular, security concerns, as the legal frameworks and enforcement mechanisms are often inadequate in developing countries (Akanfe et al., 2020; David-West et al., 2022; Pal et al., 2020). To deploy a mobile payment system, it is therefore necessary to minimise fraud in order to increase customer trust and security, as reported in existing mobile payment acceptance models (Chin et al., 2022; Jia et al., 2022; Kar, 2021; Pal et al., 2021).

The increasingly growing use of mobile payments has boosted the chances of criminals committing mobile phone fraud in an illegal effort to circumvent security measures of mobile payment services. There is consequently a lot

of pressure to investigate potential security threats that may be exploited, with the ultimate aim of preventing fraud on a mobile payment service and developing countermeasures against attacks (Chen et al., 2021; Lopez-Rojas et al., 2016; Rieke et al., 2013). Early detection of fraudulent transactions is a key task in this effort. Recent developments in mobile payment services have therefore heightened the need for automated detection systems that enable immediate detection and prevention of fraudulent transactions.

The main challenges currently facing researchers involved in detecting fraud in mobile payment transactions include: (1) extreme class imbalance (only a small proportion of customers have fraudulent intentions); (2) changing patterns of fraud over time (fraudsters are always looking for new ways to bypass systems and commit crimes); and (3) inadequate selection of performance metrics. The consequence of the first challenge is a poor user experience for legitimate customers, as the detection of fraudsters usually also implies rejecting some legitimate mobile payment transactions. The second challenge usually leads to a decrease in the performance and efficiency of the detection model. Therefore, machine learning models must be constantly updated, otherwise they will not meet their objectives. Regarding the last challenge, in some cases the providers of mobile payment systems should prefer a higher false positive rate in exchange for a lower false negative rate and vice versa. But how to choose the right ratio between these two errors remains a challenging area in the field of fraud detection in mobile payment transactions.

A relatively high detection accuracy was reported in earlier research by using both traditional supervised learning methods (Choi and Lee, 2017, 2018) and deep learning-based methods (Mubalaike and Adali, 2018; Xenopoulos, 2017). However, a major problem with this kind of application is the extreme class imbalance of transactions, with a considerable dominance of legitimate transactions in the data. This in turn leads to a poor classification performance on the minority class of fraudulent transactions. To address this issue, two approaches have been utilized. The first approach relies on under-sampling methods used to generate a balanced dataset (Pambudi et al., 2019). The main limitation of this approach is the loss of potentially important information stored in discarded legitimate transactions, which can reduce detection accuracy. Alternatively, an attempt has been made to isolate fraudulent transactions in an unsupervised fashion (Buschjäger et al., 2021), inspired by outlier detection methods. Nevertheless, a comprehensive evaluation of machine learning methods is not yet available in the literature.

Moreover, little is known about how the two approaches can be integrated to improve the detection performance. To overcome the above problems, here we propose to enhance the performance of eXtreme Gradient boosting (XGBoost), a state-of-the-art machine learning method, by including a data sampling component addressing the issue of extreme class imbalance of mobile payment transactions.

In many financial applications it is necessary to filter out unusual observations to ensure the reliability of the system and prevent attempts to maliciously use it. This is particularly useful for detecting financial fraud attempts, as their behaviour patterns differ significantly from normal financial transactions (Bernard et al., 2021). Outlier detection methods are capable of processing all available data in real time to uncover patterns that evade traditional supervised learning methods. By doing so, organised crime groups can be identified with higher accuracy and less false positives. Outlier detection methods have indeed proved effective for detecting credit card fraud detection (Carcillo et al., 2021), online banking fraud detection (Carminati et al., 2015), and health insurance fraud detection (Yamanishi et al., 2004). Overall, however, there has been limited use of these methods to detect financial fraud, although some review studies suggest that they deserve more attention because the detection performance of supervised algorithms is negatively affected by the inherently heavily imbalanced class distribution of financial fraud data (Ngai et al., 2011). The scarce use of outlier detection methods can be attributed to the difficulty of detecting fraudulent behaviour (e.g., abnormal frequency of transactions or spending behaviour) when overlapping with legitimate behaviour in datasets contaminated with outliers and noise. Moreover, several other challenges have been identified that make it the difficult to detect outliers in the financial domain. First, efficient general purpose outlier detection methods are lacking because an outlier detection method in one fraud domain may not be appropriate for other scenarios, as legitimate and fraudulent behaviour is different from domain to domain (Ahmed et al., 2016). Second, unsupervised learning is preferred as sufficient labelled data for building models are rarely available. Third, legitimate behaviour may change over time, and fraudsters try to make their activities look legitimate. To take advantages of both supervised machine learning and outlier detection methods, for the first time, we propose a semi-supervised ensemble fraud detection model combining unsupervised outlier detection and supervised XGBoost methods that exploit all transactions contained in a large, highly imbalanced mobile payment transaction dataset.

Finally, financial implications of fraud detection methods in mobile payment transactions have also been neglected in earlier research. Therefore, our third contribution is to propose a novel performance measure of cost savings that takes into account the financial implications of false positive and false negative rates of fraud detection systems. Using the PaySim dataset, our findings provide evidence for the effectiveness of both XGBoost leveraged by an under-sampling class-balancing procedure and extreme gradient boosting outlier detection (XGBOD), thus providing important tools to support operation and management of mobile payment services.

In summary, the contributions of this study are threefold:

1. Developing a novel fraud detection framework for mobile payment systems by integrating the XGBoost method with class-balancing adjustments and unsupervised outlier detection methods, making it suitable for detecting fraud in a typical class-imbalanced mobile payment scenario.
2. Proposing a novel cost savings measure to evaluate the performance of mobile payment fraud detection systems. Unlike the traditional performance measures, the proposed measure considers both the cost savings from the correct detection of fraudulent transactions and the decrease in the margin for the transactions incorrectly identified as fraudulent.
3. Using the benchmark PaySim dataset of more than 6 million mobile payment transactions, we demonstrate that the proposed fraud detection framework not only outperforms state-of-the-art fraud detection methods in terms of detection accuracy but also generates substantial financial savings to the providers of mobile payment systems.

The remainder of this paper is organized as follows. Section 2 reviews the related work on fraud detection in mobile payment transactions with respect to data sources, methods used and performance achieved in earlier studies. Section 3 outlines the proposed fraud detection framework. Section 4 provides the results of the evaluation on the PaySim dataset, robustness check, and financial implications. Section 5 concludes with providing some possible directions for future research.

2. Fraud Detection in Mobile Payment Systems – Literature Review

A considerable amount of literature has been published on financial fraud detection, see West and Bhattacharya (2016) for a review and Hajek and

Henriques (2017) for a comprehensive evaluation of financial fraud detection methods. Risk factors of financial fraud were investigated, indicating that pressure / incentive to commit fraud is the most important risk factor (Huang et al., 2017). Related studies can be broadly categorized according to the financial fraud type as follows (Onwubiko, 2020): (1) account takeover fraud, (2) payment fraud, and (3) application fraud. Onwubiko (2020) also identified four main fraud channels, namely physical, web, telephony, and mobile. Frauds in mobile payment transactions have increasingly been recognized as a major concern in finance due to recent developments in mobile payment services (Chen and Sivakumar, 2021). Therefore, security requirements must be met to address security issues related to mobile payment transactions, such as mobile malware and SMS-based attacks (Kang, 2018). Heterogeneous software and hardware mobile platforms make the security problems more challenging (Li and Clark, 2013).

Regarding the data used in previous studies and summarized in Table 1, the lack of real-world datasets has been identified as a major problem in the application domain. Therefore, most earlier research tended to generate simulated synthetic data based on features captured from real-world fraud and legitimate transactions. To do so, Rieke et al. (2013) extracted payment laundering patterns from real-world events. However, the number of instances was insufficient for efficient fraud detection, as indicated by relatively low false negative (legitimate) rates in early studies (Coppolino, 2015; Rieke et al., 2013). Considerable progress has been made by introducing the PaySim financial simulator (Lopez-Rojas et al., 2016, 2018) that resembles normal mobile transactions and injects fraudulent behaviour to produce a larger number of financial frauds. Agent-based simulations and statistical analysis confirmed that the simulated data are as prudent as the original aggregated anonymized real data, thus, representing an optimal control environment for fraud detection in mobile payment transactions. By leveraging the PaySim data, Lopez-Rojas and Barneaud (2019) demonstrated their advantages over the relatively small real-world dataset. In addition, the simulated data retained the transactions and causal dynamics of the original data. It should be however noted that by preserving the statistical properties of the real-world data, the high class imbalance in favour of legitimate transactions is also maintained in the simulated dataset.

Traditional machine learning methods with supervised or unsupervised learning are not effective in handling extreme class imbalance in the data. Although a relatively high overall accuracy was reported in several studies,

these methods performed well only in terms of majority (legitimate) class accuracy (Choi and Lee, 2017, 2018; Du et al., 2018; Zhou et al., 2018). This holds also for more recent deep learning models, such as deep belief networks (Xenopoulos, 2017) and restricted Boltzman machines (Mubalaike and Adali, 2018). To overcome this major limitation, class imbalance was first approached by using under-sampling methods and then machine learning methods were trained on the balanced dataset (Pambudi et al. 2019). Similarly, Xenopoulos (2017) used under-sampling to produce balanced bootstraps for ensemble learning, and Misra et al. (2020) and Schlör et al. (2021) applied it to generate balanced training data for deep learning-based detection models. The main drawback of the under-sampling approach is that potentially useful instances are often excluded from the training data, which can significantly degrade the detection accuracy. Alternatively, isolation-based approaches were used to approximate the data distribution and build a generative model using mixture components. This outlier detection method was successfully applied to fraud detection by Buschjäger et al. (2021).

