Staff Working Paper/Document de travail du personnel—2024-15

Last updated: May 15, 2024

Finding a Needle in a Haystack: A Machine Learning Framework for Anomaly Detection in Payment Systems

by Ajit Desai,[1] Anneke Kosse[2] and Jacob Sharples[3]

[1] Banking and Payments Department
Bank of Canada
adesai@bankofcanada.ca

[2] Bank for International Settlements
anneke.kosse@bis.org

[3] Financial Stability Department
Bank of Canada
jsharples@bankofcanada.ca

Bank of Canada staff working papers provide a forum for staff to publish work-in-progress research independently from the Bank's Governing Council. This research may support or challenge prevailing policy orthodoxy. Therefore, the views expressed in this paper are solely those of the authors and may differ from official Bank of Canada views. No responsibility for them should be attributed to the Bank.

DOI: https://doi.org/10.34989/swp-2024-15 | **ISSN** 1701-9397

Acknowledgements

The views expressed in this paper are solely those of the authors and do not necessarily reflect those of the Bank of Canada, the Bank for International Settlements (BIS), the BIS Committee on Payments and Market Infrastructures or its members. We thank Leonard Sabetti for his contribution during earlier stages of the work. We also thank Segun Bewaji, Alessio Brini, Narayan Bulusu, Ricardo de Avillez, Laura Felber, Marc Glowka, Tarush Gupta, Constanza Martinez and Ellen van der Woerd for their detailed comments. In addition, we thank participants of the following conferences for their comments and suggestions: BIS Research Webinar Series (2022), Economics of Payments XI Conference (2022), DNB Central Bankers Go Data Driven Conference (2022), Canadian Economics Association Annual Conference (2022), RBI Global Conference on Financial Resilience (2023), International Conference on Economic Modeling and Data Science (2023), the Bank of Canada and the Bank for International Settlements' Seminar on Granular Data (2023), the Second CEMLA Regional Conference of Payments and Financial Market Infrastructures (2023), and the American Economic Association Annual Meeting Poster Session (2024).

Abstract

We propose a flexible machine learning (ML) framework for real-time transaction monitoring in high-value payment systems (HVPS), which are a central piece of a country's financial infrastructure. This framework can be used by system operators and overseers to detect anomalous transactions, which—if caused by a cyber attack or an operational outage and left undetected—could have serious implications for the HVPS, its participants and the financial system more broadly. Given the substantial volume of payments settled each day and the scarcity of actual anomalous transactions in HVPS, detecting anomalies resembles an attempt to find a needle in a haystack. Therefore, our framework uses a layered approach. In the first layer, a supervised ML algorithm is used to identify and separate "typical" payments from "unusual" payments. In the second layer, only the unusual payments are run through an unsupervised ML algorithm for anomaly detection. We test this framework using artificially manipulated transactions and payments data from the Canadian HVPS. The ML algorithm employed in the first layer achieves a detection rate of 93%, marking a significant improvement over commonly used econometric models. Moreover, the ML algorithm used in the second layer marks the artificially manipulated transactions as nearly twice as suspicious as the original transactions, proving its effectiveness.

Topics: Digital currencies and fintech; Financial institutions; Financial services; Financial system regulation and policies; Payment clearing and settlement systems

JEL codes: C45, C55, D83, E42

Résumé

Nous proposons un cadre flexible d'apprentissage automatique pour surveiller les transactions en temps réel dans les systèmes de paiement de grande valeur (SPGV), lesquels représentent un élément central de l'infrastructure financière d'un pays. Notre cadre est susceptible de servir aux opérateurs et aux organes de surveillance de ces systèmes pour détecter des anomalies qui – si elles étaient causées par une cyberattaque ou une défaillance opérationnelle et passaient inaperçues – pourraient avoir de graves conséquences pour le SPGV concerné, ses participants et le système financier en général. Compte tenu du volume important des paiements à régler chaque jour et de la rareté des transactions vraiment anormales dans les SPGV, repérer des anomalies revient à chercher une aiguille dans une botte de foin. C'est pourquoi notre cadre repose sur deux niveaux de traitement. Le premier est basé sur un algorithme d'apprentissage automatique supervisé qui distingue les paiements typiques des paiements inhabituels. Seuls les paiements inhabituels sont soumis au second niveau de traitement, et passent par un algorithme non supervisé de reconnaissance des anomalies. Pour tester ce cadre, nous recourons à des données de transaction manipulées artificiellement et à des données de paiement provenant de SPGV canadiens. L'algorithme d'apprentissage automatique du premier niveau atteint un taux de détection de 93 % – résultat nettement supérieur à celui obtenu par les modèles économétriques couramment utilisés. De plus, l’algorithme du second niveau présente les transactions manipulées comme étant près de deux fois plus suspectes que les transactions originales, ce qui prouve son efficacité.

Sujets : Monnaies numériques et technologies financières; Institutions financières; Services financiers; Réglementation et politiques relatives au système financier; Systèmes de compensation et de règlement des paiements

Codes JEL : C45, C55, D83, E42

1 Introduction

High-value payment systems (HVPSs), such as Lynx in Canada, Fedwire in the US, Chaps in the UK, and Target2 in the Eurozone, are vital components of jurisdictions' financial systems. Typically, these are real-time gross settlement (RTGS) systems that process large-value transactions between financial institutions, often requiring settlement by a particular time. As such, the safety and efficiency of HVPSs are key to financial stability and economic growth. If not properly managed, an HVPS can be a source of a shock, such as payments fraud, a cyber attack, market stress, or operational problems (Chapman et al. 2015; BIS-Report 2019; FED-Report 2019; Kotidis and Schreft 2023). Moreover, as HVPSs provide a link between their participating financial institutions, they could become a channel through which shocks are transmitted across domestic or even international financial markets (Kosse and Lu 2022; Kotidis and Schreft 2023).

In particular, cyber attacks pose a growing risk to financial institutions and HVPSs.[1] For instance, in 2016, the Central Bank of Bangladesh (CBB) fell victim to a cyber heist, where hackers attempted to steal nearly one billion dollars from the CBB reserves account at the Federal Reserve Bank of New York (Bukth and Huda 2017). Similarly, cyber attacks on Mexico's interbank payment network and Banco de Chile in 2018 resulted in losses amounting to millions of dollars (Nish and Naumaan 2019).[2] Recently, Kotidis and Schreft (2023) has documented the effects of a cyber attack on a service provider to the banks participating in Fedwire and highlighted the importance of operational resilience. Moreover, simulation-based studies have shown that cyber attacks, even when targeting individual HVPS participants, can have a significant impact on the system in which they participate (Eisenbach et al. 2021; Kosse and Lu 2022; Docherty and Wang 2010). Therefore, real-time transaction monitoring for timely detection of anomalies in HVPSs is a way to enhance cyber resilience and avoid such shocks from having unintended system-wide consequences.

To enhance safety in HVPSs and, more broadly, to limit systemic risk and foster financial stability, various guidelines have been issued and many initiatives have been undertaken globally.[3] For instance, in 2016, the Bank for International Settlements (BIS) Committee on Payments and Market Infrastructures (CPMI) and the International Organisation of Securities Commissions (IOSCO) issued guidance to boost cyber resilience in financial market infrastructures (FMIs) like HVPSs.[4] This guidance stresses the importance of FMIs maintaining effective capabilities to monitor anomalous activity and outlines the tools and processes they should employ for detecting cyber incidents.

Despite these efforts, real-time transaction monitoring for anomaly detection in HVPSs faces several challenges, with the scarcity of anomalies and the absence of pre-identified examples being the primary obstacle. Additionally, the high frequency of payments, the extensive size of payment networks, the complex strategic interactions among participants,[5] and the limited availability of detailed transaction information[6] further complicate the detection process. As a result, the ability to detect unusual transactions in HPVSs is still in its early stages and faces several limitations. For instance, HVPSs with capabilities to monitor payment flows mostly rely on rule-based and ad-hoc monitoring approaches. These approaches involve a

[1] The Bank of Canada's recent financial stability review highlights that the financial sector globally has the largest share of reported cyber attacks. https://www.bankofcanada.ca/2023/05/financial-system-review-2023/.

[2] A comprehensive overview of cyber incidents involving financial institutions can be found at CEIP (2021).

[3] The Federal Reserve published strategies for improving safety in the US payment systems and provides models that help organizations classify fraudulent payment activity (FED-Report 2019). Likewise, the Eurosystem reported the development of a methodology enabling the operator to detect potentially anomalous transactions in their HVPS (TARGET-Report 2019).

[4] See https://www.bis.org/cpmi/publ/d146.pdf.

[5] In certain situations, participants might strategically alter their typical payment behavior due to liquidity constraints or to minimize payment delay costs (Bech and Garratt 2003; Castro et al. 2021). These factors are unrelated to cyber incidents or outages.

[6] Having additional information that provides details about transactions could offer supplementary features to learn and understand typical payment behavior (Glowka 2019; León 2020; Sabetti and Heijmans 2021).

delay and require prior assumptions about how anomalous payments would look. Such assumptions may not cover all forms of anomalies, as it is impossible to anticipate all potential scenarios (FED-Report 2019; Arjani et al. 2020; Arjani and Heijmans 2020). Furthermore, anomaly detection tools employed by individual system participants only capture transactions to and from that particular participant, which limits their utility for system-wide transaction monitoring.

To mitigate these challenges, in this paper we propose a flexible, centralized, and layered transaction monitoring framework for pattern recognition and anomaly detection in HVPS, leveraging data-driven and nonlinear machine learning (ML) tools. In this framework, monitoring occurs centrally at the HVPS level, utilizing transaction data from all system participants to comprehend their payment patterns. This approach contrasts with the monitoring tools of individual participants, which focus only on their own transactions. Moreover, the framework consists of multiple layers to overcome the challenges due to the extensive size of usual payments and unavailability of pre-identified anomalies. In the first layer, a supervised ML algorithm is used to classify payments by submission time, effectively screening for typical, or usual, transactions. In the second layer, only the misclassified (unusual) payments are processed through an unsupervised ML algorithm for anomaly detection.[7] Using nonlinear ML-based models in each layer allows us to learn complex patterns from a large amount of historical data, eliminating the need for predetermined rules. Moreover, the proposed ML models in our framework do not require prior assumptions about the data-generating process or the structure of the potential anomalies, allowing for a more generalized detection process. Furthermore, both layers in our framework are independent and have flexible components capable of integrating additional ML models. Therefore, they can serve as robust tools for detecting anomalies within HVPSs.

We test our framework with manipulated transactions and actual-transaction data from Canada's former and current HVPS, the Large Value Transfer System (LVTS), and Lynx, respectively.[8] The results demonstrate that the proposed framework is a promising approach for transaction monitoring and anomaly detection in HVPSs. The gradient-boosting-based ML model used in the first layer outperforms a logistic regression and other ML models by up to 44%, and it exhibits higher out-of-sample accuracy when classifying transactions on non-regular days and artificially manipulated transactions.[9] In particular, our model demonstrates a notable ability to correctly detect 93% of all artificial transactions. Moreover, the isolation forest (IF) model used in the second layer successfully assigns higher scores to manually altered anomalous transactions—on average, twice as high as their original counterparts. Also, we perform various scenario analyses, which demonstrate that the framework is flexible enough to be applied for different payment system designs and to be extended with other features to even further improve its robustness.

To gain insight into the primary features driving the anomalous payments identified by our algorithm, we employ the Shapley value-based SHAP approach (Lundberg and Lee 2017) to interpret the results of the ML models used in both layers. We find that for a given HVPS transaction, the time elapsed since the previous incoming transaction and basic transaction features, like sender-receiver pair and payment amount, effectively predict payment submission patterns. By contrast, the more complex intraday features, such as the time elapsed since specific types of incoming and outgoing transactions, assume a prominent role in identifying and isolating anomalies. We also show that the SHAP approach can help system operators in assessing the nature and severity of anomalies, enabling timely interventions and prompt action.

[7] Our two-step approach is analogous to the two-step process of airport security screening for passengers, in which the first step involves initial machine screening, and only passengers who fail this screening undergo detailed inspections.

[8] We use payments settled between January 2011 and August 2021 in LVTS and from October 2021 to August 2023 in Lynx. Note that we excluded the month of September from the dataset due to the transition from LVTS to Lynx on August 29, 2021.

[9] To prevent seasonal influences from affecting the model, we identify and exclude "special days," or non-regular, days from our training sample. This helps us avoid the model solely identifying payments on such special days as anomalous. These days include Canadian provincial holidays, US national holidays, days with known operational incidents, and the Covid-19 period.

The remainder of the paper is structured as follows. In section 2 we provide further background and explain the key contribution of our paper. Section 3 presents an overview of our proposed framework for classifying and identifying anomalies, including the ML models used in each layer. This is followed in section 4 by a description of the data and transaction features used for the analyses. The results and implications are presented in section 5, and scenario analyses showing the framework's adaptability for different payment system designs are presented in section 6. We conclude in section 7.

