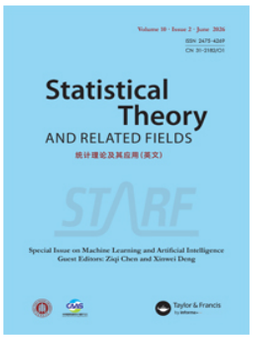

Statistical Theory and Related Fields

ISSN: 2475-4269 (Print) 2475-4277 (Online) Journal homepage: www.tandfonline.com/journals/tstf20

A new anomaly detection via multiple instance learning for sequence data with application to credit card delinquency risk control

Zhenguo Gao, Yihao Bu , Xiaoxun Li & Xiaoning Kang

To cite this article: Zhenguo Gao, Yihao Bu , Xiaoxun Li & Xiaoning Kang (2026) A new anomaly detection via multiple instance learning for sequence data with application to credit card delinquency risk control, Statistical Theory and Related Fields, 10:2, 268-284, DOI: 10.1080/24754269.2026.2652585

To link to this article: https://doi.org/10.1080/24754269.2026.2652585

Published online: 11 May 2026.

STATISTICAL THEORY AND RELATED FIELDS
2026, VOL. 10, NO. 2, 268–284
https://doi.org/10.1080/24754269.2026.2652585

A new anomaly detection via multiple instance learning for sequence data with application to credit card delinquency risk control

Zhenguo Gao [a], Yihao Bu[a], Xiaoxun Li[a] and Xiaoning Kang [b]

[a]School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, People's Republic of China; [b]Institute of Supply Chain Analytics, Dongbei University of Finance and Economics, Dalian, People's Republic of China

ABSTRACT

Anomaly detection in sequence data is widely applicable across various domains and has significant commercial value to the financial industry. This paper studies its utility as a means of controlling credit card delinquency risk. Transactions that deviate from the regular data sequence are a common precursor of payment difficulty. Current detection methods, however, do not effectively identify abnormal transactions from such data, making it difficult to control the overdue payment risk. Therefore, in this paper, we propose a Multiple Instance Learning-based Anomaly Detection (MILAD) method with well designed learning networks to address this problem. Comparing the performance of the MILAD and Deep Autoencoding Gaussian Mixture Model (DAGMM) method, which is currently the most commonly used unsupervised deep learning algorithm for credit card risk control, we observe that the proposed MILAD is able to effectively control the overdue risk by leveraging both transaction and payment information.

ARTICLE HISTORY
Received 26 May 2025
Revised 17 March 2026
Accepted 25 March 2026

KEYWORDS
Abnormal transactions; attention weights; complex sequential data analysis; overdue risk

1. Introduction

In recent years, research on anomaly detection of sequence data has gradually become a hot topic. It has a very wide range of applications in many scientific fields. Especially in the financial area, anomaly detection has great commercial value. Traditional anomaly detection methods for sequence data search for changes in several parameters of temporal data sequences, such as time series data (Chen & Gupta, 1997; Gao et al., 2020, 2019). Unlike traditional ones, anomaly detection in the financial area focuses on analysing the anomaly status of a multivariate time series data by studying the influence of anomaly samples on the abnormal state of the whole data sequence in a high-dimensional space. The motivation of this paper stems from the common credit default problem in the financial field, where the control of the overdue risk of credit cards plays a key role. However, there has not been any effective algorithm in the literature that can accurately analyse overdue risk utilizing sequence samples from credit card transactions.

CONTACT Xiaoning Kang xiaoningmike@126.com Institute of Supply Chain Analytics, Dongbei University of Finance and Economics, Dalian 116025, People's Republic of China

Most transactions from the overdue credit card users are normal with only a few abnormal ones, such as impulse purchase, fraudulent purchase. These abnormal transactions are the main causes of the overdue problem. However, most of the existing approaches for the overdue risk control of the credit cards (Bolton & Hand, 2001; Chen & Guestrin, 2016; Liu et al., 2019; Lucas & Jurgovsky, 2020) do not make full use of the transaction information, but rely too much on the business experience when conducting risk control. A great challenge of utilizing these abnormal transactions is that there are no obvious post event features for assigning the abnormality labels to the anomaly transactions. The only label information that we could use is the users' monthly overdue information. Zong et al. (2018) proposed the DAGMM algorithm which combines traditional unsupervised methods and deep autoencoders to achieve some good results. However, in practice, sample features are constructed artificially, which means that the samples may not be representative and comprehensive enough. Therefore, the difference between abnormal and normal samples is limited, such that the models can not distinguish them very well. Furthermore, this unsupervised algorithm cannot use the overdue information effectively. At present, this method only has few applications in cold-start businesses due to the absence of abnormal labels.

The characteristic of the credit card bill overdue risk detection is that the monthly bill has a label, but transactions on the bill do not have labels, which also happens in other application scenarios. The Multiple Instance Learning is a good solution to solve this kind of problem. Numerous studies have been conducted in this field, such as Carbonneau et al. (2018). The traditional Multiple Instance Learning has three steps. First, all original samples are preprocessed to obtain the feature vectors of these samples. Then the feature vectors of all samples in the bag are aggregated together into a bag vector. Finally, all aggregated bag vectors are classified by classification models. This method was first applied to test drug activity and then widely used in many common fields, including target detection, text classification, and speech classification.

Multiple Instance Learning has also been successfully applied to video anomaly detection (Sultani et al., 2018; Tian et al., 2021). While the MIL framework is shared across domains, our credit card delinquency problem presents distinct characteristics in data modality (tabular features vs. visual frames), temporal structure (discrete transactions vs. continuous frames), and output requirements (explicit instance-level pseudo-labels for identifying specific risky transactions). These differences motivate our specific design choices in MILAD for financial anomaly detection in tabular sequential data.

Under the Multiple Instance Learning framework, samples are grouped into sets, which are defined as Bags. An abnormal status label is assigned to the entire bag. However, no label is assigned to the samples in the bag. Then the relationship between the bag label and sample labels is determined on the basis of the assumption of Multiple Instance Learning. Ilse et al. (2018) proposed an Attention-based Multiple Instance Learning algorithm (ABMIL). ABMIL uses the Attention Neural Network to learn the attention weights of samples in a bag. Then the attention weights are used to aggregate samples in the bag, followed by the subsequent classification analysis. This aggregation method can assign weights to the samples in a bag, and then detect important samples based on sample weights. Inspired by their method, we utilize both individual transaction information and the overall bill overdue information simultaneously to improve the existing methods.