However, a comprehensive evaluation of state-of-the-art machine learning-based approaches exploiting under-sampling methods for handling class imbalance problem, is lacking in the literature. Hybrid semi-supervised methods taking advantage of supervised learning and unsupervised outlier detection methods have also been overlooked. Finally, only standard performance measures have been used to evaluate fraud detection performance in mobile payment systems, thus neglecting the financial implications of fraud detection.

3. Fraud Detection Framework

The proposed framework for fraud detection in mobile payment systems is presented in Fig. 1. The proposed fraud detection models are aimed to take advantage of XGBoost while overcoming the problem of extremely imbalanced classes in mobile payment transaction data. We will demonstrate that this approach is not only more accurate than supervised machine learning and outlier detection methods used in existing studies but that our approach is also more profitable in terms of the proposed cost savings measure.

3.1. Proposed Fraud Detection Models

This section outlines two fraud detection models proposed in this study. First, the eXtreme Gradient boosting (XGBoost) method, augmented with

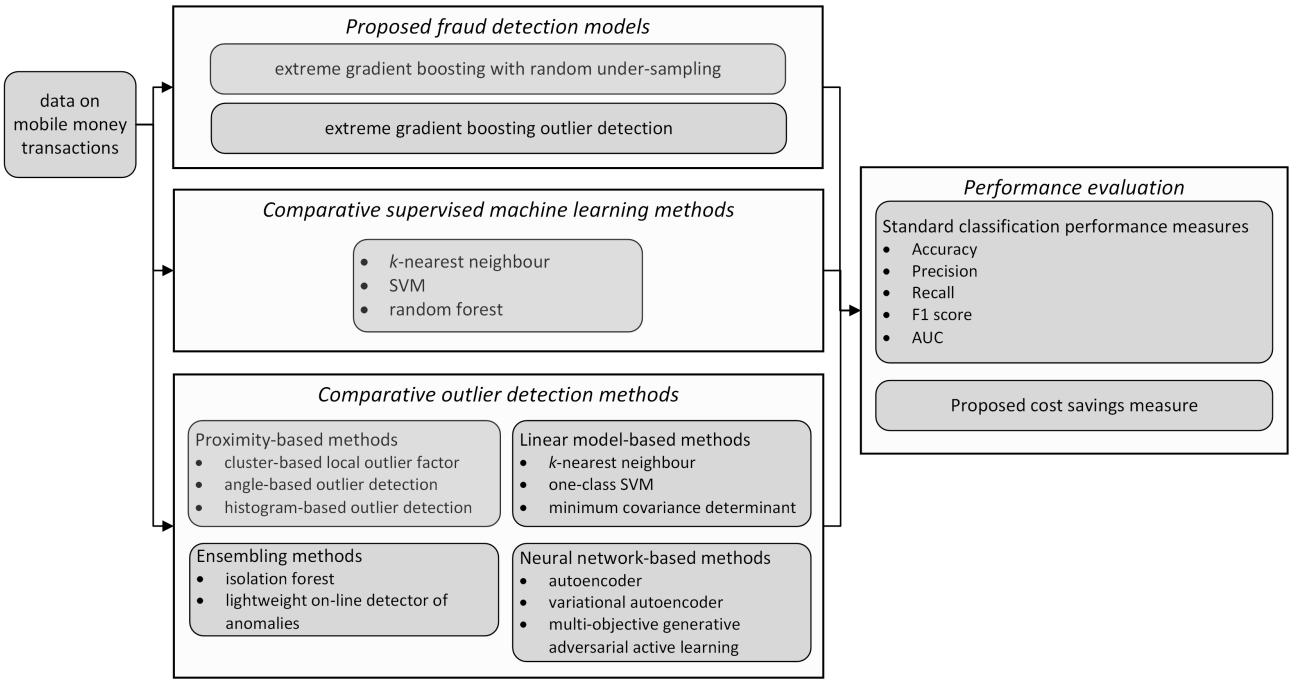

Figure 1: Fraud detection framework

Table 1: Summary of data and methods used in previous studies

Study	**Data (# fraud / legitimate)**	**Method**	**Performance**
Rieke et al. (2013)	synthetic logs (20/5,297)	predictive security analyser	FNR=0.550
Coppolino et al. (2015)	synthetic logs	Dempster-Shafer theory	FNR=0.240
Xenopoulos (2017)	PaySim (492/284,315)	ensemble of deep belief networks	Acc=89.05, AUC=0.961
Choi and Lee (2017, 2018)	Korean payment data (2,402/274,670)	unsupervised (EM, K-means, Farthest-First, X-means, MakeDensity), supervised (NB, SVM, LR, OneR, C4.5, RF)	Acc=99.97
Mubalaike and Adali (2018)	PaySim (8,213/6M)	restricted Boltzman machines	Acc=91.53
Du et al. (2018)	PaySim (8,213/6M)	SVM with LogDet regularization	Acc=97.57, AUC=0.978
Zhou et al. (2018)	Chinese bankcard enrolment (5,753/∼52M)	GB DT, LR, RF, rule-based expert	$Precision$=50.83, $Recall$=0.25
Pambudi et al. (2019)	PaySim (4,093/246,033)	RUS+SVM	$F1$=0.900, AUC=0.880
Misra et al. (2020)	PaySim (492/284,315)	Autoencoder+MLP	Acc=0.999, $F1$=0.827
Mendelson and Lerner (2020)	PaySim (8,213/6M)	cluster drift detection	AUC=0.898
Turner et al. (2021)	Bitcoin blockchain transactions	DeepWalk network analysis	-
Schlör et al. (2021)	PaySim (8,213/6M)	deep MLP with ReLU and iNALU	$F1$=0.880, AUC=0.960
Buschjager et al. (2021)	PaySim (269/572K)	generalized Isolation Forest	AUC=0.821
This study	PaySim (8,213/6M)	RUS+XGBoost, XGBOD	

Legend: Acc – accuracy, AUC – area under ROC curve, DT – decision tree, EM – expectation-maximization, $F1$ – $F1$-score (average of precision and recall), FNR – false negative (legitimate) rate, GB – gradient boosting, LR – logistic regression, MLP – multilayer perceptron, NB – Naïve Bayes, PNN – probabilistic neural network, RF – random forest, SVM – support vector machine, XGBOD – extreme gradient boosting outlier detection, and XGBoost – eXtreme Gradient boosting.

random under-sampling, is introduced to leverage both the supervised learning capability and robustness of XGBoost, a state-of-the-art machine learning method, and the data sampling component to overcome the class imbalance problem inherent in mobile payment transaction data. The second model exploits the extreme gradient boosting outlier detection (XGBOD) method, a semi-supervised algorithm that improves the performance of the XGBoost method on highly imbalanced mobile payment transaction data by introducing outlier scores obtained from multiple unsupervised outlier detection methods.

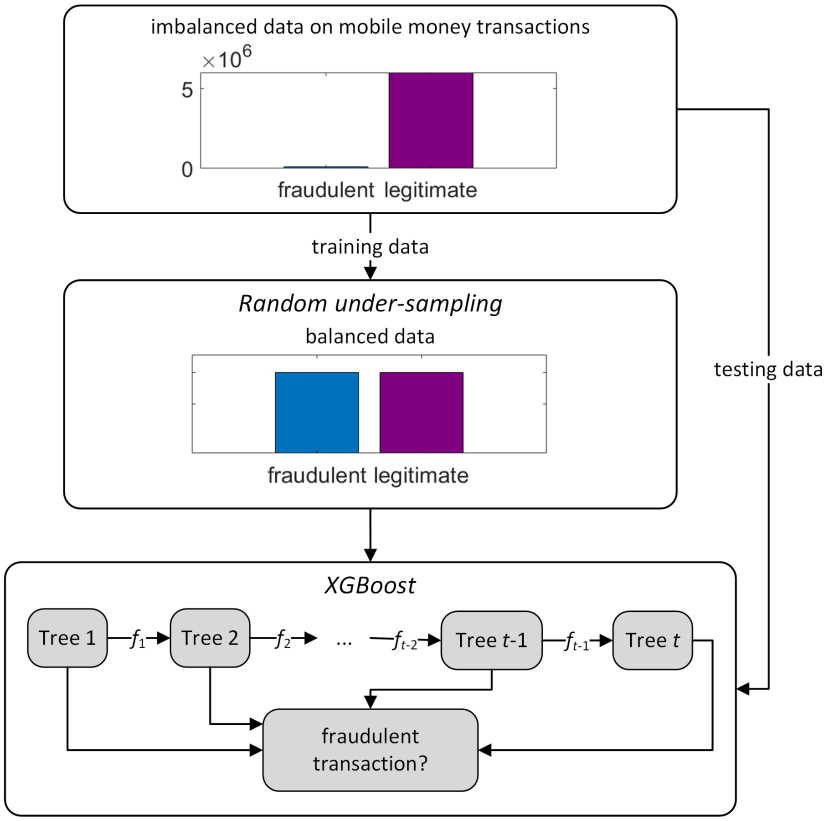

Figure 2: Flowchart of RUS-XGBoost for fraud detection

3.1.1. Extreme gradient boosting with random under-sampling

The proposed RUS+XGBoost integrates the random under-sampling (RUS) method with XGBoost, as depicted in Fig. 2. The RUS component is first used to generate balanced training samples, and XGBoost then generates additive models to produce the final prediction on whether the mobile payment transaction is fraudulent or not.

Under-sampling for handling class imbalance problem

The extremely high imbalance between legitimate and fraud classes makes detecting financial fraud a challenge (Du et al., 2018). Considering the importance of class imbalance in financial fraud detection, numerous methods have been used to improve the classification performance of supervised learning methods. In the related literature (Pambudi et al., 2019), data-level solutions have been particularly successful because they allow to address the imbalance

problem before training machine learning methods. In addition, data-level methods integrated into classifier ensembles appear to be particularly effective (Galar et al., 2012). From the data-level methods, over-sampling methods create artificial instances in the minority class to balance the training data. However, this can lead to problems of overfitting and overgeneralization as instances of the majority class are ignored. Moreover, given the gradual increase in data on financial fraud, under-sampling methods should be a better choice than their over-sampling counterpart.