2 Background and Contribution to Existing Literature

Changes in timing and patterns of HVPS transactions have historically been studied to understand the impact of economic disruptions, such as the 2008 global financial crisis (Bech and Garratt 2012; Massarenti et al. 2012; Alexandrova-Kabadjova et al. 2015; Zhang 2015), and to investigate liquidity and delay trade-offs faced by HVPS participants (Bech and Garratt 2003; McAndrews and Rajan 2000; Martin and McAndrews 2008; Castro et al. 2021; McMahon et al. 2022). However, given the importance of the availability and smooth functioning of HVPSs for the broader financial system and with cyber threats growing over time, there is a strong case for HVPS transaction monitoring in the context of operational risks, such as cyber events and operational incidents (Chapman et al. 2015; Eisenbach et al. 2021; Kotidis and Schreft 2023).

Only recently, driven by the increasing risk of cyber threats and operational outages, researchers have started developing tools for monitoring transactions. Given the high frequency of payments settled in HVPSs, this work mostly relies on advanced analytical tools using ML. For instance, Glowka (2019) applies clustering techniques on Target2 data to identify payment profiles of banks and Arévalo et al. (2022) use an ML model to identify clusters of anomalous payments in the Salvadorian payment system. Artificial neural network (ANN) models have also been increasingly used to identify anomalous payment flows between participants. For instance, Triepels et al. (2017) employ an autoencoder, an unsupervised ANN methodology that is able to recognize complex patterns, and León et al. (2020) use an ANN model for identifying patterns in participants' payment behavior. Similar models have been set up by Rubio et al. (2020) and Sabetti and Heijmans (2021) to classify and identify anomalous payments in the HVPS of Ecuador and the retail payment system in Canada, respectively.

Two common challenges faced in this research field thus far are the rarity of known anomalies and the large size and complexity of payment networks. In our paper, we develop a framework for anomaly detection that addresses these two challenges. Typically, HVPSs settle thousands of transactions without systematically monitoring for anomalies, resulting in limited or no pre-identified anomalies to learn from. Additionally, the timing of these transactions are driven by multiple factors such as business requirements, liquidity constraints, time sensitivity, and intricate bilateral and multilateral business relationships among participants. Existing work based on clustering-based methods, for instance, requires the identification of the optimal number of clusters, which may be particularly challenging with high-dimensional data and outliers. This, in turn, leads to a slow training process when using large datasets. Likewise, the main limitation of ANN models is that they are hard to train on transaction-level data, require manual aggregation, and are difficult to interpret. In contrast, the framework proposed in this paper is easier to train on high volumes of transaction-level data. By using multiple layers and a combination of supervised and unsupervised tree-based ML models, we mitigate the challenges resulting from the rarity of anomalies and the large volume of payments. Moreover, our framework provides tools for the interpretation of the results to gain insight into the influential factors driving HVPS participants' usual and anomalous payment behavior.

Given the novelties and flexibility of our proposed framework, it contributes to the broader literature on financial fraud detection and financial stability monitoring. The challenges related to dealing with large volumes of data and identifying rare anomalous events not only apply when monitoring HVPS transactions but are also relevant when identifying credit card fraud, illegal trading activities, financial statement fraud, and systemic financial instability (Ngai et al. 2011; Flood et al. 2016; Ryman-Tubb et al. 2018; James et al. 2023). Our framework can be adapted and applied to this broader set of applications.

3 Framework for Pattern Recognition and Anomaly Detection in HVPSs

Given the large size and complexity of payment networks and the rarity of known anomalies, we propose a framework that consists of two layers,[10] each using different ML models, that can be employed in near real-time (see Figure 1). In the first layer, a supervised ML classifier is used to classify settled HVPS payments based on a target variable.[11] This classifier allows us to separate the full set of HVPS payments into two subsets: correctly classified and misclassified transactions. This separation is crucial: it allows us to systematically filter out a large subset of typical transactions in the first layer such that we can focus on a smaller subset of payments in the second layer. Subsequently, we use this subset of misclassified transactions for anomaly detection with the help of an unsupervised ML algorithm in the second layer.

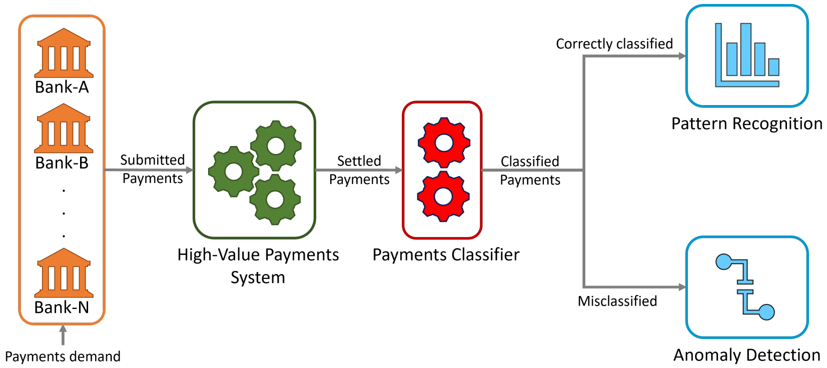


Figure 1: Schematic overview of the layered approach proposed in this paper for transaction monitoring in HVPSs. All settled HVPS payments pass through a payments classifier. Subsequently, the subset of misclassified payments is sent to the anomaly detection model, while the correctly classified payments are used to study the usual payment patterns of the HVPS participants.

The key advantage of using the layered approach, combining supervised and unsupervised ML tools, is its ability to separate typical payments in the first layer. This assists in mitigating difficulties arising from high volume and helps address the challenges due to the rarity of anomalies. Moreover, the first layer is flexible and independent from the second layer, allowing the integration of additional algorithms to enhance robustness. For instance, multiple classification algorithms can be used simultaneously, each with a different classification target to improve the performance of the first layer and ease the subsequent task of anomaly detection (see section 6.1 for an example of a joint classifier). Furthermore, our framework allows the use of the subsets of correctly classified and misclassified payments from the first layer to study the usual and unusual payment patterns of HVPS participants in the second layer. To this end, we employ the Shapley value-based (SHAP) approach to assess the impact of various transaction features on the timing of payments and to gain a deeper understanding of which features are more likely to uncover anomalies.

[10] Having sufficient pre-identified anomalies could have eliminated the need for a layered approach. However, obtaining a sufficiently large set of labeled data covering a diverse set of anomalies is challenging for many applications, including HVPS.

[11] Note that the classifier module in the first layer is adaptable; it can incorporate multiple ML algorithms and handle various target variables. Additionally, it can be expanded to perform regressions with continuous targets.

3.1 Layer 1: Payments Classifier

As a first step, we run all HVPS payments through one or more *payments classifiers*. These classifiers use supervised learning models with categorical targets, which could be binary or multi-class. The selection of target variables can be tailored based on the characteristics of the payment systems. For simplicity, we choose a binary classifier based on the payment submission period, which classifies payments as either morning or afternoon payments. Morning payments are defined as those that are submitted to the HPVS before noon, whereas afternoon payments are those submitted in the afternoon.

The model can be formulated as follows: $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\}$ represents a set of M predictors (also known as features or independent variables), each being a vector of N data points (samples), and $\mathbf{y}$ is the target variable (or dependent variable), also containing N data points. In our case, $\mathbf{y} = \mathbf{1}, \mathbf{2}$, where 1 represents the label for the set of morning payments and 2 represents the afternoon payments. We denote $\hat{\mathbf{y}}$ as the predicted target (i.e., the submission time of the payment), which can be estimated using the classification model $f : \hat{\mathbf{y}} = f(X)$.

A commonly used econometric model for such a classification is a logistic regression that analyzes the relationship between the target variable and the predictor variables and that generates probabilities for the target variable belonging to one of the two groups. Logistic regression models are generally valued for their simplicity and interpretability. However, their performance may be limited when dealing with large and complex datasets. In particular, challenges arise when there are many continuous and categorical predictors and when dealing with nonlinearity, collinearity, and outliers (Hastie et al. 2009).

In light of these challenges and given the nature of our data, we instead use a decision-tree-based ensemble learning approach. Random forest and gradient boosting models are examples of such an approach and are based on flexible nonlinear ML algorithms with a demonstrated ability to perform well for complex categorical learning tasks. These non-parametric models can effectively handle collinearity, nonlinearity, outliers, and datasets containing both continuous and categorical predictors, making them suitable for our analysis (Friedman 2001; Friedman et al. 2001). We primarily use the Light Gradient Boosted Machine (LightGBM) algorithm. While we explored various other models, including logistic regression, decision trees, random forests, and standard gradient boosting, LightGBM stands out for several reasons. It is an open-source library that allows for a fast and efficient implementation of the gradient boosting algorithm by focusing on boosting examples with larger gradients. It also adds a type of automatic feature selection that has been demonstrated to generate better predictive performance (Ke et al. 2017). Particularly for handling very large datasets like HVPS data, such a scalable and efficient model is useful for the purpose of fast training. See Appendix A for more details on the LightGBM algorithm and the implications of its use.

3.2 Layer 2: Anomaly Detection on the Subset of Missclassified Payments

In the subsequent step, we run the subset of misclassified payments from the first layer through an ML-based anomaly detection model. As there are commonly no pre-identified anomalous transactions in HVPS datasets, we are dealing with unlabelled data. Therefore, we use an unsupervised learning-based IF algorithm (Liu et al. 2008).[12]

The choice of the IF algorithm is motivated by its effectiveness in handling transaction-level data and its capacity to rank transactions according to abnormality levels. Additionally, its simplicity and shared characteristics with the chosen ML model in the first layer enhance its suitability. Like gradient boosting,

[12] We could have used unsupervised learning with the IF model without combining it with the payments classifier model as a first step. However, such a standalone IF model would exhibit relatively low accuracy, likely due to the large size and complexity of the HVPS payments. For more details, see Appendix B.

the IF algorithm uses decision trees (called isolation trees) and is based on the concept of isolating data points that are rare and significantly different from the majority of the data. The IF algorithm has been proven to be fast and memory-efficient, due to which it can handle high-dimensional and large-scale transaction-level data (Liu et al. 2008). Therefore, the IF approach is particularly well-suited for transaction monitoring and anomaly detection in HVPSs.

In short, the IF algorithm first randomly selects a subset of n data points from the set of M features ($X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\}$), each with N sample points. Next, it creates isolation trees by recursively partitioning the data until either the maximum tree depth is reached or there is just one data point left. During each partitioning step, the algorithm randomly selects a feature ($\mathbf{x}^m$ from X) and a split point along that feature to divide the data into two subsets. The algorithm repeats these steps with different sub-samples to generate multiple trees. Finally, it computes the anomaly score for each data point by calculating the average number of splits (or path length) using all isolation trees. This average path length is then transformed into an anomaly score (with lower path lengths leading to higher scores), which indicates the likelihood of that data point being an anomaly. See Appendix B for more details on the IF and the implications of its use.

3.3 Interpretation Tools for Classification and Anomaly Detection Models

We use the Shapley value-based (SHAP) approach (Lundberg and Lee 2017) to interpret the outcomes of the LightGBM and IF models in both Layer 1 and Layer 2.[13] The SHAP approach allows us to examine how the characteristics of HVPS payments influence their submission time and the identification of anomalies. The SHAP values, which we visualize in so-called dependence plots, explain how specific payment features impact participants' usual payment patterns and how they contribute to the detection of anomalies therein.

We also use the SHAP approach to study individual transactions, known as local interpretation. We do so using force plots, which visualize the marginal contribution of transaction features with the help of arrows. These arrows either push the prediction scores up (towards an afternoon prediction) or down (towards a morning prediction). The length of the arrows indicate the magnitude of the SHAP value, which is analogous to the strength of the contribution of that transaction feature to the model's prediction. In addition, we use the SHAP approach to study subsets of payments, known as global interpretation. This approach provides the average impact of transaction features on the model's prediction for that subset of transactions. The local and global interpretation analyses help us gain insight into why a particular transaction or a subset of transactions are identified as unusual. This allows us to understand the influence of specific payment features on the submission time of HVPS payments.

Note that the SHAP approach does not provide causal inference or any optimal statistical criterion (Slack et al. 2020; Molnar 2020). Instead, its purpose is to explain the marginal contribution of each transaction feature to the difference between the models' predictions and the average prediction of the entire training sample. Therefore, its primary use is intuitively explaining the predictions of the models (Lundberg and Lee 2017). Further details on the SHAP approach are provided in Appendix D.

In the following sections, for simplicity, we primarily focus on demonstrating our framework's usefulness with a binary classifier, discussed in section 3, on a transaction sample from LVTS, presented in section 4. After that, in section 6, we showcase the framework's flexibility using a joint classifier in the first layer, and we extend the framework's application to other LVTS sub-samples and a data sample from Lynx.

[13] We use TreeSHAP, a variant of SHAP, which offers computational efficiency in estimating Shapley values for tree-based ML models, such as LightGBM. Although initially designed for supervised learning models, TreeSHAP can be adapted with minor modifications for unsupervised learning-based IF models (Lundberg and Lee 2017).