In this paper, we propose an anomaly detection algorithm based on the Multiple Instance Learning technique, named MILAD (Multiple Instance Learning for Anomaly Detection). MILAD is based on the information from a sequence of samples, which can make full use of

both the sample and sequence information. MILAD is designed for solving anomaly detection problems in a wide range of applications, not limited to the risk control of credit cards. In the experiments of this paper, MILAD is able to control the overdue risk from the transaction and produces more accurate and effective results. MILAD outperforms the most commonly used algorithm, DAGMM, in terms of several major evaluation criteria and provides a better performance in model interpretation.

The rest of the paper is organized as follows. The proposed MILAD model and its algorithm are introduced in Section 2, as well as the computational details of each module and their parameter optimization techniques. Section 3 conducts several experiments on the application dataset and compares the performance of the proposed MILAD with that of the DAGMM algorithm. Section 4 summarizes the paper.

2. Methodology

2.1. Model and notation

Suppose $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_J\}$ is a time-dependent multivariate sequence, where $\mathbf{x}_j \in \mathbb{R}^d$ is a sample of the sequence at time $j = 1, \ldots, J$. $y_j \in \{0, 1\}$ is the hidden label indicating the status of the sample $\mathbf{x}_j$, and 1 means abnormal. y_j is the hidden state of the sample that is predicted from the following model,

$$y_j = \begin{cases} 1, & \text{if } f(x_j) \geq \delta, \\ 0, & \text{else}, \end{cases} \qquad j = 1, \ldots, J, \tag{1}$$

where $f : \mathbb{R}^d \to \mathbb{N}$ is a classifier based on feature mapping. We need to estimate the hidden state y_j of each sample. δ is the threshold parameter discriminating the abnormal status of samples. An appropriate δ should be chosen according to the practical situation. Then the abnormal state label Y of the sequence X is modelled as

$$Y = \begin{cases} 1, & \text{if } \mathcal{F}(y_1, \ldots, y_j, \ldots, y_J) \geq \Delta, \\ 0, & \text{else}, \end{cases} \tag{2}$$

where $\mathcal{F} : \mathbb{R}^J \to \mathbb{R}$ is a function used to estimate the overall anomaly state of a data sequence. Δ is the threshold to discriminate the overall anomaly state of the data sequence.

Figure 1 is the flowchart of our entire modelling framework. The framework is composed of three steps. The first step is the Multiple Instance Learning based on the sample information and the sequence information using the Attention mechanism. The second step is the anomaly label estimation of all samples in the data sequence according to the result from the previous Multiple Instance Learning procedure. The third step is the sequence anomaly detection procedure based on the estimated abnormal labels of samples using binary supervised learning method. Models are trained by the common optimization algorithm Adam (Kingma & Ba, 2014).

Algorithm 1 is the computational flow of our proposed method MILAD. It is a Multiple Instance Learning-based method, which can effectively associate the unknown sample label y_j with the known sequence label Y through the Attention network mechanism and achieve efficient modelling processes eventually. The MILAD algorithm constructs a risk analysis model $\mathcal{F}$ based on sample anomaly detection in a data sequence. In practice, taking the credit card overdue risk prediction businesses as an example, we can use the model $\mathcal{F}$ to evaluate

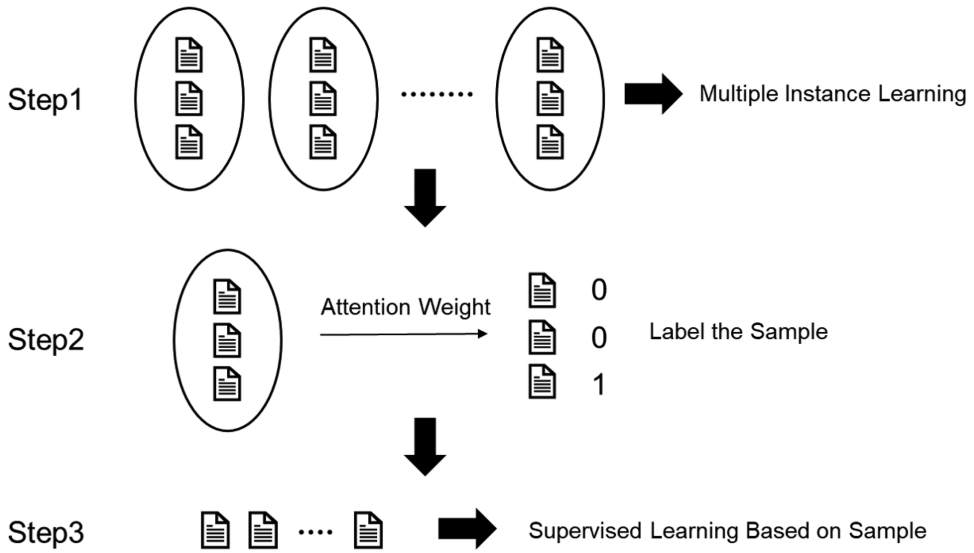

Figure 1. The framework of the MILAD algorithm.

card holders' overdue risk based on their transaction vector $\mathbf{x}' \in \mathbb{R}^d$. We can predict the overdue risk probability p' through the model $\mathcal{F}$, and finally determine whether to decline the transaction $\mathbf{x}'$ based on the actual needs of the business. In this way we can directly control the overdue risk from the dimension of transactions. Comparing with traditional approach, controlling overdue risk based on the MILAD algorithm is much more convenient in practice.

Algorithm 1 MILAD

INPUT: The multi time series sample bag $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_J\}$ which is a collection of time series of length J, $\mathbf{x}_j \in \mathbb{R}^d$.

Step 1: Multiple Instance Learning

Use Algorithm 2 to estimate the classification probability P of the bag, the attention weight w_j^* of samples in the bag, and the abnormal probability p_j of samples in the bag.

Step 2: Sample Anomaly Detection

Based on the Multiple Instance Learning results from Step 1, detect the abnormal state of each sample $\mathbf{x}_j$ in the bag, and get the sample anomaly state set $S = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_J\}$ using Algorithm 4.

Step 3: Sequence Anomaly Detection

Based on the abnormal sample detection results from Step 2, use the classification method (e.g., Xgboost) to estimate the abnormal state Y of the sample bag X.