The RUS method used in this study enables controlling for the number of samples selected from the original data. RUS is a non-heuristic method that randomly selects a data subset from the majority class, which is computationally effective and enables sampling heterogeneous data (Haixian et al., 2017).

Extreme gradient boosting

XGBoost is a computationally efficient and scalable implementation of gradient boosted decision trees that build additive models in a stepwise fashion. The overall error is minimized incrementally by introducing additive models based on the errors obtained in the previous steps. This results in an ensemble of base learners with better prediction ability than the individual classifiers. This is achieved by gradually improving the accuracy, low tree depth and equal contribution of the base learners to the final combined model. To further improve robustness to noise and overfitting, gradient boosting was augmented with a random sampling scheme (stochastic gradient boosting). XGBoost is an enhanced implementation with a more regularized model to control overfitting. The objective function of XGBoost to be minimized is given as follows (Chen and Guestrin, 2016):

$$obj^{(t)} = \sum_{i=1}^{n} (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{t=1}^{T} \Omega(f_t), \tag{1}$$

where y_i is the target value of the i-th instance, $\hat{y}_i^{(t)}$ is its predicted value at the t-th iteration, $f_t(x_i)$ is the additive decision tree model greedily added to improve the model performance, and $\Omega(f_t)$ is a regularization term penalizing the model complexity. The goal of this regularization procedure is to compress the weights for many features to zero to perform feature selection, which is advantageous when dealing with high-dimensional data. Therefore, XGBoost is currently one of the best performing classifiers across domains and has been successfully applied to insurance fraud detection (Dhieb et al.,

2019).

3.1.2. Extreme gradient boosting outlier detection model

The XGBOD method (Zhao and Hryniewicki, 2018) is a semi-supervised ensemble algorithm integrating multiple unsupervised outlier detection algorithms and an XGBoost classifier, as illustrated in Fig. 3. First, unsupervised methods are used to obtain data representations in terms of transformed outlier scores (TOS). Second, a feature selection method is used to reduce the TOS feature space so that only relevant TOS are retained. Then, the outlier score matrix is combined with the original features to produce a combined feature space. An improved feature space is thus generated, and the XGBoost classifier is used in this feature space to produce the final outlier scores for each mobile payment transaction. The advantage of this approach is its good predictive ability, which is due to its robustness to overfitting and data imbalance.

In the proposed XGBOD-based fraud detection model, a variety of unsupervised outlier detection methods (presented in subsection 3.2.2) are used to produce the TOS features. To maintain the balance between their diversity and accuracy, the balance selection algorithm (Zhao and Hryniewicki, 2018) is used to perform TOS selection. This algorithm applies a discounted accuracy function $\Psi(TOS_i)$ to pick the subset of p most relevant TOS. The function is defined as follows:

$$\Psi(TOS_i) = \frac{AUC_i}{\sum_{i,j=1}^{k} \mid \rho(TOS_i, TOS_j) \mid}, \tag{2}$$

where AUC_i is the AUC performance of the i-th outlier detection method, and $\rho(TOS_i, TOS_j)$ denotes the Pearson correlation coefficient between a pair of TOS.

3.2. Machine Learning Methods for Comparative Evaluation

In this section, we present the machine learning methods used for comparative evaluation in detecting fraud in mobile payment transactions. The methods can be broadly divided into (1) machine learning methods with supervised learning that address the class imbalance problem typical for financial fraud detection data, and (2) outlier detection methods.

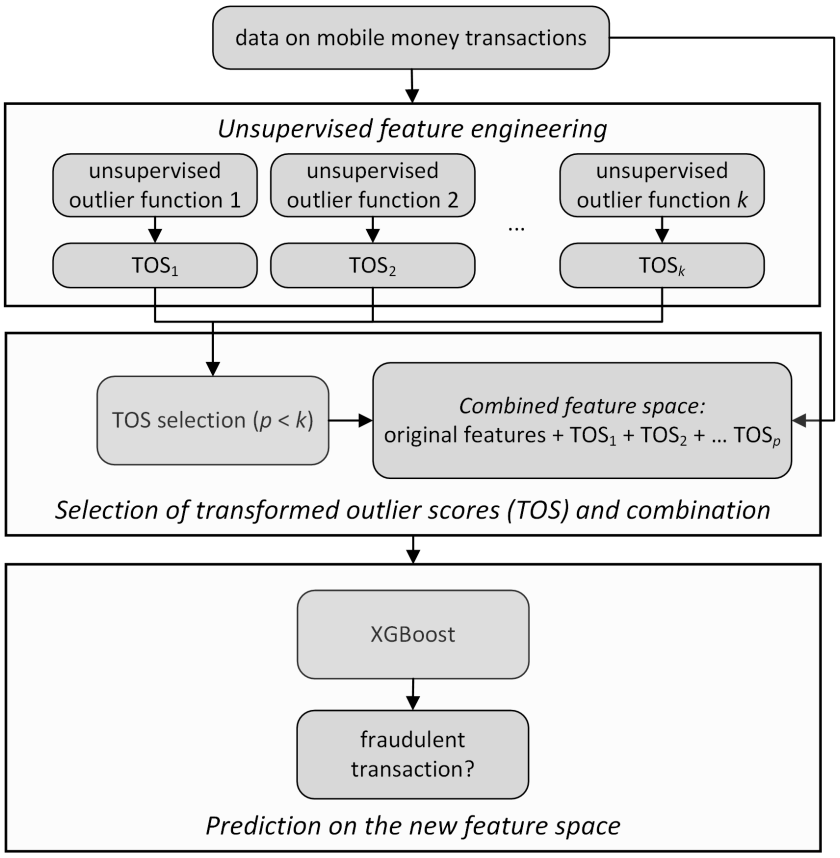

Figure 3: Flowchart of XGBOD for fraud detection

3.2.1. Supervised learning methods for imbalanced data

k-nearest neighbour classifier

The k-nearest neighbour (k-NN) method is an instance-based non-parametric classifier that uses training instances for comparison purpose. An instance is classified considering its k most-similar instances (typically in terms of Euclidean distance) using a majority vote. This simple approach proved to be accurate in a comparative analysis of machine learning methods for highly imbalanced credit card fraud detection (Awoyemi et al., 2017). In financial fraud detection, it is assumed that fraud instances are far from the samples of the legitimate class. Therefore, k-NN can be effectively used even in unsupervised outlier detection mode (Ramaswamy et al., 2000).

Support vector machine

SVM is a particularly effective classifier for financial fraud detection due to its capacity to deal with high-dimensional data (Du et al., 2018; Pambudi et al., 2019; Seera et al., 2021). The SVM algorithm aims to find the optimal separating hyperplane that maximizes the margin between instances from different classes. The decision boundary is represented by a subset of the data known as support vectors. Finding the parameters of the hyperplane is an optimization problem that takes into consideration both, minimizing the training error and maximizing the margin. To handle nonlinear relationships in the data, kernel functions (e.g., linear, polynomial or radial basis functions) are employed to map the classification problem from the original feature space to a new feature space of higher dimension where linear separation is possible.

Random forest

Random forest (RF) integrates multiple decision tree predictors trained independently on different data samples. This allows to generate a number of trees, ensuring that the generalization error converges to a certain limit. Another major advantage of RF is its non-differentiable decision boundary. In addition, random feature selection is used to split the nodes in each tree, making the RF classifier more robust to noise. The application of RF in financial fraud detection is particularly effective when the class distribution is imbalanced because its hierarchical structure enables learning patterns from both classes (Nami and Shajari, 2018). These advantages explain the good performance of RF on financial fraud detection tasks (Zhou et al., 2018).

3.2.2. Outlier detection methods

Outlier detection is typically conducted using unsupervised machine learning methods. The methods presented in this section are trained to represent

the legitimate data using clusters of similar data observations. Then, an unseen instance is assigned a score that is compared to a threshold representing the decision boundary separating legitimate instances from outliers.

The evaluation conducted in this study contains four types of outlier detection methods, namely (1) proximity-based methods, (2) linear model-based methods, (3) ensembling methods, and (4) neural network-based methods.

Proximity-based methods

To detect outliers, proximity-based methods investigate the neighbourhood of each data instance. For example, the local outlier factor (LOF) method (Breunig et al., 2000) uses the Euclidean distance between the data instance and its closest neighbour to obtain an outlier score. In the k-NN method (KNN) (Ramaswamy et al., 2000), a partition-based algorithm is first used to identify candidate partitions containing outliers, and then the distances of instances from these partitions are calculated to detect outliers. An important advantage of proximity-based methods is their independence of the data distribution. In other words, no a priori knowledge about the data distribution is required. However, these methods usually do not scale well for high-dimensional data. To reduce the sensitivity of LOF to the curse of dimensionality, the cluster-based local outlier factor (CBLOF) method (He et al., 2003) replaces closest neighbours with closest clusters, and the angle-based outlier detection (ABOD) method (Kriegel et al., 2008) replaces distances with the angular radius and variance of each data vector. The histogram-based outlier detection (HBOS) method assumes independence of features to score instances in linear time and is thus computationally more efficient compared to nearest-neighbour-based methods. However, HBOS fails in detecting local outliers because the density estimation produced by histograms does not allow modelling local outliers.

Linear model-based methods

Linear model-based methods rely on the construction of decision boundary separating instances in the legitimate class from the rest of the input data space. The one-class SVM (OCSVM) method (Schölkopf et al., 2000) constructs a separating hyperplane in high-dimensional space by minimizing the structural risk to capture regions of data belonging to the legitimate class. To prevent overfitting, this method allows a certain percentage of data instances (regularization parameter) to fall outside the separation boundary. The minimum covariance determinant (MCD) method (Hardin and Rocke, 2004) combine a multivariate location and scale estimator with a ro-

bust clustering algorithm so that the determinant of the covariance matrix is minimized for each cluster. This method is first trained to fit a minimum covariance determinant model and then the outlier score is calculated using the Mahalanobis distance. However, problems can arise when clusters overlap significantly, leading to poor convergence of the algorithm.