4 Data

4.1 LVTS Transaction Data

We test our proposed framework using around ten years of historical transaction data from the Canadian HVPS, Large Value Transfer System (LVTS), from January 2012 to August 2021. Our focus is on payments settled during the system's regular business hours, which is from 6 am to 6 pm. As such, we cover approximately 85 million transactions exchanged among 17 participants. The daily average transaction value is approximately 135 billion Can$ over a daily average transaction volume of 35,000 transactions.

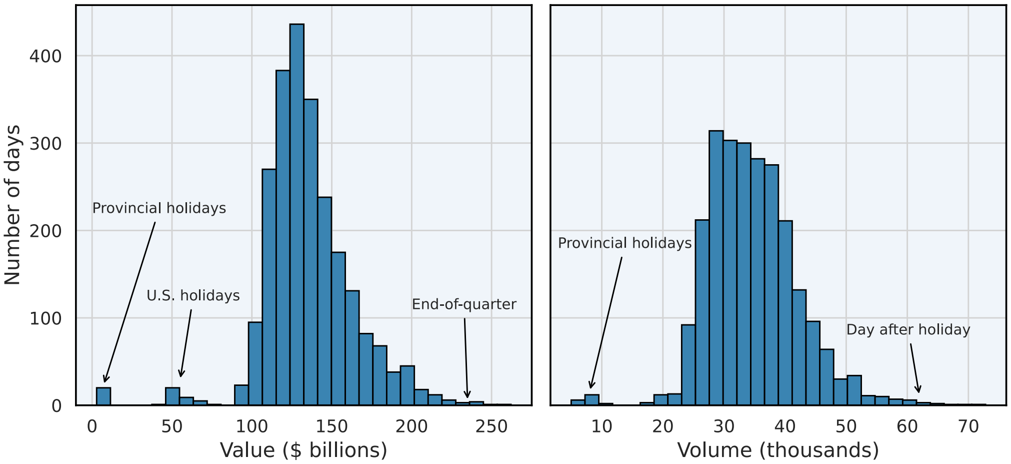
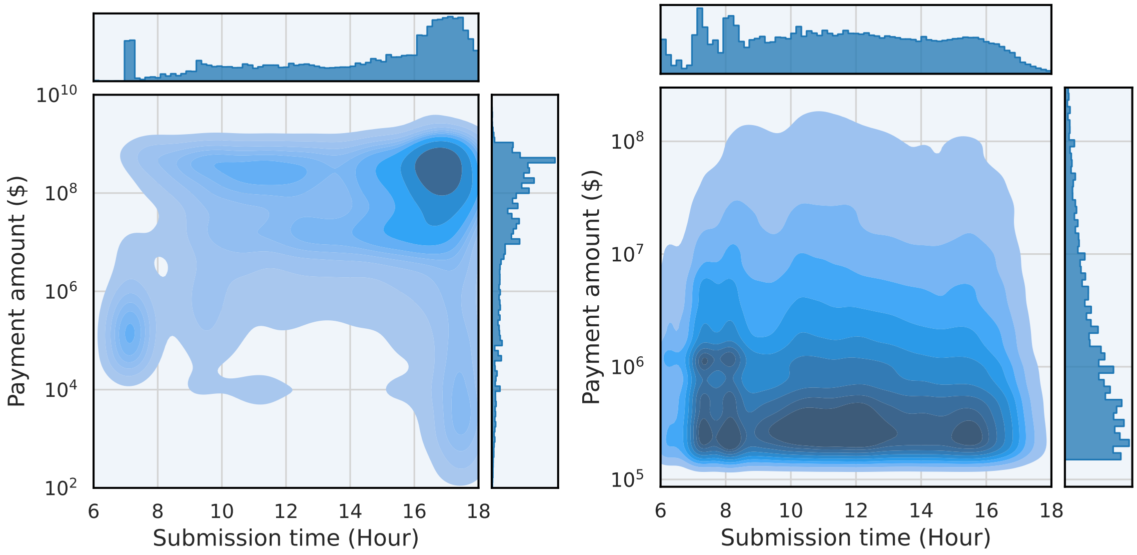


Figure 2: Distribution of daily value (in Can$) and volume (number of transactions) settled in LVTS between January 2012 and August 2021. Arrows highlight special days, such as Canadian provincial holidays, US national holidays, and days after a holiday.

Our sample contains various "special days." These are days when the system is open but only processing a small number of payments, such as Canadian provincial holidays and US national holidays. We call these "partial holidays." Also, there are days when the total value of payments is higher than usual, such as days after national holidays (which we call "post-closure" days) and at the end of the quarter. These special days explain the tails in Figure 2. Other special days observed in our sample are days with operational incidents. On these days, one or more LVTS participants could not send and/or receive payments for a certain amount of time because of technical difficulties. Also note that our sample covers the onset of the Covid-19 pandemic, during which participants changed their payment behavior because of policy changes (Chaudhry et al. 2021). As further discussed below, we use the data from these different types of "special days" to test our models' ability to identify changes in payment patterns and to isolate anomalous transactions.

LVTS settles transactions through two tranches: tranche 1 (LVTS-T1) and tranche 2 (LVTS-T2). The settlement mechanism of LVTS-T1 is similar to a traditional real-time gross settlement (RTGS) system in that transactions are settled using the senders' own collateral. By contrast, LVTS-T2 has components of an RTGS and a deferred net settlement (DNS) system that settles transactions using a joint collateral pool extended by the receiving participants (Arjani and McVanel 2006). Given the fundamental difference between these two LVTS tranches, we divide the dataset into LVTS-T1 and LVTS-T2 subsets. Figure 3 shows how the payment patterns differ between the two tranches. LVTS-T1 is primarily used for high-value payments, which are mainly submitted during the final opening hours of the system. By contrast, LVTS-T2 settles lower-value payments, the submissions of which are spread throughout the day. In our analyses, we include all LVTS-T1 payments[14] and only those LVTS-T2 that are larger than 150,000 Can$. We do so for reasons of simplicity and to demonstrate the effectiveness of our framework in capturing especially high-value anomalous payments, as these will have the largest impact if left undetected.



(a) LVTS-T1

(b) LVTS-T2

Figure 3: Marginal and joint distribution of payment amount and submission time by LVTS tranche. Darker areas in the joint distribution indicate a higher density of payments. Note that the payment amount is based on a log scale.

4.2 Lynx Transaction Data

In August 2021, Lynx replaced LVTS and became Canada's new HVPS. Lynx is an RTGS and offers a new centralized queuing and liquidity savings mechanism (LSM). Also, Lynx's queuing policy is designed to encourage the use of the LSM. This, in turn, is expected to influence participants' payment patterns (Desai et al. 2023). We use Lynx data from October 2021 to August 2023 for our analysis. During this period, Lynx settled an average of 40,000 transactions per day, most of which were settled using the LSM. Note that the majority of this sample coincides with the Covid-19 pandemic. As we do with the LVTS-T2 data, we only focus on those Lynx transactions that are larger than 150,000 Can$.

4.3 Transaction Features

For each payment in our LVTS and Lynx datasets, we extract a set of features that we use in layer 1 for the *payments classifier* to classify payments and in layer 2 for isolating anomalies. These features include basis payment characteristics, as well as more complex indicators, such as the temporal relation between payments and the liquidity position of participants. We believe that these features effectively capture the intricate dynamics of payment patterns. See Table 1 for the comprehensive list of transaction features used in our analyses. These features can be categorized into the following subsets:

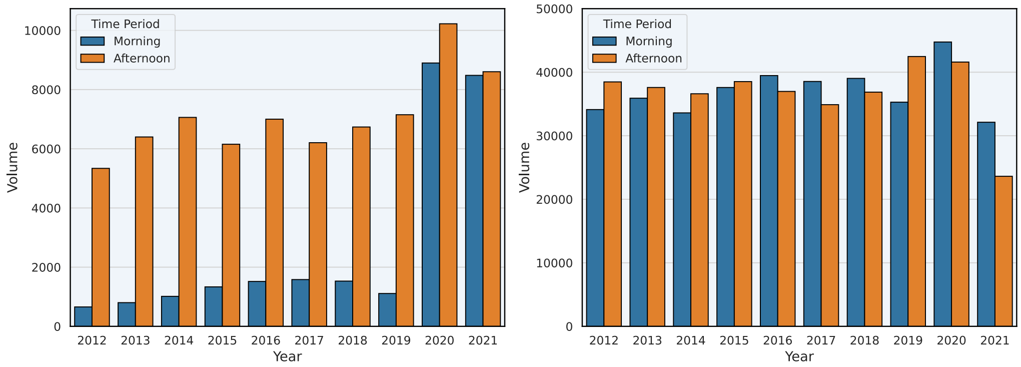
- Basic transaction features, such as sender, receiver, and amount.

[14] In the LVTS-T1 sample we do not include the payments sent by the Bank of Canada.

- Liquidity features, such as total collateral pledged by the sender, sender's credit limits, and system-level liquidity.
- Timestamp features, such as the month of the year, week of the year, and day of the month.
- Intraday features, such as the time elapsed since the last payment was sent to any receiver (or the same receiver), the time elapsed since the last payment was received (from any receiver or the same receiver), and the time elapsed since the start of the current period.

Certain transaction features, such as sender, receiver, and payment type, are categorical, whereas others, such as payment amount, collateral, and liquidity, are continuous of nature. Additionally, certain features, such as collateral, overnight rate, and specific timestamp features, do not change throughout the day. The mixed nature of these features introduces additional complexity during the training of the model, which may affect the performance of traditional models (Hastie et al. 2009). It is also for this reason that we use decision-tree-based models.

We use "period" as the target variable for the payments classifier. Payments submitted between 6 am and noon are classified as morning payments, and those submitted between noon and 6 pm are defined as afternoon payments. Figure 4 shows the yearly number of morning and afternoon payments by LVTS tranche. Before the start of the Covid-19 pandemic, LVTS-T1 payments were mostly submitted in the afternoon. However, during the pandemic, participants started to submit their payments both in the morning and afternoon (Figure 4, left). LVTS-T2 transactions have generally been submitted both in the morning and afternoon over the entire sample period, except for some changes in 2021 (Figure 4, right).



(a) LVTS-T1

(b) LVTS-T2

Figure 4: Number of payments submitted in the morning versus afternoon. The LVTS-T1 sample contains all LVTS-T1 payments. The LVTS-T2 sample includes all LVTS-T2 payments with a value greater than 150K Can$.

4.4 Training and Test Data

To test the ML models' performance in layer 1 and layer 2 of our framework, we divide our data samples into training and test sets. The training set is used to estimate the ML models and to tune their parameters. The "special days" are excluded from this set. By contrast, the test set is used for the out-of-sample evaluation of the models and therefore includes all transactions, those from both "normal" and "special" days. By doing so, the payments classifier is trained to classify the training data as accurately as possible and then tested out of sample on a wider range of transactions.

Table 1: List of features extracted for each of the transactions settled in LVTS.

Transaction feature	**Description**
Binary target variable: Period	Submission time: 6am–noon (morning) or noon–6pm (afternoon)
Sender	ID of LVTS participant sending the payment
Receiver	ID of LVTS participant receiving the payment
Amount	Amount of the transaction (in Can$)
Rounding	Rounded (zero cents) vs non-rounded payments
Payment type	Type 1 for interbank and type 2 for client-driven payments
Overnight loan	Whether the payment is for an overnight loan
Loan repayment	Whether the transaction is a repayment of an overnight loan
Settlement balance	Size of overnight settlement balance of sender (in Can$)
Collateral	The collateral pledged by the sender
Liquidity	System-wide available liquidity (for all LVTS participants)
BCL	Sender's bilateral credit limit for LVTS-T2 payments
MCL	Sender's multilateral credit limit for LVTS-T1 and LVTS-T2 payments
Rate	Overnight money market rate on previous day
Settlement time: Month	Month in which the payment was settled
Settlement time: Month-day	Day on which the payment was settled
Settlement time: Week	Week of the year the payment was settled
Settlement time: Week-day	The specified day of the week
Settlement time: Hour-time	Seconds elapsed since the start of the latest hour
Settlement time: Period-time	Seconds elapsed since the start of the period (morning or afternoon)
Multilateral-time-sender	Seconds elapsed since the last payment sent
Multilateral-time-receiver	Seconds elapsed since the last payment received
Multilateral-type-time-sender	Seconds elapsed since last same-type payment sent
Multilateral-type-time-receiver	Seconds elapsed since last same-type payment received
Bilateral-time-sender	Seconds since last payment sent from same sender to same receiver
Bilateral-time-receiver	Seconds since last payment received by same sender from same receiver
Bilateral-type-time-sender	Seconds since last same-type payment from sender to receiver
Bilateral-type-time-receiver	Seconds since last same-type payment received from sender to receiver

As a second testing procedure, we also run the models on a sample of artificial anomalies. These anomalies are generated by manually altering certain transaction features, such as increasing the actual payment amount or changing the original submission time from morning to afternoon. These intentionally designed artificial transactions exhibit significant deviations from typical transaction behavior, allowing us to assess the model's ability to detect such outliers. For further details on the artificial transactions sample, see Appendix C.

