Targeted Detection of Anomalous Merchants on Integrated Payment Platforms via Multifaceted Transaction Representation Learning

Guanyu Lu
School of Data Science and Engineering
East China Normal University
Shanghai, China
gylu@stu.ecnu.edu.cn

Xiang Lin
School of Data Science and Engineering
East China Normal University
Shanghai, China
51205903046@stu.ecnu.edu.cn

Martin Pavlovski
Temple University
Philadelphia, PA, USA
martin.pavlovski@temple.edu

Xinyu Zhang
Shanghai Shouqianba Internet
Technology Co., Ltd.
Shanghai, China
yudi@shouqianba.com

Fang Zhou*
School of Data Science and Engineering
East China Normal University
Shanghai, China
fzhou@dase.ecnu.edu.cn

***Abstract*—Integrated payment platforms have significantly improved the convenience of daily life, yet they also present a fertile ground for fraudulent behavior. This paper focuses on the detection of anomalous merchants at the transaction level on such platforms, as locating specific anomalous patterns at such a granular level aids in taking corresponding security measures. However, in an integrated payment scenario, a limited number of imprecise labels are accessed at the merchant level rather than the transaction level, thus rendering transaction-level anomaly detection quite difficult. Meanwhile, the collected data comprises not only normal merchants and target anomalies (of interest) but also non-target anomalies (of lesser interest). To address these challenges, we adopt a two-step approach. First, we cluster merchants exhibiting similar behaviors and filter out potential non-target anomalies to better understand the transactional patterns among normal merchants. Then, we learn transaction representations encapsulated within hyperspheres, considering three key aspects: transaction context, historical information, and merchant information; and leverage such representations to determine anomaly scores for individual transactions. Real-world transactions from an integrated payment platform were used in the experiments. The results demonstrate that our model outperforms several state-of-the-art baselines, with an average AUPRC improvement of 10.5%-11.6%, 16.5%-16.7%, and 3.7%-5.4% in the three discovered merchant clusters.**

***Index Terms*—anomaly detection, integrated payment platform, transaction representation learning**

I. Introduction

Despite the development of mobile payment methods in financial technology, a single mobile payment application (e.g., WeChat or Alipay) cannot satisfy the payment preferences of different users. Integrated payment platforms integrate the payment services of more than one bank or a non-banking financial institution to meet the diverse needs of consumers and accommodate their payment patterns. A large number of merchants in such platforms generate tens of millions of transactions on a daily basis, nevertheless, with anomalous behaviors hidden in some of them. Thus, effective anomaly detection is crucial for creating a more secure and reliable payment environment on integrated payment platforms.

Some recent works, such as [1], [2], are focused on anomaly detection within an integrated payment setting. However, these studies cannot pinpoint which transactions within anomalous merchants are truly abnormal. To implement appropriate security measures effectively, it is crucial to precisely identify anomalous transactions. Taking a detected anomalous merchant as an example, it may generate hundreds or even thousands of transactions daily, but only a few transactions are anomalous. The exhaustive scrutiny of each transaction for such anomalous merchants demands a significant investment of human power and time. Therefore, detecting anomalous merchants at the transaction level becomes critical. Furthermore, in the integrated payment platform, there exist two types of anomalies: one comprises target anomalies (of interest), such as fraud and gambling recharge, which have the potential to inflict significant economic losses; the other type involves non-target anomalies (of lesser interest), such as cash out and click farming, posing minimal threats to the platform. The platform requires precise and targeted detection of these anomalous merchants that are of interest [1]. However, existing anomaly detection methods [3]–[7] are prone to interference from non-target anomalies, leading to many false positives.

Detecting anomalous merchants at the transaction level poses three primary challenges: (1) **Imprecise data labels.** Labels are defined on a merchant level (sourced from consumer feedback and judgments based on industry rules) instead of a transaction level. These merchant-level labels only indicate

*Corresponding author.

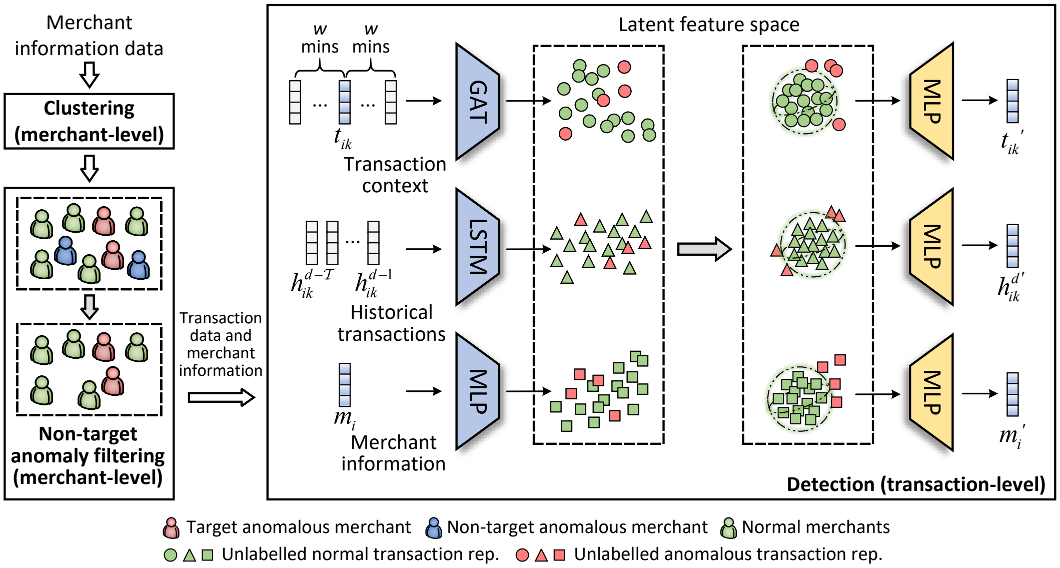

Merchant information data
Clustering (merchant-level)
Non-target anomaly filtering (merchant-level)
Transaction data and merchant information
w mins
w mins
t_{ik}
Transaction context
GAT
h_{ik}^{d-T}
h_{ik}^{d-1}
Historical transactions
LSTM
m_i
Merchant information
MLP
Latent feature space
MLP
MLP
MLP
t_{ik}'
$h_{ik}^{d'}$
m_i'
Detection (transaction-level)
Target anomalous merchant
Non-target anomalous merchant
Normal merchants
Unlabelled normal transaction rep.
Unlabelled anomalous transaction rep.