Ensembling methods

Isolation Forest (Liu et al., 2008) aims to separate outliers from the rest of the data samples. To calculate an isolation score for the data instances, random forest is employed. The method assumes that outliers are susceptible to isolation and, therefore, can be isolated closer to the root of the tree. Specifically, the average path length from the root of the trees can be used obtain the isolation score. Isolation trees are thus able to build sub-models on different data samples while maintaining low computational complexity and the ability to scale to handle large volumes of data and high-dimensional problems. Similarly, lightweight on-line detector of anomalies (LODA) comprises a collection of weak learners represented by one-dimensional histograms approximating probabilities of random data projections. The use of sparse projections makes LODA robust to both the large number of samples and missing data, allowing the detection of anomalous samples in real-time (Pevny, 2016).

Neural network-based methods

Neural network-based methods utilize feature learning to reduce dimensionality. An autoencoder is an unsupervised neural network capable of nonlinear dimensionality reduction and reproducing input data vectors. Sakurada and Yairi (2014) showed that autoencoder (AE) can be successfully applied to outlier detection. To detect outliers in financial fraud, AEs can be trained to learn legitimate behaviour and compute a reconstruction error representing the outlier score (Sakurada and Yairi, 2014). To achieve robustness in learning disentangled representations, variational autoencoder (VAE) was proposed that utilizes both the joint data distribution and their latent generative factors (Burgess et al., 2018). VAE represents a probabilistic graphical model whose posterior distribution is estimated using a neural network. The outlier score of VAE is calculated as the reconstruction probability. Recently, generative adversial networks (GANs) have been deployed to unsupervised outlier detection. Specifically, multi-objective generative adversarial active learning (MO-GAAL) uses GANs to sample informative potential outliers following a mini-max game between a discriminator and a generator (Liu et al., 2019). Thus, GANs assist the discriminative algorithm in finding a boundary that can effectively separate fraudulent outliers from legitimate

normal data. This has been exploited in several studies on financial fraud (Sethia et al., 2018; Delecourt and Guo, 2019).

3.3. Performance Evaluation

In many related studies (Du et al., 2018; Misra et al., 2020; Mubalaike and Adali, 2018), the ratio of correctly classified transactions to the total number of transactions (i.e., accuracy) has been used as the evaluation measure. However, in the scenario of class-imbalanced data, this measure fails to detect well the model performance for the minority (fraud) class.

As noted in previous research (Lopez-Rojas and Barneaud, 2019), an inherent problem in detecting financial fraud that needs to be addressed is the unknown distribution and impact of all fraudulent transactions. In the absence of an adequate measure of fraud detection performance, existing fraud detection approaches rely on traditional measures of classification performance. The most desirable performance measure is the ability to correctly identify fraudulent transactions (true positive rate). In addition, minimizing false positive and false negative transaction rates (see confusion matrix in Table 2) is also a desirable quality of fraud detection systems, especially in a changing fraudulent environment. Here, we use these standard classification measures to evaluate the performance of fraud detection models. The true positive rate (*Recall*) is defined as the number of transactions correctly identified as fraudulent as a percentage of all fraudulent transactions as follows:

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

where TP and FN are the numbers of true positive and false negative transactions. The false positive rate (FPR) is the number of transactions incorrectly identified as fraudulent as a percentage of all legitimate transactions:

$$FPR = \frac{FP}{FP + TN}, \tag{4}$$

where FP and TN are the numbers of false positive and true negative transactions. The false negative rate (FNR) is the number of transactions incorrectly identified as legitimate as a percentage of all fraudulent transactions:

$$FNR = \frac{FN}{TP + FN} = 1 - Recall. \tag{5}$$

Table 2: Confusion matrix for fraud detection

Prediction/Target	**Positive**	**Negative**
Positive (fraudulent transaction)	TP	FP
Negative (legitimate transaction)	FN	TN

In reality, financial institutions try to reduce the risk of fraud while trying to comply with regulations, but *Recall* is difficult to estimate in the real world because FN is unknown (hidden fraud). Therefore, financial institutions can only calculate *Precision* (i.e., the number of transactions correctly identified as fraudulent as a percentage of all transactions that are expected to be fraudulent) (Lopez-Rojas and Barneaud, 2019):

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

Previous studies have also considered the $F1$ measure (Pambudi et al., 2019; Schlör et al., 2021), defined as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{7}$$

The area under the receiver operating characteristic curve (AUC) has also been used as a more appropriate measure for fraud detection in mobile payment transactions due to its robustness to imbalanced data (Buschjäger et al., 2021; Mendelson and Lerner, 2020). AUC can be defined as the probability that a fraud detection model ranks a randomly selected fraudulent transaction higher than a randomly selected legitimate transaction, as follows:

$$AUC = \int_0^1 Recall(T) \times \frac{d}{dT} FPR(T) dT, \tag{8}$$

where T is the cut-off point.

3.4. Cost savings measure

In addition to the traditional performance measures above, here we propose a measure of cost savings measure to account for the financial implications of fraud detection models. The proposed cost savings measure was inspired by profit-based loan default prediction systems, considering potential returns and losses (Papouskova and Hajek, 2019; Ye et al., 2018). On the

one hand, correct detection of a fraudulent transaction leads to the following cost savings:

$$CS_{TP} = \sum_{i=1}^{n}(TP_i \times A_i \times 3.36) - (TP_B \times A_F \times 3.36), \tag{9}$$

where CS_{TP} are cost savings from TP transactions, TP_i is the i-th transaction of TP, A_i is the amount of the i-th transaction, TP_B is the number of TP transactions detected by the reference fraud detection system, and A_F is the average amount of fraudulent transactions. We also took into account that fraud now costs financial institutions \$3.36 for every dollar lost to fraud and that the current average percentage of successful fraud attempts is 48% (i.e., TP_B=0.52)[1].

On the other hand, mobile transactions generate a revenue margin of 3.5% on average [4]. Therefore, we also considered the cost of FP transactions, estimated as the decrease in the margin for these transactions:

$$Cost_{FP} = (TN \times A_L \times 0.035) - \sum_{j=1}^{m}(FP_j \times AT_j \times 0.035), \tag{10}$$

where $Cost_{FP}$ is cost of FP transactions, FP_j is the j-th FP transaction, A_L is the average amount of legitimate transactions, and AT_j is the amount of the j-th transaction. The total cost savings CS_{total} is then calculated as:

$$CS_{total} = CS_{TP} - Cost_{FP}. \tag{11}$$

Note that the proposed measure is expressed in financial terms and is instance-dependent (with respect to the amount of each transaction), allowing for a direct interpretation by financial institutions.

4. Experimental Results and Analysis

4.1. Data

Consistent with most previous studies (Buschjäger et al., 2021; Du et al., 2018; Xenopoulos, 2017), we used the PaySim dataset[2] in this study. The main objective of the simulations performed by Lopez-Rojas and his research

[1]https://chainstoreage.com/study-fraud-costs-increased-73-year-over-year-us-retailers
[2]https://www.kaggle.com/ealaxi/paysim1

team (Lopez-Rojas et al., 2016; Lopez-Rojas et al., 2018; Lopez-Rojas and Barneaud, 2019) was to replicate typical fraud scenarios that have similar statistical characteristics to the original mobile payment transaction data. To this end, different types of fraudulent transactions were injected, including cash-in (increasing account balance), cash-out (withdrawing cash), payment (paying for goods or services), transfer (to another user) and debit (sending money to a bank account). PaySim simulated 743 time steps, representing thirty days of real-time data. To introduce fraudulent behaviour into the system, 1,000 fraudsters were included with a 3% probability of committing fraud at any time step. A total of 6,362,620 mobile transactions were involved in the dataset, of which 8,213 were fraudulent. Table 3 provides descriptive statistics of the dataset, and Fig. 4 shows the numbers and amounts of transactions in time steps.

We opted for this dataset for several reasons (Lopez-Rojas and Barneaud, 2019). First, real-time historical data do not include enough fraudulent transactions. Therefore, some previous studies have considered all abnormal transactions to be fraudulent (Choi and Lee, 2017). Second, privacy protections prevent companies from making datasets public. Third, fraudulent behaviour is adaptive, making it difficult to create sufficiently diverse real-world fraud data. In addition, a similar approach based on typical real attack scenarios was taken in studies related to online banking fraud detection (Carminati et al., 2015).