5 Results of Proposed Framework Using Canadian LVTS-T1 Data

5.1 Layer 1 - Payments Classification

We estimated the *payments classifier* with LVTS-T1 settlement data from January 2012 to August 2021, using the model outlined in section 3.1 and the training procedure described in section A.1. Subsequently, we tested its out-of-sample performance with artificial anomalies on a sample containing both normal and special days (as described in section 4.4).

Table 2: Out-of-sample model performance using different models[*]

Model	**Accuracy (in %) on clean data**[a]	**Detection rate (in %) on artificial data**[b]
Logistic Regression	67.8	66.4
Decision Tree	92.3	82.6
Gradient Boosting	94.6	76.8
Random Forest	96.7	94.4
Light Gradient Boosting	97.6	92.2

[*] All models are trained using LVTS-T1 settlement data from January 2012 to August 2021 on clean (normal) days.

[a] Out-of-sample accuracy of models when classifying transactions into morning and afternoon payments on normal or clean days only. Note that the accuracy provides a general measure of how well a binary classification model is performing. It is calculated as the ratio of correctly predicted transactions (from both the morning and afternoon periods) to the total number of transactions in the sample.

[b] Ability to detect artificially manipulated transactions out of sample, calculated as the fraction of manually altered transactions correctly identified by the classifier.

The results demonstrate that the gradient-boosting-based LightGBM model outperforms the logistic regression model by up to 44% when classifying payments into morning and afternoon payments (Table 2). The LightGBM also performs better than other ML models, such as the decision tree, random forest, and gradient boosting. Specifically, it achieves 97.6% accuracy during the out-of-sample testing on the clean sample, with only 2.4% of transactions being misclassified (see column one). Moreover, when testing the model with artificially manipulated anomalous transactions (column two), LightGBM correctly detects 92.2% of payments out of sample, which is slightly lower than the random forest model, which achieved 94.4% accuracy, but better than other models.

The misclassification rate of LightGBM increases when using the datasets that include special days (Figure 5, left). Approximately 10% of payments are misclassified when tested on the dataset that includes partial holidays. This percentage increases to 46% when testing it on the dataset covering the Covid-19

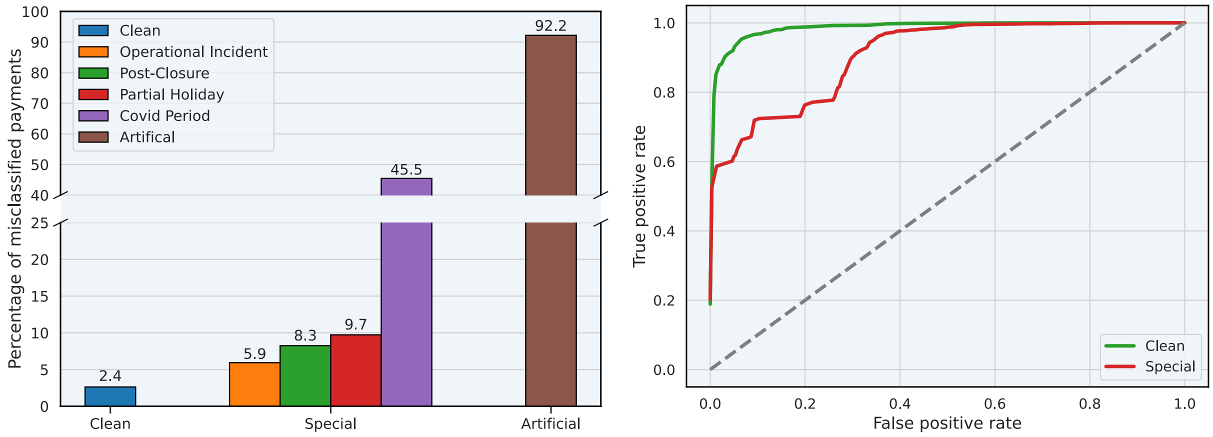


Figure 5: Left: Out-of-sample performance of payments classifier using LightGBM, as measured by the fraction of misclassified payments on the clean (normal) days, special days, and a sample of artificial transactions. Right: Receiver operating characteristic (ROC) curve showing the LightGBM's performance at various threshold values between 0 and 1.

period. This demonstrates that LVTS participants exhibited different payment patterns on these days and during this period, and proves that the model is capable of learning participants' "usual" payment behavior and detecting deviations in it.

An alternative way to assess the model's performance is with the receiver operating characteristic (ROC) curve (Figure 5, right). The fact that the area underneath the green curve is larger than that underneath the red curve reveals that the model performs significantly better when using the clean (normal) data instead of the data with special days. This again demonstrates the model's ability to detect deviations from LVTS participants' typical payment behavior.

5.2 Layer 2 - Anomaly Detection

As proposed in Section 3.2, we subsequently use the set of misclassified transactions to train the isolation forest (IF) algorithm to detect anomalies. We assess the model using artificially manipulated transactions and their corresponding original counterparts (i.e., the real transactions).

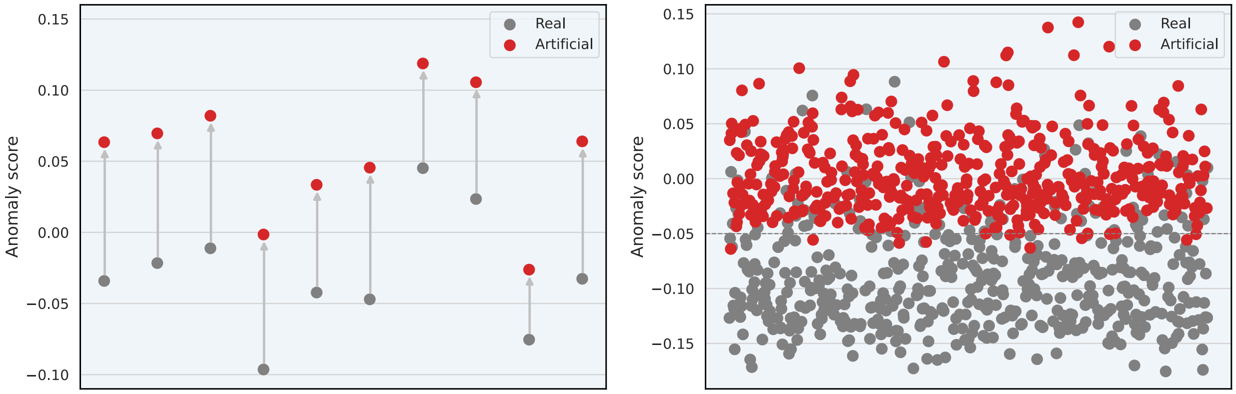


Figure 6: Anomaly detection using the IF model on LVTS-T1 data: Anomaly scores when using the set of transactions before (real, shown in gray) and after (artificial, shown in red) manipulation. A transaction with a high (positive) score is likely more anomalous than a transaction with a low (negative) score.

The anomaly scores generated by the IF algorithm are useful in evaluating the model's performance: transactions with higher and positive anomaly scores indicate greater anomalous behavior compared to those with lower and negative scores. The results presented in Figure 6 show that the IF algorithm successfully assigns higher anomaly scores to the manually generated anomalous transactions than to the real transactions. On average, the model assigns approximately a twice-as-high anomaly score to manually altered transactions than to the original data (see right panel). As such, the IF algorithm proves to be a successful tool to detect unusual transactions and to prioritize further investigation of the most likely suspicious transactions.[15]

5.3 Interpretation of Layer 1 Model Predictions - Payments Classifier

5.3.1 Role of payment features in participants' usual payment behavior

The training results of the payments classifier also allow us to gain further insights into how specific payment features influence the submission time of HVPS payments. The average SHAP values across the training sample for each feature as depicted in Figure 7 show that the time elapsed since the previous payment received and the time elapsed since the start of the period contribute most towards the model's capability to correctly predict payments. Other important predictors are the amount of the payment, the amount of collateral pledged by the payer, and recipient-specific features. The model's reliance on the time elapsed since the previous payment received suggests that incoming payments play an important role in participants' behavior. This aligns with earlier research that demonstrates the presence of substantial payment coordination between HVPS participants (see e.g., McAndrews and Rajan 2000; Becher et al. 2008; Alexandrova-Kabadjova et al. 2023).

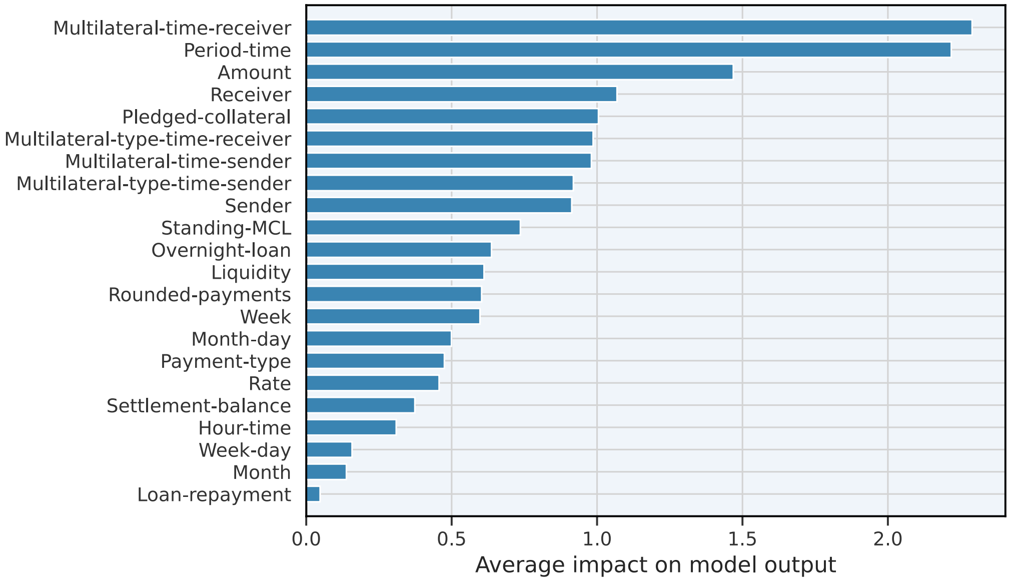


Figure 7: LVTS-T1: SHAP global feature importance, measured by the mean absolute SHAP values for each transaction in the training sample. The features are ranked from high (top) to low (bottom) based on the average impact on the model outcomes.

Figure 8 illustrates how the predictions of the payments classifier are influenced by the amount and type of the payment. Customer-driven payments (type 2, shown in red) are more likely to be classified as morning

[15] Note that the accuracy is reduced when relying solely on the unsupervised IF model without using the preceding supervised classifier model in the first layer . See Appendix B.2 for further details.

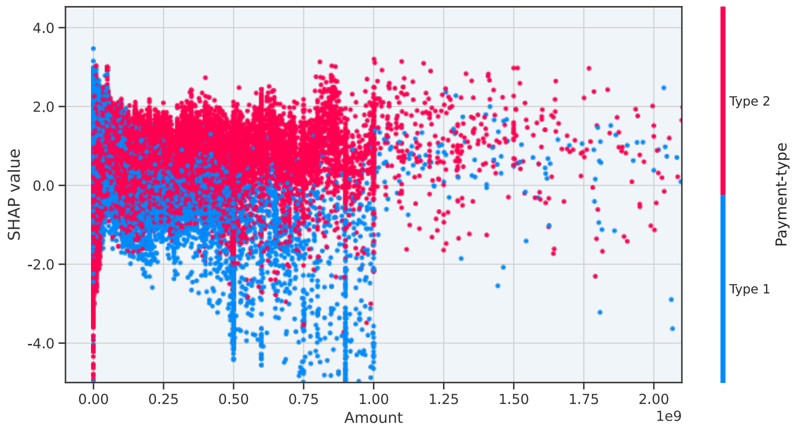


Figure 8: Dependence plot showing the SHAP values for payment amount (in Can$) by payment type for each transaction. Type 1 are inter-bank payments (shown in blue) and type 2 are client-driven payments (shown in red).

payments (negative SHAP value) when small, whereas they are predicted as afternoon payments when large (positive SHAP value). By contrast, interbank payments (type 1, shown in blue) are more commonly submitted in the afternoon when small and sent in the afternoon when large. Interestingly, for payments larger than 1 billion Can$, the payment type no longer strongly contributes to the model's prediction. Most high-value payments of either type tend to be submitted in the afternoon, with the exception of a few large interbank payments being submitted in the morning.

5.3.2 Role of Payment Features in the Prediction of Individual Payments

Figure 9 shows the SHAP force plots for one particular transaction, with ($f(x)$) being the prediction score. A negative score indicates the morning period, while a positive score indicates the afternoon. The upper plot is based on the actual features of the transaction, whereas the bottom plot was generated after manipulating its period-time feature (from 7750 to 100 seconds). The actual submission period (morning) is correctly predicted by the model, as indicated by the negative (-4.25) prediction score. The bottom panel, however, shows that the model successfully misclassifies and predict the manipulated transaction as an afternoon payment (with a higher and positive score of 3.02). Moreover, the large red "period-time" arrow pushes the model towards an afternoon prediction, which indicates that it is this period-time feature that makes the model predict it as an afternoon payment.

