Hindawi
Security and Communication Networks
Volume 2021, Article ID 9941464, 9 pages
https://doi.org/10.1155/2021/9941464

Research Article

Explainable Fraud Detection for Few Labeled Time Series Data

Zhiwen Xiao and Jianbin Jiao

School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, BJ10, Beijing, China

Correspondence should be addressed to Jianbin Jiao; jiaojb@ucas.ac.cn

Received 26 March 2021; Revised 21 May 2021; Accepted 6 June 2021; Published 14 June 2021

Academic Editor: Liguo Zhang

Fraud detection technology is an important method to ensure financial security. It is necessary to develop explainable fraud detection methods to express significant causality for participants in the transaction. The main contribution of our work is to propose an explainable classification method in the framework of multiple instance learning (MIL), which incorporates the AP clustering method in the self-training LSTM model to obtain a clear explanation. Based on a real-world dataset and a simulated dataset, we conducted two comparative studies to evaluate the effectiveness of the proposed method. Experimental results show that our proposed method achieves the similar predictive performance as the state-of-art method, while our method can generate clear causal explanations for a few labeled time series data. The significance of the research work is that financial institutions can use this method to efficiently identify fraudulent behaviors and easily give reasons for rejecting transactions so as to reduce fraud losses and management costs.

1. Introduction

Fraud detection is an important technology for identifying abnormal behaviors in the financial field. It aims to detect fraudsters who have no intention to perform and to terminate transactions with potential default risks in advance for avoiding losses. Fraudsters hide deceptive and destructive intentions under disguised compliance behaviors [1] and use flexible and fickle tricks to evade verification from expert experience. Thus, unforeseen frauds pose a serious threat to the normal operation of financial institutions. Therefore, the development of fraud detection technology based on machine learning has gradually become an important consensus in the financial field to reduce financial risks.

Until recently, many technologies based on graph [2], SVM [3], neural networks [4, 5], and even ensemble methods [6] have been developed mature fraud detection mechanism. However, while literatures develop lots of very complex proposals, the financial industry needs comprehensible models to be used in practice, so the empirical usefulness of complex learners is reduced [7]. The reason is that financial institutions must be able to adequately explain the decision it made, especially the reasons for the refusal of the transaction [8]. Providing explainable results is a legal obligation of financial institutions in some countries [9] and an important basis for assisting credit operators in understanding the default factors and making correct decisions. Therefore, financial institutions should consider developing explainable fraud detection methods so as to present all parties to the transaction with significant causal identification results.

However, the known explainable machine learning methods have no impressive performance when applied to fraud identification. The reason is that in most practical financial scenarios, malicious fraud is not a common example. The cost of label collection, especially the collection of default labels, is very expensive. Especially for financial scenarios with a long transaction cycle, institutions have to wait until the end of the contract to fully affirm their willingness to perform in the transaction, so as to obtain a positive posterior label [10]. It means that in a large amount of data derived from historical transaction records, there are only a relatively small number of “good” samples and a smaller number of “bad” samples and most of the remaining samples are unlabeled. Therefore, it is difficult to develop fraud detection technology based on supervised learning on historical transaction data of known labels. In addition, unsupervised technology does not require knowledge of labels. However, most unsupervised methods are based on the assumption that fraud performance is an outlier in the distribution of transaction behavior data [11]. This assumption weakens the ability to recognize deliberate concealment or disguise. In summary, it is worth exploring to construct an explainable fraud detection classifier by integrating the complementary methods of these two technologies.

Our main contribution is to propose an explainable classification method by improving the multiple instance learning (MIL) framework so as to realize fraud detection for time series data with few labeled. Different from traditional methods, the one-to-one correspondence between samples and labels is not sensitive in MIL. Under the improved MIL framework, fraud detection on tracklets of time series data can obtain acceptable explanations for the prediction of each sample.

The remainder of this paper is organized as follows. Section 2 reviews the principles of MIL. We introduce the details of the method to enhance the explainability of MIL in Section 3. In Section 4, we verify the performance of the proposed method and compare it with some existing explainable technologies. Finally, our conclusions are provided in Section 5.

2. Related Work

MIL is a relatively novel weakly supervised machine learning method. It can achieve considerable performance when training datasets with poor label quality [12]. The MIL method arranges the training set in several labeled groups, which are called instance bags, and builds a classifier for the labels of bags. MIL treats a single instance in the training set as a subvector of the feature vector set in bags and only supervises the entire multibag. The training dataset of MIL can be expressed as follows:

$$X = \{(B_1, Y_1), (B_2, Y_2), \ldots, (B_n, Y_n)\}, \tag{1}$$

where $B_n \subseteq B$ represents the nth bag in the dataset and $Y_n \in \{-1, 1\}$ represents the label of the bag B_n in the binary classification problem.