Table 3: Attributes in the PaySim dataset

Attribute	**Mean value / Range**
Step	1-743
Type of transaction	cash-out (35%), cash-in (34%), transfer and debit (31%)
Amount of transaction	180K
Customer name	6.35M unique values
Initial balance	834K
New balance	855K
Recipient name	2.72M unique values
Initial balance of the recipient	1.1M
New balance of the recipient	1.22M
Fraud	0 (legitimate 6.36M) / 1 (fraud 8.2K)

4.2. Experimental Setup

For data partitioning, we randomly created training and testing data with a 3:1 ratio (75% training data, 25% testing data). To ensure reliable performance evaluation, we repeated this process five times. Since the performance

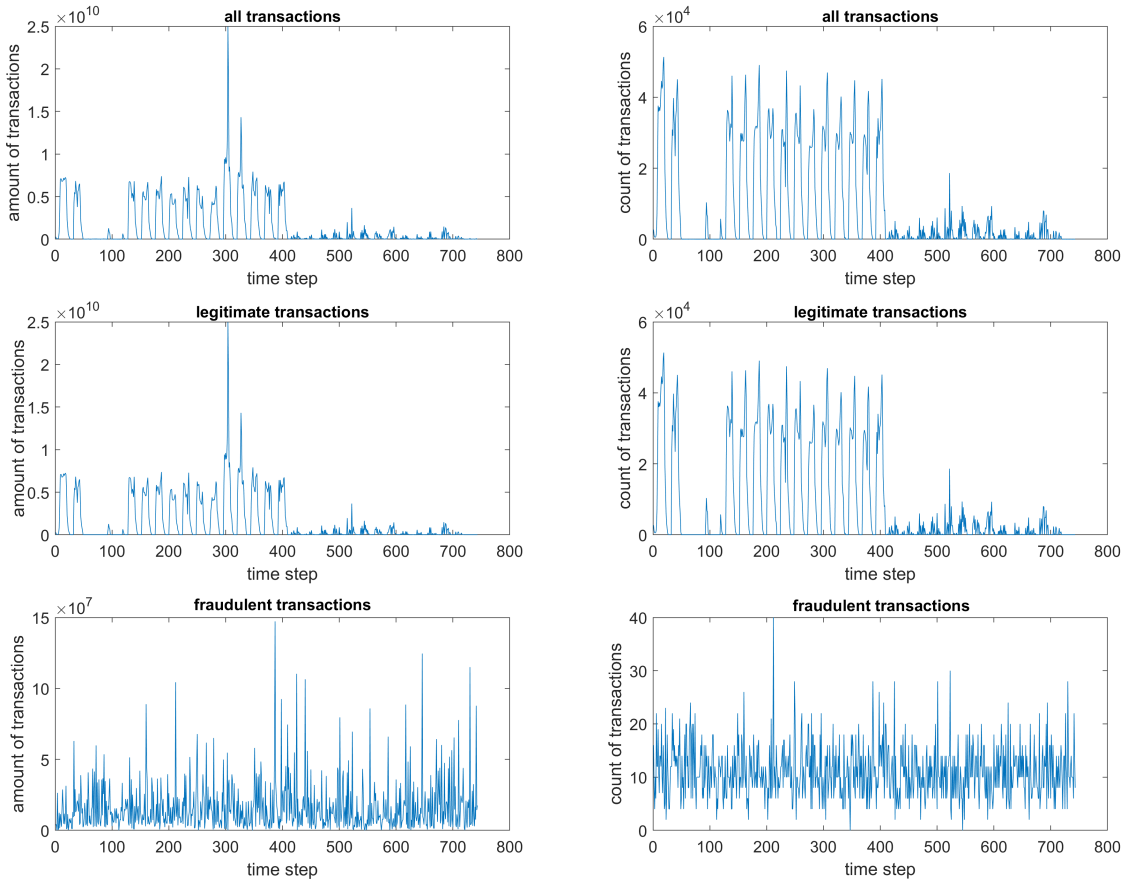

Figure 4: Visualization of amounts and counts of transactions in the PaySim dataset

of the fraud detection methods strongly relies on their hyperparameter selection, we then conducted their optimal selection using 5-fold cross-validation on the training data (for the list of hyperparameters and their values, see Appendix 1). Then, we performed fraud detection in mobile payment transactions using the above supervised learning and outlier detection methods. For experiments, we used the following implementations: (1) supervised learning methods in the Python library Scikit-Learn 0.23.0, (2) the RUS algorithm available in the library Imbalanced-Learn 0.6.2, and (3) the outlier detection methods available in the library PyOD (Zhao et al., 2019). The performance of the methods was evaluated using the measures defined in the following subsection.

4.3. Empirical Results

We performed empirical experiments using the PaySim dataset. This section consists of four subsections. First, we investigate the performance of supervised learning methods and the effect of random under-sampling on their effectiveness. Second, the performance of outlier detection methods is evaluated. Third, the financial consequences of the fraud detection models are evaluated. Finally, the robustness of the models is tested using a credit card fraud dataset.

4.3.1. Supervised learning methods

In the first set of experiments, we compared the performance of four supervised learning methods (XGBoost, k-NN, SVM, and RF), without using RUS, to obtain baseline performance. Table 4 shows the testing results of overall accuracy Acc, AUC, $F1$, $Precision$ and $Recall$. The values of performance measures were obtained as the average of five experiments. For each performance measure, the number in bold represents the best value among the tested methods. The non-parametric Wilcoxon test was performed on the performance measure values obtained in the five experiments to statistically compare the performance between the best performing method and the remaining methods. Significantly similar results at the 5% level with respect to AUC and $F1$ are marked with an asterisk.

In terms of accuracy, all the supervised learning methods used performed well. However, as noted above, the extreme class imbalance suggests that this evaluation measure is not as relevant in this case. As for the AUC measure, XGBoost was superior to the other methods, indicating a well-balanced performance for both legitimate and fraud classes. The good balance between

Precision and *Recall* caused XGBoost to achieve the best results also in terms of $F1$ measure. By contrast, SVM and k-NN performed well only with respect to Precision and Recall, respectively, making them unsuitable methods for fraud detection in mobile payment transactions. Overall, these results indicate that only XGBoost without class-balancing adjustment is suitable for detecting fraud in such a class-imbalanced scenario.

Then, we investigated the effect of the RUS under-sampling procedure on the performance of the supervised learning methods. On the one hand, Table 4 shows that RUS greatly improved the values of AUC for SVM, RF and XGBoost. On the other hand, there was a considerable deterioration in $F1$, which can be attributed to the lower *Precision* achieved at the cost of higher *Recall*. In other words, RUS caused almost all fraudulent transactions to be detected, but this was accompanied by a substantial increase in the number of FP transactions. This resulted in a bias for the minority class while reducing the accuracy for the majority class. It is worth noting that we also experimented with other heuristic-based under-sampling methods, such as edited neatest neighbour and Tomek links, to address the class imbalance problem but without improvement in detection performance. Finally, it should be noted that the execution time (training time + testing time) was substantially reduced by using RUS. For example, RUS+XGBoost was computationally most efficient with 2.38 seconds compared to 207.02 seconds required for XGBoost without using RUS.

Table 4: Fraud detection performance of supervised learning methods

Method	***AUC***	***F1***	***Acc***	***Precision***	***Recall***	***Execution time [s]***
k-NN	0.9313	0.1588	0.9881	0.0873	0.8744	4,581.4
SVM	0.6543	0.4655	0.9991	**0.9474**	0.3086	12,082.9
RF	0.8961	0.8394*	0.9996	0.9146	0.7756	1,196.2
XGBoost	0.9350	**0.8410***	**0.9998**	0.8794	0.8059	207.0
RUS+k-NN	0.8996	0.0405	0.9475	0.0207	0.8516	145.3
RUS+SVM	0.8344	0.0321	0.9431	0.0164	0.7255	1,041.5
RUS+RF	0.9933*	0.2305	0.9914	0.1303	0.9947	12.6
RUS+XGBoost	**0.9955***	0.2812	0.9934	0.1637	**0.9976**	2.4

Notes: The best results are in bold, * statistically similar at 5% as the best performer in bold. The experiments were performed on Intel® Core™ i5-8400 CPU @ 2.8GHz, 32 GB RAM with six cores on a Windows 10 oper. system in the Python libraries Scikit-Learn 0.23.0 and Imbalanced-Learn 0.6.2.

4.3.2. Outlier detection methods

In the seconds run of experiments, the performance of XGBOD was evaluated compared with other outlier detection methods. Table 5 shows that XG-

BOD significantly outperformed the remaining methods in terms of AUC and $F1$. In addition, XGBOD was also dominant with respect to both *Precision* and *Recall*, indicating excellent performance on both classes.

Table 5: Fraud detection performance of outlier detection methods

Method	***AUC***	***F1***	***Acc***	***Precision***	***Recall***	***Execution time [s]***
ABOD	0.8353	0.0680	0.9953	0.0675	0.0685	2,646.5
CBLOF	0.8593	0.0822	0.9954	0.0829	0.0822	41.3
HBOS	0.7731	0.0077	0.9951	0.0078	0.0076	4.1
LODA	0.6818	0.1060	0.9954	0.1026	0.1096	14.8
Isolation Forest	0.8358	0.0189	0.9964	0.0307	0.0137	189.9
KNN	0.8618	0.1260	0.9957	0.1288	0.1233	1,948.5
MCD	0.7705	0.1084	0.9956	0.1087	0.1081	127.4
OCSVM	0.6732	0.0273	0.9951	0.0272	0.0274	802.9
AE#	0.8050	0.0869	0.9954	0.0870	0.0868	931.1
VAE#	0.8050	0.0869	0.9954	0.0870	0.0868	2,922.9
MO-GAAL	0.9071	0.6059	0.9980	0.5902	0.6225	13,184.4
XGBOD	**0.9958**	**0.8737**	**0.9994**	**0.9942**	**0.7793**	4,256.3

Notes: The best results are in bold, # The experiments were performed on GPU NVIDIA GeForce GTX 1060 6GB, 1280 cores on a Windows 10 oper. system in the Python library PyOD.

These results can be explained by the semi-supervised learning approach used in the XGBOD method. This is because, unlike other outlier detection methods, XGBOD leverages the labels assigned to mobile transactions. In addition, the transactions contained in the majority class of legitimate transactions are fully utilized by the multiple unsupervised outlier detection methods that produce outlier scores in XGBOD. The XGBoost algorithm applied in the improved XGBOD feature space exhibits good robustness to overfitting and data imbalance, and outperforms the supervised learning methods reported in Table 4 in terms of AUC and $F1$. However, it should be admitted that the drawback of XGBOD is the longer execution time, on average 4,256.25 seconds.

4.4. Financial Impact of Fraud Detection

To investigate the financial consequences of the evaluated fraud detection systems, we used the performance measures defined in Eqs. (9)-(11). Table 6 shows the average financial performance of all methods in terms of cost savings from TP transactions, cost of FP transactions and total cost savings. To calculate these results, we used the average amounts of fraudulent and legitimate transactions in the data, i.e., $A_F = 1{,}468{,}000$ and $A_L = 178{,}200$.