Fig. 1: The workflow of TransAD.

whether a merchant has engaged in illegal activities, such as fraud or gambling recharge, but it is unclear which transaction involved anomalous behavior. Therefore, supervised anomaly detection methods [6]–[8] are inapplicable to unlabeled transaction data. (2) **Lack of prior information on non-target anomalous merchants.** Due to the wide variety of non-target anomaly categories covered and the lesser interest in these anomalies, acquiring all types of labeled non-target anomalies proves quite difficult [1]. (3) **Diverse business activities of merchants.** For example, breakfast-serving merchants have fewer active hours of successful transactions and higher transaction density in unit time than merchants offering services throughout the day.

In this work, we introduce TransAD (Transaction-level Anomaly Detection), a model for transaction-level detection of target anomalous merchants (illustrated in Fig. 1), whose detection component is designed based on the concept of one-class classification. To address the challenges posed by diverse merchant activities and non-target anomalies in unlabeled datasets, we apply clustering and non-target anomaly filtering to obtain homogeneous data to learn better representations of normal transactions. The former groups together merchants with similar transaction patterns, while the latter reduces the interference of non-target anomalies in target anomaly detection by leveraging the transaction distribution differences among merchants. The detection component learns transaction representations encapsulated within hyperspheres and considers three aspects derived from our data analysis (Section III-B): (1) *Transaction context.* For normal merchants, a given transaction and its time-adjacent transactions are typically a result of similar payment behavior. If a transaction differs significantly from its transaction context in terms of features (e.g., amount or payment type), it is more likely to be flagged as anomalous. (2) *Historical transactions.* Normal merchants' transactions within a specific historical period typically exhibit a pattern of regularity, whereas anomalous transactions deviate from these patterns (e.g., by occurring during periods with no prior transactions in the merchant's history). (3) *Merchant information.* Merchant information can enhance anomaly detection when transaction information alone is insufficient. For instance, a transaction's amount significantly differing from a merchant's daily average can be indicative of an anomalous transaction. These aspects allow for leveraging richer information to assist in learning transaction representations.

To verify the effectiveness of TransAD, we conducted experiments on a real-world dataset collected from an integrated payment platform. The results indicate that, compared with several state-of-the-art baselines, TransAD exhibits superior performance in terms of AUPRC, with an improvement of 10.5%-11.6%, 16.5%-16.7%, and 3.7%-5.4% on average for the three transaction clusters defined in Section III-B, respectively. In addition, TransAD can also identify more target anomalies among the top-99 results compared to its alternatives. This will reduce the burden of manual verification for the business personnel.

In brief, the main contributions of this paper are summarized as follows:

- To the best of our knowledge, this work is the first to achieve target anomalous merchant detection at a transaction level (involving large-scale data processing) in an integrated payment platform.
- We propose a novel anomaly detection framework, TransAD, to alleviate the deterioration of detection performance due to various challenges (e.g., imprecisely assigned labels, absent prior knowledge of non-target anomalies, and diverse merchant activities) with data from integrated payment platforms.
- Experimental results obtained on a real-world dataset collected from an integrated payment platform demonstrate that TransAD has a higher practical value compared to state-of-the-art baselines.

II. Related Work

Collecting large-scale labeled anomaly data in the real world is difficult and costly, rendering fully supervised anomaly detection impractical. Hence, over the past decades, research has pivoted to unsupervised methods, typically based on isolation concept [4], [9], density estimation [10], and probability distribution [11]. Yet these methods often fail to detect anomalies resembling normal patterns, leading to the application of deep learning. Autoencoder-based methods [12]–[14] learn latent representations and detect anomalies based on reconstruction errors, while generative adversarial learning [15], [16] uses generators and discriminators to separate outliers through reconstruction errors. However, the above schemes tend to neglect between-samples relationship, which introduces certain limitations. Recently, diffusion models, such as LMD [17] and DiffAD [18], have made progress in anomaly detection tasks. However, diffusion model-based anomaly detection algorithms are mainly applied to image data and not applicable to tabular data.

Although collecting large-scale labeled data in anomaly detection is challenging, obtaining a small amount of labeled data in some scenarios is possible. Hence, semi-supervised methods can capture additional information from such readily accessible labeled data to further improve anomaly detection performance. *(1) On the one hand*, some scholars regard normal data as the target class and generate a description for that class, thereby distinguishing the target class from other classes [19]. DeepSVDD [20] constructs a hypersphere of normal data and determines anomalies based on samples' distances to the hypersphere's center. OCAN [21] trains a discriminator on generated malicious samples at sparse boundaries of benign samples. Nonetheless, these one-class methods require highly pure training data, making them unsuitable for the diverse merchant activities in integrated payment scenarios. *(2) On the other hand*, some methods focus primarily on anomalous data, such as PU learning-based methods [22], [23], but these methods struggle with minor differences between normal and anomalous patterns in large datasets. Some state-of-the-art approaches [2], [6], [7], [24] use a small amount of labeled anomalies to guide model training, but are only capable of detecting anomalous merchants without identifying specific anomalous transactions. Models such as TitAnt [25] aim at detecting online real-time anomalous transactions by utilizing labeled anomalous transaction records; however, transaction-level labels are unavailable in the integrated payment scenario studied in this work.

III. Methodology

A. Problem Definition

Let $M = \{m_1, m_2, \ldots, m_l\}$ be a set containing l merchants, where $M^A \subset M$ is a set of labeled target anomalous merchants, and $M^N \subset M$ is a set of unlabeled merchants, such that $\left|M^A\right| \ll \left|M^N\right|$. The corresponding transaction set of the merchants is $T = \{T_1, T_2, \ldots, T_l\}$, where $T_i = \{t_{i1}, t_{i2}, \ldots, t_{ir}\}$ indicates that a merchant m_i on a given day conducts r transactions. Note that we do not know which transactions in the dataset are anomalous.

Let C_{ik} be a set of transactions adjacent in time to t_{ik} within a context window (w minutes) and $\left\{h_{ik}^{d-\mathcal{T}}, \ldots, h_{ik}^{d-1}\right\}$ are the statistics of transactions that occurred around the same period as t_{ik} in the past $\mathcal{T}$ days (e.g., if t_{ik} occurred at 8:03 AM, then $\left\{h_{ik}^{d-\mathcal{T}}, \ldots, h_{ik}^{d-1}\right\}$ contains the statistics of transactions of merchant m_i that occurred between 8:00 AM and 9:00 AM in the past $\mathcal{T}$ days).