OUTPUT: Y and S

It is important to note that the current MILAD framework focuses on single-month risk prediction, where each bag (monthly billing cycle) is treated independently. The model predicts whether a user will be delinquent in month t based on their transactions in month t, without explicitly modelling temporal dependencies across multiple months. This design choice is motivated by the practical need for banks to assess current-month risk based on recent transaction patterns. While credit card delinquency can be influenced by cumulative effects across months, the single-month prediction task remains a common and valuable problem in credit risk management. Future extensions could incorporate multi-month temporal modelling to capture longer-term behavioural patterns.

2.2. Multiple instance learning

In Model (1), the classification model f is built upon the feature information of sequence samples in the bag. However, the anomaly state label y_j of the sequence samples is generally unknown. Therefore, we cannot perform any supervised learning directly. To effectively solve this problem, we use the Multiple Instance Learning approach. Then the relationship between the bag label and sample labels is determined by the assumption of the Multiple Instance Learning.

As mentioned in Foulds and Frank (2010), there are two different assumptions: the Standard Assumption and the Collective Assumption. The Standard Assumption is that each sample in the bag has its own label, the label of the bag is negative if all samples in the bag are negative, and the label of the bag is positive if there is at least one positive sample in the bag. The Collective Assumption states that the label of a bag cannot be determined by any single sample, but by the interactions between samples and the cumulative effect of some samples in the bag. Therefore, we propose two types of designs for the Transformer Network T: the **Basic method** and the **Self-Attention based method**. The Basic method is adaptive to the standard assumption, while the Self-Attention based method is designed for the collective assumption, which has more practical usages. Figure 2 is the network structures of the Multiple Instance Learning model with both the Basic method and the Self-Attention method provided in the Transformer Network module.

The Multiple Instance Learning model is composed of the following four parts: feature transformation network T, attention network W, aggregation network A, and classification network C. Algorithm 2 is the proposed Multiple Instance Learning Algorithm. We use the Attention mechanism to adaptively aggregate the samples in the bag Y_i. Since the feature

Algorithm 2 The Multiple Instance Learning

INPUT: The multi time series sample bag $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J\}$.

Step 1: Randomly initialize the weights of the parameters in the T, W, A, C network;

Step 2: Transform the original sample through the network T and obtain the transformed vector $\mathbf{h}_j = T(\mathbf{x}_j)$ through the Basic method or the Self-Attention method (Algorithm 3).

Step 3: Calculate the attention weight w_j^* by (6) for all samples in the bag through the network W.

Step 4: Obtain the sample aggregation Z through the network A by (7).

Step 5: Obtain the abnormal probability P of the sample bag X by (8) through the network C.

OUTPUT: P

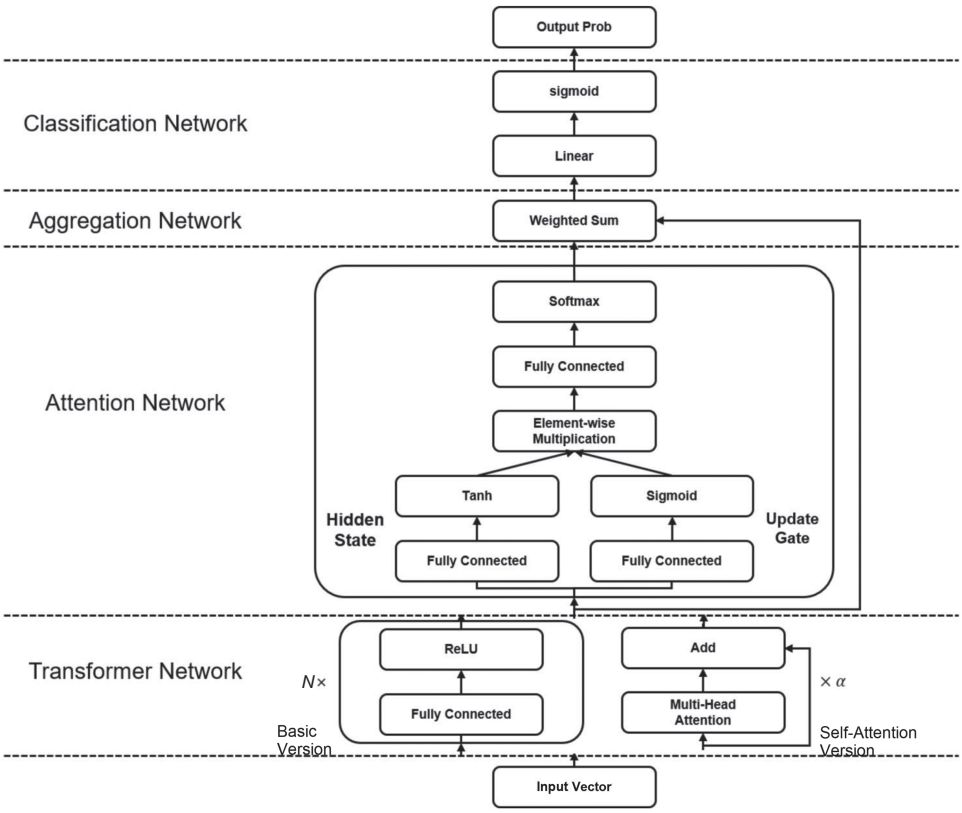

Figure 2. Multiple Instance Learning Network Structure. The symbols (T), (W), (A), and (C) represent the Transformer Network, Attention Network, Aggregation Network, and Classification Network respectively.

extractor and classifier are conducted with neural networks, it allows to establish an end-to-end model to make the whole model to be more auto adaptive. Meanwhile, each step of the model is built upon neural networks, which makes the back propagation algorithm available for parameters optimization. All parameters are optimized by minimizing the Logarithmic loss

$$L(\Theta) = \frac{1}{N}\sum_{i=1}^{N}(Y_i \ln P_i + (1 - Y_i)\ln(1 - P_i)),$$

where N is the sample size of the training data, and P_i is the anomaly probability of the i-th bag.

Feature Transformation Network T The function of the feature transformation network T is designed to conduct a feature extraction and transformation on the original features. There are two approaches based on different assumptions: the basic method and the Self-Attention method.