In general, supervised learning methods outperformed outlier detection methods in terms of overall cost savings, which can be attributed to the high

Table 6: Financial impact of fraud detection methods

Method	CS_{TP}*	$Cost_{FP}$	CS_{total}
k-NN	3,576.4	120.3	3,456.1
SVM	-2,135.4	218.3	-2,135.6
RF	2,575.1	923.1	2,574.2
XGBoost	3,630.7	380.5	3,630.3
RUS+k-NN	3,443.3	519.3	2,924.0
RUS+SVM	2,155.9	561.4	1,594.5
RUS+RF	4,903.3	85.8	4,817.5
RUS+XGBoost	**4,932.9**	65.9	**4,866.9**
ABOD	-4,556.9	23.5	-4,580.4
CBLOF	-4,418.7	22.6	-4,441.3
HBOS	-5,171.0	24.2	-5,195.2
LODA	-4,142.3	23.8	-4,166.2
Isolation Forest	-5,109.6	10.7	-5,120.3
KNN	-4,004.2	20.7	-4,024.9
MCD	-4,157.7	22.2	-4,179.7
OCSVM	-4,971.4	24.3	-4,995.7
AE	-4,372.6	22.6	-4,395.3
VAE	-4,372.6	22.6	-4,395.3
MO-GAAL	1,031.6	10.7	1,020.9
XGBOD	2,612.9	**0.1**	2,612.8

Notes: * amounts are given in mil. units of an African currency that could not be disclosed by data providers.

Recall values of supervised learning methods. Note that cost savings from TP transactions were considered to have a stronger financial impact on total cost savings compared to FP transactions. In contrast, XGBOD delivered the lowest costs associated with FP transactions, which is related to its high *Precision* performance. Surprisingly, SVM and unsupervised outlier detection methods used in previous studies (Buschjäger et al., 2021; Du et al., 2018) did not perform well in terms of financial impact and provided negative overall cost savings due to their low *Recall* values.

4.5. Comparison with State-of-the-art Methods

To further show the effectiveness of the proposed fraud detection model, the obtained AUC was compared with that of previous studies that examined the same dataset (Table 7). The best AUC performance thus far reported was achieved using SVM with LogDet regularization (Du et al., 2018). Our result in Table 4 obtained for SVM confirm that equipping SVM with LogDet regularization improves the AUC performance. Indeed, the traditional SVM method is reportedly sensitive to outliers and noisy data (Shajalal et al., 2021). Table 7 also shows that deep neural networks performed well in previous studies (Schlör et al., 2021; Xenopoulos, 2017). However, their

performance is limited by the relatively low number of fraudulent transactions in the dataset. By contrast, the worst performance was reported for the Isolation Forest method (Buschjäger et al., 2021). Note that the results for Isolation Forest obtained here (Table 5) are consistent with those from Buschjäger et al. (2021). The results in Table 7 suggest that the proposed XGBoost-based models perform better than those used in previous studies in terms of AUC, which can be attributed to their good scalability and efficient processing of sparse data.

Table 7: Comparison of fraud detection performance of the proposed XGBoost-based models with the results of previous studies

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

Notes: The best results are in bold.

4.6. Robustness Check on Bank Payment Datasets

To confirm the obtained performance evaluation, we checked the robustness of the considered fraud detection methods using a bank payment dataset. The BankSim dataset[3] (Lopez-Rojas and Axelsson, 2014) was generated using a multi-agent simulation based on a sample of transactional data from a Spanish bank. The dataset was validated using statistical techniques and social network analysis of customer-merchant relationships, thus approximating key features of real bank payment frauds. Each transaction was characterized by payment amount (in EUR), customer and merchant zip codes, customer gender and age, and merchant category (e.g., fashion, technology, transport, and travel). A total of 594,643 transaction records were included, of which 7,200 were fraudulent transaction. The simulation was run for 180 steps representing months. Thieves were injected to steal or clone an average of three credit cards at each step and conduct approximately two fraudulent transactions per day. The result of the simulation is depicted in Fig. 5.

[3]https://www.kaggle.com/ealaxi/banksim1

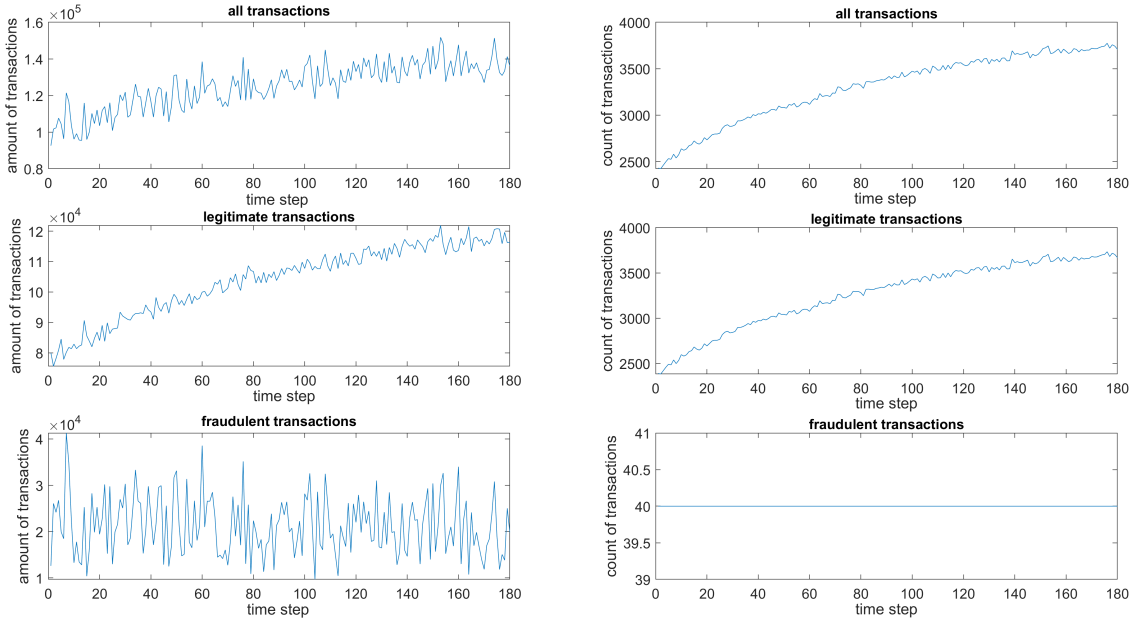

Figure 5: Visualization of amounts and counts of transactions in the BankSim dataset

The BankSim dataset provides a benchmark for detecting fraud in bank payment transactions, as several recent studies have shown (Cui et al., 2021; Vaughan, 2020). As a robustness check, we trained the evaluated models on the BankSim dataset using the same experimental setup as for the PaySim dataset. Note that the sampling process and data collection system was unique and heterogeneous for both dataset, which allowed us to verify the robustness of the tested fraud detection models. The results in Table 8 suggest that the under-sampling procedure is not as effective for smaller financial fraud datasets, improving the performance of supervised learning methods only in terms of AUC. In contrast, the performance of unsupervised outlier detection methods substantially improved compared to the large PaySim dataset, suggesting their poor scalability. Overall, XGBoost and XGBOD performed well in terms of both AUC and $F1$ measures, indicating their good robustness to data size.

4.7. Discussion

Prior studies (Buschjäger et al., 2021; Pambudi et al., 2019) have noted the importance of addressing the problem of extreme class imbalance in mo-

Table 8: Fraud detection performance on the BankSim dataset

Method	***AUC***	***F1***	***Acc***	***Precision***	***Recall***
k-NN	0.9466	0.2029	0.9089	0.1131	**0.9851**
SVM	0.7723	0.6849	0.9941	**0.9208**	0.5451
RF	0.8973	0.8145	0.9957	0.8332	0.7966
XGBoost	0.9112	**0.8391**	**0.9963**	0.8240	0.8240
RUS+k-NN	0.9454	0.3216	0.9535	0.1941	0.9371
RUS+SVM	0.9433	0.3379	0.9571	0.2065	0.9291
RUS+RF	0.9746	0.4674	0.9738	0.3073	0.9754
RUS+XGBoost	0.9774	0.4898	0.9760	0.3266	0.9789
ABOD	0.9852	0.5039	0.9877	0.5032	0.4995
CBLOF	0.9688	0.6072	0.9902	0.6069	0.6074
HBOS	0.9340	0.1490	0.9787	0.1488	0.1488
LODA	0.7272	0.0743	0.9770	0.0736	0.0736
Isolation Forest	0.9647	0.3932	0.9851	0.3974	0.3867
KNN	0.9894	0.5957	0.9899	0.5956	0.5961
MCD	0.9695	0.6922	0.9923	0.6928	0.6944
OCSVM	0.4431	0.0077	0.9758	0.0075	0.0075
AE	0.9350	0.3861	0.9848	0.3829	0.3813
VAE	0.9351	0.3863	0.9848	0.3829	0.3813
MO-GAAL	0.9367	0.3510	0.9807	0.3029	0.4173
XGBOD	**0.9968**	0.7893	0.9953	0.8018	0.7084

bile payment transactions. Therefore, our first set of experiments was designed to investigate the effect of under-sampling the majority class of legitimate transactions on the performance of supervised learning methods. Consistent with Pambudi et al. (2019), we observed that the detection performance improved for most of the machine learning methods, especially for the proposed RUS+XGBoost fraud detection model. In contrast to earlier findings (Buschjäger et al., 2021), however, the second set of experiments did not detect any evidence for the effectiveness of outlier detection methods. However, when conducted in a semi-supervised manner, the proposed XGBOD detection model was found to be superior even to the supervised learning methods. Finally, the financial consequences of the fraud detection models were examined to provide guidance on how to set up the right performance metrics for fraud detection in mobile payment transactions. This experiment addressed the need for an adequate measure of fraud detection performance as raised in recent research (Lopez-Rojas and Barneaud, 2019). We found that RUS+XGBoost performed best in terms of cost savings from correctly detecting fraudulent transactions, while XGBOD minimized the cost of false positive transactions.

