Learning Transactional Behavioral Representations for Credit Card Fraud Detection

Yu Xie, Guanjun Liu, *Senior Member, IEEE*, Chungang Yan, Changjun Jiang,
MengChu Zhou, *Fellow, IEEE*, and Maozhen Li

***Abstract*—Credit card fraud detection is a challenging task since fraudulent actions are hidden in massive legitimate behaviors. This work aims to learn a new representation for each transaction record based on the historical transactions of users in order to capture fraudulent patterns accurately and, thus, automatically detect a fraudulent transaction. We propose a novel model by improving long short-term memory with a time-aware gate that can capture the behavioral changes caused by consecutive transactions of users. A current-historical attention module is designed to build up connections between current and historical transactional behaviors, which enables the model to capture behavioral periodicity. An interaction module is designed to learn comprehensive and rational behavioral representations. To validate the effectiveness of the learned behavioral representations, experiments are conducted on a large real-world transaction dataset provided to us by a financial company in China, as well as a public dataset. Experimental results and the visualization of the learned representations illustrate that our method delivers a clear distinction between legitimate behaviors and fraudulent ones, and achieves better fraud detection performance compared with the state-of-the-art methods.**

***Index Terms*—Attention, credit card fraud detection, long short-term memory (LSTM), transactional behavioral representations.**

Nomenclature

D	Transaction dataset.
U	User set.
G	Transactional expansion set.
m	Number of original features in a transaction record.
n	Number of transactions.
x_i^u	ith transaction record of a given user u.
g_i^u	Expansion of a transaction record x_i^u.
L_u	Set of transaction records of a given user u.
n_u	Number of transaction records of a given user u.

I. Introduction

With the development of technology, credit card payment has become a popular payment mode worldwide. However, transaction frauds often occur since fraudsters can utilize some technological means (e.g., Trojan horse and credential stuffing attacks) to embezzle card accounts for the use of unauthorized funds [1]. According to the Nilson Report, the fraud-related losses worldwide were about \$35 billion in 2020 [2]. Therefore, it is necessary for credit-card-related companies to take measures to combat credit card fraud.

When a credit card fraud detection model is constructed based on machine learning, the feature representation of transaction records is the core of model training [3]. Numerous previous studies designed such models by using the original features, such as transaction time, location, and amount in transaction records or some aggregated features as their transactional representations [4], [5], [6], [7], [8], [9], [10]. However, the information provided by such features is insufficient to identify frauds [11]. Although the aggregated features can represent some trading habits of users, some more important behavioral information, e.g., the behavioral changes of users, is not fully represented [12].

Usually, fraudulent behaviors hide behind massive legitimate behaviors [13]. Fraudsters pretend to trade like legitimate users in order to hoodwink a detection system. As a result, they are considered legitimate users by the system, especially when only the transactional behavior of one single transaction record of the fraudster (i.e., a snapshot of the fraudster) is considered by such a detection system [14]. Therefore, we believe that frauds can be better identified when considering the consecutive transactions of users instead of just taking a snapshot into account. Moreover, repeated behaviors of users can be used to help a detection model distinguish fraudulent transactional behaviors from legitimate ones, but are not well considered in the existing models.

Manuscript received 14 June 2021; revised 16 November 2021, 29 March 2022, and 23 June 2022; accepted 18 September 2022. Date of publication 5 October 2022; date of current version 5 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2100801, in part by the Shanghai Science and Technology Committee under Grant 22511105500, and in part by the Bengbu University 2021 High-Level Scientific Research and Cultivation Project under Grant 2021pyxm04. *(Corresponding authors: Changjun Jiang; MengChu Zhou.)*

Yu Xie, Guanjun Liu, Chungang Yan, and Changjun Jiang are with the Key Laboratory of Embedded System and Service Computing, National (Province-Ministry Joint) Collaborative Innovation Center for Financial Network Security, Tongji University, Shanghai 201804, China (e-mail: tjxy@tongji.edu.cn; liuguanjun@tongji.edu.cn; yanchungang@tongji.edu.cn; cjjiang@tongji.edu.cn).

MengChu Zhou is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Maozhen Li is with the Department of Electronic and Electrical Engineering, Brunel University London, UB8 3PH Uxbridge, U.K., and also with the Key Laboratory, Tongji University, Shanghai 201804, China (e-mail: maozhen.li@brunel.ac.uk).

Digital Object Identifier 10.1109/TNNLS.2022.3208967

Based on the above idea, this work aims to learn a new representation for each transaction record from the aspect of historical consecutive transactions of users. Indeed, several previous studies have attempted to learn such a representation by utilizing a sequence learning model for a sequence prediction task [15], [16], [17]. However, they have some limitations. For instance, long short-term memory (LSTM) and its variants are generally used, but assume that the behavioral changes caused by different intervals between the consecutive time steps are fixed, and time dependencies are relatively short [18]. Since the correlation between current transactions and historical transactions is not fully considered, they cannot well mine the features of transactional behaviors.

In order to adapt to the nonfixed interval between continuous time steps in transactions, we augment a time-aware gate to LSTM and make it act on other control gates to learn the behavioral changes caused by different time intervals. For example, there exist the following two cases: Case 1 where user u_i has conducted two transactions within an interval of six months and Case 2 where user u_j has conducted two transactions within a time interval of 2 min. It is obvious that the two kinds of behavioral changes are different since their firing intervals are different, but this difference was not considered by the previous sequential learning models. For Case 1, an ideal model should pay more attention to the current transaction because the transactional behavior of u_i possibly has changed greatly, but, on the contrary, it should pay more attention to the previous transactions for Case 2, and if user u_j conducts multiple transactions in a short time, it should give an early warning to u_j even when they all seem legitimate. This work also designs a current-historical attention module to build up the connections between current transactions and historical ones, and capture possibly repeated behaviors. Each historical attention module is regarded as an attention mechanism for the current time step. In addition, this work designs an interaction module to enable neural units of each time step to learn more comprehensive and reasonable representations.

A real-world credit card transaction dataset and a public one are used in our experiments. The results show that, compared with the state-of-the-art methods, our method can achieve better performance. By visualizing the low-dimensional embedding vectors of legitimate and fraudulent behavioral representations, it can be seen that the representations learned by our method can better distinguish fraudulent and legitimate transactions.

The rest of this article is organized as follows. Section II presents the related work. The preliminary is given in Section III. Section IV illustrates the proposed model. The experimental setup and results are introduced in Sections V and VI, respectively. Section VII concludes this article.

II. Related Work

A. Recurrent Neural Networks

One of the limitations of the traditional neural network is that it cannot learn coherently from long sequences. The recurrent structures of solving this problem are neural networks with recurrent units (i.e., RNN), allowing information to persist for a long time [19]. In the past few years, the application of RNN to various problems achieved incredible success: speech recognition, language modeling, translation, image subtitle, and so on, and their application scope is still expanding [20], [21], [22], [23]. One of the attractions of RNNs is that they have abilities to connect previous information to the current task. For example, using previous video frames may assist in understanding the current frame.