Given a transaction t_{ik} of a merchant m_i, its adjacent transaction set C_{ik} and historical information $\left\{h_{ik}^{d-\mathcal{T}}, \ldots, h_{ik}^{d-1}\right\}$, the goal is to predict its anomaly score $S(t_{ik})$. If $S(t_{ik}) > \tau_S$, then t_{ik} is considered a target anomalous transaction, where τ_S is an anomaly score threshold.

For a merchant m_i, if there exists a target anomalous transaction in T_i, m_i is considered a target anomalous merchant and $Y\left(m_i\right) = 1$; otherwise $Y\left(m_i\right) = 0$.

B. Data Analysis

The objects of our data analysis are the catering merchants in China using a certain integrated payment platform. Fig. 2 shows the density distribution of the data features. We found that anomalous merchants have the following characteristics.

(1) Dense transactions (Fig. 2(a)): The number of transactions of anomalous merchants within ten minutes is higher than that of normal merchants. The time interval between transactions of normal merchants is typically several minutes or more as customers need to select the product they wish to purchase and then open a QR code or a mini program (that is, a mobile app that can be utilized without the need for downloading and installing an entire shopping app) to complete the payment. In contrast, anomalous merchants may generate multiple transactions in an instant (usually less than one second) due to simultaneous payments by numerous people.

(2) High number of non-local transactions (Fig. 2(b)): The number of non-local transactions of anomalous merchants within ten minutes is higher than that of normal merchants. Most transactions of normal merchants are offline, but anomalous merchants tend to conduct online transactions, resulting in more non-local transactions.

(3) Irregular transaction time (Fig. 2(c)): The transaction frequency peaks of normal catering merchants in China occur regularly from 6:00 to 8:00, 11:00 to 13:00, and 17:00 to 19:00. We found that anomalous merchants also had a large number of transactions in other periods. That being said, the transaction peaks of anomalous merchants occur irregularly, e.g., they may occur after lunch or dinner time.

(4) High transaction amount (Fig. 2(d)): We use min-max normalization to normalize transaction amounts in the interval [0,1]. The peak of the transaction amount in normal merchants is distributed between 0.4 and 0.5. We discovered that the transaction amount peak is distributed around 0.7 for anomalous merchants; thus, the amount of some anomalous transactions is higher than that of the merchant's historical transactions.

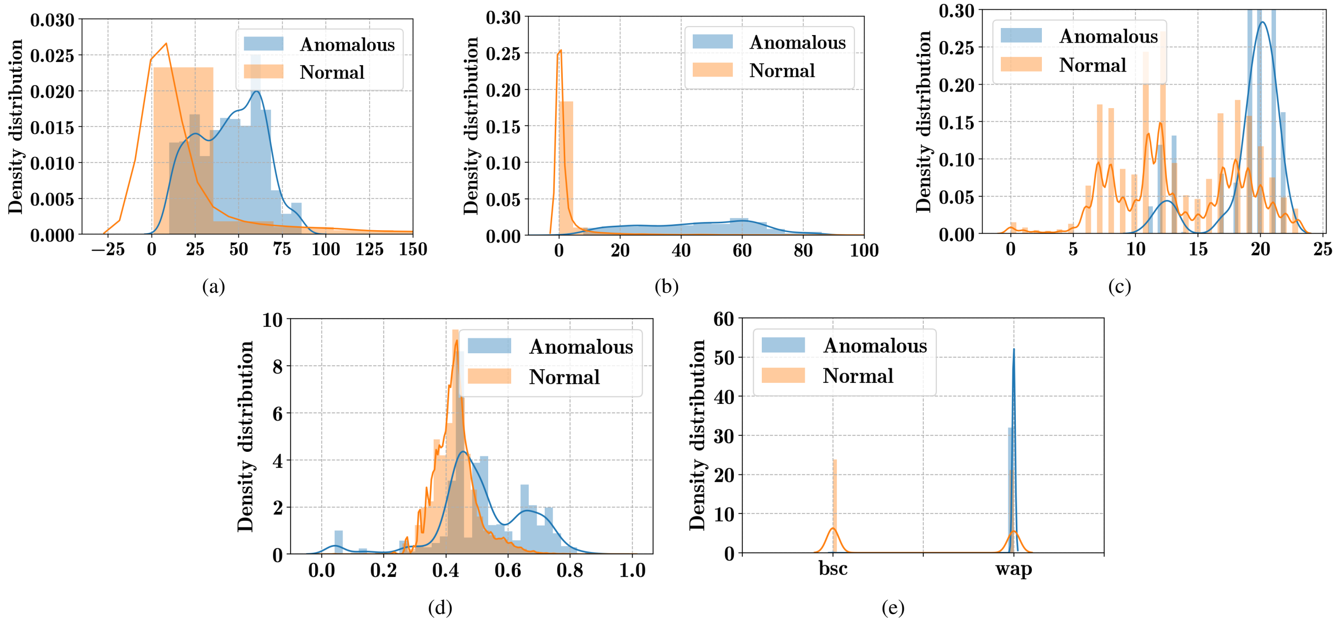


Fig. 2: Density distribution of data features with respect to (a) the number of total transactions within ten minutes, (b) the number of non-local transactions within ten minutes, (c) transaction time, (d) transaction amount and (e) payment type.

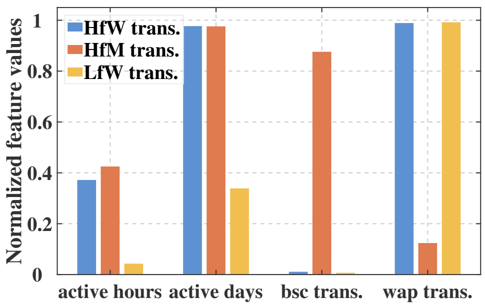


Fig. 3: The values of the most discriminative features of the three cluster centroids.

(5) Tendency of using a mini program for payment (Fig. 2(e)): A *bsc* (business-scan-consumer) transaction is paid by a barcode or QR code, while a *wap* (wireless application protocol) transaction is paid by a mini program. Normal merchants involve *bsc* and *wap* transactions, but the payment type for anomalous transactions is more inclined to mini-program payment.