Since the MIL method was proposed, research studies on this theoretical framework have produced many technologies. In the process of performing the classification task, the MIL method can be divided into two categories according to the position of extracting the feature information from the bag-space learning paradigm and the instance-space learning paradigm.

2.1. The Bag-Space Learning Paradigm. The bag-space learning paradigm takes each bag as an independent individual to extract information and assign class labels. Based on the global information in the bag level space, the bag-space learning paradigm tries to find a hyperplane that can separate the bags in the nonvector space so as to achieve an effective classification. For this reason, many research studies have focused on how to measure the distance or similarity between arbitrary bag-spaces. The Hausdorff distance [13], which is the Euclidean distance between the closest instance in two bags, was introduced to measure the distance between bags. Subsequent classification research at the bag level derived the embedded-space learning paradigm, which maps the bag-space to a single feature vector. The feature vector tried to express the whole information about a bag, and each feature vector has an associated label. In this paradigm, the original bag-space is mapped to an embedding space vector, and the classifier is trained in this new space. It converts the original problem into a standard supervised learning problem effectively and then applies any standard classifier for training. In the process of mapping bag-space to vector, the dimensionality of the embedding space is much higher than the number of training bags. Therefore, the study of embedding space learning has focused on the feature selection [14, 15].

2.2. The Instance-Space Learning Paradigm. The instance-space learning paradigm is based on the classifier in the instance space. After the instances have the classification result, the label of the package is determined according to the concept (label of instance). This paradigm infers from the instance level label to the bag level label, and there is an assumption about the relationship between the bag label and the instance label in the training set. In the standard MIL assumption, each instance has a hidden label. If and only if the bag contains at least one positive instance, the bag is marked as a positive bag. If and only if all the negative instances in the bag are negative, the bag is marked as a negative bag. The multi-instance bag classifier $F(B_i)$ is expressed by the following equation:

$$F(B_i) = \begin{cases} +1, & \exists x_n^i \in B_i : f(x) = +1, \\ -1, & \text{otherwise}, \end{cases} \tag{2}$$

where x_n^i is the nth instance in the bag B_i and $f(x)$ represents the discriminant function for inferring the label of the instance in the feature space. The generalization of the standard hypothesis leads to the collective assumption [16]. The label of a bag is determined by the multiclass label (concept) in the instance level. The expression is shown by the following equation:

$$F(B_i) = \begin{cases} +1, & \forall c \in \Pi^+ : \sigma_c \leq \sum_{x \in B_i} f_c(x) = 1, \\ -1, & \text{otherwise}, \end{cases} \tag{3}$$

where c is a concept, $\Pi^+ \in \Pi$ is a collection of positive concepts, σ is a predefined threshold, and $f_c(x)$ is the discriminant function under the concept c. Based on the above assumption, the classic mi-SVM method [17] trained a SVM classifier to update the label after the instance label has been initialized. This step is performed until the label no longer changes. The trained classifier is used to predict the label of instances. The improved method [18] calculated the probability of instance selection based on the training data gathered in a random subspace and used these probabilities to create a classifier pool for training subinstances. This method does not need to make any prior assumptions about the data structure and the proportion of instances in the bag.

When predicting time series data based on the MIL method, the similarity loss across bags [19] is introduced to model the sequential constraints between the news published on different days. This approach minimizes the total loss to obtain the probability of each news being a precursor. An improved MIL method [20] based on radial basis function (RBF) extracts features from transaction data to predict the likelihood of default based on behavior. An extend MIL method [21] was proposed to evaluate credit scores by transactional data and static individual information. This method considers the dynamic transactional data and cost-sensitive problem simultaneously. Essentially, the bag level label in MIL is pushed by the subinstance level label, and then the causal relationship between a bag label and subinstance label can be expressed by backtracking from the bag-space to the instance space. This shows that the instance-space learning paradigm is easier to enhance the explainability of the MIL algorithm. However, no relevant research has attempted to explore the prediction of time series data by the MIL methods from the perspective of enhancing model explainability. The main reason is that insufficient instance labels cannot be inferred using effective supervised learning methods, and it is not easy to train a classifier in the instance space. In summary, in the instance-space learning paradigm, how to obtain an instance label is the key to an explainable MIL classifier.