However, RNN cannot well solve the problem of long-term dependence. Thus, many variants are proposed to improve RNN, e.g., LSTM is a very successful recurrent neural network and widely used in many tasks [24]. LSTM is explicitly designed to avoid long-term dependency problems since remembering long-span information is its default behavior. It contains a cell state and three control gates, namely, forget gate, input gate, and output gate. Another innovative variant of RNN is the gated recurrent unit (GRU) [25]. It combines forget and input gates into an update gate and also incorporates cell states and hidden states. The resulting model is simpler and more popular than the standard LSTM model. In addition, there are some other effective variants. Update gate RNN (UG-RNN) [26] is designed to cope with long sequences. This method adds a single gate to decide whether a given layer should be hidden or updated, so as to improve the learning performance and trainability of long sequences. The work [27] proposes a new LSTM variant, Time LSTM (T-LSTM), to model sequential actions of users for recommendation. It equips LSTM with the designed time gates to capture users' interests. HAINT-LSTM [18] modifies the forget gate of traditional LSTM by considering the frequency of calls. A self-historical attention mechanism is added to allow long-term dependencies and more external information to be considered in the transmission of neural units, such as profiles of users. NHA-LSTM is an extension of HAINT-LSTM, where an improved network embedding method, FraudWalk, is proposed to construct embedding for nodes in interactive networks, and thus, it can reveal potential group frauds [28]. The literature [16] employs LSTM to incorporate transaction sequences. Some new features are added based on some predefined rules. The time between two consecutive transactions in minutes (time delta) is employed as an additional feature to support time normalization on inputs. The work [29] proposes a novel recurrent unit called time-aware LSTM to handle irregular time intervals in longitudinal patient records. A long and short-term memory is added in the recurrent unit to record user behaviors. The work [30] adds a time attention module in a recurrent neural unit and uses time data to extract the characteristics of the click behaviors.

In summary, the following aspects are not fully considered in previous neural-network-based transaction fraud detection tasks: 1) behavioral changes of users caused by different transaction time intervals are not well captured; 2) the connections between current and historical transactions are not fully captured; and 3) employing the combination of multiple information to learn more comprehensive and reasonable features is imperative.

B. Credit Card Fraud Detection

A typical fraud detection system generally consists of an automatic fraud detection model and a manual analysis process operated by business investigators [31]. The automatic fraud detection model is to monitor all incoming transactions and score them, which is generally produced by some data mining techniques [32], [33]. The manual procedure is that the business investigators review the suspicious transactions with high fraud scores marked by an automatic fraud detection model and then provide feedbacks (fraudulent or legitimate) [8]. The construction of an automatic fraud detection model can be based on expert- or data-driven methods or their combination [34]. The expert-driven methods attempt to identify specific fraud scenarios by analyzing historical frauds and then form some rules representing some fraud modes [35]. The data-driven methods are generally based on machine learning algorithms to train a fraud detection model [36]. For example, the literature [4] combines supervised and unsupervised learning techniques for constructing credit card fraud detection models and presents a number of criteria to compute outlier scores at different levels of granularity. The work [37] proposes an online boosting approach by coupling it with the extremely fast decision tree as the base learner in order to ensemble them into a single online strong learner for credit card fraud detection. The study [38] trains a generative adversarial network to output mimicked minority class examples, which are then merged with training data into an augmented training set so that the effectiveness of a classifier can be improved. However, most of them only use the original features of transaction records to train a model. The information provided by these original features fails to well reflect the characteristics of transactions, and thus, the performance of the trained model is not too good [12].

There have been some methods for transactional representations that are based on transaction aggregation strategies. The study [3] uses an aggregation strategy to add new features to original transaction records. The aggregation is to group the transactions made in recent hours according to the related cardholder id and transaction type. Then, the number of these grouped transactions, the total amounts spent on these transactions, and their average amount in different time windows are calculated as new features. Several studies [11], [39], [40], [41], [42] followed the method in [3]. For example, the aggregation method in [11] depends not only on cardholder id and transaction type but also on the country and merchant code, thus leading to a much richer feature space than that in [3]. They also create a new set of features based on analyzing the periodic behavior of the time of a transaction using the Von Mises distribution, which can analyze the periodicity of transactional behaviors of users. Several studies employ the representations of this method as their transactional representations to deal with the fraud detection problem by combining different machine learning algorithms [13], [14], [43], [44]. The work [45] presents a framework based on the hidden Markov model to associate the likelihoods of a transaction with its previous transactions sequence. These likelihoods are used as additional features for fraud detection.

TABLE I
ORIGINAL FEATURES OF TRANSACTION DATASETS

Attributes name	Description
User id	Users unique identity
Pay_single_limit	The limit on the amount of a single transaction for a user
Pay_accumulate_limit	The limit of the user's daily accumulated transaction amount
Common phone	User's usual mobile phone number
Is_common_ip	Is IP commonly used when the transaction occur
Card area	The area for the card
Trade date	Date of the transaction
Trade time	Time of the transaction
Trade amount	Amount of the transaction
Card balance	Account balance before payment
Transaction object	Recipient number of a person or a business
Client mac	The client MAC address when the transaction occur
Label	Legitimate or fraudulent transaction

Although these aggregated features can provide rich information for the detection model, original classifiers limit the ability of the detection model to automatically learn complicated transactional behaviors of users.

III. Preliminaries

In this section, our real-world transaction dataset is introduced. Then, exploratory and theoretical analyses are exhibited to reveal the transactional behavioral characteristics of fraudsters, and the motivation of our method is elaborated. Transactional expansion is finally presented.

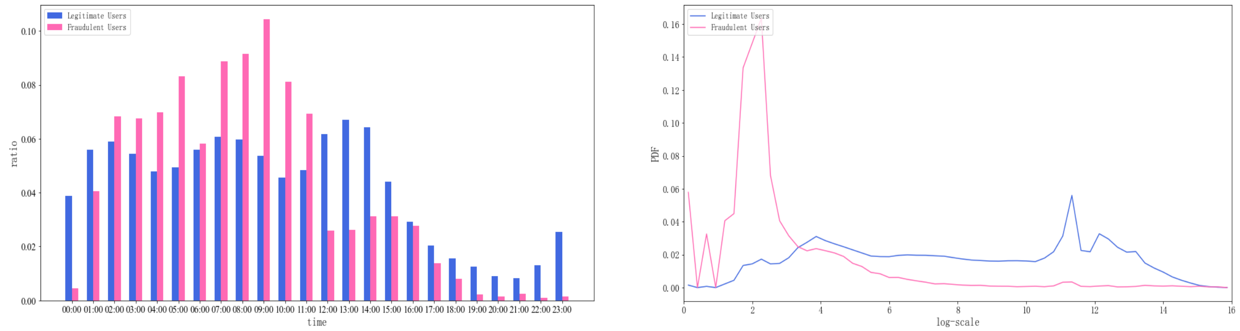
A. Description of Real-World Dataset

We have obtained a large real-world transaction dataset provided by a financial institution in China. The dataset contains 5.12 million online transaction records of 107 192 users, including 4.98 million legitimate transactions and 0.14 million fraudulent ones. Each transaction record has 12 attributes (i.e., original features), as shown in Table I. Each transaction record contains a label, where *1* represents the fraudulent transaction and *0* represents the legitimate one.

B. Exploratory Analysis of Transactional Behaviors

Based on the real-world transaction dataset, we can reveal the characteristics of fraudsters by answering: 1) do users have specific time regularity in their transactional activities? and 2) do fraudsters have any unique behavioral patterns?

1) *Attribute of Trade Time of Users:* We separate the transactions of legitimate users and fraudsters according to



(a)

(b)

Fig. 1. Explorative analysis of transactional behaviors. (a) *Trade time* of users' transactions. (b) Distribution of the time between two consecutive transactions.

their labels, count the ratio of the number of transactions per half-hour to the total number of transactions, and display them in a histogram. As shown in Fig. 1(a), the vast majority of fraudulent transactions occur between 1:30 A.M. and 12:00 P.M., and there are almost no fraudulent transactions between 6:30 P.M. and 1:00 A.M., with the largest number of transactions occurring at 9:30 A.M. However, legitimate users conduct a mass of transactions after 12:00 P.M., and there are many transactions between 6:00 P.M. and 1:00 A.M. This means that the distribution of *trade time* of legitimate users is relatively uniform, while the fraudsters mainly trade in a fixed period of time.