Besides, due to the different types of catering (such as desserts, bars, and hot pots) operated by merchants on the integrated payment platform, multiple business activity patterns (Fig. 3) exist among those merchants. We clustered merchants using K-means to explore different merchant activity patterns and found that there are three groups hidden in the data. Fig. 3 presents the most discriminative features of the three cluster centroids, i.e., the features with significant differences between their values. According to the characteristics shown by the features, we define these three clusters as high-frequency *wap* (HfW) transactions, low-frequency *wap* (LfW) transactions, and high-frequency mixed (HfM) transactions (including both *wap* and *bsc* transactions). Merchants in the HfW transaction cluster and HfM transaction cluster are more active than those in the LfW transaction cluster. The fraction of *bsc* transactions in the HfM transaction cluster is relatively high, while most of the transactions in the HfW transaction cluster and the LfW transaction cluster are *wap* transactions.

C. Proposed Model

In this section, we present the proposed TransAD model, which includes three components: clustering, non-target anomaly filtering, and detection. Since merchants on the platform conduct various business activities, we first apply K-means to group merchants with similar transaction patterns into the same cluster. Each cluster may include normal merchants, a portion of non-target anomalies, and a small number of target anomalies. Second, we filter out non-target anomalies as much as possible through the filtering component. Our model is designed to focus on identifying target anomalies that present a critical threat to maintaining a secure and reliable payment environment. Third, the model detects anomalies from three aspects: transaction context, historical transactions, and merchant information. Below we describe the non-target anomaly filtering and detection procedures in detail.

1) Non-target Anomaly Filtering: After the clustering process, the unlabeled data within each group comprises a substantial number of normal merchants, a fraction of non-target anomalies, and possibly several target anomalies. Considering that non-target anomalies can affect the learning of normal patterns, thereby complicating the identification of target anomalies, we apply filtering components to effectively emilinate non-target anomalies.

Inspired by iForest [9], we first introduce the isolation score $IS(m_i)$ to separate target and non-target anomalies from normal merchants. $IS(m_i)$ is formulated as

$$IS\left(m_i\right) = 2^{-\frac{E\left(h\left(m_i\right)\right)}{T(n)}}, \tag{1}$$

where $T(n) = 2H(n-1) - \frac{2(n-1)}{n}$. Note that n represents the sample size, $h(m_i)$ is the number of edges m_i passes through from the root to a leaf node in a totally random tree, $T(n)$ is the average path length for which a totally random tree search fails, and $H(n-1)$ can be approximated by $ln(n-1) + 0.577$ (Euler's constant).

Next, we consider the similarity score $SS\left(m_i\right)$, which measures the degree of similarity between merchants to separate target anomalies from the rest of the merchants.$SS\left(m_i\right)$ is calculated as the minimum Euclidean distance between the selected and anomalous merchants, i.e.,

$$SS\left(m_i\right) = e^{-d_i}, \tag{2}$$

such that $d_i = \min_{k \in \{1,\ldots,A\}} \left(m_i - \mu_k\right)^2$, where A is the number of anomalous merchants and μ_k is the vector representation of the k^{th} anomalous merchant.

The $IS(m_i)$ scores of target and non-target anomalies are higher than those of normal merchants, while $SS(m_i)$ scores of non-target anomalies are lower than those of target anomalies. For each cluster k, We complete the filtering of non-target anomalies according to: $IS(m_i) > \tau_{is}$ and $SS(m_i) < \tau_{ss}$, where the isolation threshold τ_{is} is the average value of the $IS(m_i)$ of anomalous merchants, while the similarity threshold τ_{ss} is a fixed quantile value of $SS\left(m_i\right)$.

2) Detection: According to the descriptive analysis in Section III-B, we design a detection mechanism through learning transaction representations encapsulated within hyperspheres from three aspects: transaction context, historical transactions, and merchant information.

Transaction Context. This module focuses on learning transaction representations by taking into account the context of a transaction, which encompasses temporally adjacent transactions. From Fig. 2, it can be inferred that the transaction context usually contains rich information that can assist in learning the representation of the transaction to which they are adjacent. A similar concept is leveraged in Graph Attention Networks (GAT) [26], which apply attention mechanisms to graph neural networks to assign different weights to neighboring nodes and thus better capture contextual information. Therefore, we consider GAT as the encoder of the transaction context module. Each transaction can be represented as a node and linked with other transactions that occurred at adjacent timesteps. The decoder is a feedforward neural network that reconstructs the latent representation of a transaction through two fully connected layers. The encoder and decoder of the transaction context module are formulated as

$$Z_{t_{ik}} = \mathrm{GAT}\left(t_{ik}, C_{ik}\right), \quad t'_{ik} = g\left(Z_{t_{ik}}\right), \tag{3}$$

where t_{ik} is the original input of the k^{th} transaction of the merchant m_i to the graph attention layer, C_{ik} is a set of neighbors describing the context of t_{ik}, $Z_{t_{ik}}$ is the latent feature representation generated by the graph attention network, and t'_{ik} is the reconstructed representation of t_{ik}.

Historical Transaction. Although the transaction context focuses on information about transactions at adjacent timesteps, more details about a given transaction can be gleaned from historical transactions, as the difference between a transaction and its preceding ones can reflect the possibility of it being anomalous. Considering that Long-Short Term Memory (LSTM) [27] has been shown to achieve excellent performance in learning long-term dependencies in various application domains, we leverage LSTM as the encoder in the historical transaction module. Statistics of transactions that occurred around the same time as a given transaction in the past $\mathcal{T}$ days are input into the LSTM encoder to learn the representation of that transaction. Likewise, a fully connected network acts as a decoder for reconstructing historical transaction inputs. The encoder and decoder of the historical transaction module are formulated as

$$Z_{h_{ik}^{d-j}} = \mathrm{LSTM}\left(Z_{h_{ik}^{d-1-j}}, h_{ik}^{d-j}\right) (j \in \{\mathcal{T}-1, \mathcal{T}-2, \ldots, 1\}),$$
$$h_{ik}^{d\,\prime} = g\left(Z_{h_{ik}^{d-1}}\right), \tag{4}$$

where $Z_{h_{ik}^{d-j}}$ is the hidden layer output of the LSTM unit, $Z_{h_{ik}^{d-1}}$ is the latent feature representation generated by the final LSTM unit, h_{ik}^{d-j} is the historical transaction representation on the j^{th} day before t_{ik}, and $h_{ik}^{d\,\prime}$ is the reconstructed representation of the historical transactions.

Merchant Information. The above two modules consider the context and historical transactions over a certain time period from the perspective of a given transaction. However, considering only transaction information is sometimes not enough to identify anomalies, and specific merchant information needs to be considered. Therefore, we utilize merchant information to assist the learning of transaction representations and apply fully connected networks as the encoder and decoder, respectively. The encoder and decoder of the merchant information module are formulated as

$$Z_{m_i} = f\left(m_i\right), \quad {m_i}' = g\left(Z_{m_i}\right), \tag{5}$$

where m_i is the merchant information corresponding to the given transaction, Z_{m_i} is the latent representation of the merchant corresponding to the given transaction, and ${m_i}'$ is the reconstructed vector representation of m_i.