3. MIL Classification Method with Enhanced Explainability

In this section, the explainable MIL framework adopts the instance-space learning paradigm and a self-training semisupervised learning method. The calculation framework is shown in Figure 1:

Aiming at the learning of few labels at the instance space, the proposed method regards each input sample as a bag and splits the original behavior trajectory of each sample into tracklets at a specific time interval as a feature space in instance level. Each tracklet is an instance in the bag. We use the affinity propagation (AP) algorithm to improve the self-training method based on the long short-term memory (LSTM) model. It iteratively trains the classifier by learning the global information in the instance space and marks unlabeled instances.

In the initialization of the instance label Y^{ins}, the proposed method takes the label of the original training set as the bag label Y^{bag} and assigns it to all the instances in this bag. Among them, the instances in the unlabeled package are initialized as negative instances (no fraud). Clustering methods fit well for obtaining the hidden structure information in feature space [22]. Considering unlabeled tracklets, we cluster behavior tracklets on a low-dimensional manifold space. The clustering results are used to determine the behavior clusters of each tracklet; then, the existing labeled tracklets are used in the cluster to define the label of all tracklets in this cluster. The clustering diagram of the AP algorithm in the instance space is shown in Figure 2.

The clustering result makes the tracklet X^{ins} gather in a cluster $C_z \in C$ with the same behavior. Then, each instance represented by the tracklet can be labeled by clustering. The instance of cluster center is considered to be a behavior prototype $P_z \in P$.

When using the AP algorithm to perform unsupervised clustering of instances, the similarity between input data points is used as a clustering measure. The AP algorithm records the similarity in a matrix S, and the expression of elements in the matrix is shown as follows:

$$s(i,k) = -x_i - x_k^2. \tag{4}$$

The element $s(i,k)$ can be regard as the distance from the kth data point to the ith data point which can be a cluster center point (prototype). The AP algorithm does not preset the number of clusters and cluster centers in the initial stage but treats all data as candidate cluster centers and selects high-quality cluster centers as prototype instances through information exchange. The selection process of the cluster center point is realized by the matrix R (which represents responsibility) and the matrix A (which represents availability) where the element $r(i,k)$ in matrix R is defined as the degree that how the data point k can be used as the cluster center of the data point i. The element $a(i,k)$ in matrix A is defined as the degree that how the data point i selects the data point k as its cluster center suitability. The iterative process is as follows:

$$r_{t+1}(i,k) \longleftarrow \begin{cases} s(i,k) - \max\limits_{j\neq k}\{r_t(i,j) + a_t(i,j)\}, & i \neq k, \\ s(i,k) - \max\limits_{j\neq k}\{S(i,j)\}, & i = k, \end{cases} \tag{5}$$

$$a_{t+1}(i,k) \longleftarrow \begin{cases} \min\left\{0, r_{t+1}(k,k) + \sum\limits_{j\neq i,k} \max\{0, r_{t+1}(j,k)\}\right\}, & i \neq k, \\ \sum\limits_{j\neq i,k} \max\{0, r_{t+1}(j,k)\}, & i = k. \end{cases} \tag{6}$$

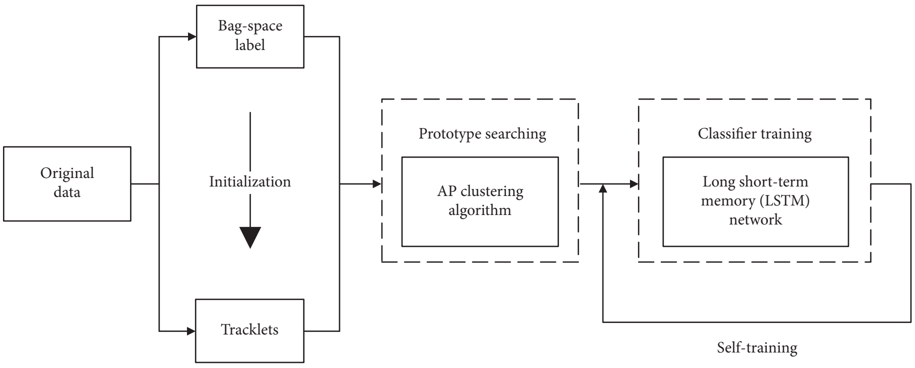

Figure 1: The computational framework of the MIL method with enhanced explainability.

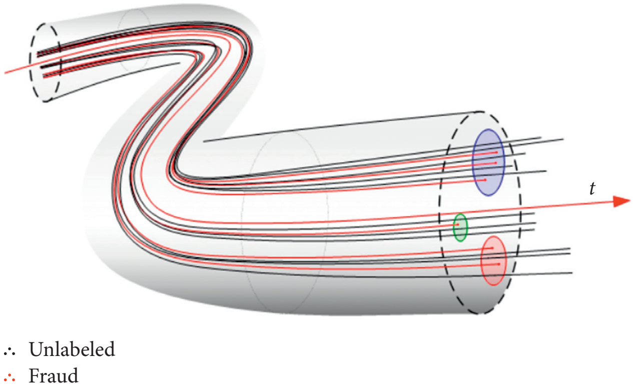

Figure 2: Low-dimensional manifold clustering of behavior tracklets in the instance space.