2) *Behavioral Patterns of Fraudsters:* In order to further explore the differences in behavioral patterns between legitimate users and fraudsters, we first transform *trade time* of each transaction record into a time stamp and then calculate the time between two consecutive transactions for legitimate users and fraudsters separately. Next, we get the bins of time intervals and calculate the probability value of each bin, i.e., the ratio of the number of intervals in a bin to the total number of intervals of legitimate users or fraudsters is counted, where the horizontal axis represents the logarithm of the consecutive time intervals. As shown in Fig. 1(b), the vast majority of fraudsters conduct transactions within a short time interval, which indicates that the fraudsters tend to engage in intensive fraudulent activities. However, legitimate users span a wide range of time intervals. In fact, short intervals of consecutive transactions indicate that the users are engaged in intensive and planned activities, and there is a strong behavioral dependence between such transactions.

Based on the above analysis of the transactional behavioral characteristics, we conclude that:

1) The transactions of fraudsters usually take place in a fixed period of time.
2) The activities of fraudsters are manifested as mostly intensive and planned transactions.
3) There is a great difference between the transactional behaviors of fraudsters and those of legitimate users.

They motivate us to develop an effective and high-performance model to conduct credit card fraud detection.

C. Theoretical Explanation

Data mining theories and technologies play an important role in credit card fraud detection, as it is generally applied to extract and reveal the hidden fraud patterns behind a large number of transactions, where feature extraction is an effective and representative one [40], [46]. Traditional methods use statistical analysis and probability theory to extract and identify useful information, and subsequently generate transactional representations [47]. With the popularity of neural networks, more and more studies employ neural-network-based techniques to learn comprehensive transactional representations [28], [30]. Compared with traditional regression analysis, neural networks have no mode limitation in analysis and can be detected automatically, especially when there are correlations among transactions [48], [49]. In a word, neural-network-based feature extraction techniques can be employed to analyze the correlation among transactions and learn new transactional representations to change the original feature space for getting richer information [18], which enhances the separability of fraudulent and legitimate transactions. From the perspective of credit card fraud detection, fraudsters generally conduct continuous, intensive, and planned transactions, which means that the transaction behaviors of fraudsters and legitimate users are very different in the time interval. In addition, with the passage of time, transaction behaviors of users will change, i.e., the concept drift problem exists in fraud detection, which means that the behavioral changes caused by different intervals between the consecutive time steps cannot be fixed. Thus, in our work, we propose a neural-network-based feature extraction technique to construct our detection model, which can learn informative transactional behavioral representations to better identify frauds.

D. Transactional Expansion

Given a user $u \in U$ and u's transaction records $L_u = \{x_1^u, x_2^u, \ldots, x_{n_u}^u\}$, each $x_i^u = (x_{i_1}^u, x_{i_2}^u, \ldots, x_{i_m}^u)$ is an m-dimensional vector, and m is the number of original features of

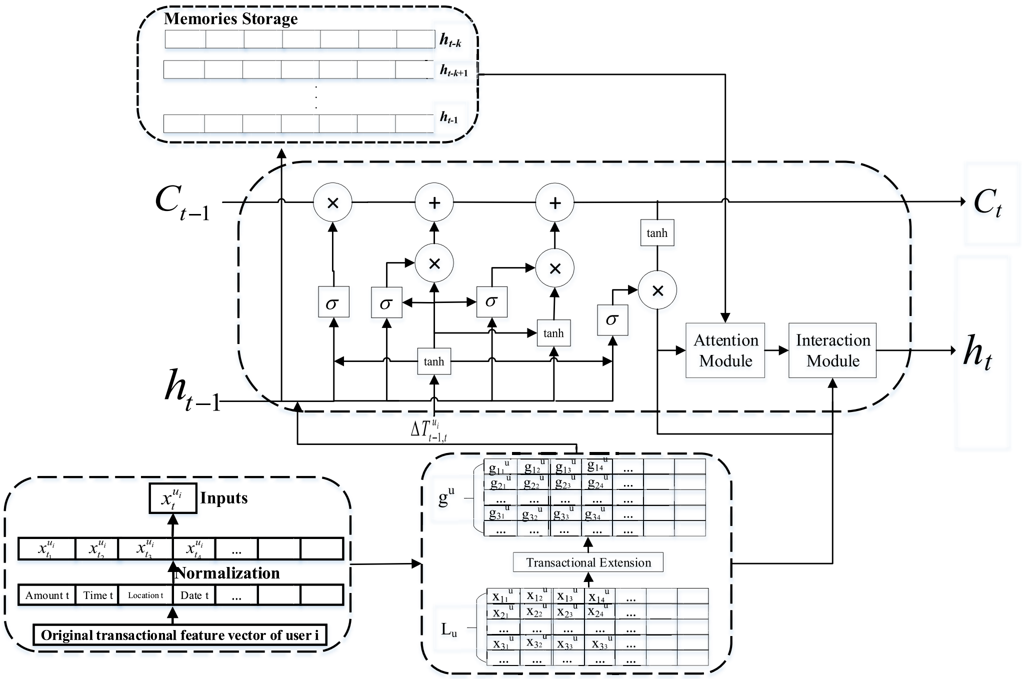

Memories Storage
h_{t-k}
h_{t-k+1}
h_{t-1}
C_{t-1}
C_t
×
+
+
×
×
×
σ
σ
σ
σ
tanh
tanh
tanh
tanh
Attention Module
Interaction Module
h_t
h_{t-1}
$\Delta T^{u_i}_{t-1,t}$
$x_t^{u_i}$ Inputs
$x_{t_1}^{u_i}$
$x_{t_2}^{u_i}$
$x_{t_3}^{u_i}$
$x_{t_4}^{u_i}$
...
Normalization
Amount t
Time t
Location t
Date t
...
Original transactional feature vector of user i
g^u
g_{11}^u g_{12}^u g_{13}^u g_{14}^u ...
g_{21}^u g_{22}^u g_{23}^u g_{24}^u ...
g_{31}^u g_{32}^u g_{33}^u g_{34}^u ...
Transactional Extension
L_u
x_{11}^u x_{12}^u x_{13}^u x_{14}^u ...
x_{21}^u x_{22}^u x_{23}^u x_{24}^u ...
x_{31}^u x_{32}^u x_{33}^u x_{33}^u ...

Fig. 2. Illustration of the proposed TH-LSTM.

transaction records. To enable the model to learn the transactional behavioral representation for each transaction record from the user's historical transactions, transactional expansion is performed for each transaction record.