3) Training and Testing: We describe the training and testing procedures of the proposed model TransAD below. After preprocessing, the training data is input to the clustering component and grouped into multiple clusters (three in our case, as determined by the elbow method) by merchant activities. Then, the data in the three clusters are separately entered into the filter to remove non-target anomalies. The data that excludes non-target anomalies is finally passed to the detection component, which learns representations of normal

TABLE I: Descriptive statistics of the collected dataset.

Data	Size		
	Training set	Validation set	Testing set
Unlabeled transaction data	15,077,652	15,503,142	14,851,966
Unlabeled merchants	171,366	173,504	173,491
Anomalous merchants	0	43	88

transactions encapsulated within hyperspheres to obtain sphere center representations in the three respective latent feature spaces. The initial model parameters are continuously updated to minimize the reconstruction loss,

$$\mathcal{L}_p = \frac{1}{n}\sum_{n}\left(\left\|{t_{ik}}' - t_{ik}\right\|^2 + \beta\left\|{h_{ik}^{d}}' - h_{ik}^{d}\right\|^2 + \gamma\left\|{m_i}' - m_i\right\|^2\right), \tag{6}$$

where n is the sample size, β and γ are trade-off parameters. After updating the parameters, the sphere centers of the three aspects in the latent feature space are calculated as $c_t = \frac{1}{n}\sum_n Z_{t_{ik}}$, $c_h = \frac{1}{n}\sum_n Z_{h_{ik}^{d-1}}$, and $c_m = \frac{1}{n}\sum_n Z_{m_i}$.

Thereafter, the detection component continues by adjusting the parameters to propel the latent feature representations of normal transactions closer to the sphere centers. Its loss function is formulated as

$$\mathcal{L}_f = \frac{1}{n}\sum_{n}\left(\left\|Z_{t_{ik}} - c_t\right\|^2 + \beta\left\|Z_{h_{ik}^{d-1}} - c_h\right\|^2 + \gamma\left\|Z_{m_i} - c_m\right\|^2\right). \tag{7}$$

After the model's training is completed, the preprocessed testing data is first divided into three merchant clusters according to the distance between the merchant activity features and the centroids. Then, non-target anomalies are filtered out from the clustered testing data. Finally, for a given transaction from the testing data, its anomaly score is obtained based on the distances between the latent feature representations of that transaction, its transaction history and its corresponding merchant, and their respective center points, i.e.:

$$S(t_{ik}) = \left\|Z_{t_{ik}} - c_t\right\|^2 + \beta\left\|Z_{h_{ik}^{d-1}} - c_h\right\|^2 + \gamma\left\|Z_{m_i} - c_m\right\|^2. \tag{8}$$

We determine a threshold based on a separate validation set, and transactions with scores exceeding the threshold are considered anomalies. Ultimately, merchants involved in abnormal transactions were identified as targeted anomalous merchants.

IV. Experiments

A. Setup

1) Dataset: To evaluate the performance of TransAD, we used a real-world dataset from an integrated payment platform provided by a company. Table I summarizes more specific statistics about the dataset. The training, validation, and testing datasets include the platform's unlabeled data from September 2, 2021, September 3, 2021, and September 4, 2021, along with their preceding seven-day periods, respectively. The business personnel verified 131 anomalous merchants from July to September 2021. The validation and testing sets contain 43 and 88 labeled anomalies, respectively.

We extracted the following three groups of features.

(1) **Transaction features** describe the relevant information about transactions conducted at a given merchant. We extracted fifteen features that reflect three aspects: **a.** Transaction state features including transaction status (success, refund, or failure), payment type (barcode, QR code, or mini program), and transaction channel (such as credit card, Alipay, WeChat Pay, etc.); **b.** Transaction spatiotemporal information indicative of whether the transaction is non-local, whether the transaction occurs at night, and whether the transaction occurs during meal times; **c.** Transactional amount features encompassing the transaction amount itself, along with a series of associated features indicating whether the transaction amount is a multiple of ten, whether the transaction amount is a multiple of one hundred, whether the transaction amount contains decimals, whether the integer part of the transaction amount is a multiple of ten, whether the transaction amount is within five CNY of a multiple of hundred, whether the transaction amount is less than one CNY, whether the transaction amount is one cent CNY, and whether the transaction amount is in a special form (e.g., numbers such as 666, 888, or 998, which are considered lucky numbers in Chinese culture).

(2) **Historical transaction features** are aggregates of the transaction features described in (1), represented by transaction statistics (such as average or maximum) calculated on a weekly basis.

(3) **Merchant information features** reflect the merchants' scale, business rules, and management level. We extracted twelve features, categorized into four groups: **a.** Merchant scale features that include the number of merchant stores and the number of cities where a merchant's stores are located. **b.** Merchant transaction frequency which refers to the fraction, the numbers of hours, and the numbers of days of a merchant's successful transactions in the past seven days. **c.** The features related to merchants' transaction spatiotemporal information including the concentration of payer IP addresses in the past seven days per merchant, the number of cities where merchants conducted stores' code transactions in the past seven days, the fraction of nighttime transactions in the past seven days for merchants, and the fraction of mealtime transactions in the past seven days per merchant. **d.** Features associated with payment type proportions for merchants, which encompass the fraction of payments through barcode/QR code (*bsc* transactions) in the past seven days, the fraction of payments through a mini-program (*wap* transactions) in the past seven days, and the fraction of Alipay and WeChat payments in the past seven days.

2) Data Preprocessing: We present the preprocessing of the raw data below. (1) We normalize the transaction amount using the min-max value of successful transactions in the merchant's monthly history. (2) Since exceeding the credit card limit will result in a failed transaction, we delete the failed record if a successful transaction exists within 5 minutes before or after a failed transaction occurred. (3) A *bsc* transaction should be a local transaction; thus, we correct the consumer's IP address if it is inconsistent with the merchant store's IP address. (4)

TABLE II: F1-Score and AUPRC performance of TransAD, the baselines and ablated models.