Similarly, Figure 10 shows the force plots comparing an actual transaction (upper panel) with one where we manually lowered its amount (from 11.8 thousand to 5.2 billion) (bottom panel). The model correctly classifies the actual transaction as a morning transaction (with a negative score of -10.34) and misclassifies the manipulated one as an afternoon transaction (with a positive score of 0.21). The payment amount is indeed the most prominent feature driving this misclassification, as illustrated with the relatively large right-pushing arrow.

These examples indicate that the payments classifier model has successfully learned the typical payment behavior of LVTS-T1 participants and that it is able to identify transactions that deviate from this typical behavior. Moreover, the SHAP values provide a useful tool to better understand why the model is predicting a particular payment as such. This not only provides insight into the drivers behind LVTS-T1 participants' true behavior, but also serves as a robustness check of the model's specification.

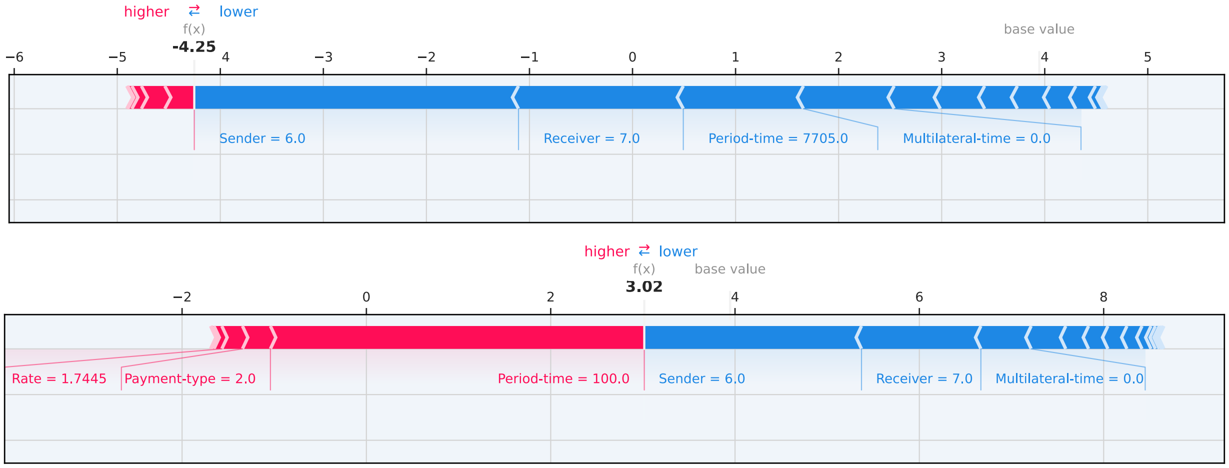


Figure 9: LVTS-T1: SHAP force plots showing the marginal contribution of each predictor for a chosen transaction. The red arrows are positive SHAP values (pushing the prediction towards an "afternoon" prediction), and the blue arrows are negative SHAP values (pushing the prediction towards the morning period). $f(x)$ is the prediction score generated by the model for that transaction, and the base value is the average of all predictions over the entire sample. The feature values shown in red and blue are the predictor values for that transaction. The original transaction was settled in the morning and correctly classified as such by the model (see negative prediction score in the upper panel). After manipulating the period-time feature, the transaction is misclassified as an afternoon payment (see positive prediction score in the bottom panel), which is mainly driven by the amended period-time.

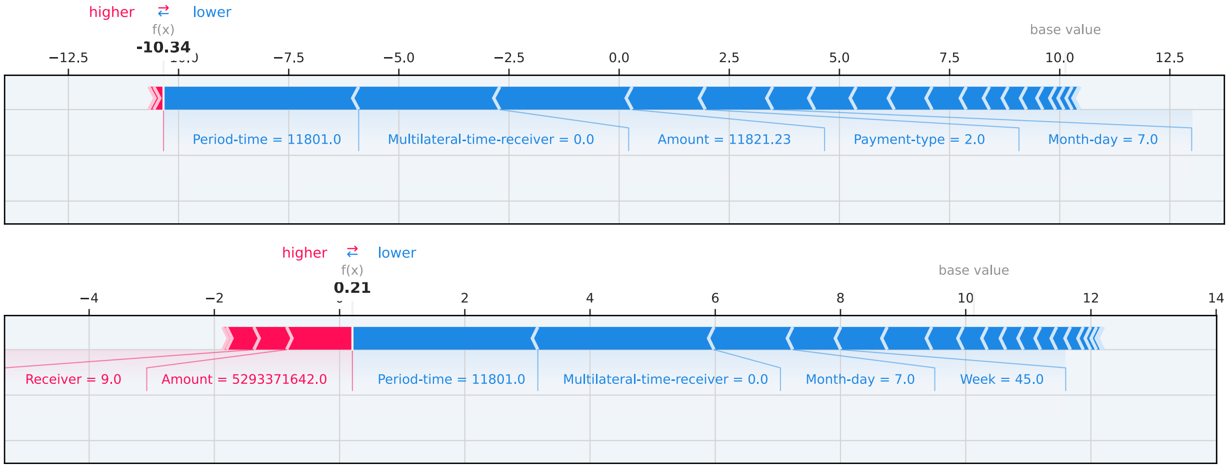


Figure 10: LVTS-T1: SHAP force plots showing the feature contribution for a chosen transaction. The original transaction was settled in the morning and correctly classified as such (see negative prediction score in upper panel). After manipulating the payment value, the model classifies the transaction as an afternoon payment (see positive prediction score at the bottom). The payment value is the dominant feature driving the misclassification.

5.4 Interpretation of Layer-2 Model Predictions - Anomaly Detection

From Figure 11 it is evident that the intraday time features contribute the most to the identification of anomalous payments. This is slightly different from the features contributing to the outcomes of the payments classifier in the first layer (see Figure 7). This suggests that the more complex intraday time features, such as the time passed since the last incoming payment of the same kind, the time elapsed since the last outgoing payment, and the time since since the last incoming payment of any kind, are crucial for identifying and isolating anomalies from the subset of misclassified transactions. Other features that the IF model heavily draws on when identifying anomalies include the payment value, the credit position of the sender, and the liquidity available in the system.

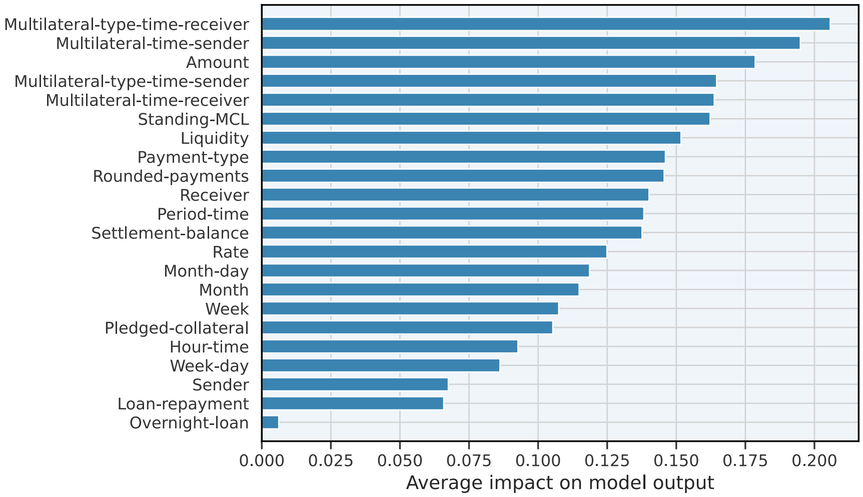


Figure 11: Anomaly detection model using LVTS-T1 data: SHAP global feature importance measured as mean absolute SHAP values for each transaction in the misclassified subset of transactions. The features are ranked from high (top) to low (bottom), based on the average impact on the model's output.

6 Results of Alternative Applications

6.1 Model Training and Evaluation Using Joint Payment Classifiers

To demonstrate the flexibility of our two-layered framework, we introduced a second payments classifier module that operates parallel to the original classifier. This effectively creates a joint classifier. The second classifier is trained using the payment type (interbank vs client-driven) as the target variable,[16] while the first classifier still focuses on the submission period (morning vs afternoon). As such, the joint classifier classifies payments based on both the payment timing and the type of payment. It can be formulated as follows: let $\mathbf{y_1}$ be the first target variable (submission time) and $\mathbf{y_2}$ the second target variable (payment type). We can denote $\hat{\mathbf{y}}_1$ and $\hat{\mathbf{y}}_2$ as the predicted targets, which can be estimated using the classification models $f_1 : \hat{\mathbf{y}}_1 = f_1(X)$ (classifier 1) and $f_2 : \hat{\mathbf{y}}_2 = f_2(X)$ (classifier 2).

We train and test the joint-classifier model on the LVTS-T1 dataset using the same procedures as above and as visualised in Figure 12: all transactions that are misclassified based on either one of the target variables are automatically sent to the anomaly detection module. Subsequently, only those correctly classified based

[16] See Appendix A.2 for more details on the second payments classifier type.

on both dimensions are used for pattern recognition. The results (see Appendix A.2) demonstrate how using classifiers that capture multiple payment features at the same time can enhance the model's accuracy and robustness. In our tests, the joint classifier was able to correctly catch 96% of the manipulated transactions, which is higher than using either the original (92%) or the alternative classifier (66%) independently. The use of multiple payments classifiers can also improve the performance of the anomaly detection model in the second layer. On average, the IF model assigns a 13% higher anomaly score to manually altered transactions when used in conjunction with the joint-classifier model, compared to the standalone classifier-1 model.

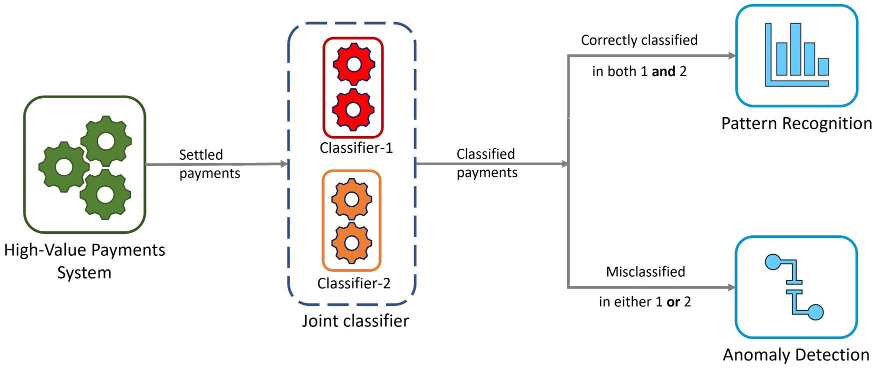


Figure 12: Schematic overview of alternative layered approach for pattern recognition and anomaly detection using a joint classifier. The payments classifier 1 is trained with the submission time (morning vs afternoon) as the target variable, while classifier 2 is trained with the payment type (interbank vs client-driven) being the target variable.

6.2 Model Training and Evaluation Using LVTS-T2 Data

To assess the extent to which the proposed single-classifier model can be applied to payment systems or settlement mechanisms other than LVTS-T1, we trained and tested it on the LVTS-T2 dataset, using the data and procedure outlined in sections 4.1 and A.1. Overall, the model is able to detect 96.4% of the artificially manipulated LVTS-T2 transactions (Figure 13, left panel). These results indicate that the binary-payments-classifier model is able to handle artificial transactions pretty well when applied to another type of settlement mechanism. However, its ability to capture more subtle deviations in payment patterns, such as those on special days, is reduced when testing it beyond the LVTS-T1 data: the misclassification rate using the clean sample (14%) is only slightly better than when using the special days samples (15% to 19%). In particular, the number of misclassified payments using the clean sample is higher when using the LVTS-T2 data (14%) than when using the LVTS-T1 set (2.4%). This could be attributed to the sheer volume and variety of payments processed through LVTS-T2. So although our proposed classifier can be applied to payment systems beyond that of LVTS-T1, the set of transaction features used for the classification would need to be tailored to generate the same level of performance.

The time elapsed since the start of the period is the most important transaction feature that the model uses to classify payments, followed by the time elapsed since the last incoming payment from the same counterpart. This "bilateral" transaction feature is more influential than the "multilateral" transaction features that were prominent when using the LVTS-T1 sample. This highlights that bilateral relationships are important in LVTS-T2, with participants attaching higher value to coordinating payments on a bilateral than on a multilateral basis (as in LVTS-T1). This is in line with what one would expect, as the risk management in LVTS-T2 is heavily based on bilateral credit limits (BCL) (see Arjani and McVanel (2006)).