For the ith transaction record x_i^u in L_u, we first construct a transaction group g_i^u with $d+1$ elements

$$g_i^u = \left\{x_{i-d}^u, x_{i-d+1}^u, \dots, x_i^u\right\}$$

where d represents the memory size and g_i^u represents the memory of x_i^u. In order to equalize the lengths of all groups, i.e., when i is less than or equal to d, we replicate the earliest transaction record of the user. For example, for x_1^u, we have $g_1^u = \{x_1^u, x_1^u, \dots, x_1^u\}$ in which the number of x_1^u is $d+1$; for x_2^u, we have $g_2^u = \{x_1^u, \dots, x_1^u, x_2^u\}$ in which the number of x_1^u is d; and for x_d^u, we have $g_d^u = \{x_1^u, x_1^u, x_2^u, \dots, x_{d-1}^u, x_d^u\}$ in which the number of x_1^u is 2. In other words, for the kth element x_{i-d+k}^u in g_i^u, $k \in (0, d)$, if $i-d+k \le 1$, then the element is x_1^u. The idea of this design is inspired by the conclusion of [16]: compared with the recent transactions, the earliest transaction has less impact on the current transaction and, thus, has a weaker impact on learning the relationships between the current transaction and its recent ones. Therefore, $L_u = \{x_1^u, x_2^u, \dots, x_{n_u}^u\}$ of user u is represented as a set of transaction groups, i.e., $g^u = \{g_1^u, g_2^u, \dots, g_{n_u}^u\}$ is the transactional expansion of L_u.

It is worth mentioning that, for a transaction dataset, some users have thousands of historical transactions, while some have dozens only. If d equals the maximum number of a user's transactions, then, for each transaction, its memory can be expressed as the transactional behavior of all its historical transactions. For a natural language processing task, many studies set the length of each sentence to be the longest one, so as to unify the length of inputs [50], [51], [52]. However, for many users, the long memory enables the earliest transactional behavior to get more attention if we set d as the number of all transactions, which contradicts the common view that the current transaction of a user is generally similar to her/his recent transactional behavior. In addition, if d is set too small (i.e., a short memory), e.g., $d = 2$, it is difficult for this case to reveal the historical transactional behaviors. Thus, the representations cannot be learned sufficiently from the historical transactions of users. Therefore, we choose $n/|U|$ as the value of d, where n is the number of transactions in the dataset and $|U|$ is the number of users. We demonstrate the rationality of this setting latter.

IV. Proposed Model

After giving the exploratory analysis of transactional behaviors of users and motivation of this work, we propose a new time-aware historical-attention-based LSTM (TH-LSTM) model, as shown in Fig. 2, to learn the transactional behavioral representation for each transaction record of users.

A. Structure of Recurrent Unit

Our exploratory analysis results reveal that fraudsters tend to perform intensive and planned activities in a short period of time, while the time spans of legitimate users are wider. Therefore, it is essential to leverage the time intervals between consecutive transactions to express the motivation and relevance of the transactional behaviors of users. One of the limitations of traditional LSTM is that it only records the chronological order of the inputs but fails to consider the changes in transactional behaviors caused by different time intervals, which means that all the previous behaviors have the same impact on the current transaction [53]. In fact, if the last transaction occurred too long ago, the transactional behavior of users may have changed a lot, and then, it may not be as influential as their recent behavior. Thus, we try to design a kind of recurrent unit that can capture the different changes in behaviors according to the varied length of the time intervals and transfer the information to the next recurrent unit. Therefore, we embody this idea in a time-aware gate of the recurrent unit. At each time step, a time-aware gate receives its previous hidden state information, the current transaction, and the time interval between the current and previous transactions of a user and interacts with other control gates. The longer the time interval, the less information in the previous time step is considered, and the more information about the current step is stored in the memory of the recurrent unit. On the contrary, the shorter the time interval, the more information about the previous time step is stored in the memory of the recurrent unit.

Thus, our basic recurrent unit is designed as follows:

$$s_t = \mathbf{tanh}(W_{sh}h_{t-1} + W_{sx}x_t + W_{st}\,\Delta\mathrm{T}_{t-1,t} + b_s) \tag{1}$$

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + W_{fs}s_t + b_f) \tag{2}$$

$$\text{s.t. } W_{fs} < 0$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + W_{is}s_t + b_i) \tag{3}$$

$$\mathcal{T}_t = \sigma(W_{\mathcal{T}h}h_{t-1} + W_{\mathcal{T}x}x_t + W_{\mathcal{T}s}s_t + b_{\mathcal{T}}) \tag{4}$$

$$\zeta = \mathbf{tanh}(W_{uh}h_{t-1} + W_{ux}x_t + W_{us}s_t + b_u) \tag{5}$$

$$c_t = f_t * c_{t-1} + i_t * \zeta + \mathcal{T}_t * s_t \tag{6}$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + W_{os}s_t + b_o) \tag{7}$$

$$\tilde{h}_t = o_t * \mathbf{tanh}(c_t) \tag{8}$$

where h_{t-1} represents the output of the recurrent unit at the $(t-1)$th time step and x_t is the input of the recurrent unit at the tth time step. $\Delta\mathrm{T}_{t-1,t} = \mathrm{Timestamp}_t - \mathrm{Timestamp}_{t-1}$ is the value of time interval between the tth and $(t-1)$th transactions. f_t, i_t, $\mathcal{T}_t$, and o_t mean a forget gate, an input gate, a time-aware gate, and an output gate, respectively. s_t represents a time-aware state where the information of time interval is stored. ζ is the generated candidate cell state, and c_t is the new cell state. $\tilde{h}_t$ is the candidate hidden state of the recurrent unit at the tth time step. W_{sh}, W_{fh}, W_{ih}, $W_{\mathcal{T}h}$, W_{uh}, W_{oh}, W_{sx}, W_{fx}, W_{ix}, $W_{\mathcal{T}x}$, W_{ux}, W_{ox}, W_{st}, W_{fs}, W_{is}, $W_{\mathcal{T}s}$, W_{us}, and W_{os} are the weight matrices that can be randomly initialized and then jointly learned during the learning process. Their dimensions are based on the number of features of inputs (vocabulary_size) and the number of nodes in neural units (num_nodes). b_s, b_f, b_i, $b_{\mathcal{T}}$, b_u, and b_o are the bias vectors of dimensions $\mathbb{R}^{1\times num\_nodes}$. Parameter settings are described in Section V-D. σ represents a sigmoidal nonlinear function, and **tanh** represents a tanh function [54]. $*$ is an elementwise product.

B. Current-Historical Attention Module

As mentioned in Section I, we can find intensive and planned transactional behaviors from the transactions of fraudsters. In addition, behavioral periodicity exists in transactions of legitimate users. Thus, we try to reveal the correlation between historical transactions and current one of users as context information.

Recently, attention mechanisms have become a research hotspot in industrial and academic domains [55], [56], [57], [58], [59]. Practitioners in the field of artificial intelligence have widely applied attention-based models to speech recognition [21] and natural language processing [22], [60], achieving great success. Their initial purpose is to solve the problem of multidependencies and measure the correlation among elements [23]. It is also essential to expose the similarities and multidependencies among transactions in this study's context. Inspired by the philosophy of such a mechanism, we design a current-historical attention module in our TH-LSTM to capture the repeated activities and behavioral periodicity by building up the connections between current and historical transactions of users. Considering the information transmission characteristics of our designed recurrent unit, the current-historical attention module can further capture the regularity that may be forgotten by the recurrent unit in the long historical memories

$$q_t = \mathbf{Concat}(\tilde{h}_t, c_t) \tag{9}$$

$$o_{t,i} = \mathbf{tanh}(W_{aq}q_t + W_{ah}h_i + b_a) \tag{10}$$

$$\alpha_{t,i} = \frac{\exp((o_{t,i})^T v_t)}{\sum_{i=t-d}^{t-1}\exp((o_{t,i})^T v_t)} \tag{11}$$

$$e_t = \sum_{i=t-d}^{t-1}\alpha_{t,i} * h_i \tag{12}$$

where $i \in \{t-d, t-d+1, \ldots, t-1\}$ is the index of the historical state of the recurrent unit. Thus, $h_i = \{h_{t-d}, h_{t-d+1}, \ldots, h_{t-1}\}$ are historical memories stored by each time step, where d represents the memory size. q_t is the tth state of the recurrent unit. Here, the candidate hidden state of the recurrent unit $\tilde{h}_t$ and the new cell state c_t at the tth time step are concatenated as q_t [18]. In order to score each historical state h_i, we compare it with the current state of recurrent unit q_t to get the scores $o_{t,i}$ and further weight $o_{t,i}$ to avoid the model paying too much attention to one behavior while ignoring others, where v_t is the importance weights that measure the importance of $o_{t,i}$ [59]. $o_{t,i}$ and v_t are multiplied and then normalized as the final attention weights. e_t is the weighted sum of all historical memories and represents the correlation between historical behaviors and current transactions, which contains the regularity of a user's historical behaviors and the influence of those on current transaction. W_{aq}, W_{ah}, and b_a are the weight matrices and bias, which can be randomly initialized and then jointly learned during a learning process.