Clusters		HfW transaction			HfM transaction			LfW transaction		
Metrics		F1-Score	AUPRC	AUPRC-lift	F1-Score	AUPRC	AUPRC-lift	F1-Score	AUPRC	AUPRC-lift
Models	iForest	0.006	0.002	0.115	0.002	0.001	0.167	0.014	0.004	0.051
	OCAN	0.005	0.002	0.115	0.007	0.002	0.166	0.021	0.011	0.044
	OCSVM	0.030	0.012	0.105	**0.040**	0.003	0.165	0.087	0.018	0.037
	DeepSVDD	0.007	0.003	0.114	0.003	0.001	0.167	0.034	0.017	0.038
	FROCC	0.001	0.001	0.116	0.001	0.001	0.167	0.004	0.001	0.054
	TransAD$_{-\mathrm{h}}$	0.126	0.098	0.019	0.003	0.060	0.108	0.063	0.033	0.022
	TransAD$_{-\mathrm{m}}$	0.016	0.004	0.113	0.006	0.002	0.166	0.010	0.027	0.028
	TransAD$_{-\mathrm{h},-\mathrm{m}}$	0.010	0.002	0.115	0.018	0.016	0.152	0.068	0.022	0.033
	TransAD	**0.157**	**0.117**		0.004	**0.168**		**0.107**	**0.055**	

We use one-hot encoding for the categorical features.

3) Baselines: We compared the performance of our model and state-of-the-art anomaly detection methods.

- **iForest** [4] detects anomalies based on how many steps are needed to isolate instances using isolation trees.
- **OCSVM** [28] finds an optimal hyperplane in the latent feature space to achieve the maximum separation of the target data and the coordinate origin.
- **DeepSVDD** [20] builds a hypersphere to contain as much normal data as possible.
- **OCAN** [21] generates anomalous data where the density of normal data is sparse.
- **FROCC** [3] randomly projects normal data onto a set of unit vectors and uses the boundary of the projected region to identify anomalous samples.

4) Metrics and Parameter Settings: To compare the effectiveness of the proposed model and that of the baselines, we analyze the metrics derived from the confusion matrices calculated for each model, involving F1-Score and Area Under the Precision-Recall Curve (AUPRC). TransAD was trained with batches of 512 samples (determined based on validation set performance), optimizing it loss functions (see *Training and Testing* in Section III-C) over 10 and 20 iterations respectively. For OCSVM, we employed a radial basis function kernel with a coefficient of 0.0001 and capped the iterations at 20, while other hyperparameters were set using their default values in scikit-learn. As for the remaining baselines, we used their open-source implementations with default hyperparameters, with DeepSVDD and OCAN having the same batch size and iterations as our model's. The experiments were carried out on an Alibaba Cloud DSW server featuring an Intel Xeon Platinum 8269CY CPU, running Ubuntu 18.04, with 60 GB of memory.

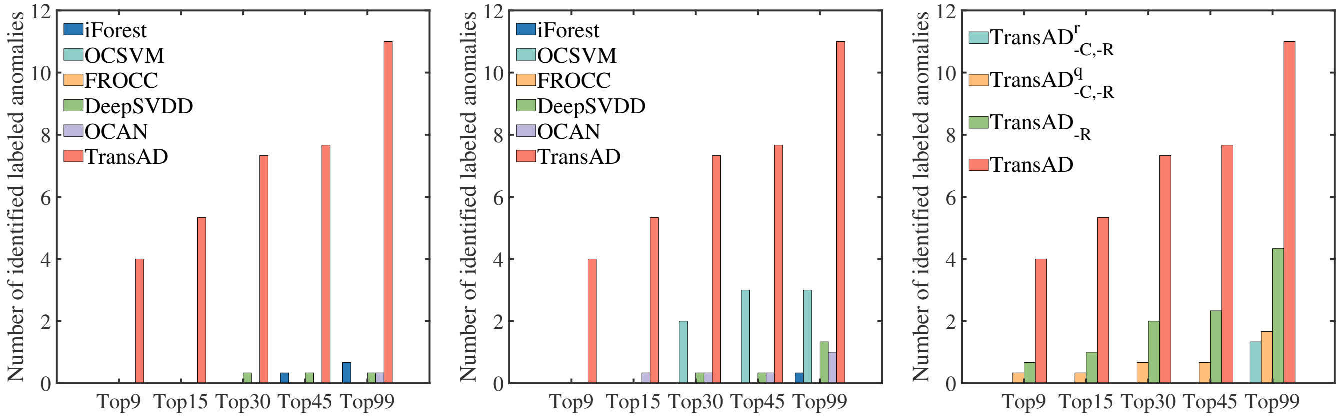
B. Results and Discussion

In the experiments, we focus on the following **research questions (RQs)**. **RQ1:** What is the overall performance of TransAD compared to the baselines? **RQ2:** How effective are the clustering and filtering components of the model? **RQ3:** What are the contributions of different aspects in the detection component? **RQ4:** How sensitive is TransAD to the trade-off parameters β, γ, and the selected context window in the detection component?

1) Overall performance (RQ1): Table II includes the results of our model and the baselines obtained for the three clusters with respect to F1-Score and AUPRC. The F1-Score and AUPRC results for both the baseline models and TransAD are relatively low due to the highly imbalanced characteristic of the real-world dataset we utilized, which contains only 0.05% anomalies in the testing data. For instance, AUPRC can reflect the identification of anomalous merchants when the data is unbalanced. TransAD exhibits satisfactory overall performance and absolute superiority in terms of AUPRC, with an improvement of 10.5%-11.6%, 16.5%-16.7%, and 3.7%-5.4% on average for the three clusters, respectively. The baseline methods can only achieve better results with respect to F1-Score for a single cluster, which has certain limitations.