In the process of iterate operators, the damping factor $\lambda \in (0,1)$ is introduced to perform a weighted summation of the values before the operator iteration. It preserves the effective information generated during the last iteration and avoids the numerical oscillations during the iteration. The weighting calculation is shown in the following equations:

$$r_{t+1}(i,k) = \lambda \times r_t(i,k) + (1-\lambda) \times r_{t+1}(i,k), \quad (7)$$

$$a_{t+1}(i,k) = \lambda \times a_t(i,k) + (1-\lambda) \times a_{t+1}(i,k). \quad (8)$$

After the iteration, for any data point i, the maximum in $r_t(i,k) + a_t(i,k)$ of the data point k is selected as a cluster center point. Therefore, the element $S(k,k)$ on the main diagonal of the similarity matrix is bias parameters, which should be set to a larger value.

Next, we train the LSTM classifier iteratively. In each iteration, we train the LSTM classifier M_c and predict instance to update Y^{pre} after training. In the next iteration, each instance label-set Y_i^{pre} from a negative bag will be corrected to a negative label $y_i^{\text{bag}-}$ of its bag. Then, we use the cluster prototype label to correct the label of the instances in this cluster. Finally, the classifier M_c is retrained and updated in the training set with the updated labels. The self-training process is stopped when the prototype label-set Y^{pre} changes no longer significantly.

The pseudocode of the self-training method is shown in Algorithm 1:

We improve the hypothesis based on the inference method of fraud in the actual field. The hypothesis is that when at least one kind fraud is detected in the transaction, the sample is a positive instance of fraud. The classifier $F(B_i)$ is shown as the following equation:

$$F(B_i) = \begin{cases} +1, & \exists x_n^i \in B_i : f(x) \neq -1, \\ -1, & \text{otherwise}. \end{cases} \quad (9)$$

It can be observed from the expression that the proposed learning framework obtains the bag label by predicting the label of the tracklets. So, the instance label can be the reason for the bag label. The causality can be explained as follows: when the label of the sample is negative, it means that no abnormal behavior is detected; when the label of the sample is positive, there has been a fraudulent behavior belonging to a certain prototype in the sample at least for a period of time. In summary, when classifying time series datasets

```
Input: Instance-set $X^{\text{ins}}$ and bag label-set $Y^{\text{bag}} = \{y_1^{\text{bag}}, y_2^{\text{bag}}, \ldots, y_n^{\text{bag}}\}$
Output: Instance label-set $Y^{\text{ins}} = \{y_1^{\text{ins}}, y_2^{\text{ins}}, \ldots, y_n^{\text{ins}}\}$
  $X^{\text{ins}}$ clustering by the AP algorithm;
  Get cluster $C = \{C_1, C_2, \ldots, C_z\}$ and instance prototype $P = \{P_1, P_2, \ldots, P_z\}$;
  Initialize instance label $Y_i^{\text{ins}} \longleftarrow y_i^{\text{bag}}$ $(i \leq n)$;
  repeat
  $Y_i^{\text{pre}} \longleftarrow y_i^{\text{bag}-}$ p.s. negative bag label correction
  $Y_{C_k}^{\text{pre}} \longleftarrow y_{P_k}^{\text{pre}}$ p.s. prototype label correction
  LSTM classifier $M_c$ training in dataset $(X, Y^{\text{pre}})$
  Predict $X^{\text{ins}}$ by $M_c$, update $Y^{\text{pre}}$
  until $Y^{\text{pre}}$ not change
```

Algorithm 1: The self-training algorithm of the LSTM classification model.

represented by transaction behavior data, the expression of causality is the description of features of tracklets in the time-space. In the proposed ML framework, the MIL method can propose an explanation for bag label by predicting instances label, thereby improving the explainability of classification results for few labeled time series data.

4. Experimental Results and Analysis

In this section, we evaluate the performance of our proposed explainable method. We try to verify and answer two main questions through experiments. The first question is whether this method can provide better performance for real-world fraud detection tasks. The second question is whether this method can maintain a considerable performance in the few labeled training set. For this purpose, we have selected four classic methods as benchmarks for comparison with the proposed methods, namely: SVM, random forest (RF), AP clustering algorithm, and HOBA [5] method where the SVM technique is the most widely financial fraud detection technique used in data mining [23] and the RF model has good performance in many classification problems [24]. The AP clustering algorithm was selected to verify the effectiveness of the self-training process in our proposed method. The recently developed HOBA (homogeneity-oriented behavior analysis) has achieved outstanding experimental performance comparing with many related studies.