C. Interaction Module

Previous models focus on the last memory transferred from the recurrent unit but do not fully consider the interaction of multiple representative information [18]. In order to enable the model to obtain a comprehensive and reasonable judgment and learn more effective transactional behavioral representations, we regard e_t as context information, $\tilde{h}_t$ as the regularity and motivation of the user, and g_t^u as the original information and further interact them to get the transactional behavioral representation of each transaction record

$$h_t = \mathbf{tanh}\big(W_h \tilde{h}_t + W_e e_t + W_g g_t^u + b_h\big) \tag{13}$$

where g_t^u is the transactional expansion of x_t^u. W_h, W_e, and W_g are the weight matrices that are randomly initialized and then jointly learned during a learning process. b_h is the bias. $\tilde{h}_t$, e_t, and g_t^u are first multiplied with their corresponding weight matrix, respectively, and then, the sum of them is fed into a one-layer multilayer perception (MLP) to generate the output of the current recurrent unit, which is passed to the next recurrent unit. Based on the above process, the whole recurrent neural unit in our model is finished.

D. Learning Process

For transaction dataset D, each transaction record x_i^u is first transformed into transactional expansion g_i^u and then fed to the model. After d time steps where d is the memory size, the output h_i of the last recurrent unit is regarded as the transactional behavioral representation of a transaction record x_i^u, i.e., x_i^u is represented as h_i. Then, considering that credit card fraud detection is a binary classification task, we add a classification layer to the model

$$\hat{y}_i = \sigma(W_i h_i + b_i), \quad i \in \{1, 2, \ldots, n\}. \tag{14}$$

The cross-entropy loss [61] is employed as a loss function

$$\text{Loss} = -\sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{15}$$

where y_i is the true label of transaction record x_i^u and $\hat{y}_i$ is the predicted one. W_i and b_i are a weight matrix and bias. Therefore, the parameters of the model can be jointly learned during a learning process.

E. Training Algorithm

Algorithm 1 illustrates the training process of our method. The Adam optimization algorithm [62] is employed to minimize the loss function since it is commonly used in credit card fraud detection.

Algorithm 1 Training Algorithm of TH-LSTM

```
Inputs: Transaction dataset: $D$; Batch generater: $B$; Network parameters: $\theta$; Size of each mini-batch: $batch\_size$; Number of batches: $n\_batch$; Number of epochs: $n\_epoch$;
Output: Trained model $M$;
Conduct the transactional expansion: $D \longrightarrow G$;
Initialize the network parameters $\theta$;
for each $j \in \{1, 2, \ldots, epoch\}$ do
    for each $k \in \{1, 2, \ldots, num\_batch\}$ do
        $batch = next(B(batch\_size, G))$;
        for each $i$ in $batch$ do
            Compute $h_i$ and $\hat{y}_i$ by (13) and (14)
        end for
        Compute the loss by (15)
        Minimize the loss by Adam and update the network parameters $\theta$
    end for
end for
return $M$;
```

V. Experimental Setup

In this section, the benchmark methods are first introduced. Then, we show our data partition. In addition, the evaluation criteria of credit card fraud detection and the parameter settings of the proposed model are presented.

A. Benchmark Methods

1) *LSTM:* This method is an improved structure of RNN. It is mainly used to solve the problem of gradient disappearance and gradient explosion in the process of long sequence training, obtaining better performance than vanilla RNN in dealing with longer sequences. A memory cell and three control gates (forget, input, and output ones) are the basic components in LSTM [24].
2) *GRU:* This method is also a widely used RNN variant. In the structure of recurrent neural networks, LSTM and GRU both use a gating mechanism, but the latter uses fewer parameters and has a faster training speed. It contains a reset gate determining how to combine the new input information with the previous memory and an update gate defining how much of the previous memory is saved to the current time step [25].
3) *Bidirectional RNN (Bi-RNN):* The basic idea of Bi-RNN is that each training sequence consists of two RNNs: forward connections and backward ones. Forward ones help a model learn from previous representations, and backward ones help a model learn from future representations [63].
4) *UG-RNN:* This innovative method begins with a vanilla RNN and adds a single gate to decide whether to carry over the hidden state or update, which improves the learning performance and trainability for long sequences [26].
5) *Aggregated-Feature-Based LSTM (A-LSTM):* This method integrates the state-of-the-art feature aggregation strategies and phrases the fraud detection problem as a sequence classification task. LSTM network is employed in A-LSTM to incorporate transaction sequences. A new feature called tdelta is presented as an additional feature. Their aggregation functions are restricted to two functions that are proposed in their cited work: the total amount spent and the number of transactions. Thus, we use these two features as aggregation functions in our experiments. They also compute such pairs (sum_{S_k} and $count_{S_k}$) for each element in the power set of country, merchant category, card entry mode, and a time horizon of 24 h. Thus, we chose some similar features in our dataset, i.e., card area, transaction object, and client mac as our feature set, when we conduct our experiments [16].

6) *Spatiotemporal Attention-Based Neural Network (STA-NN):* This method presents an attention-based 3-D convolution neural network for credit card fraud detection. Temporal aggregated features and spatial aggregated ones are generated, i.e., original transaction records are aggregated into tensor format $\chi \in \Re^{N_1,N_2,N_3}$, where N_1, N_2, and N_3 denote the dimensions of temporal, spatial, and feature slices, respectively [17].
7) *Homogeneity-Oriented-Behavior-Analysis-Based Feature Engineering Framework (HOBA):* This method presents a feature engineering framework based on homogeneity-oriented behavior analysis (HOBA) to generate feature variables for the fraud detection model. Deep learning techniques are incorporated into the fraud detection system to deliver good fraud detection performance [12].
8) *Aggregated-Feature-Based CNN (A-CNN):* This method proposes a CNN-based fraud detection framework to extract the intrinsic patterns of fraud behaviors. It transforms transaction records into a feature matrix for each transaction, and thus, the inherent interactions in time series can be extracted for the CNN model. A new feature called trading entropy is proposed to extract more complex frauds [15].
9) *Historical Attention-Based and Interactive LSTM (HAINT-LSTM):* This method modifies the forget gate of the traditional LSTM since the authors consider the frequency of calls. A self-historical attention mechanism is added to allow long-time dependencies, and more external information (such as personal profiles) is considered in the transmission of neural units [18].
10) *T-LSTM:* This method adds specific inner gating units in LSTM to extract the interests of users for the recommendation, which are controlled by the time interval between two actions. It has been widely used in predicting users' next actions in recommendation systems [27].
11) *Time Attention-Based Heterogeneous RNN (TAH-RNN):* This method adds a time attention module in a recurrent neural unit and uses time data to extract the characteristics of the click behavior. The intuition of this idea is that the time interval of click behavior denotes the degree of interest or familiarity of a user [30].