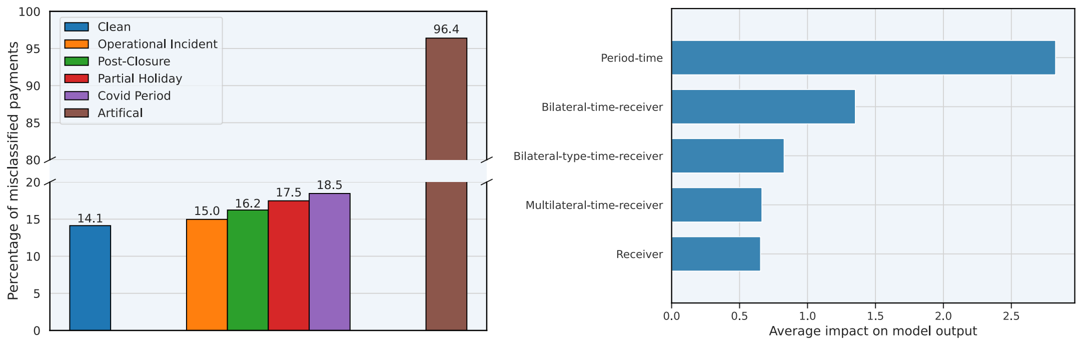


Figure 13: Results using LVTS-T2 data. Left panel: Out-of-sample performance of payments classifier using LightGBM, as measured by the fraction of misclassified payments on the clean (normal) days, special days, and a sample of artificial transactions. Right: Key features contributing to the LightGBM model's predictions based on mean absolute SHAP values.

6.3 Model Training and Evaluation on Lynx

We also trained and tested the single-classifier model on the Lynx dataset. When interpreting the results and comparing these with the results of the LVTS datasets, it is important to keep in mind that (i) Lynx is an RTGS system with an LSM, so of a different design than both LVTS-T1 and LVTS-T2, (ii) the size of the sample is much smaller and only ranges from October 2021 to August 2023, and (iii) the Lynx data contains all HVPS transactions in Canada, i.e., all transactions that used to be sent through either LVTS-T1 or LVTS-T2 (Desai et al. 2023).

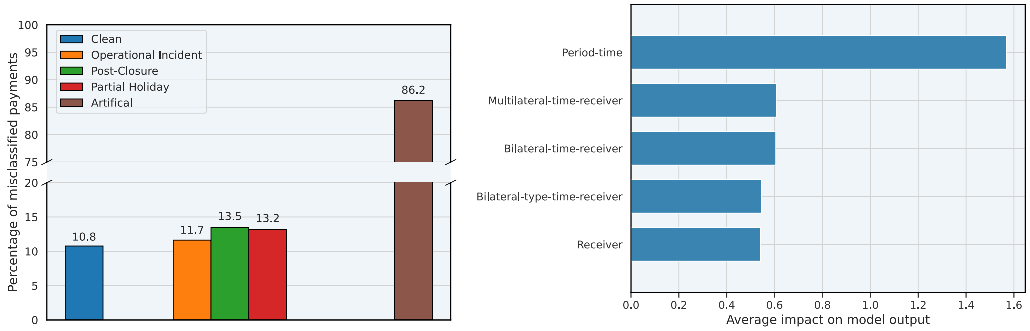


Figure 14: Results using Lynx data. Left panel: Out-of-sample performance of the payments classifier using LightGBM, as measured by the fraction of misclassified payments on the clean (normal) days, special days, and a sample of artificial transactions. Right: Key features contributing to the LightGBM model's predictions based on mean absolute SHAP values.

Overall, our LightGBM model is able to detect 86.2% of the artificially manipulated transactions when applying it to the Lynx data. This is pretty good, especially given the newness of the system and the small size of the dataset, which make it more challenging to effectively capture usual payments behavior. This may also explain why the payments misclassification rate is only slightly lower when using the normal sample than when including the special days (Figure 14, left).

The right panel of Figure 14 shows the key features that the Lynx payments classifier relies on during its training. Like the LVTS-T2 sample, "period-time" is the most prominent feature. However, in Lynx, not only the "bilateral" timing features but also the "multilateral" timing features are key predictors that the payments classifier relies on when classifying payments into morning and afternoon payments. This is in line with the RTGS nature of Lynx (Desai et al. 2023).

7 Conclusions and Practical Implementation

In this paper we propose a novel ML framework for real-time transaction monitoring in HVPSs. We use advanced data-driven ML algorithms to learn the "typical" payment patterns of HVPS participants and to isolate anomalous transactions when deviations are spotted. The key strength of our framework is that it is able to address the challenging task of identifying anomalies in large and high dimensional sets of payments data, particularly when anomalies are not known a priori. The framework derives its strength from its layered approach. In the first layer, we train a supervised ML-based payments classifier to efficiently screen typical transactions. This payments classifier then streamlines the subsequent anomaly detection task in the second layer, for which we use an unsupervised ML-based IF model. We show that our framework can be applied for different types of HVPSs. Also, we demonstrate that it can be extended with additional algorithms, such as with joint classifiers, to further enhance its robustness.

Due to its flexibility, the ML framework proposed in this paper can be used for different applications in finance. Payment system operators and overseers may implement it to detect cyber attacks or operational outages, which, if left undetected, could have serious implications for the HVPS, its participants, and the financial system more broadly. Moreover, the framework could be used to detect early signs of financial stress at individual institutions or the unusual behavior of their clients. Recent bank runs that triggered the failure of Silicon Valley Bank and Signature Bank in the United States and the acquisition of Credit Suisse in Switzerland, for example, emphasized the importance of timely monitoring and heightened vigilance in order to safeguard financial market infrastructures. While trained on HVPS data, our framework could also be applied to other types of payment systems that generate large and highly dimensional datasets. For example, banks and other payment service providers could use it for anti-money laundering transaction screening to identify suspicious behavior and financial crime.

From a practical standpoint, the initial training of the ML models in our framework may demand a substantial time investment and computational resources. The models may also require regular retraining and updates to ensure that they remain attuned to evolving payment patterns. That said, once in place and up to date, they allow for real-time (fast) and computationally efficient (utilizing fewer resources) monitoring and predictions. Moreover, the Shapley value-based (SHAP) tools can assist operators in their regular evaluation of the framework and provide them with an initial indication of where flagged anomalies may be coming from or what they are hinting at, thus aiding a timely identification and resolution.

Despite the numerous advantages of our ML framework for HVPS transaction monitoring, there is room for improvement, which can be explored in future research. In particular, further work may consider some of the challenges that have surfaced in our work. First, the number of transaction features automatically increases with the number of HVPS participants and payment types, which increases the complexity of model training and the predictions. This scalability challenge could be addressed by applying feature selection techniques or by exploring alternative approaches that handle large sets of features. Second, the classification of a large number of anomalous transactions could prevent payment system operators from reviewing them and taking action in a timely manner. This may be addressed by improving the accuracy of the model through feature engineering and additional training. Finally, it could be challenging to accurately classify edge cases along the classification boundary, such as, in our case, payments close to noon. This could be addressed by using a regression model in the first layer of the framework.

References

Alexandrova-Kabadjova, B., A. Badev, S. B. Bastos, E. Benos, F. Cepeda-Lopez, R. Garratt, R. Heijmans, A. Kosse, A. Martin, T. Nellen, T. Nilsson, J. Paulick, A. Pustelnikov, F. Rivadeneyra, M. R. do Coutto Bastos, and S. Testi (2023). Intraday liquidity around the world. *BIS Working Paper* (No 1089). doi: www.bis.org/publ/work1089.htm.

Alexandrova-Kabadjova, B., A. Serguieva, R. Heijmans, and L. Garcia-Ochoa (2015). Direct participants' behavior through the lens of transactional analysis: The case of SPEI. In *Proceedings of the International Conference on Social Modeling and Simulation*, pp. 205–215.

Arjani, N. and R. Heijmans (2020). Is there anybody out there? Detecting operational outages from large value transfer system transaction data. *Journal of Financial Market Infrastructures 8*(4). doi: 10.21314/JFMI.2019.118.

Arjani, N. and D. McVanel (2006). A primer on Canada's large value transfer system. Technical report, Bank of Canada. https://www.bankofcanada.ca/wp-content/uploads/2010/05/lvts_neville.pdf.

Arjani, N., L. Sabetti, and F. Li (2020). Monitoring intraday liquidity risks in a real-time gross settlement system. *Journal of Financial Market Infrastructures 9*(3). doi: 10.21314/JFMI.2021.012.

Arévalo, F., P. Barucca, I.-E. Téllez-León, W. Rodríguez, G. Gage, and R. Morales (2022). Identifying clusters of anomalous payments in the Salvadorian payment system. *Latin American Journal of Central Banking 3*(1). https://www.sciencedirect.com/science/article/pii/S2666143822000059.

Bech, M. L. and R. Garratt (2003, April). The intraday liquidity management game. *Journal of Economic Theory 109*(2), 198–219.

Bech, M. L. and R. J. Garratt (2012). Illiquidity in the interbank payment system following wide-scale disruptions. *Journal of Money, Credit and Banking 44*(5), 903–929.

Becher, C., M. Galbiati, and M. Tudela (2008). The timing and funding of CHAPS Sterling payments. *FRBNY Economic Policy Review 14*(2), 113–133.

BIS-Report (2019). Reducing the risk of wholesale payments fraud related to endpoint security: A toolkit. Technical report, Bank for International Settlements. https://www.bis.org/cpmi/publ/d188.pdf.

Bukth, T. and S. S. Huda (2017). *The soft threat: The story of the Bangladesh bank reserve heist.* SAGE Publications: SAGE Business Cases Originals.

Castro, P. S., A. Desai, H. Du, R. Garratt, and F. Rivadeneyra (2021). Estimating policy functions in payments systems using reinforcement learning. Technical report, Bank of Canada Working Paper 2021-7.

CEIP (2021). Timeline of cyber incidents involving financial institutions. Technical report, Carnegie Endowment for International Peace. https://carnegieendowment.org/specialprojects/protectingfinancialstability/timeline#MexicanBankTheft2018.

Chapman, J., J. Chiu, S. Jafri, and H. Pérez Saiz (2015). Public policy objectives and the next generation of CPA systems: An analytical framework. Technical report, Bank of Canada.

Chaudhry, A., A. Kosse, and K. Sondergard (2021). Behaviour in the Canadian large-value payment system: Covid-19 vs. the global financial crisis. Technical report, Bank of Canada.

Chen, T., T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, et al. (2015). Xgboost: Extreme gradient boosting. *R package version 0.4-2 1*(4), 1–4.

Desai, A., Z. Lu, H. Rodrigo, J. Sharples, P. Tian, and N. Zhang (2023). From LVTS to Lynx: Quantitative assessment of payment system transition in Canada. *Journal of Payments Strategy & Systems 17*(3), 291–314.

Docherty, P. and G. Wang (2010). Using synthetic data to evaluate the impact of RTGS on systemic risk in the Australian payments system. *Journal of Financial Stability 6*(2), 103–117.

Eisenbach, T. M., A. Kovner, and M. J. Lee (2021). Cyber risk and the US financial system: A pre-mortem analysis. *Journal of Financial Economics 145*(3), 802–826.

FED-Report (2019). Federal Reserve fraudclassifier model. Technical report, Federal Reserve. https://fedpaymentsimprovement.org/.

Flood, M. D., V. L. Lemieux, M. Varga, and B. W. Wong (2016). The application of visual analytics to financial stability monitoring. *Journal of Financial Stability 27*, 180–197.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics. New York: Springer. doi: 10.1007/978-0-387-84858-7.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29*(5), 1189–1232. doi: 10.1214/aos/1013203451.

Glowka, M. (2019). Profiling banks: How to use cluster analysis with payment system data. *Journal of Financial Market Infrastructures 8*(2), 21–45. doi: 10.21314/JFMI.2019.118.

Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The elements of statistical learning: Data mining, inference, and prediction*, Volume 2. New York, Springer.

James, R., H. Leung, and A. Prokhorov (2023). A machine learning attack on illegal trading. *Journal of Banking & Finance 148*, 106735.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pp. 3146–3154. doi: https://lightgbm.readthedocs.io/en/latest/.

Kosse, A. and Z. Lu (2022). Transmission of cyber risk through Canadian wholesale payments system. Technical report, Bank of Canada Working Paper. https://doi.org/10.34989/swp-2022-23.

Kotidis, A. and S. Schreft (2023). The propagation of cyberattacks through the financial system: Evidence from an actual event. Technical report, Federal Reserve System Working Paper 2022-25.

León, C. (2020). Detecting anomalous payments networks: A dimensionality-reduction approach. *Latin American Journal of Central Banking 1*(1-4), 100001.

León, C., P. Barucca, O. Acero, G. Gage, and F. Ortega (2020). Pattern recognition of financial institutions' payment behavior. *Latin American Journal of Central Banking 1*(1-4), 100011.

LightGBM-Documentation (2022). LightGBM's official documentation. https://lightgbm.readthedocs.io/en/latest/.

Liu, F. T., K. M. Ting, and Z.-H. Zhou (2008). Isolation forest. In *2008 eighth IEEE international conference on data mining*, pp. 413–422. IEEE.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence 2*(1), 2522–5839. doi: 10.1038/s42256-019-0138-9.

Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc.

Martin, A. and J. McAndrews (2008). Liquidity-saving mechanisms. *Journal of Monetary Economics 55*(3), 554–567.

Massarenti, M., S. Petriconi, and J. Lindner (2012). Intraday patterns and timing of TARGET2 interbank payments. *Journal of Financial Market Infrastructures 1*(2), 3–24.