In order to compare the performance of these classifiers intuitively, we have selected representative metrics for commonly evaluating classifiers. Accuracy (Acc) is a standard performance indicator used to compare classifiers, $F1$-score ($F1$) is the harmonic mean of Precision and Recall, and AUC represents the area under the ROC curve. Because AUC does not include category distribution or misclassification costs, it is widely used to evaluate models trained on unbalanced datasets. In the classification problem, the calculation of AUC refers to the existing method, and the calculation is shown in the following equation:

$$\text{AUC} = \frac{1}{2}(1 + \text{TPR} - \text{FPR}), \tag{10}$$

where TPR is the true positive rate and FPR is the false positive rate. These three metrics can reflect the overall performance of the model. In addition, Recall and Precision are both important evaluation metrics in fraud detection tasks. Recall can reflect the ability to identify fraud risks, while Precision can reflect the discrimination cost of the classifier. The experiment compares the metrics of the proposed method with that of other benchmark classifiers to fully observe the performance of the proposed method.

We used different datasets to verify the two problems (details in this section below), but two datasets were preprocessed in the same way. We excluded duplicates, outliers, and accounts with no transactions in datasets. We sum the transaction data by date according to the timestamp and generate the two-dimensional feature vector of the Month × Date as the input of the proposed method. Furthermore, when training these benchmark classifiers, we have to reduce the dimensionality of two datasets to match the input requirements of benchmark classifiers. The detailed results of the two experiments will be introduced separately.

4.1. Performance Analysis in Fraud Detection. For the first question, we used a private credit card transaction details provided by an anonymous financial institution. Some fraudulent transactions were marked based on real investigations during the performance period, and the sample labels are incomplete. The dataset used in the experiment included a total of more than 5 million transaction records of 8057 accounts in 573 days, of which 1228 accounts (15.2%) show clear fraud during the performance period and 537 accounts have good performance labels and remaining accounts are unlabeled. We selected 100 positive samples and 100 negative samples with clear labels as the validation set and the remaining data as the training set.

After ten independent runs in different data partitions, the experimental results of each method are statistically analysed by the average values. The proposed method is compared with other comparison methods in identifying fraud categories (positive instances). The results are shown in Table 1.

In order to compare the performance of each classifier more intuitively, we show the fitted ROC curve in Figure 3.

We can observe that the improved MIL method based on self-training is close to the performance of the compared HOBA method. The performance of the RF model on this

Table 1: The experimental result on the real-world dataset.

	$F1$	Acc ± std (%)	Recall (%)	Precision (%)	AUC
AP	0.5911	58.58 ± 1.24	61.25	58.25	0.5852
SVM	0.5698	61.24 ± 3.96	52.15	64.58	0.6788
RF	0.6596	67.35 ± 0.56	63.28	69.37	0.6122
HOBA	0.8377	84.52 ± 1.34	80.25	87.91	0.8424
Ours	0.8317	82.53 ± 1.56	84.13	82.39	0.8360

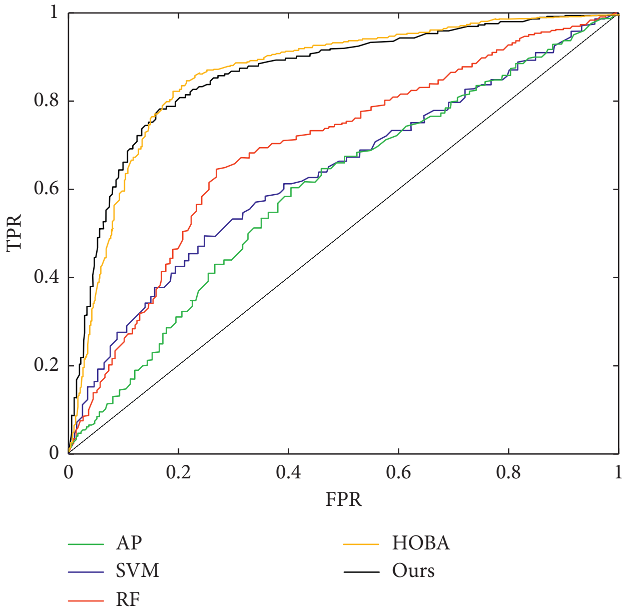

Figure 3: ROC curve of each classifier on the real-world dataset.

dataset is still better than that of the other two traditional methods, which is consistent with most previous research results. $F1$-score, accuracy, and AUC value can reflect that the overall performance of the two methods which are significantly ahead of other classifiers. Among them, the proposed method has the highest Recall rate; it means the proposed method is more conducive to the financial business that cannot accept false rejections. The comparison results show that it is difficult to classify transaction data composed of time series data by using traditional methods. Our proposed method achieves considerable performance almost the same as the state-of-the-art method when dealing with such tasks, and it is worth mentioning that our method is explainable. In summary, the effectiveness of the proposed explainable method is verified in predicting actual time series data.