B. Data Partition

In the construction of a credit card fraud detection model, uncertainty lies in the great change in terms of the accuracy of the trained model in practical applications [13]. For instance, a trained model performs well on transactions that occurred in the past month but not in the current month. This case is mainly caused by the diversity of users: 1) the types of users are diverse; 2) the behavior of users is changing with time; and 3) fraudsters are constantly improving their fraud strategies in order to deceive a fraud detection system In order to verify a model through more reasonable and dependable experimental results, and get closer to the real scene, we divide the training set and the test set according to the *trading date* of a transaction dataset. Specifically, as shown in Table II, we first utilize data in Jan. as the training set and data in Feb. as the test set. Then, data in Feb. are used as the training set, data in Mar. as the test set, and so on.

TABLE II
DATA PARTITION

Dataset	#Legitimate	#Fraudulent	#Total
Jan.	656416	28175	666591
Feb.	657899	33762	691661
Mar.	226206	10601	264807
Apr.	1229764	23271	1243035
May.	1189117	27122	1216299
Jun.	1017816	24898	1042714

TABLE III
CONFUSION MATRIX OF CLASSIFICATION

	Actual positive	Actual negative
Predicted positive	T_P	F_P
Predicted negative	F_N	T_N

C. Evaluation Criteria

Considering that fraud detection is regarded as an imbalanced classification task [64], the following commonly used evaluation criteria are employed to evaluate the proposed model:

$$\text{Precision } (P_r) = \frac{T_P}{T_P + F_P}$$
$$\text{Recall } (R_e) = \frac{T_P}{T_P + F_N}$$
$$F_1 = \frac{2P_r R_e}{P_r + R_e}$$

which are extracted by a confusion matrix as shown in Table III. Specifically, precision (P_r) refers to how many detected suspicious transactions are fraudulent. Recall (R_e) means how many fraudulent transactions are identified from all fraudulent transactions in the transaction dataset. F_1 comprehensively evaluates the performance of a fraud detection model. AUC, which represents the area under the ROC curve, is also a commonly used evaluation criterion in imbalanced classification tasks. Thus, it is also considered in our credit card fraud detection task to evaluate our model.

D. Parameter Settings

In our experiments, the size of the minibatch is set to 1000. The numbers of units in TH-LSTM and training epochs in the training algorithm are set to 64 and 100, respectively. The number of transactional features (vocabulary_size) is 11, as shown in Table I, where *User id* is used for transactional expansion and dropped before inputting into the model. Besides, the dropout technique [65] is used to prevent overfitting and the dropout ratio is set to 0.8. The learning rate in the Adam optimization algorithm is set to 0.001.

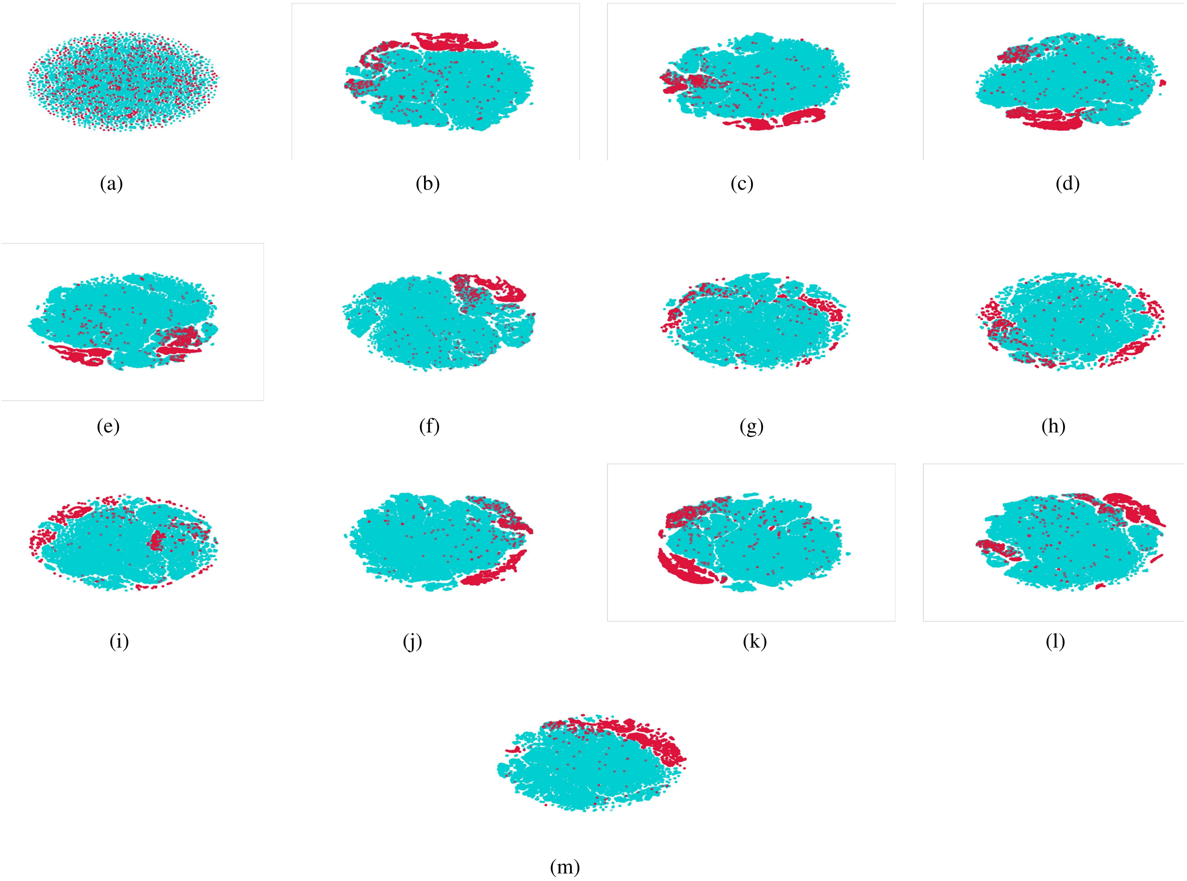


Fig. 3. Explorative view of transactional behaviors (blue and red colors (dots) represent legitimate and fraudulent transactions, respectively). (a) Original features of transaction records. (b) STA-NN. (c) A-LSTM. (d) HOBA. (e) A-CNN. (f) LSTM. (g) GRU. (h) Bi-RNN. (i) UG-RNN. (j) HAINT-LSTM. (k) T-LSTM. (l) TAH-LSTM. (m) TH-LSTM.

VI. Experimental Results

In this section, the visualizations of the varied transactional representations are first presented to give a more intuitive view of the learning effectiveness among tested models. Then, we evaluate the performance of the proposed model and the benchmark methods. The Holm test is employed to further compare the experimental results between the proposed model and its peers. An ablation study is performed to verify the effectiveness of each module of our model. We visualize the normalized attention weights to provide the interpretability of an attention module. In addition, we evaluate the sensitivity of memory size in the proposed model.