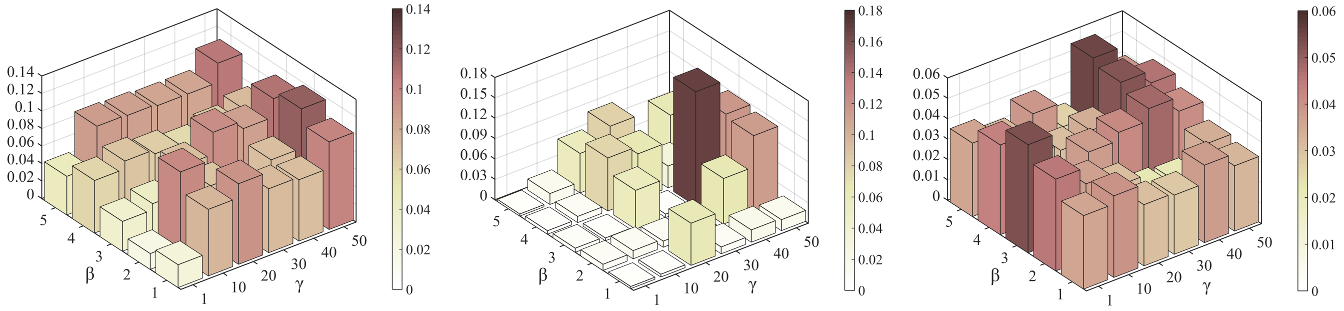
Apart from the results of F1-Score and AUPRC, we also illustrate the advantage of our model from the perspective of business personnel verification. Since the verification process is manual and time-consuming, the number of predicted anomalies that can be verified is very limited. We focus on inspecting the top N ($N \in \{9, 15, 30, 45, 99\}$) predicted anomalous merchants, sorted by their anomaly scores. Fig. 4(a) compares the number of labeled target anomalies identified by TransAD and the baselines without clustering. At $N = 9$, our model can already identify 4 labeled anomalies. As N increases to 99, iForest, DeepSVDD, and OCAN identified 0.667, 0.333, and 0.333 labeled anomalies on average, while OCSVM and FROCC were unable to identify any labeled anomalies. Nevertheless, our model can identify 11 labeled anomalies.

Next, we applied the clustering mechanism described in Section III-C to each baseline. Fig. 4(b) shows a comparison of TransAD and the baselines with clustering in identifying labeled target anomalies, and our model still outperforms the state-of-the-art. Combining the observations from Fig. 4(a) and Fig. 4(b), the baselines whose performance is affected after incorporating the clustering mechanism are iForest, OCSVM, OCAN, and DeepSVDD. iForest's ability to identify anomalies deteriorated. Thus, it is more appropriate for iForest to use the entire dataset for training. The results of OCAN, OCSVM, and DeepSVDD improved because they are based on the one-class classification method, which is aided by the clustering mechanism as it groups merchants with similar transaction patterns together.



(a) (b) (c)

Fig. 4: The number of labeled anomalies in top N predicted anomalous merchants identified by TransAD and (a) the baselines without clustering, (b) the baselines with clustering and (c) its three variants.



(a) (b) (c)

Fig. 5: AUPRC results obtained with different trade-off parameters in the (a) HfW, (b) HfM and (c) LfW transaction cluster.

2) Effect of clustering and filtering components (RQ2): We experimented with three variants, $\mathrm{TransAD}^{\mathrm{r}}_{-\mathrm{C},-\mathrm{R}}$, $\mathrm{TransAD}^{\mathrm{q}}_{-\mathrm{C},-\mathrm{R}}$, and $\mathrm{TransAD}_{-\mathrm{R}}$, to assess the importance of the clustering and filtering components. $\mathrm{TransAD}_{-\mathrm{C},-\mathrm{R}}$ includes only the detection component. Constrained by computational resources and considering the complexity of the model, we selected 20,000 random (for $\mathrm{TransAD}^{\mathrm{r}}_{-\mathrm{C},-\mathrm{R}}$) or high-quality (for $\mathrm{TransAD}^{\mathrm{q}}_{-\mathrm{C},-\mathrm{R}}$) merchants for training, respectively. High-quality merchants are considered those that were online for more than four months and for which the number of successful transaction days in the past four months was greater than or equal to 100 days. $\mathrm{TransAD}_{-\mathrm{R}}$ excludes the filtering component. Fig. 4(c) presents the number of labeled anomalies identified by TransAD and its three variants. The comparison of the results of $\mathrm{TransAD}^{\mathrm{r}}_{-\mathrm{C},-\mathrm{R}}$ and $\mathrm{TransAD}^{\mathrm{q}}_{-\mathrm{C},-\mathrm{R}}$ demonstrates that selecting high-quality merchants helps the model to identify anomalies better. Moreover, after incorporating the clustering mechanism, the model is able to scale up to larger amounts of data as each cluster can be trained in parallel. Compared to $\mathrm{TransAD}_{-\mathrm{C},-\mathrm{R}}$, $\mathrm{TransAD}_{-\mathrm{R}}$ identifies more labeled anomalies. Lastly, after filtering, non-target anomalies are removed, and thus the detection rate has further improved.

3) Ablation Study (RQ3): To investigate the impact of the different aspects (used in the detection component) on the model's anomaly detection ability, we tested three ablation models: $\mathrm{TransAD}_{-\mathrm{h}}$ that excludes the historical transaction aspect, $\mathrm{TransAD}_{-\mathrm{m}}$ that excludes the merchant information aspect, and $\mathrm{TransAD}_{-\mathrm{h},-\mathrm{m}}$ that retains only the transaction context aspect. Refer to Table II for the detailed experimental results. TransAD exhibits the best AUPRC performance and introduces improvements of 1.9%-11.5%, 10.8%-16.6%, and 2.2%-3.3% in the HfW transaction, HfM transaction, and LfW transaction clusters, respectively. Therefore, learning historical transactions and merchant information has the least impact on the LfW transaction cluster. TransAD also attains excellent results in terms of F1-Score, particularly on the HfW transaction cluster. In summary, this ablation study confirms the importance of the historical transaction and merchant information modules in the detection component.

4) Parameter sensitivity (RQ4): We studied the effect of the trade-off parameters β ($\beta \in \{1, 2, 3, 4, 5\}$) and γ ($\gamma \in \{1, 10, 20, 30, 40, 50\}$) on the detection performance, and Fig. 5 show the AUPRC values obtained on the three clusters under different choices of the trade-off parameters. We discovered that larger values of γ tend to achieve optimal

TABLE III: AUPRC performance and the average degree in the constructed graph using different context windows.

Clusters	AUPRC (sparsity)						
	1 min	2 mins	3 mins	4 mins	5 mins	10 mins	15 mins
HfW trans.	0.109 (0.449)	0.089 (0.830)	0.060 (1.190)	0.110 (1.540)	0.078 (1.886)	**0.117 (3.573)**	0.105 (5.208)
HfM trans.	0.083 (0.553)	0.070 (1.077)	0.002 (1.569)	0.058 (2.042)	0.089 (2.506)	**0.168 (4.727)**	0.060 (6.851)
LfW trans.	0.035 (0.209)	0.035 (0.330)	0.023 (0.419)	0.037 (0.497)	**0.055 (0.569)**	0.047 (0.894)	0.032 (1.193)

detection results. But in Fig. 5(a), we have chosen a smaller β value; that is, a lower weight value of the historical transaction module achieved better results. Thus, historical transaction information plays a less critical role in HfW transactions.