4.2. Performance Analysis in Few Labeled Dataset. For the second question, we compared the influence of the number of labels in the same dataset on our proposed method. Due to the inherent privacy nature of financial transactions, no transaction dataset is legally published. This prevents us from collecting sufficient labeled transaction data. For this reason, our comparative experiment is verified on the dataset generated by the PaySim simulator. The PaySim simulator based on the agent-based simulation technology framework, combined with the application of mathematical statistics [25], proved that the simulation data can be used as the original dataset for research. The generated dataset contains a total of 1,852,392 transaction records from nearly 1,000 accounts for more than 700 days, of which 9,651 fraudulent transaction data (0.5%) were randomly mixed. The mixing of fraudulent data has resulted in 25% of the samples being fraudulent. The dataset divides the last 6 months of transaction details into verification set. The training set contains 1,296,674 transaction records from 870 accounts, while the test set contains 555,718 transaction records from 218 accounts.

We randomly hide the labels of 50% of the accounts to construct a compared dataset. Considering that each benchmark classifier is based on supervised learning, we only input labeled data for the benchmark learner and input all data for the proposed method. After ten independent runs in different data partitions, the experimental results of each method are statistically analysed by the average values. The performance of each classifier on the PaySim simulation dataset is shown in Table 2.

Table 2: The experimental results on the PaySim simulation dataset.

	$F1$	Acc ± std (%)	Recall (%)	Precision (%)	AUC
AP	0.5481	80.22 ± 2.87	59.47	50.82	0.7006
SVM	0.5911	75.68 ± 2.39	48.91	69.17	0.7343
RF	0.6837	82.74 ± 1.39	59.60	80.18	0.8171
HOBA	0.6439	83.95 ± 2.12	63.23	66.67	0.7658
Ours	0.7789	83.63 ± 1.08	59.04	98.72	0.8910

Table 3: The experimental results on the PaySim simulation dataset with missing labels.

	$F1$	Acc ± std (%)	Recall (%)	Precision (%)	AUC
AP	0.4233	60.49 ± 3.74	32.28	61.46	0.6083
SVM	0.5911	51.57 ± 2.42	25.32	54.13	0.5275
RF	0.6837	58.96 ± 2.47	31.22	61.46	0.5982
HOBA	0.3543	53.46 ± 1.74	54.12	27.18	0.5346
Ours	0.7789	84.58 ± 2.27	61.92	86.69	0.8511