McAndrews, J. and S. Rajan (2000). The timing and funding of Fedwire funds transfers. *Economic Policy Review 6*(2).

McMahon, C., D. McGillivray, A. Desai, F. Rivadeneyra, J.-P. Lam, T. Lo, D. Marsden, and V. Skavysh (2022). Improving the efficiency of payments systems using quantum computing. *arXiv preprint arXiv:2209.15392*.

Molnar, C. (2020). *Interpretable machine learning.* Lulu.com.

Ngai, E. W., Y. Hu, Y. H. Wong, Y. Chen, and X. Sun (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.

Nish, A. and S. Naumaan (2019). *The cyber threat landscape: Confronting challenges to the financial system*. Carnegie Endowment for International Peace. https://carnegieendowment.org/files/02_19_Nish_Naumaan_Fin_Threats_final.pdf.

Osborne, M. J. and A. Rubinstein (1994). *A course in game theory.* MIT Press.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Rubio, J., P. Barucca, G. Gage, J. Arroyo, and R. Morales-Resendiz (2020). Classifying payment patterns with artificial neural networks: An autoencoder approach. *Latin American Journal of Central Banking 1*(1-4), 100013.

Ryman-Tubb, N. F., P. Krause, and W. Garn (2018). How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence 76*, 130–157.

Sabetti, L. and R. Heijmans (2021). Shallow or deep? Training an autoencoder to detect anomalous flows in a retail payment system. *Latin American Journal of Central Banking 2*(2), 100031.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games 2*(28), 307–317.

Slack, D., S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.

TARGET-Report (2019). TARGET annual report. Technical report, European Central Bank. https://www.ecb.europa.eu/pub/pdf/targetar/ecb.targetar2019.en.pdf.

Triepels, R., H. Daniels, and R. Heijmans (2017). Anomaly detection in real-time gross settlement systems. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 3: ICEIS*, pp. 433–441. INSTICC: SciTePress.

Zhang, N. (2015). Changes in payment timing in Canada's large value transfer system. Technical report, Bank of Canada Working Paper. https://www.bankofcanada.ca/2015/06/working-paper-2015-20/.

A Gradient Boosting Machines

Gradient boosting (GB) is a decision tree (DT)-based ensemble learning approach. It is a general technique of boosting in which a sequence of weak learners (e.g., DTs) are built on a repeatedly modified version of the training set. The data modification at each boosting interaction consists of applying weights to each of the training samples, and for successive iterations, the sample weights are modified. Basically, the next learner is fit on the residual of the previous learner (Friedman 2001; Friedman et al. 2001).

Let $X = \mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M$ represent M features, each with N training samples. The target variable y_i denotes the true label for the i-th instance in binary classification (class 1 or 2). Similarly, let p_i represent the predicted probability of class 1 for the i-th instance. Then the objective function L can be defined as

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right].$$

To perform gradient boosting, we need to compute the gradient of the objective function with respect to the predicted probabilities p_i, which is given by

$$\frac{\partial L}{\partial p_i} = -\frac{1}{N}\left(\frac{y_i}{p_i} - \frac{(1-y_i)}{(1-p_i)}\right).$$

To train this model, during each iteration of the gradient boosting process, a weak learner (a DT) is trained using a subset of the dataset at the start to minimize the gradient of the objective function. By computing the difference between predicted values and the actual targets, pseudo-residuals are derived. Subsequently, another weak learner is trained on these pseudo-residuals from the previous step, aiming to minimize prediction errors. This iterative process continues, with each new learner focusing on the combined model's errors from preceding iterations. The predicted probabilities generated by each new weak learner are incrementally added to the predictions from prior iterations using a specified step size (learning rate) to update the model. This iterative procedure persists, often running for a predefined number of boosting rounds or until a designated stopping criterion is reached. Note that the final prediction for binary classification is often obtained by applying a threshold (e.g., 0.5) to the predicted probabilities. If p_i is greater than the threshold, the example is classified as class 1; otherwise, it is classified as class 2.[17]

Tree-based GB is a popular ML algorithm. However, it is computationally expensive in terms of efficiency and scalability when the feature dimension M is high and the sample size N is large. To tackle this problem, Ke et al. (2017) propose an improved and efficient implementation of GB called LightGBM, which (i) uses a histogram-based approach for tree construction to reduce memory usage and speed up training, (ii) employs gradient-based optimization to excludes a significant proportion of data instances with small gradients and speeds up the training process, and (iii) uses an exclusive predictors bundling technique to bundle mutually exclusive features to reduce the number of features. Using these techniques in the LighGBM model, they achieve a 20 times speedup with almost the same accuracy on multiple public datasets compared to the GB model. Moreover, in many cases LighGBM has been shown to perform similar to or, in some cases better than, other boosting algorithms such as XGBoost (Chen et al. 2015). In this paper we use the open-source LightGBM Python-package available at LightGBM-Documentation (2022).

[17] Note that our procedure involves two simplifying assumptions. First, we reduce the granularity of information in the dependent variable by categorizing payments as either morning or afternoon transactions. Second, we set arbitrary thresholds to classify predicted values into morning or afternoon payments, irrespective of the model's confidence level. For instance, even if a payment is identified with a 51% probability of being an afternoon transaction, we classify it as such. We intend to relax some of these simplifications in future work.

A.1 Gradient Boosting Classifier Model Training

The payments classifier is trained using transaction features as predictors and the payments submission period (morning or afternoon) as the target variable. To train a classifier model like LightGBM, a sequence of weak learners (e.g., DTs) is built, with each learner trained on a modified version of the training set. In boosting, the next learner fits on the residual of the previous one, and this process involves applying weights to each training sample. These sample weights are modified in successive iterations (Friedman et al. 2001).

For model training, tuning, and out-of-sample performance evaluation, we use the standard *k*-fold cross-validation technique. This is an iterative re-sampling procedure that uses a small number of subsets (*k*) of the randomly shuffled dataset for in-sample training and out-of-sample testing. This procedure is commonly used to evaluate ML models (Hastie et al. 2009). Figure 15 provides a schematic overview of the *k*-fold cross-validation procedure, using *k* set at 5 for illustrative purposes. The dataset is initially shuffled and then divided into two subsets: a training set (for in-sample model training, tuning, and validation) and a test set (for out-of-sample model prediction and evaluation). Subsequently, the in-sample training set is further divided into five folds, each of which is split into a training subset and a validation subset. In each iteration, a unique validation subset (highlighted in blue) is chosen, while the remaining data forms the training subset. A model is estimated on the training subset, evaluated on the validation set, and the score is retained. The scores gathered from each of the *k* folds contribute to model parameter selection. Finally, the model's performance is tested on the out-of-sample test set (highlighted in orange).

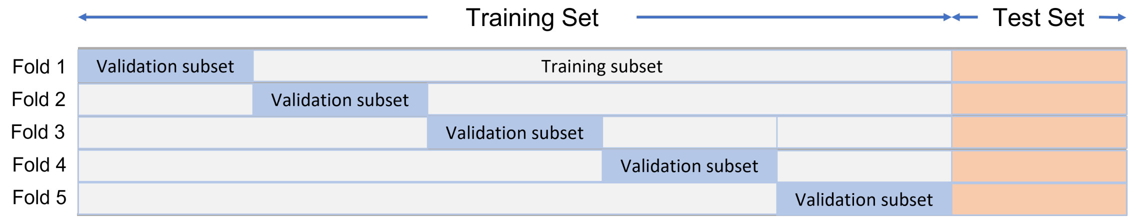


Figure 15: A schematic overview of a *k*=5-fold cross-validation procedure for the training and tuning of the payments classifier model. First, the dataset is divided into a training set and a test set. Next, for each fold, the training set is split into a training subset (highlighted in light gray) and a validation subset (highlighted in blue). Finally, the out-of-sample test set (highlighted in orange) is used for the prediction.

A.2 Model Training and Evaluation Using Alternative Payments Classifiers

In this section, we train an alternate payments classifier using the payment type as the target variable (rather than the payment period). As laid out in section 4, our dataset contains type 1 (interbank) and type 2 (client-driven) payments. We test this alternative payments classifier using the same transaction features and the same LVTS-T1 sample.

Overall, the share of misclassified payments is larger when using the alternative payments classifier compared to the original one, which classified payments based on their submission time. Additionally, the alternative classifier shows smaller differences in misclassification rates between the clean (approximately 10%) and special-days samples (averaging about 14%). Furthermore, the misclassification rate on the artificial sample is around 80%, which is also lower than that of the original classifier. This suggests that the original classifier is more effective in correctly classifying payments for the LVTS-T1 dataset. Consequently, it can be inferred that the payments classifier model is effective but requires a fine-tuning of its parameters, particularly the target variable for the specific system.

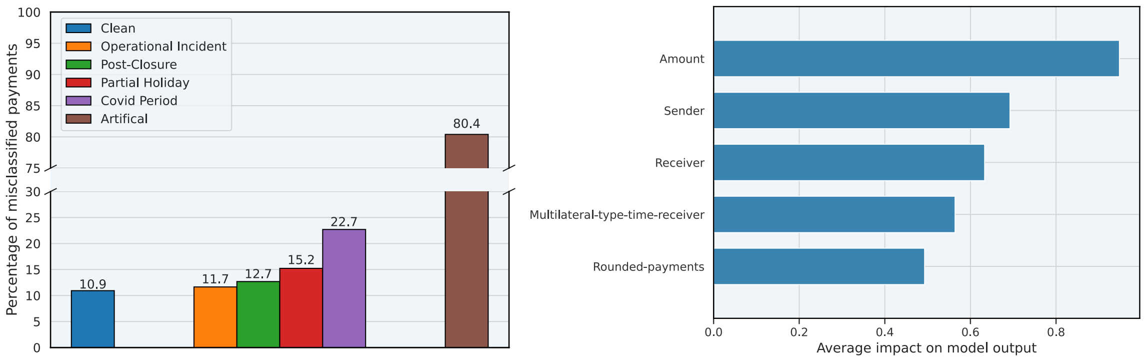


Figure 16: Alternate classifier: (left) Out-of-sample performance of payments classifier measured as the misclassification rate on the clean and special-days transactions. (Right) A few key features were identified and ranked using SHAP global feature importance—measured as mean absolute Shapley values for each instance in the training sample.

The alternative classifier merely learns participants' behavior based on some of the same key features as the original classifier, such as the payment value, sender, receiver, and multilateral-time (see Figure 16, right panel). However, there is also a notable difference; the alternative classification based on the type of payment is also driven by whether the transaction is rounded or not. This reflects the fact that the model correctly captured the fact that most type 1 (interbank) payments are rounded.

B Isolation Forest

The isolation forest (IF) is an unsupervised anomaly detection algorithm introduced by Liu et al. (2008). Unlike other modeling approaches that focus on profiling normal instances to identify anomalies, the IF focuses on directly isolating anomalous instances. It is designed with using the concept that anomalies are rare and distinct points in the feature space. The IF algorithm relies on decision trees known as isolation trees (iTrees). The number of partitions required to isolate a point can be interpreted as the length of the path from the root to a terminating node within the tree.

The IF algorithms can be explained using the following steps (see original paper for complete algorithm (Liu et al. 2008): 1. From a given training dataset, a random sub-sample is selected and assigned to an iTree. 2. By selecting random subsets of features from the chosen sub-sample of data, the iTrees are grown using a random threshold at each split (any value in the range of minimum and maximum values of the selected features). 3. During each split, if the value of a chosen data point is less than the selected threshold, it goes to the left branch otherwise it goes to the right. This process is repeated until each data point is completely isolated or until the maximum (predefined) depth is reached. 4. The above steps are repeated to construct many iTrees by choosing different random subsets of features and sub-samples of data. 5. The anomaly score is then assigned to each of the data points based on the depth of the tree required to isolate that data point.

To elaborate further on the construction of an iTree in steps 2 and 3, the algorithm recursively divides the feature dataset by randomly selecting an attribute and a split value. This division continues until either the node contains only one instance or all data at the node share the same values. Once an iTree is fully grown, each point in the training dataset is isolated at one of the external nodes. Intuitively, data points that are more anomalous are typically easier to isolate in this process, resulting in smaller path lengths in the tree. The path length of each data point in the training set is defined as the number of edges that the point traverses from the root node to reach an external node (final node). Consequently, the anomaly score for each data point is computed based on the depth of the tree required to isolate that specific point.

In this paper, we employ scikit-learn's Python library to use the IF (Pedregosa et al. 2011). In this implementation, the anomaly score of an input sample is computed as the mean anomaly score of the trees in the forest. The lower anomaly score represents the more abnormal data point. The measure of normality of an observation given a tree is the depth of the leaf containing this observation, which is equivalent to the number of splittings required to isolate this point.