The basic approach assumes that there is no interaction effect and no structural information between samples in the bag. Therefore, sample $\mathbf{x}_j$ can be transformed into a feature vector $\mathbf{h}_j$ directly using a two-layer fully connected network,

$$\mathbf{h}_j = W_2^\top \sigma(W_1^\top \mathbf{x}_j + \mathbf{b}_1) + \mathbf{b}_2,$$

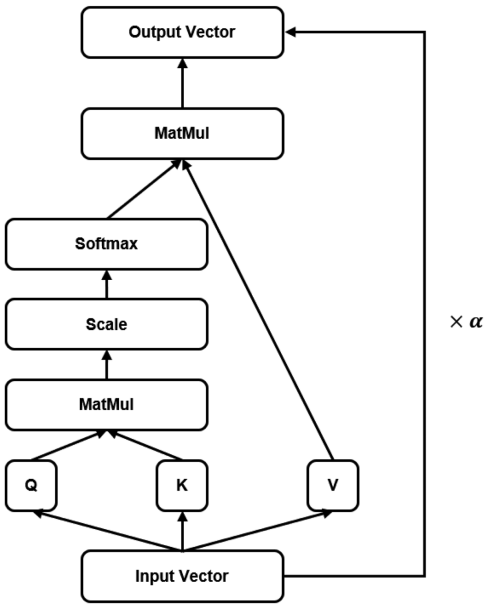

Figure 3. The structure of the Soft-Transformer.

where $\sigma(\cdot)$ is the activation function.

However, in practice, there exist various interaction effects between samples. In order to learn the interaction effectively, Vaswani et al. (2017) introduced a Transformer framework based on the Attention mechanism to obtain the interaction information between sequences composed of words. Based on this method, Rymarczyk et al. (2021) proposed a Soft-Transformer framework. Figure 3 is the schematic diagram of the Soft-Transformer. The Soft-Transformer transforms samples into feature vectors before the Attention-based Multiple Instance Learning, which can explore the interactive information between samples more effectively. Let $\mathbf{x}_j$ be the sample vector, and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_J]^\top$ be the sample matrix composed of vectors. Firstly, we use weight matrices $W_Q^{d\times d_1}$, $W_K^{d\times d_1}$, $W_V^{d\times d_2}$ to calculate the corresponding matrices $Q^{J\times d_1}$ (Query), $K^{J\times d_1}$ (Key), $V^{J\times d_2}$ (Value), where $Q = XW_Q$, $K = XW_K$, $V = XW_V$. We usually make $d_1 = d_2$. Then we have

$$W_A = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_1}}\right). \tag{3}$$

Finally we get the transformed sample matrix:

$$T = [\mathbf{t}_1, \ldots, \mathbf{t}_J]^\top = W_A V, \tag{4}$$

where $\mathbf{t}_j$ is the transformed sample vector. Then we can perform the Multiple Instance Learning in the same way. Using this method we can get the transformed sample vector and the interaction information between samples. However, after transformation, the actual meaning of the vector is different from those for the original samples. The subsequent sample weights do not represent the importance of the samples anymore, and cannot be used to discriminate critical samples. Therefore, we use the Soft-Transformer to transform the output vector $\mathbf{t}_j$

into $\mathbf{h}_j$.

$$\mathbf{h}_j = \mathbf{x}_j + \alpha \mathbf{t}_j, \tag{5}$$

where α is a learnable parameter. The Soft-Transformer ensures that the weight of the transformed vector retains its ability to reflect the importance of the samples after considering the interaction information in the analysis. In the subsequent analysis, we perform a critical sample discrimination based on attention weights. Algorithm 3 is the proposed transformation algorithm based on the Self-Attention method.

Algorithm 3 The Feature Transformation Algorithm Based on the Self-Attention Method

INPUT: The multi time series sample bag $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_J\}$.

Step 1: Rewrite the input sample vectors in matrix form:
$X = [\mathbf{x}_1, \ldots, \mathbf{x}_J]^\top$.

Step 2: Obtain $Q = XW_Q, K = XW_K, V = XW_V$ matrices.

Step 3: Calculate attention weight W_A by (3).

Step 4: Calculate transformed matrix T by (4).

Step 5: Calculate the transformed feature vector $\mathbf{h}_j$ by (5) through the Soft-Transformer.

OUTPUT: $\mathbf{h}_j$.

Attention Network W The attention network W is constructed to learn the attention weights of samples in the bag. The attention weights are estimated through a two-layer gated neural network module with

$$\begin{aligned} \mathbf{v}_j &= \tanh(V^\top \mathbf{h}_j), \\ \mathbf{u}_j &= \text{Sigmoid}(U^\top \mathbf{h}_j), \\ w_j &= W_a^\top (\mathbf{v}_j \odot \mathbf{u}_j), \\ w_j^* &= \frac{\exp\{w_j\}}{\sum_{i=1}^{J} \exp\{w_i\}}, \end{aligned} \tag{6}$$

where $\mathbf{v}_j$ is the hidden state, $\mathbf{u}_j$ is the updated gate state, $\odot$ represents the element-wise multiplication of the vector, w_j is the attention weight, and w_j^* is the normalized attention weight by Softmax.

Aggregation Network A The aggregation network aggregates all samples in the bag. After calculating the attention weight of each sample through the Attention Network W, we can estimate the feature vector $Z_i \in \mathbb{R}^d$ of the bag by calculating the weighted sum of the sample vectors in the bag,

$$Z_i = \sum_{j=1}^{J} w_j^* \mathbf{h}_j. \tag{7}$$

where w_j^* is the attention weight of each sample in the bag, and $\mathbf{h}_j$ is the transformed feature vector obtained from the Transformer Network T. This aggregation strategy is known as **attention-based weighted pooling**, where the bag representation is computed as a weighted sum of instance features with learned attention weights. This approach is more suitable for credit risk assessment than alternative pooling strategies such as max pooling (which assumes

only the single most anomalous instance determines the bag label) or mean pooling (which treats all instances equally and fails to identify critical transactions). Attention-based pooling allows the model to learn which transactions are most indicative of risk while still considering the overall transaction pattern, which is essential to handle highly imbalanced data where the anomaly rate is only 1.70%.