A. Visualization of the Varied Transactional Representations

In order to show the learned transactional behavioral representations more intuitively, we visualize the varied transactional representations of different models on a 2-D space by T-SNE [66], as shown in Fig. 3. Fig. 3(a) is the original transactional representations, i.e., the original features of transaction records. Fig. 3(b)–(l) shows the learned transactional representations of the benchmark methods. Fig. 3(m) shows the learned transactional behavioral representations of our proposed model. The blue dots represent legitimate transactions, and the red dots represent fraudulent ones. As shown in Fig. 3(a), the representations of legitimate transactions and fraudulent ones are all mixed together, which means that the original transactional representations fail to disclose the behavioral characteristics of legitimate users and fraudsters. Fig. 3(b)–(l) illustrates that, although the benchmark methods can identify some characteristics of users' behaviors, they still fail to group the behaviors of fraudsters together, which means that their learned transactional representations cannot well identify the fraudulent behaviors precisely. However, from Fig. 3(m), we can see that the fraudulent behaviors are gathered together and represented in a group, and there is a clear separation between legitimate and fraudulent behaviors, which means that our proposed model can well identify the characteristic of the most users' behaviors by the learned transactional behavioral representations.

B. Credit Card Fraud Detection

We compare the proposed model with its peers via a real-world transaction dataset and a public one, which aims to further evaluate the effectiveness of the learned transactional behavioral representations and the proposed model for credit card fraud detection. The experimental results on the real-world transaction dataset are shown in Table IV. The best results are highlighted in bold, and the higher, the better.

TABLE IV
CREDIT CARD FRAUD DETECTION PERFORMANCE OF 15 METHODS

Train	Test	Criteria	STA-NN	A-LSTM	HOBA	A-CNN	LSTM	GRU	Bi-RNN	UGRNN	HAINT-LSTM	T-LSTM	TAH-LSTM	TH-LSTM
Jan.	Feb.	P_r	0.761	0.831	0.828	0.835	0.874	0.837	0.864	0.823	0.844	0.847	0.883	**0.921**
		R_e	0.960	0.963	0.956	0.952	0.823	0.911	0.933	**0.989**	0.917	0.925	0.922	0.972
		F_1	0.849	0.892	0.888	0.890	0.848	0.872	0.897	0.898	0.879	0.884	0.902	**0.946**
		AUC	0.874	0.920	0.913	0.916	0.905	0.943	0.956	0.969	0.946	0.926	0.952	**0.980**
Feb.	Mar.	P_r	0.884	0.914	**0.963**	0.949	0.919	0.913	0.879	0.871	0.962	0.917	0.915	0.939
		R_e	**0.729**	0.668	0.642	0.661	0.609	0.692	0.705	0.656	0.539	0.696	0.678	0.711
		F_1	0.784	0.772	0.770	0.779	0.732	0.787	0.782	0.749	0.691	0.791	0.779	**0.810**
		AUC	**0.859**	0.810	0.837	0.839	0.797	0.837	0.840	0.816	0.767	0.840	0.844	0.850
Mar.	Apr.	P_r	0.586	0.564	0.571	0.572	0.552	0.585	0.483	0.594	0.576	0.578	0.568	**0.613**
		R_e	0.990	**0.996**	0.995	0.994	0.995	0.959	0.958	0.954	0.995	0.995	0.994	**0.996**
		F_1	0.736	0.720	0.726	0.726	0.710	0.727	0.642	0.732	0.730	0.731	0.723	**0.759**
		AUC	0.957	0.960	0.957	0.954	0.954	0.943	0.923	0.941	0.958	0.958	0.956	**0.964**
Apr.	May.	P_r	0.839	0.941	0.954	0.905	0.768	0.831	0.928	0.891	0.971	0.872	0.973	**0.982**
		R_e	**0.945**	0.840	0.818	0.812	0.943	0.893	0.894	0.834	0.829	**0.945**	0.849	0.887
		F_1	0.889	0.888	0.881	0.856	0.847	0.861	0.910	0.862	0.894	0.907	0.907	**0.932**
		AUC	0.919	0.919	0.907	0.941	0.955	0.938	0.945	0.911	0.913	**0.965**	0.933	0.945
May.	Jun.	P_r	0.970	0.918	0.872	0.893	0.937	0.784	0.916	0.845	**0.971**	0.854	0.824	0.946
		R_e	0.858	0.947	0.945	**0.972**	0.856	0.941	0.909	0.969	0.910	0.930	0.925	**0.972**
		F_1	0.911	0.932	0.907	0.931	0.895	0.856	0.912	0.903	0.940	0.891	0.872	**0.959**
		AUC	0.928	0.956	0.935	0.946	0.925	0.958	0.950	0.975	0.953	0.956	0.953	**0.983**
Average value		P_r	0.808	0.834	0.838	0.831	0.810	0.790	0.814	0.805	0.865	0.814	0.833	**0.880**
		R_e	0.896	0.883	0.871	0.878	0.845	0.879	0.880	0.880	0.838	0.898	0.874	**0.908**
		F_1	0.834	0.841	0.834	0.836	0.806	0.821	0.829	0.829	0.827	0.841	0.837	**0.881**
		AUC	0.907	0.913	0.910	0.919	0.907	0.924	0.923	0.922	0.907	0.929	0.928	**0.944**
Average rank		P_r	7.400	7.400	6.200	6.400	7.000	8.600	7.800	8.600	3.600	6.800	6.200	**2.000**
		R_e	5.200	4.600	6.800	6.800	8.200	7.400	7.000	6.800	8.800	4.600	7.400	**2.400**
		F_1	5.800	6.600	7.600	6.800	11.000	8.200	5.600	6.600	6.600	5.400	6.200	**1.000**
		AUC	7.400	6.400	8.800	7.400	8.800	6.400	6.000	7.000	7.200	3.800	5.400	**1.600**
Training efficiency (ms)			96.86	107.62	102.95	99.67	72.69	113.09	80.09	90.47	95.42	130.26	114.62	100.97
Inference speed (ms)			0.027	0.027	0.027	0.027	0.027	0.027	0.027	0.018	0.027	0.027	0.036	0.027

The results are based on the mean across ten runs. In terms of P_r and R_e, the proposed model performs the best in three different test sets and also on *average value*, which means that it can well identify most fraudulent transactions with only a few false positives. As for F_1, the proposed model performs best in all tested sets, and for AUC, the proposed model performs the best in three different test sets and also on *average value*. Training efficiency and inference speed (the average time of training the model and detecting transactions) are shown at the bottom of Table IV. As shown in Table IV, although the time-aware gate, the current-historical attention module, and the interaction module are designed in our neural unit, the training time of the model is not significantly improved, which means that our model improves the detection performance, while the training efficiency is not reduced, i.e., our proposed architecture greatly improves the fraud detection performance, and the increase in the model complexity does not reduce the training efficiency.

We also compare the proposed model with its peers via a public dataset containing 284 807 transactions made by European cardholders in 2013. The aggregated-feature-based detection methods (A-LSTM, HOBA, and A-CNN) are not suitable for the dataset because they need specific feature names, but the dataset does not provide them. As shown in Table V, our proposed model still performs best in terms of P_r, R_e, F_1, and AUC.

Overall, in this work, we can conclude that:

1) Different time intervals lead to different changes in the transactional behaviors of users. Prior aggregated features are generated and added explicitly as additional features based on the time window or time intervals [12], [15], [16]. We think that the behavioral changes cannot be fully revealed by the basic feature engineering, i.e., the aggregated-feature-based detection model.
2) In the recent research, although the recurrent networks and attention mechanism are not very interpretable, they have achieved higher performance in speech recognition, language modeling, translation, and image subtitles [16], [21], [22], [23], and their application scope is also expanding. For credit card fraud detection, recurrent networks can well reveal the sequential transactional behaviors of users and achieve higher performance based on our previous work [20].