B.1 Isolation Forest Model Training

The IF algorithm is trained using a set of transaction features from the subset of misclassified transactions identified in the first layer. In this step, several binary decision trees are constructed during the training phase. This involves using a random sub-sample of the data and selecting a feature subset at random. After the training, each data point (i.e., transaction) is assigned an *anomaly score* based on the aggregation of the depth of the trees required to classify that transaction. For instance, if a smaller tree (i.e., a tree with fewer splits) is needed to segregate the transaction, then a transaction gets a higher anomaly score than those that require deeper trees (i.e., with more splits). This is illusrated in Figure 17 where the 🔴 transaction can be isolated from the other transactions (🔵) with fewer splits than the 🟢 transaction can.

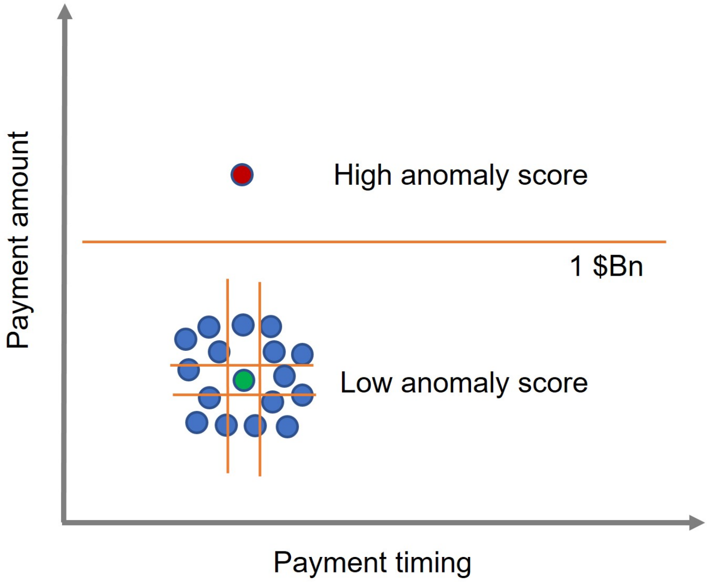


Figure 17: Stylized example of anomaly detection using the IF algorithm. Although both the 🔴 and 🟢 transactions are submitted at the same time, the 🔴 transaction is larger than 1 billion Can$, whereas the 🟢 and all 🔵 transactions are smaller than that. Consequently, the 🔴 data point can be isolated from the 🔵 points with fewer splits than the 🟢 data point can. The model will therefore assign a higher anomaly score to the 🔴 transaction than to the 🟢 transaction.

Parameter selection of unsupervised models is challenging due to the lack of target variables. Therefore, we use an ad-hoc approach for tuning the IF model. We fine-tune the model to maximize the average anomaly score on a predefined fraction of the dataset. Different sets of model parameters are selected, and for each set, we calculate the average anomaly score for the designated subset (consisting of transactions with high anomaly scores). Eventually, we adopt the model with the parameters that yield the highest average anomaly score within that subset. As illustrated in Figure 17, we would select the model attributing the highest anomaly score to the 🔴 data point.

B.2 Performance of the Standalone Isolation Forest Model

As an alternative to our two-layered approach, we directly trained the unsupervised learning-based IF model on the LVTS-T1 training data, eliminating the preceding payments classifier model. The results indicate that the average anomaly scores for artificially manipulated transactions are lower when employing the standalone IF model (i.e., when not integrated into the two-layer framework), compared to its integration within our framework. This suggests that the IF model within the framework may be more effective in isolating anomalies. This improved performance is likely attributed to the increased complexity resulting from the sheer volume of daily typical transactions within HVPSs. The effective screening of these typical transactions by the first layer streamlines subsequent anomaly detection in the second layer. This underscores the necessity of a two-layer approach and the significance of an efficient payments classifier in the initial layer.

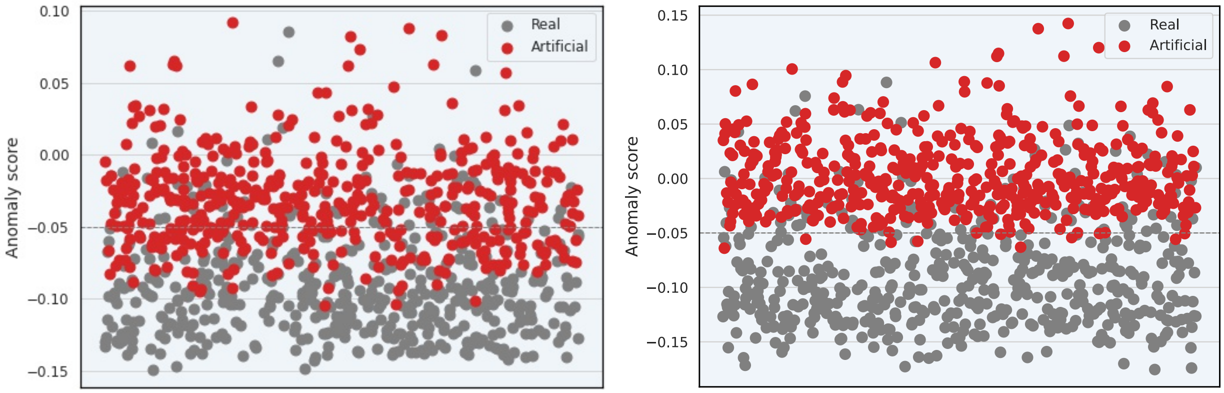


Figure 18: Performance comparison of the IF model (left) when trained independently on the entire dataset without the first layer and (right) as part of the framework in the second layer, using only misclassified samples from the first layer. The out-of-sample anomaly score on the set of transactions before (real) and after (artificial) manipulation. A transaction with a high (+ve) score is likely more anomalous than a transaction with a low (-ve) score.

C Artificial Anomalous Transactions Sample

The artificial anomalous transactions in our dataset are crafted by manipulating specific transaction attributes. For instance, we experiment with altering payment amounts or shifting payment submission times across different periods. Specifically, we employ the following steps to attain the sample of artificial transactions:

- We select 500 random payments from the out-of-sample subset.
- In this subset of payments, the amount featured is altered according to a uniform distribution of 5 to 10 billion Can$.
- Similarly, the period-time feature is altered according to a uniform distribution between 0 and 1000 seconds. This will generate more payments during the early parts of the periods.

Altering payment amounts to higher values and adjusting the period-time to lower values results in generating high-value payments sent earlier in the periods, which is rare in reality. The distribution of real and artificially manipulated transaction samples is depicted in Figure 19, which includes histograms comparing real and artificially generated subsets of transactions. The left side of the figure illustrates the impact of altering payment amounts, while the right side shows the effects of manipulating transaction submission times for LVTS-T1 payments. These artificial transactions are intentionally designed to exhibit substantial deviations from typical transaction behavior. We employ these extreme cases to assess whether our model has the capability to detect such outliers. Note that for the purpose of this test case, only two features are altered. However, it is possible to simultaneously modify more features.

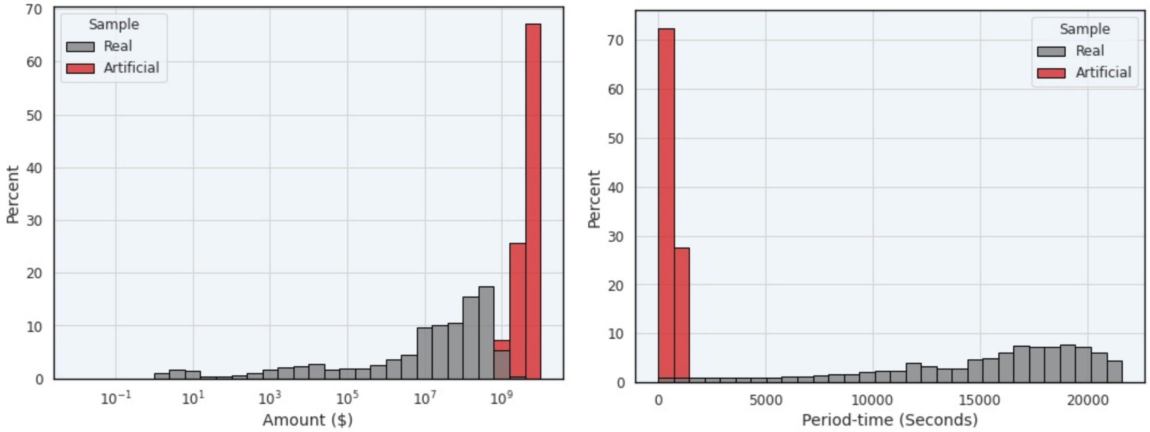


Figure 19: Histograms to compare artificial transaction samples (red) and their real counterparts (gray). These samples are generated by manipulating payment amounts (on the left) and submission periods (on the right) for LVTS-T1 payments.

D The Shapley Values and SHAP for Model Interpretation

The Shapley values is a method from coalitional game theory that provides a way to fairly distribute the *payout* among the *players* by computing the average marginal contribution of each player across all possible coalitions (Shapley 1953; Osborne and Rubinstein 1994).

For a coalitional game, (N,v), where N is a finite set of players indexed by i and v is the utility function or payoff function, the Shapley value can be obtained by this theorem, which satisfies the symmetry, dummy, and additivity axioms (Osborne and Rubinstein 1994):

$$\phi_i(N,v) = \underbrace{\frac{1}{N!}\sum_{S\subseteq N\setminus\{i\}}}_{\text{average over all } S} \underbrace{|S|!\,\Big(|N|-|S|-1\Big)!}_{\text{possible coalitions}} \underbrace{\Big[v(S\cup\{i\})-v(S)\Big]}_{\text{marginal value}}.$$

At a high level, the above equation can be split into three parts. The last part of the equation (the marginal value) gives the marginal contribution of an individual player i, when added to the coalition S that does not have i. The middle part shows how to compute different possible ways in which we could have formed the coalitions. Then, we take an average of possible ways that we could have done the marginal value calculation. The SHAP proposed by Lundberg et al. 2020 uses the Shapley values to explain the model predictions in terms of the marginal contribution of each predictor. The SHAP specifies the explanation of model $\mathscr{F}$ as a linear model of coalitions:

$$\mathscr{F}(S) = \phi_0 + \sum_{i=1}^{M} \phi_i S_i, \tag{1}$$

where $S \in \{0,1\}^M$ is the coalition vector with maximum M coalitions and ϕ_i the Shapley value for ith player.

In S the entry 1 means the corresponding player is present and 0 means the player is absent.

To illustrate, consider the binary classification problem as a *game*. Then the Shapley values can be used to fairly distribute the *payout* (= the prediction) among the *players* (= the predictors). Note: for the computation of the Shapley values in the SHAP, the zero means the corresponding predictor is absent. In that case, the absent predictors' value is replaced by a random value from its sample (Lundberg et al. 2020; Molnar 2020). The procedure is further illustrated as follows:

1. Consider a binary classification (class 0 or 1) problem with three predictors (Figure 20).
2. The average prediction (or base value) of the model is 0.5 (i.e., a 50-50 chance of being in either class). For the current transaction, our model predicts class 1.
3. By computing the Shapley values for all possible coalitions among three predictors, we can explain the difference between the actual prediction (1.0) and the base value (0.5) in terms of each predictor's marginal contribution.
4. In our example, predictor 1 increases the chances of class 1 by 20 percentage points, predictor 2 pushes it down by 10 points, and predictor 3 contributes +40 points. Thus, together they increase the prediction by 50 points from the average prediction, leading to the final prediction of class 1.

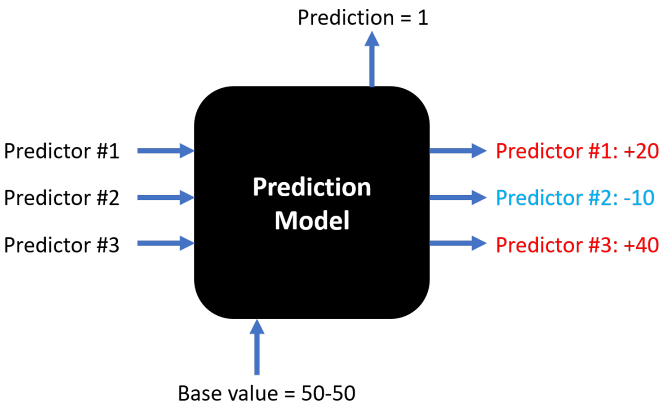


Figure 20: The SHAP explainer provides the marginal contribution of each predictor.

The SHAP values tell us which predictor contributes the most in the current instance of the prediction, that is, a local interpretation. Similarly, by using the Shapley values for each instance in the sample, we can get the average contribution of each predictor for that sample. That would give us a global interpretation of the model in terms of its feature importance. However, it is important to remember that these are only for the chosen model, and they do not explain the causality.

The SHAP package developed by Lundberg and Lee 2017 provides various tools to visualize the Shapley values computed for various ML models commonly used for predictions. For instance, the feature importance plots and summary plots (Figure 7 and Figure 11) are useful for global model interpretations. The force plots (Figure 9 and Figure 10) are useful for local interpretation, that is, at each instance of prediction. Also, the dependence plots (Figure 8) could be valuable for understanding the relationships between given predictors and the targets. The SHAP, although a powerful model-agnostic ad-hoc tool developed based on theoretical foundations for model interpretability, has some shortcomings, and it should be used with caution (Molnar 2020; Slack et al. 2020).