Classification Network C The classification network classifies the abnormal status of the bag. After the aggregation of samples in the bag, the classification problem is turned into a traditional binary supervised learning problem. To deal with features extracted from the Neural Network, a fully connected (FC) layer network together with the Sigmoid activation function is used to calculate the anomaly classification probability P_i of the i-th bag, where

$$P_i = \text{Sigmoid}(W_C^\top Z_i + b). \tag{8}$$

2.3. Sample anomaly detection

After the Attention-based Multiple Instance Learning, we obtain the probability P_i of the label of the bag to be 1, and the attention weight w_{ij}^* of each sample in the bag. Unlike the traditional Multiple Instance Learning, the estimated attention weights of the samples are more important here, which can be used to detect the key samples in the bag. That is, which sample in the bag has a significant impact on the abnormal status of the bag. The samples with larger attention weights have a greater impact on bags, and these samples are likely to be the key samples which lead to the bag abnormality. Therefore, we can combine the prediction results of the bag and the estimated attention weights together to predict the anomaly status of each samples in the bag i. Let $p_{ij} = P_i w_{ij}^*$ be the probability of sample $\mathbf{x}_j$ in the i-th bag being abnormal. By choosing an appropriate threshold δ, the abnormality status of the j-th sample $\hat{y}_{ij} = 1$ if $p_{ij} \geq \delta$.

Algorithm 4 Sample Anomaly Detection Algorithm

INPUT: The multi time series sample bag $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_J\}$, the bag prediction probability P and the attention weights w_j^*, the empirical threshold δ.

Step 1: Calculate the abnormal probability of the sample $p_j = P w_j^*$.

Step 2: Discriminate the abnormal samples:

$$\hat{y}_j = 1 \quad \text{if} \quad p_j \geq \delta.$$

OUTPUT: The sample anomaly state set $S = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_J\}$.

2.4. Sequence anomaly detection

After the sample anomaly detection procedure, we get the anomaly set $S = \{\hat{y}_1, \ldots, \hat{y}_J\}$, which contains the pseudo labels of all samples in the bag. Then we can do the binary supervised learning to estimate the abnormal state Y of the sample bag in Model (2) using the common classification approaches. In this paper, we adopt the Xgboost algorithm.

3. Experiments

3.1. Data preprocessing

Since payment data often contains sensitive private information about individuals or institutions, and only banks and other related institutions have access to it. Therefore the acquisition of such public dataset is quite limited. The lack of available effective public datasets is also a challenge for researches in this field. In this work, we evaluate the performance of the proposed MILAD algorithm on a commonly used real dataset, which is the Credit Card Fraud Detection (CCFD) data (Dal Pozzolo et al., 2015). The CCFD dataset is composed of transactions of credit card users in Europe in September 2013. This dataset includes 284807 transactions, where 492 are abnormal transactions. It is a highly imbalanced dataset, which only has 0.17% of abnormal transactions. To deal with this highly imbalance problem of the data, we use the common undersampling method to sample 10% of normal transactions. Then the abnormal rate increases to 1.70%. Due to the privacy issues in this field, this dataset cannot provide the original transaction features and the user information. The data contains 28 principal component features, $\{V_1, \ldots, V_{28}\}$, which are transformed from the original features, the transaction amount, and the anomaly label of each transaction.

In order to make the dataset suitable for solving our problem, we have to generate a new dataset through the following data generating mechanism based on the original CCFD data. We randomly select a certain number of transactions from the original dataset to form a sample bag, then take each sample bag as the user's transaction set X_i, and then label the bag according to the sample label in the bag. The labelling process of the sample bag is based on these two assumptions of the Multiple Instance Learning, which are the standard assumption and collective assumption. In the subsequent data analysis we assume that there are no available labels for the samples in the bag. Generating the dataset in this way can effectively mimic our desired scenario in which we have the label for user's transaction set, but lack the labels for each transaction in the bag. The labelling rules for the sample bags are as follows. Under the Standard Assumption, as long as there is a sample $\mathbf{x}_{ij} \in \mathbb{R}^d$ whose label y_{ij} is abnormal in the sequence set $X_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{ij}, \ldots, \mathbf{x}_{iJ}\}$, the label of the overall sequence $Y_i = \min\{1, \sum_{j=1}^{J} y_{ij}\}$. Under the Collective Assumption, only when the sum of the amount of abnormal samples reaches a certain threshold, the label of the overall sequence $Y_i = 1$, if $\frac{\sum_{j=1}^{J} y_{ij} v_{ij}}{\sum_{j=1}^{J} v_{ij}} \geq \Delta$, where v_{ij} is the transaction amount of the sample $\mathbf{x}_{ij} \in \mathbb{R}^d$, that is $v_j \in \{x_{j1}, \ldots, x_{jd}\}$. The parameter $\Delta \in (0, 1)$ is determined according to the specific application scenario.

For the convenience of the experiment, we assume the sample size in each bag is the same when generating the dataset. Under the standard assumption, we use the probability, which reflects the anomaly status of the sample, to rank the samples in the bag, rather than discriminating the samples. In the subsequent discrimination analysis, the threshold δ in Model (1) is chosen to be the one which makes the highest F1 score in the training set. Under the collective assumption, we need to consider the proportion of the abnormal transaction amount among all transactions in the bag. When the proportion of abnormal transaction amount reaches the threshold $\Delta = 0.1$, we will consider the user's transaction bag to be overdue. Under each assumption, we have $N_1 = 200$ bags for training, $N_2 = 50$ bags for test. The size of the bag is $J = 10$. The anomaly rate of bags is 16.6% for the standard assumption and 5.8% for the collective assumption. Table 1 summarizes the experiment data.

Table 1. Dataset under the standard and collective assumptions.

Assumption	Type	Number of bags	Bag size	Feature dimension	Anomaly rate
Standard	Training	200	10	28	16.6%
	Test	50	10	28	16.6%
Collective	Training	200	10	28	5.8%
	Test	50	10	28	5.8%

We show the performance of the proposed method on both the standard assumption and collective assumption. We first evaluate the sample anomaly detection performance, and then analyse the performance of the sequence anomaly detection. In the sample anomaly detection part, we compare MILAD with the most commonly used unsupervised anomaly detection algorithm DAGMM in the financial field in terms of common model evaluation criteria (Precision, Recall, F1 score, AUC), as well as the interpretability of these two methods. In the Sequence Anomaly Detection part, we first built an idealized model, which is a model constructed based on the ideal assumption that the hidden labels are all available, hereafter denoted by the **Ideal** model. We use the Ideal model as the benchmark since it always has the best performance among all possible methods. We use AUC to evaluate model performances.

The computational resources of our experiments are *Windows 10, Intel(R) Core(TM) i5-9300H, GeForce GTX 1650 GPU, 16GB Ram*. We use *Python 3.8* under *Tensorflow 2.5.0* environment.