We then evaluated the effect of the temporal context window, a key parameter in the transaction context module, and set its width to $\{1, 2, 3, 4, 5, 10, 15\}$. A wider context window promotes more neighboring transactions. Table III lists the AUPRC values obtained in the three clusters using different context windows. Next to each AUPRC value, we provide the average degree in the graph constructed by the transaction context module. Optimal AUPRC is achieved in the HfW and HfM transaction clusters in case a context window of 10 minutes is used. Due to the high frequency of transactions in these two clusters, a wider context window would provide more contextual information relevant to learning a better representation of a given transaction. However, the graph constructed based on a 15-minute context window is already denser, which increases the complexity of the model. On the other hand, the window size required to reach an optimal AUPRC for the LfW transaction cluster is narrower. The reason is that merchants in this cluster are inactive, and transactions occur less frequently; thus, the constructed transaction graph becomes more sparse.

V. Conclusion

We introduced TransAD, a novel transaction-level anomaly detection model that encompasses clustering, non-target anomaly filtering, and multi-aspect detection based on transaction representation learning, aimed to address various challenges posed by massive integrated payment data, and locates specific anomalous transactions to facilitate taking corresponding security measures. TransAD effectively alleviates the shortcomings of state-of-the-art baselines on a real-world integrated payment platform data, while identifying considerably more labeled anomalies, which economizes unnecessary labor resources for manual verification.

Acknowledgment

This work was supported by Shanghai “Science and Technology Innovation Action Plan” Project (No.23511100700).

References

[1] G. Lu, F. Zhou, M. Pavlovski, and et al., “A robust prioritized anomaly detection when not all anomalies are of primary interest,” in *ICDE*, 2024, pp. 775–788.

[2] W. Zong, F. Zhou, M. Pavlovski, and et al., “Peripheral instance augmentation for end-to-end anomaly detection using weighted adversarial learning,” in *DASFAA*, 2022, pp. 506–522.

[3] A. Bhattacharya, S. Varambally, A. Bagchi, and et al., “Fast one-class classification using class boundary-preserving random projections,” in *SIGKDD*, 2021, pp. 66–74.

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *TKDD*, vol. 6, no. 1, pp. 1–39, 2012.

[5] F. V. Massoli, F. Falchi, and et al., “Mocca: Multilayer one-class classification for anomaly detection,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, pp. 2313–2323, 2022.

[6] G. Pang, A. van den Hengel, C. Shen, and et al., “Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data,” in *SIGKDD*, 2021, pp. 1298–1308.

[7] G. Pang, C. Shen, and A. van den Hengel, “Deep anomaly detection with deviation networks,” in *SIGKDD*, 2019, pp. 353–362.

[8] C. Liu, Q. Zhong, X. Ao, and et al., “Fraud transactions detection via behavior tree with local intention calibration,” in *SIGKDD*, 2020, pp. 3035–3043.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *ICDM*, 2008, pp. 413–422.

[10] B. Liu, P.-N. Tan, and J. Zhou, “Unsupervised anomaly detection by robust density estimation,” in *AAAI*, vol. 36, 2022, pp. 4101–4108.

[11] Z. Li, Y. Zhao, X. Hu, and et al., “Ecod: Unsupervised outlier detection using empirical cumulative distribution functions,” *TKDE*, vol. 35, no. 12, pp. 12 181–12 193, 2022.

[12] J. Audibert, P. Michiardi, F. Guyard, and et al., “Usad: Unsupervised anomaly detection on multivariate time series,” in *SIGKDD*, 2020, pp. 3395–3404.

[13] D. L. Aguilar, M. A. Medina-Pérez, O. Loyola-Gonzalez, and et al., “Towards an interpretable autoencoder: A decision-tree-based autoencoder and its application in anomaly detection,” *TDSC*, vol. 20, no. 2, pp. 1048–1059, 2022.

[14] T. Kieu, B. Yang, C. Guo, and et al., “Robust and explainable autoencoders for unsupervised time series outlier detection,” in *ICDE*, 2022, pp. 3038–3050.

[15] X. Chen, L. Deng, F. Huang, and et al., “Daemon: Unsupervised anomaly detection and interpretation for multivariate time series,” in *ICDE*, 2021, pp. 2225–2230.

[16] Z. Zhang, W. Li, W. Ding, and et al., “Stad-gan: unsupervised anomaly detection on multivariate time series with self-training generative adversarial networks,” *TKDD*, vol. 17, no. 5, pp. 1–18, 2023.

[17] Z. Liu, J. P. Zhou, Y. Wang, and et al., “Unsupervised out-of-distribution detection with diffusion inpainting,” in *ICML*, 2023, pp. 22 528–22 538.

[18] X. Zhang, N. Li, J. Li, and et al., “Unsupervised surface anomaly detection with diffusion probabilistic model,” in *CVPR*, 2023, pp. 6782–6791.

[19] H. Deng and X. Li, “Anomaly detection via reverse distillation from one-class embedding,” in *CVPR*, 2022, pp. 9737–9746.

[20] L. Ruff, R. Vandermeulen, N. Goernitz, and et al., “Deep one-class classification,” in *ICML*, 2018, pp. 4393–4402.

[21] P. Zheng, S. Yuan, X. Wu, and et al., “One-class adversarial nets for fraud detection,” in *AAAI*, vol. 33, 2019, pp. 1286–1293.

[22] H. Ju, D. Lee, J. Hwang, and et al., “Pumad: Pu metric learning for anomaly detection,” *Information Sciences*, vol. 523, pp. 167–183, 2020.

[23] L. Perini, V. Vercruyssen, and J. Davis, “Learning from positive and unlabeled multi-instance bags in anomaly detection,” in *SIGKDD*, 2023, pp. 1897–1906.

[24] G. Pang, C. Shen, H. Jin, and et al., “Deep weakly-supervised anomaly detection,” in *SIGKDD*, 2023, pp. 1795–1807.

[25] S. Cao, X. Yang, C. Chen, J. Zhou, X. Li, and Y. Qi, “Titant: Online real-time transaction fraud detection in ant financial,” *PVLDB*, vol. 12, no. 12, pp. 2082 – 2093, 2019.

[26] P. Veličković, G. Cucurull, A. Casanova, and et al., “Graph attention networks,” in *ICLR*, 2018, p. 1–12.

[27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] B. Schölkopf, R. C. Williamson, A. Smola, and et al., “Support vector method for novelty detection,” *NeurIPS*, vol. 12, p. 582–588, 1999.