Based on the experimental results of this study, we propose the following suggestions for mobile payment systems.

Firstly, the providers of mobile payment systems should pay more at-

tention to recent developments in the machine learning research. Specifically, XGBoost enhanced with class-balancing or outlier detection methods should be applied to effectively handle the extreme class imbalance problem in the data and accurately detect fraud in mobile payment transactions. RUS+XGBoost is particularly recommended for its low execution time, indicating its capability for real-time fraud detection.

Secondly, cost savings and transaction costs should be considered when implementing fraud detection systems in mobile payment systems. For fraud detection models in mobile payment systems, these evaluation metrics are critical due to the high costs associated with mobile payment default. The proposed cost savings measure can be used for this purpose as it offers providers appropriate guidance for making decisions on the selection cost-effective fraud detection systems.

The importance of accurate and cost-effective fraud detection systems has dramatically increased during the COVID-19 pandemic because many emerging and developing countries used mobile money transfer to provide COVID-19 aid (Blumenstock, 2020). Indeed, mobile payment systems provide a fast and scalable solution while complying with social distancing measures, which encouraged government-to-person mobile payments. To enable sustainable solutions for mobile money transfer, fraud detection technologies represent a critical component of the frameworks for sustainable government-to-person mobile money transfers proposed in response to COVID-19 (Davidovich et al., 2020).

Finally, our results suggest that unsupervised outlier detection methods are not appropriate for fraud detection in mobile payment transactions. The current study was unable to evaluate the use of the fraud detection system in a real environment because the number of labelled instances is insufficient in existing real-world data. Instead, we experimented with a controlled environment with fraudulent behaviour injected into the data to obtain a well-performing fraud detection system. However, we believe that the accuracy of the proposed fraud detection system would not deteriorate in real-world applications as the data used in this study are based on the real-world anonymized data. To further improve the detection accuracy and to assist the providers of mobile payment systems with the development of fraud detection systems, large labelled real-world data should be collected and made available to enable effective training of state-of-the-art supervised learning methods.

5. Conclusion

In this paper, we have proposed an XGBoost-based fraud detection framework while considering the financial impact of fraud detection. The findings from this study make several noteworthy contributions to the current literature. First, the XGBoost model was combined with under-sampling to effectively address the problem of extreme class imbalance and avoid overfitting. Second, to fully exploit the large amount of underlying data, unsupervised outlier detection methods were integrated into the XGBoost-based model. The comparison of the XGBoost-based fraud detection performance with various state-of-the-art machine learning methods confirmed that we have found a cutting-edge solution for fraud detection in mobile payment systems. Our findings also suggest a role for the proposed model in promoting cost savings of fraud detection systems. Taken together, our results strongly argue against a major role of single machine learning methods and unsupervised outlier detection methods in fraud detection of mobile payment transactions, implying that ensemble XGBoost-based methods are preferable.

In the future research, ensemble methods combined with alternative under-sampling and unsupervised outlier detection methods should be further investigated, including automatic optimization of outlier detection ensembles (Reunanen et al., 2020) and the XGBoost method enhanced with weighted and focal losses (Wang et al., 2020). Unfortunately, it was not possible to investigate our model's robustness against different mobile payment transaction data distributions due to privacy concerns and other limitations of existing datasets. Therefore, further data would be needed to evaluate model robustness, including testing the feasibility of transfer learning across multiple datasets. The proposed fraud detection models should also be applied to solving related fraud detection problems, such as credit card and loan frauds, which also exhibit class imbalance characteristics and large real-world datasets are available for these problems (West and Bhattacharya, 2016). Other possible application fields of the proposed model include credit scoring (default prediction) (Mahbobi et al., 2021), direct marketing (Wong et al., 2020), and customer churn prediction (Wong et al., 2020). An issue that was not addressed in this study was the interpretability property of the fraud detection models. Therefore, further research might explore the tradeoff between achieving a high detection accuracy while maintaining interpretability (Hajek, 2019). Finally, the current investigation was limited by the use of the cost savings measure only to evaluate the trained model, and thus not in

the objective function of the fraud detection model. Future research should therefore examine the performance of fraud detection models using the cost savings measure as the objective function. This could lead to our model delivering even greater cost savings to the end user.

Acknowledgements

This article was supported by the scientific research project of the Czech Sciences Foundation Grant No. 19-15498S.

Conflict of Interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Ahmed, M., Mahmood, A.N., & Islam, M.R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems, 55*, 278–288.

[2] Akanfe, O., Valecha, R., & Rao, H.R. (2020). Assessing country-level privacy risk for digital payment systems. *Computers & Security, 99*, 102065

[3] Awoyemi, J.O., Adetunmbi, A.O., & Oluwadare, S.A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. *IEEE International Conference on Computing, Networking and Informatics*, ICCNI 2017, IEEE, p. 1–9.

[4] Bansal, S., Bruno, P., Denecker, O., & Niederkorn, M. (2019). *Global Payments Report 2019: Amid Sustained Growth, Accelerating Challenges Demand Bold Actions.*

[5] Bernard, P., De Freitas, N.E.M., & Maillet, B.B. (2021). A financial fraud detection indicator for investors: an IDeA. *Annals of Operations Research*, 1–24.

[6] Blumenstock, J. (2020). Machine learning can help get COVID-19 aid to those who need it most. *Nature* 13.7.2020, 1–3.

[7] Breunig, M.M., Kriegel, H.P., Ng, R.T., & Sander, J. (2020). LOF: Identifying density-based local outliers. In: *2000 ACM SIGMOD International Conference on Management of Data - SIGMOD '00*, New York, New York, USA, p. 93–104.

[8] Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in β-VAE. In: *Proc. of the 31st Conference on Neural Information Processing Systems*, p. 1–11.

[9] Buschjäger, S., Honysz, P.J., & Morik, K. (2021). Randomized outlier detection with trees. *International Journal of Data Science and Analytics*, 1–14.

[10] Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences, 557*, 317–331.

[11] Carminati, M., Caron, R., Maggi, F., Epifani, I., & Zanero, S. (2015). BankSealer: A decision support system for online banking fraud analysis and investigation. *Computers & Security, 53*, 175–186.

[12] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In: *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, p. 785–794.

[13] Chen, Y., & Sivakumar, V. (2021). Invesitigation of finance industry on risk awareness model and digital economic growth. *Annals of Operations Research*, 1–22.

[14] Chen, S., Yuan, Y., Luo, X.R., Jian, J., & Wang, Y. (2021). Discovering group-based transnational cyber fraud actives: A polymethodological view. *Computers & Security, 104*, 102217.

[15] Chin, A. G., Harris, M. A., & Brookshire, R. (2022). An empirical investigation of intent to adopt mobile payment systems using a trust-based extended valence framework. *Information Systems Frontiers, 24*, 329–347.

[16] Choi, D., & Lee, K. (2017). Machine learning based approach to financial fraud detection process in mobile payment system. *IT CoNvergence PRActice (INPRA), 5*(4), 12–24.

[17] Choi, D., & Lee, K. (2018). An artificial intelligence approach to financial fraud detection under IoT environment: A survey and implementation. *Security and Communication Networks, 2018*, 5483472.

[18] Coppolino, L., D'Antonio, S., Formicola, V., Massei, C., & Romano, L. (2015). Use of the Dempster–Shafer theory to detect account takeovers in mobile money transfer services. *Journal of Ambient Intelligence and Humanized Computing, 6*(6), 753–762.

[19] Cui, J., Yan, C., & Wang, C. (2021). ReMEMBeR: Ranking metric embedding-based multicontextual behavior profiling for online banking fraud detection. *IEEE Transactions on Computational Social Systems, 8*(3), 643–654.

[20] David-West, O., Oni, O., & Ashiru, F. (2022) Diffusion of innovations: Mobile money utility and financial inclusion in Nigeria. Insights from agents and unbanked poor end users. *Information Systems Frontiers*, 1–21.

[21] Davidovic, S., Nunhuck, S., Prady, D., Tourpe, H., & Anderson, E. (2020). Beyond the COVID-19 crisis: a framework for sustainable government-to-person mobile money transfers. *IMF Working Papers, 198*, 1–38.

[22] Delecourt, S., & Guo, L. (2019). Building a robust mobile payment fraud detection system with adversarial examples. In: *2019 IEEE 2nd Int. Conf. on Artificial Intelligence and Knowledge Engineering (AIKE)*, IEEE, p. 103–106.

[23] Dhieb, N., Ghazzai, H., Besbes, H., & Massoud, Y. (2019). Extreme gradient boosting machine learning algorithm for safe auto insurance operations. In: *2019 IEEE International Conference on Vehicular Electronics and Safety*, ICVES 2019, IEEE, p. 1–5.

[24] Du, J.Z., Lu, W.G., Wu, X.H., Dong, J.Y., & Zuo, W.M. (2018). L-SVM: A radius-margin-based SVM algorithm with LogDet regularization. *Expert Systems with Applications, 102*, 113–125.

[25] Franque, F. B., Oliveira, T., & Tam, C. (2022). Continuance intention of mobile payment: TTF model with Trust in an African context. *Information Systems Frontiers*, 1–19.

[26] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 42*(4), 463–484.

[27] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications, 73*, 220–239.

[28] Hajek, P., & Henriques, R. (2017). Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods. *Knowledge-Based Systems, 128*, 139–152.

[29] Hajek, P. (2019). Interpretable fuzzy rule-based systems for detecting financial statement fraud. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*, AIAI 2019, Springer, p. 425–436.

[30] Hardin, J., & Rocke, D.M. (2004). Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics and Data Analysis, 44*(4), 625–638.

[31] He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters, 24*(9–10), 1641–1650.