3.1.1. Sample anomaly detection under the standard and collective assumptions

Table 2 is the network structures of the experiments. For DAGMM, we use the same network structure under both assumptions, and we also use the same hyperparameter settings in Zong et al. (2018) ($\lambda_1 = 0.1, \lambda_2 = 0.005$). For MILAD, the Basic method is adopted under the standard assumption, and the Self-Attention method is adopted under the collective assumption. FC(a, b, c) is a full connection network, where a and b are the number of input and output neurons, and c is the activation function.

Figure 4 shows the loss function curves of these two algorithms with respect to 1000 epochs in the training processes. We can see that the DAGMM algorithm converges after 1000 epochs. Therefore we choose the model after the 1000 epochs of training as the final DAGMM model. For the MILAD algorithm, since we treat each bag as a sample group, the sample size is relatively small. It can be seen that the model is over fitted after 30 epochs of training under the standard assumption, and 10 epochs of training under the collective assumption. Therefore, under the standard assumption, we choose the model after 30 epochs

Table 2. Network structures under two assumptions.

Method	Layer	Structure
DAGMM	Compression Network (Encoder)	FC(28, 16, tanh) $\rightarrow$ FC(16, 4, tanh) $\rightarrow$ FC(4, 1, none)
	Compression Network (Decoder)	FC(1, 4, tanh) $\rightarrow$ FC(4, 16, tanh) $\rightarrow$ FC(16, 28, none)
	Estimate Network	FC(3, 10, tanh) – Dropout(0.2) $\rightarrow$ FC(10, 2, Softmax)
MILAD (Standard)	Transformer Network	FC(28, 16, ReLU) $\rightarrow$ FC(16, 8, ReLU)
	Attention Network	FC(8, 8, tanh) $\odot$ FC(8, 8, Sigmoid) $\rightarrow$ FC(8, 1)
	Classification Network	FC(8, 1, Sigmoid)
MILAD (Collective)	Transformer Network	Refer to the structure in Algorithm 3, where $d = 8$
	Attention Network	FC(8, 8, tanh) $\odot$ FC(8, 8, Sigmoid) $\rightarrow$ FC(8, 1)
	Classification Network	FC(8, 1, Sigmoid)

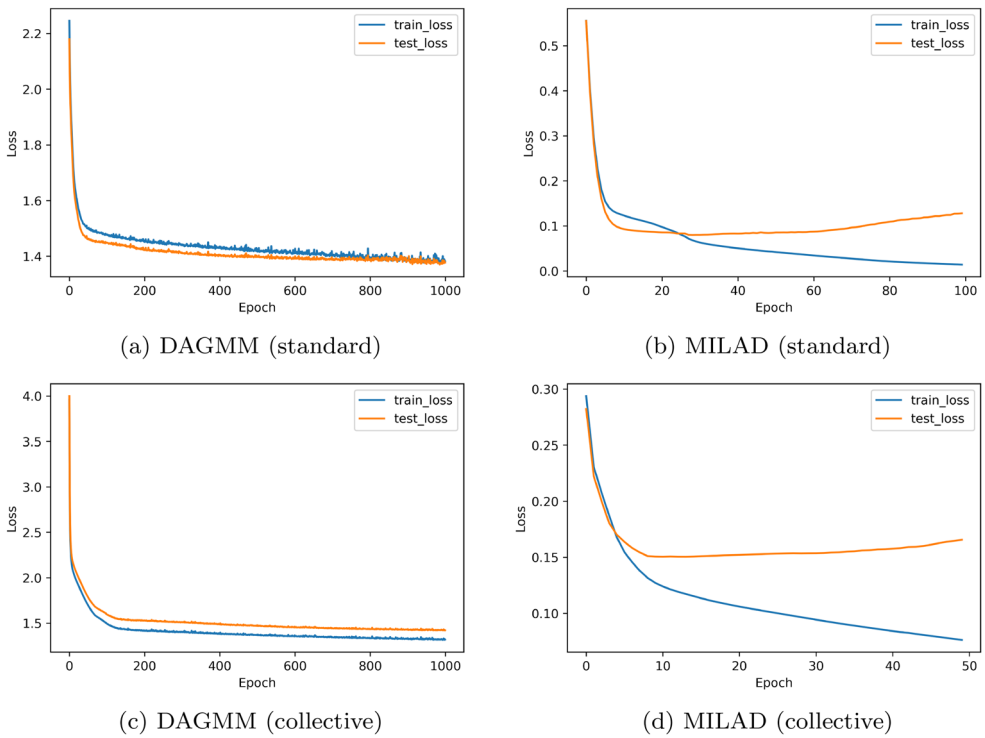

Figure 4. Loss curves under two assumptions. (a) DAGMM (standard). (b) MILAD (standard). (c) DAGMM (collective) and (d) MILAD (collective).

of training as the final MILAD model, while under the collective assumption, we choose the model after 10 epochs of training as the final MILAD model.

The final output of the DAGMM model is an energy score based on the sample's probability density within the Gaussian Mixture Model. Samples in low-density regions receive high energy scores and are considered anomalies. In order to compare it with the anomaly probability computed from the MILAD method, we use the function $f(x) = 1 - \frac{2}{\pi}\arctan(x)$ to convert this unbounded energy score into the probability ranged in (0,1). Figure 5 is the ROC curve of the model trained by the DAGMM algorithm and the MILAD algorithm. It can be seen that the performance of the MILAD algorithm is significantly better than that of the DAGMM algorithm under both assumptions.

Table 3 is the comparison matrix in several common model evaluation criteria. It can be seen that MILAD is significantly better than DAGMM in terms of these common model evaluation criteria, such as Precision, Recall, F1 score and AUC. This is because DAGMM is an unsupervised learning method, while MILAD is a supervised learning algorithm, which can effectively utilize the label information of the bag for complex data through the Attention based Multiple Instance Learning approach, and outperforms the unsupervised learning method. Therefore it is reasonable that MILAD achieves better performance, and is more useful in practice.