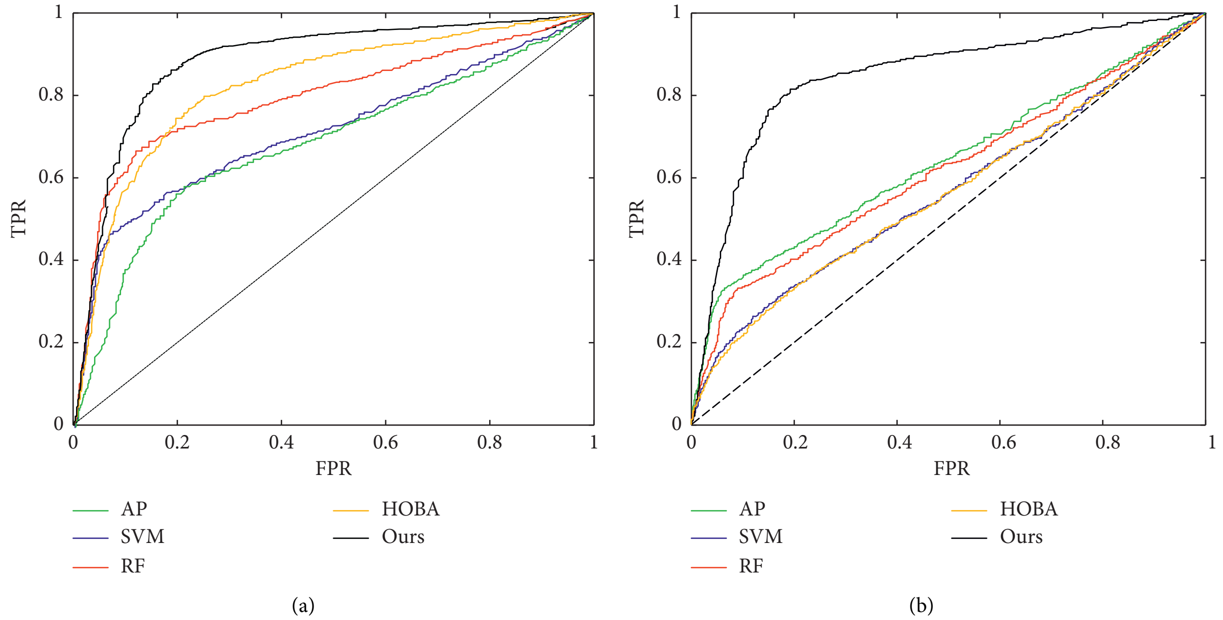

Figure 4: (a) ROC curve of each classifier on the PaySim simulation dataset; (b) ROC curve of each classifier on the PaySim simulation dataset with missing labels.

The performance of each classifier on the PaySim simulation dataset with 50% missing labels is shown in Table 3.

The fitted ROC curves of each classifier on the two comparison datasets are shown in Figure 4.

We observe that our proposed method provides better experimental results on the PaySim simulation dataset. The performance of the HOBA method is much better than that of the other three benchmark classifiers. The proposed method has the highest Precision rate in experiments. It proves that the proposed model can detect more than half of the suspected fraud under the expected low FPR rate. After some labels were hidden, the performance of all benchmark classifiers dropped significantly and the performance of the HOBA method decreased the most. However, the proposed method still maintains significant performance on the dataset with missing labels, which shows that our improved self-training model effectively learns the hidden fraud features in the dataset through a semisupervised method. From these experimental results, it can be observed that the LSTM model of self-training in the MIL framework has stronger fraud detection capabilities for few labeled data in the real financial field.

Table 4: The explainable results of the proposed classifier.

Sample ID	Predicted label	Explanation
****6293	Compliance	No abnormal behavior detected
****8428	Compliance	No abnormal behavior detected
****7570	Fraud	Type III prototype in August
****2085	Fraud	Type VI prototype in May and type II prototype in June

4.3. Explanation Analysis. The method we propose can give intuitive and concise reasons for the classification prediction results of every testing samples. The explainable predicted results of several examples are shown in Table 4.

Among them, when the label is Compliance, there is only one reason for no abnormal behavior detected. However, there are many reasons for Fraud label; this is a combination of multiple fraud categories. From the MIL framework, we observed that the category label of the sample is determined by the multi-instance bag label, and the reason is the concept in bag, for example, sample with ID ****7570; its bag label is Fraud; the reason is expressed as "August feature in the bag is marked as type III." It means that a sample with with ID ****7570 transaction behavior is suspected to be a fraudulent prototype of Type III in August. Therefore, our proposed method provides an explainable prediction method for the real-world time series data through the self-training LSTM prediction model with the AP clustering algorithm in the MIL framework. In addition, with the in-depth application of the proposed method, fraud prototypes can introduce descriptions based on expert experience to achieve a more vivid explanation of the predicted results.

5. Conclusions

In this paper, we proposed a fraud detection method with enhanced explainability in the MIL framework, which incorporates the AP clustering method in the self-training LSTM prediction model. Compared with previous work, we focus on the actual problems of real financial data and obtain a classifier with high predictive performance and clear causal explanation on a few labeled dataset.

The empirical research is based on two datasets, compared with three benchmark classifiers and variated the proposed method from two aspects. First, the real dataset from an anonymous organization is used to evaluate the overall performance of the proposed method. Compared with other classifiers, the proposed method is more effective in predicting actual transaction data. Then, the data generated by the PaySim simulator are used to verify the performance changes in the case of hiding labels. When 50% of account labels are artificially hidden, the proposed method still maintains good predictive performance even when the benchmark classifiers generally drop in performance. It verifies that the proposed method can effectively learn and distinguish the fraud features hidden in the dataset. The empirical analysis results provide trustable evidence, which proves in two steps that our proposed method can complete the classification task with significant performance advantages.

As far as we know, among many fraud detection methods for transaction data, this research is one of the few classification techniques that can obtain a clear casual explanation. The significance of our work is that financial institutions can efficiently identify fraudulent behaviors and easily give reasons for rejecting of transactions so as to reduce the fraud losses and management costs. However, our work still has limitations in the prediction problem for large-scale datasets. The complexity of the AP algorithm leads to higher requirements for computing resources. Therefore, in future work, we hope to explore possible combinations of more advanced clustering algorithms and deep learning to develop more efficient fraud detection methods.