[32] Huang, S. Y., Lin, C. C., Chiu, A. A., & Yen, D. C. (2017). Fraud detection using fraud triangle risk factors. *Information Systems Frontiers, 19*(6), 1343–1356.

[33] Iman, N. (2018). Is mobile payment still relevant in the fintech era?. *Electronic Commerce Research and Applications, 30*, 72–82.

[34] Jia, L., Song, X., & Hall, D. (2022). Influence of habits on mobile payment acceptance: An ecosystem perspective. *Information Systems Frontiers, 24*, 247–266.

[35] Jocevski, M., Ghezzi, A., & Arvidsson, N. (2020). Exploring the growth challenge of mobile payment platforms: A business model perspective. *Electronic Commerce Research and Applications, 40*, 100908.

[36] Kang, J. (2018). Mobile payment in Fintech environment: trends, security challenges, and services. *Human-Centric Computing and Information Sciences, 8*(1), 1–16.

[37] Kar, A. K. (2021). What affects usage satisfaction in mobile payments? Modelling user generated content to develop the "digital service usage satisfaction model". *Information Systems Frontiers, 23*(5), 1341–1361.

[38] Kriegel, H.P., Schubert, M., & Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. In: *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 444–452.

[39] Li, Q., & Clark, G. (2013). Mobile security: A look ahead. *IEEE Security and Privacy, 11*(1), 78–81.

[40] Liu, F.T., & Ting, K.M., Zhou, Z.H. (2008). Isolation forest. In: *IEEE Int. Conf. on Data Mining*, ICDM, IEEE, p. 413–422.

[41] Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., & He, X. (2019). Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering, 32*(8), 1517–1528.

[42] Lopez-Rojas, E., Elmir, A., & Axelsson, S. (2016). Paysim: A financial mobile money simulator for fraud detection. In: *28th European Modeling and Simulation Symposium*, EMSS 2016, Dime University of Genoa, Larnaca, p. 249–255.

[43] Lopez-Rojas, E.A., & Axelsson, S. (2014). Banksim: A bank payments simulator for fraud detection research. In: *the 26th European Modeling and Simulation Symposium (EMSS)*, p. 144–152.

[44] Lopez-Rojas, E.A., Axelsson, S., & Baca, D. (2018). Analysis of fraud controls using the PaySim financial simulator. *International Journal of Simulation and Process Modelling, 13*(4), 377–386.

[45] Lopez-Rojas, E.A., & Barneaud, C. (2019). Advantages of the PaySim simulator for improving financial fraud controls. *Advances in Intelligent Systems and Computing, 998*, 727–736.

[46] Mahbobi, M., Kimiagari, S., & Vasudevan, M. (2021). Credit risk classification: an integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*, 1–29.

[47] Mendelson, S., & Lerner, B. (2020). Online cluster drift detection for novelty detection in data streams. In: *Proc. of the 19th IEEE International Conference on Machine Learning and Applications*, ICMLA 2020, p. 171–178.

[48] Misra, S., Thakur, S., Ghosh, M., & Saha, S.K. (2020). An autoencoder based model for detecting fraudulent credit card transaction. *Procedia Computer Science, 167*, 254–262.

[49] Mubalaike, A.M., & Adali, E. (2018). Deep learning approach for intelligent financial fraud detection system. In: *UBMK 2018 3rd Int. Conf. on Computer Science and Engineering*, p. 598–603.

[50] Nami, S., & Shajari, M. (2018). Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. *Expert Systems with Applications, 110*, 381–392.

[51] Ngai, E.W.T., Hu, Y., Wong, Y.H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems, 50*(3), 559–569.

[52] Onwubiko, C. (2020). Fraud matrix: a morphological and analysis-based classification and taxonomy of fraud. *Computers & Security, 96*, 101900.

[53] Pal, A., De, R., & Herath, T. (2020). The role of mobile payment technology in sustainable and human-centric development: evidence from the post-demonetization period in India. *Information Systems Frontiers, 22*(3), 607–631.

[54] Pal, A., Herath, T., De, R., & Rao, H. R. (2021). Is the convenience worth the risk? An investigation of mobile payment usage. *Information Systems Frontiers, 23*(4), 941–961.

[55] Pambudi, B.N., Hidayah, I., & Fauziati, S. (2019). Improving money laundering detection using optimized support vector machine. In: *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems*, ISRITI 2019, p. 273–278.

[56] Papouskova, M., & Hajek, P. (2019). Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decision Support Systems, 118*, 33–45.

[57] Pevny, T. (2016). Loda: Lightweight on-line detector of anomalies. *Machine Learning, 102*(2), 275–304.

[58] Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data sets. In: *Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data*, p. 427–438.

[59] Reunanen, N., Räty, T., & Lintonen, T. (2020). Automatic optimization of outlier detection ensembles using a limited number of outlier examples. *International Journal of Data Science and Analytics, 10*, 377–394.

[60] Rieke, R., Zhdanova, M., Repp, J., Giot, R., & Gaber, C. (2013). Fraud detection in mobile payments utilizing process behavior analysis. In: *2013 Int. Conf. on Availability, Reliability and Security*, ARES 2013, p. 662–669.

[61] Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: *Proc. of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, p. 4–11.

[62] Seera, M., Lim, C.P., Kumar, A., Dhamotharan, L., & Tan, K.H. (2021). An intelligent payment card fraud detection system. *Annals of Operations Research*, 1–23.

[63] Sethia, A., Patel, R., & Raut, P. (2018). Data augmentation using generative models for credit card fraud detection. In: *4th International Conference on Computing Communication and Automation (ICCCA)*, IEEE, p. 1–6.

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

[65] Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J., Platt, J., & Holloway, R. (2000). Support vector method for novelty detection. In: *Advances in Neural Information Processing Systems*, MIT Press, p. 582–588.

[66] Shajalal, M., Hajek, P., & Abedin, M. Z. (2021). Product backorder prediction using deep neural network on imbalanced data. *International Journal of Production Research*, 1–18.

[67] Turner, A., Mccombie, S., & Uhlmann, A. (2021). Follow the money: Revealing risky nodes in a Ransomware-Bitcoin network. In: *Proc. of the 54th Hawaii Int. Conf. on System Sciences*, p. 1560–1572.

[68] Vaughan, G. (2020). Efficient big data model selection with applications to fraud detection. *International Journal of Forecasting, 36*(3), 1116–1127.

[69] Verkijika, S.F. (2020). An affective response model for understanding the acceptance of mobile payment systems. *Electronic Commerce Research and Applications, 39*, 100905.

[70] Wang, C., Deng, C., & Wang, S. (2020). Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. *Pattern Recognition Letters, 136*, 190–197.

[71] West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: A comprehensive review. *Computers & Security, 57*, 47–66.

[72] Wong, M. L., Seng, K., & Wong, P. K. (2020). Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. *Expert Systems with Applications, 141*, 112918.

[73] Xenopoulos, P. (2017). Introducing DeepBalance: Random deep belief network ensembles to address class imbalance. In: *2017 IEEE Int. Conf. on Big Data, Big Data 2017*, p. 3684–3689.

[74] Yamanishi, K., Takeuchi, J.I., Williams, G., & Milne, P. (2004). Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery, 8*(3), 275–300.

[75] Ye, X., Dong, L.A., & Ma, D. (2018). Loan evaluation in P2P lending based on random forest optimized by genetic algorithm with profit score. *Electronic Commerce Research and Applications, 32*, 23–36.

[76] Zhao, Y., & Hryniewicki, M.K. (2018). XGBOD: Improving supervised outlier detection with unsupervised representation learning. In: *Proc. of the Int. Joint Conf. on Neural Networks*, p. 1–8.

[77] Zhao, Y., Nasrullah, Z., & Li, Z. (2019). PyOD: A Python toolbox for scalable outlier detection. *Journal of Machine Learning Research, 20*(96), 1–7.

[78] Zhou, H., Chai, H.F., & Qiu, M.L. (2018). Fraud detection within bankcard enrollment on mobile device based payment using machine learning. *Frontiers of Information Technology and Electronic Engineering, 19*(12), 1537–1545.

Appendix 1: Settings of machine learning methods

Method	**Parameters**
ABOD	contamination = the proportion of frauds in the training dataset, neighbours $k = \{5, 10\}$
CBLOF	number of clusters = 8, clustering estimator = K-means, alpha = 0.9
HBOS	alpha = 0.1
LODA	number of bins = 10, number of random cuts = 100
Isolation Forest	number of estimators = {100, 200}
KNN	neighbours $k = \{2, 3, 5\}$, radius = 1.0
MCD	contamination = the proportion of frauds in the training dataset
OCSVM	kernel function: {linear, polynomial, RBF with gamma = 0.01}, nu = 0.1
AE	hidden activation = ReLU, optimizer = adam, epochs = 100, dropout rate = 0.2, L2 regularizer = 0.2, hidden neurons = [8, 4, 4, 8]
VAE	hidden activation = ReLU, optimizer = adam, epochs = 100, gamma = 1.0, dropout rate = 0.2, L2 regularizer = 0.1, encoder neurons = [8, 4, 2], decoder neurons = [2, 4, 8]
MO-GAAL	contamination = the proportion of frauds in the training dataset, number of sub generators = 10, learning rate of the discriminator = 0.01, learning rate of the generator = 0.0001, epochs = 20
XGBOD	estimator list = {ABOD, CBLOF, HBOS, LODA, Isolation Forest, KNN, MCD, OCSVM, AE, VAE}, $p = 5$, learning rate = 0.1
k-NN	$k = 2, 3, 5$
SVM	complexity parameter $C = 1$, kernel function: {linear, polynomial, RBF with gamma = 0.01}
RF	number of trees = {100, 200}
XGBoost	booster = gbtree, eta = 0.3, gamma = 0, maximum depth of a tree = {3, 6, 9}, sampling method = uniform, lambda = 1, alpha = 0
RUS	sampling strategy = {0.5, 0.75, 1.0}