It is worth noting that the precision of DAGMM is notably low (less than 0.1) in Table 3. This is expected because DAGMM is trained without access to any labels (neither instance-level nor bag-level), relying solely on the data's underlying density structure to identify

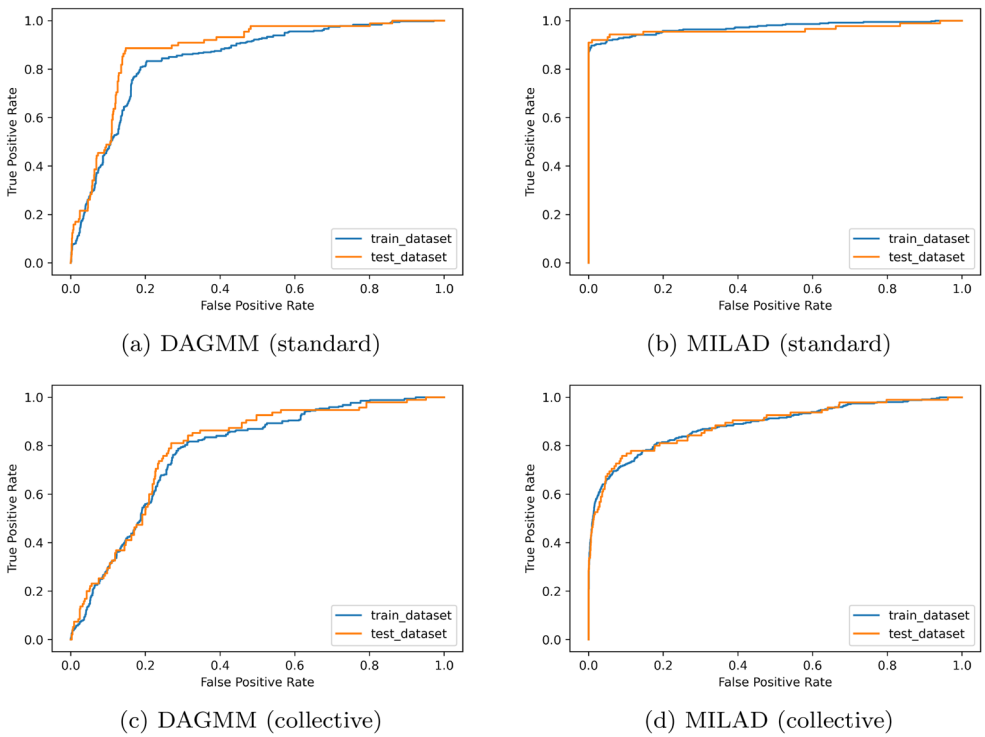

Figure 5. ROC curves under two assumptions. (a) DAGMM (standard). (b) MILAD (standard). (c) DAGMM (collective) and (d) MILAD (collective).

Table 3. Model comparison under two assumptions.

Assumption	Type	Method	Precision	Recall	F1-score	AUC
Standard	Training	DAGMM	0.0899	0.3722	0.1449	0.8397
		MILAD	**1.0000**	**0.8639**	**0.9270**	**0.9717**
	Test	DAGMM	0.0971	0.4545	0.1600	0.8769
		MILAD	**0.9302**	**0.9091**	**0.9195**	**0.9627**
Collective	Training	DAGMM	0.0580	0.2232	0.0921	0.7725
		MILAD	**0.6273**	**0.4000**	**0.4885**	**0.8854**
	Test	DAGMM	0.0561	0.2526	0.0918	0.7856
		MILAD	**0.5781**	**0.3895**	**0.4654**	**0.8878**

anomalies. Given that the dataset is highly imbalanced (only 1.70% true anomalies), it is extremely difficult for an unsupervised method to accurately identify this tiny fraction of positive samples, leading to a high number of false positives and thus very low precision. This result precisely demonstrates the limitation of unsupervised approaches and justifies the need for our proposed MILAD method, which can effectively leverage the available bag-level supervision.

To show the interpretability of the MILAD algorithm, we also check these abnormal sample bags to see whether the method can identify the abnormal samples in the bag that cause the bag abnormality. Table 4 is the samples' anomaly state (y_j) and their attention weights (w_j^*) of four randomly selected anomaly sample bags under the standard and collective assumptions. The attention weights $\{0.75; 0.78; (0.30, 0.27); (0.20, 0.21)\}$ of the abnormal samples that cause the abnormality of the entire bag are significantly larger than other samples in

Table 4. Attention weights of two randomly selected cases under two assumptions.

Assumption	Case	Infor	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Standard	I	Label	0	0	0	0	0	**1**	0	0	0	0
		Weight	0.03	0.04	0.02	0.00	0.02	**0.75**	0.05	0.02	0.04	0.03
	II	Label	0	0	0	**1**	0	0	0	0	0	0
		Weight	0.03	0.01	0.03	**0.78**	0.00	0.00	0.00	0.01	0.10	0.03
Collective	I	Label	**1**	**1**	0	0	0	0	0	0	0	0
		Weight	**0.30**	**0.27**	0.03	0.05	0.03	0.07	0.06	0.05	0.04	0.10
	II	Label	0	0	0	0	**1**	0	0	0	0	**1**
		Weight	0.07	0.04	0.12	0.06	**0.20**	0.06	0.11	0.11	0.02	**0.21**

Table 5. AUC under the standard and collective assumptions.

Assumption	Type	Ideal	MILAD	RNN	IF	RF	LSTM-AE	OC-SVM	DAGMM
Standard	Training	1	1	1	–	1	–	–	1
	Test	0.98	**0.98**	0.94	–	0.77	–	–	0.87
Collective	Training	1	1	1	–	1	–	–	1
	Test (Mean)	0.97	**0.97**	0.94	0.86	0.77	0.76	0.74	0.35
	Test (Std)	0.05	0.03	0.07	0.05	0.08	0.17	0.08	0.12

IF, LSTM-AE, and OC-SVM were evaluated only under the Collective Assumption as requested by the reviewer. '–' indicates that the method was not evaluated under this assumption.

the same bag. This result is consistent with our experiment setups, which fully demonstrates the outstanding interpretability of our MILAD method.

3.1.2. Sequence anomaly detection

For sequence anomaly detection we adopt the Xgboost algorithm, which is a commonly used binary supervised learning method in this field. In practice, there are only a few bags tending to be abnormal and overdue. Therefore the binary classification problem we are dealing with is a highly unbalanced data analysis problem in the experiment. For the highly unbalanced data problem, the Precision, Recall and F1 criteria are highly dependent on the threshold we use, where the threshold is mainly decided based on certain business demands in practice. Therefore the AUC is commonly used as the model evaluation criterion to evaluate the model performance, since it is a model evaluation criterion with respect to all possible threshold. All models are trained to achieve their best performances.