Data Availability

The desensitized sensitive data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Wu, Y. Xu, and J. Li, "Feature construction for fraudulent credit card cash-out detection," *Decision Support Systems*, vol. 127, Article ID 113155, 2019.

[2] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo, "Fraud detection: a systematic literature review of graph-based anomaly detection approaches," *Decision Support Systems*, vol. 133, Article ID 113303, 2020.

[3] N. Rtayli and N. Enneya, "Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization," *Journal of Information Security and Applications*, vol. 55, Article ID 102596, 2020.

[4] A. Rb and S. K. Kr, "Credit card fraud detection using artificial neural network," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 35–41, 2021.

[5] X. Zhang, Y. Han, W. Xu, and Q. Wang, "HOBA: a novel feature engineering methodology for credit card fraud detection with a deep learning architecture," *Information Sciences*, vol. 557, pp. 302–316, 2021.

[6] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit card fraud detection using pipeling and ensemble learning," *Procedia Computer Science*, vol. 173, pp. 104–112, 2020.

[7] R. Florez-Lopez and J. M. Ramon-Jeronimo, "Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5737–5753, 2015.

[8] J. N. Crook, D. B. Edelman, and L. C. Thomas, "Recent developments in consumer credit risk assessment," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1447–1465, 2007.

[9] M. B. Gorzałczany and F. Rudziński, "A multi-objective genetic optimization for fast, fuzzy rule-based credit classification with balanced accuracy and interpretability," *Applied Soft Computing*, vol. 40, pp. 206–220, 2016.

[10] P. Jrana and J. Baria, "A survey on fraud detection techniques in ecommerce," *International Journal of Computer Applications*, vol. 113, no. 14, pp. 5–7, 2015.

[11] F. Carcillo, Y.-A. Le. Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, vol. 557, pp. 317–331, 2019.

[12] M.-A. Carbonneau, V. Cheplygina, E. Grange, and G. Gagnon, "Multiple instance learning: a survey of problem characteristics and applications," *Pattern Recognition*, vol. 77, pp. 329–353, 2017.

[13] Z. Zhouyu Fu, A. Robles-Kelly, and J. Jun Zhou, "MILIS: multiple instance learning with instance selection," *Ieee*

Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 958–977, 2011.

[14] L. Yuan, J. Liu, and X. Tang, "Combining example selection with instance selection to speed up multiple-instance learning," *Neurocomputing*, vol. 129, pp. 504–515, 2014.

[15] L. Yuan, X. Wen, H. Xu, and L. Zhao, "Multiple- instance learning with empirical estimation guided instance selection," in *Proceedings of 2018 24th International Conference on Pattern Recognition*, pp. 770–775, Beijing, China, August 2018.

[16] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.

[17] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proceedings of the 15th International Conference on Neural Information Processing Systems*, pp. 577–584, MIT press, Cambridge, MA, USA, January 2002.

[18] M.-A. Carbonneau, E. Granger, A. J. Raymond, and G. Gagnon, "Robust multiple-instance learning ensembles using random subspace instance selection," *Pattern Recognition*, vol. 58, pp. 83–99, 2016.

[19] Y. Ning, S. Muthiah, H. Rangwala, and N. Ramakrishnan, "Modeling precursors for event forecasting via nested multi-instance learning," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1095–1104, San Francisco, CA, USA, 2016.

[20] T. Zhang, W. Zhang, W. Xu, and H. Hao, "Multiple instance learning for credit risk assessment with transaction data," *Knowledge-Based Systems*, vol. 161, pp. 65–77, 2018.

[21] W. Zhang, "Cost-sensitive multiple-instance learning method with dynamic transactional data for personal credit scoring," *Expert Systems with Applications*, vol. 157, Article ID 113489, 2020.

[22] C. He, J. Shao, J. Zhang, and X. Zhou, "Clustering-based multiple instance learning with multi-view feature," *Expert Systems with Applications*, vol. 162, Article ID 113027, 2020.

[23] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: a comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, Article ID 100402, 2021.

[24] H. D. Nayak, F. Deekshita, L. Anvitha, A. Shetty, D. J. D'Souza, and M. P. Abraham, "Fraud detection in online transactions using machine learning approaches-a review," *Advances in Intelligent Systems and Computing*, pp. 589–599, 2021.

[25] E. Lopez-Rojas, A. Elmir, and S. Axelsson, "Paysim: a financial mobile money simulator for fraud detection," in *Proceedings of the Annual Simulation Symposium*, pp. 249–255, Larnaca, Cyprus, September 2016.