To provide a comprehensive comparison, in addition to DAGMM, we include five additional baseline methods: (a) **Random Forest (RF)**: treating the flattened sequence (all samples in a bag concatenated) as a single feature vector for classification; (b) **Recurrent Neural Network (RNN)**: using an LSTM network to process the sequence of samples within each bag for bag-level classification; (c) **Isolation Forest (IF)**: an unsupervised tree-based anomaly detector trained on instance-level features without using any labels; (d) **One-Class SVM (OC-SVM)**: an unsupervised method that learns a decision boundary around normal instances; (e) **LSTM Autoencoder (LSTM-AE)**: an unsupervised sequence-to-sequence autoencoder that detects anomalies based on reconstruction error. These baselines represent both supervised approaches (RF, RNN) that directly work on bag-level tasks and unsupervised approaches (IF, OC-SVM, LSTM-AE) that do not leverage any label information.

The AUC results are shown in Table 5. Under the Collective Assumption with 400 training bags and 100 test bags (averaged over 5 random seeds), MILAD achieves AUC = 0.97, matching the Ideal model's performance. This demonstrates that our three-step MIL framework effectively converts weak bag-level supervision into strong instance-level predictions.

Among supervised methods, RNN achieves AUC = 0.94, showing that end-to-end sequence learning is effective but still falls short of MILAD. RF performs moderately (AUC = 0.77), as it loses sequential structure by flattening the data.

Among unsupervised methods, Isolation Forest achieves the best performance (AUC = 0.86), likely due to its tree-based ensemble approach being robust to high-dimensional features. LSTM-AE (AUC = 0.76) and One-Class SVM (AUC = 0.74) show moderate performance. However, all unsupervised methods fall substantially below supervised methods, with a gap of at least 11 percentage points compared to MILAD. This demonstrates the critical importance of leveraging weak bag-level supervision. DAGMM performs worst (AUC = 0.35), confirming that purely unsupervised deep learning methods struggle with highly imbalanced data without any label guidance.

Notably, LSTM-AE shows high variance (std = 0.17), suggesting that reconstruction-based anomaly detection is sensitive to data distribution and less reliable for practical deployment.

We can conclude that MILAD is more feasible than other methods under both standard and collective assumption.

4. Conclusion

In this paper, we focus on the anomaly state evaluation of the data sequence caused by the abnormal samples contained in it. We propose a anomaly detection algorithm MILAD based on the Multiple Instance Learning techniques. We apply the proposed method to the delinquency risk detection in the credit card industry. The empirical results demonstrate that MILAD overcomes many shortcomings that existing methods have through its use of the sample information and the sequence anomaly information simultaneously to effectively identify abnormal samples. The proposed method can help financial institutions to control the overdue risk based on transactions directly and effectively.

Note that, abnormality is rare in most practical problems; that is there only exit a few abnormal sample bags. Therefore, the data of these binary classification problem are most likely to be highly imbalanced data. In that case, we should adopt the imbalance data analysis techniques. To evaluate the model performance for the imbalanced data, AUC will be a good choice.

It is important to note that the current MILAD framework treats each monthly billing cycle (bag) independently and does not model temporal dependencies across multiple months. While this single-month risk prediction is a common and practical problem in credit risk management, credit card delinquency is often the result of cumulative effects across multiple months (e.g., a user gradually increasing spending over several months before defaulting). Future research could extend MILAD to capture multi-month temporal dependencies by (1) treating each month as a 'super-instance' and applying a second-level MIL or RNN across months; (2) using a hierarchical model with MILAD at the transaction level (within-month) followed by an LSTM or Transformer at the month level (across-month); or (3) incorporating historical features (e.g., cumulative spending, payment history) as additional bag-level features. Such extensions would enable the model to capture longer-term behavioural patterns and potentially improve risk prediction accuracy.

Acknowledgments

The authors would like to thank the providers (Dal Pozzolo et al., 2015) of all datasets used in this work. The authors would like to thank the editors and reviewers for their valuable comments, which have significantly contributed to improving the manuscript.

Author contributions

CRediT: **Zhenguo Gao:** Conceptualization, Methodology, Supervision, Writing – original draft; **Yihao Bu:** Formal analysis, Investigation; **Xiaoxun Li:** Software, Validation; **Xiaoning Kang:** Supervision, Writing – review & editing

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Kang's research is supported by Social Science Foundation of Liaoning Province [grant number L25ATJ001].

ORCID

Zhenguo Gao https://orcid.org/0000-0003-1592-4495
Xiaoning Kang http://orcid.org/0000-0003-0394-6240

References

Bolton, R. J., & Hand, D. J. (2001). Unsupervised profiling methods for fraud detection. In *Credit Scoring and Credit Control VII* (pp. 235–255).

Carbonneau, M.-A., Cheplygina, V., Granger, E., & Gagnon, G. (2018). Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, *77*, 329–353. https://doi.org/10.1016/j.patcog.2017.09.023

Chen, J., & Gupta, A. K. (1997). Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, *92*(438), 739–747. https://doi.org/10.1080/01621459.1997.10474026

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence* (pp. 159–166).

Foulds, J., & Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, *25*(1), 1–25. https://doi.org/10.1017/S0269888909990293

Gao, Z., Du, P., Jin, R., & Robertson, J. (2020). Surface temperature monitoring in liver procurement via functional variance change-point analysis. *The Annals of Applied Statistics*, *14*, 143–159. https://doi.org/10.1214/19-AOAS1297

Gao, Z., Shang, Z., Du, P., & Robertson, J. L. (2019). Variance change point detection under a smoothly-changing mean trend with application to liver procurement. *Journal of the American Statistical Association*, *114*(526), 773–781. https://doi.org/10.1080/01621459.2018.1442341

Ilse, M., Tomczak, J., & Welling, M. (2018). Attention-based deep multiple instance learning. In *International Conference on Machine Learning* (pp. 2127–2136).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L., & Qi, Y. (2019). Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 4424–4431).

Lucas, Y., & Jurgovsky, J. (2020). Credit card fraud detection using machine learning: A survey. *arXiv:2010.06479*.

Rymarczyk, D., Borowa, A., Tabor, J., & Zielinski, B. (2021). Kernel self-attention for weakly-supervised image classification using deep multiple instance learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1721–1730).

Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6479–6488).

Tian, Y., Pang, G., Chen, C., Singh, R., Verjans, J. W., & Carneiro, G. (2021). Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4975–4986).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.