Thus, according to the analysis of experimental results: 1) although basic feature engineering is simple and efficient in fraud detection, it still not very sufficient in revealing

TABLE V

PERFORMANCE OF DIFFERENT MODELS ON A PUBLIC DATASET

Criteria	STA-NN	LSTM	GRU	Bi-RNN	UGRNN	HAINT-LSTM	T-LSTM	TAH-LSTM	TH-LSTM
P_r	0.502	0.501	0.501	0.500	0.501	0.501	0.500	0.501	**0.504**
R_e	0.664	0.560	0.608	0.771	0.675	0.682	0.830	0.888	**0.996**
F_1	0.571	0.529	0.550	0.607	0.575	0.578	0.624	0.641	**0.669**
AUC	0.499	0.499	0.500	0.502	0.501	0.500	0.499	0.501	**0.504**

TABLE VI

p-VALUE OF HOLM TEST (SIGNIFICANCE LEVEL $= 0.1$) BETWEEN TH-LSTM AND ITS PEERS

	STA-NN	A-LSTM	HOBA	A-CNN	LSTM	GRU	Bi-RNN	UGRNN	HAINT-LSTM	T-LSTM	TAH-LSTM
TH-LSTM (P_r)	0.14305	0.14305	0.21466	0.21466	0.17000	0.04180	0.09878	0.04180	0.48290	0.17648	0.21466
TH-LSTM (R_e)	0.70921	0.70921	0.33917	0.33917	0.09673	0.28486	0.33917	0.33917	0.04802	0.70921	0.28486
TH-LSTM (F_1)	0.10589	0.09841	0.02972	0.06807	0.00013	0.01592	0.10589	0.09841	0.09841	0.10589	0.09841
TH-LSTM (*AUC*)	0.08705	0.15825	0.01505	0.08781	0.01592	0.15825	0.15825	0.12069	0.08781	0.29258	0.19126

behavioral changes and fraudulent behaviors and 2) our proposed model is superior to basic feature engineering.

C. Statistical Test

To further compare the results between the proposed model with its peers, both the Friedman test [67] and the Holm test [68] are performed, where the proposed model is used as a control method. Usually, a small p-value indicates the fact that one algorithm is significantly better than another. The significance level is set as 0.1. The average rank is shown at the bottom of Table IV, and the p-value of the Holm test for comparisons is shown in Table VI. The performance of the proposed model is significantly better than its peers in terms of P_r, F_1, and AUC. Considering that it performs the best on *average value* and *average rank* in terms of all evaluation criteria, we can conclude that its performance is significantly better than its peers.

D. Ablation Study

To further prove the performance of the proposed model, an ablation study is performed to verify the effectiveness of each module of our model. Specifically, we test the model containing only the time-aware gate, the current-historical attention module, and the interaction module (abbreviated as M1, M2, and M3 for convenience), respectively. M0 represents the traditional LSTM. As shown in Table VII, M1 + M2 + M3 gets the best performance followed by M1 + M2, and both M1 and M2 greatly improve the detection performance of M0. These experiments mean that each module of the proposed model well extracts the transactional behaviors of users, and the integration of these modules obtains the best detection performance by integrating the learned transactional behavioral representations.

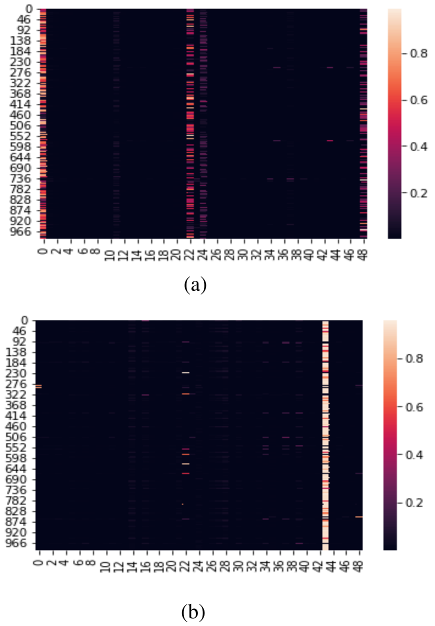
TABLE VII

ABLATION STUDY

Test	Criteria	M0	M1	M2	M1+M2	M1+M2+M3
Feb.	P_r	0.868	0.857	0.886	0.926	0.925
	R_e	0.825	0.984	0.973	0.955	0.977
	F_1	0.846	0.916	0.928	0.940	0.950
Mar.	P_r	0.919	0.941	0.929	0.928	0.925
	R_e	0.602	0.618	0.662	0.694	0.711
	F_1	0.727	0.746	0.773	0.794	0.804
Apr.	P_r	0.554	0.575	0.582	0.590	0.608
	R_e	0.994	0.996	0.995	0.995	0.996
	F_1	0.711	0.729	0.734	0.741	0.755
May.	P_r	0.796	0.823	0.853	0.886	0.879
	R_e	0.903	0.907	0.914	0.962	0.994
	F_1	0.846	0.863	0.882	0.922	0.933
Jun.	P_r	0.858	0.905	0.949	0.920	0.942
	R_e	0.935	0.924	0.917	0.978	0.980
	F_1	0.895	0.915	0.939	0.948	0.961

E. Interpretability of Attention Weights

As mentioned in Section III-B, legitimate transactional behaviors differ much from fraudulent ones, which is also the motivation for designing a current-historical attention module. This module extracts the relationships between the users' historical transactions and the current one, and adds this information to the transactional representations. In order to evaluate the effectiveness of our attention module and provide the interpretability of the periodicity of users' behaviors, a heatmap is employed to visualize the normalized attention weights for both legitimate and fraudulent transactions. We randomly select 2000 transactions with 1000 legitimate ones and 1000 fraudulent ones. Each row in Fig. 4 represents a transaction record, and the vertical axis denotes the number of attention steps. As shown in Fig. 4, there is a great difference between the transactional behaviors of fraudsters and legitimate users. The periodic behaviors are normalized to some time steps in the attentional feature space. In the representations of attentional feature space, although the attentional feature space is not very interpretable, we can observe which time steps are more important for the identification of current transactions, i.e., whether there are obvious periodic transactional behaviors. Specifically, if there are many time steps that have an influence on the current transaction, we believe that this user has periodic transactional behavior. It can be seen from Fig. 4(a) that the motivation of the current



(a)

(b)

Fig. 4. Visualization of normalized attention weights. (a) Legitimate transactions. (b) Fraudulent transactions.

transaction is well-captured by the current-historical attention module in steps 1, 11, 22, 24, and 48, i.e., some regular trading habits of legitimate users are extracted by the module. For fraudulent behaviors, the prominent periodicity is not found in Fig. 4(b), but the recent intensive fraudulent behaviors are extracted by the attention module at step 43, which is also consistent with our conclusion in the exploratory analysis that the recent activities of fraudsters are intensive and planned.

Overall, since the researchers give different opinions on the interpretability of an attention module [69], [70], [71], and our purpose is to learn effective transactional behavioral representations to better detect frauds, we mainly focus on the following aspects.

1) The heatmap difference between fraudsters and legitimate users is obvious, which shows that our current-historical attention module can extract the different behavioral patterns of both users.
2) After the current-historical attention module maps $\tilde{h}_t$ to a new feature space, since the features of this feature space are not very interpretable, we try to visualize the normalized attention weights to verify whether the current-historical attention module can extract the periodicity of legitimate users and intensive behaviors of fraudsters that are presented in exploratory analysis of transactional behaviors.
3) Although the transactional representations learned by the current-historical attention module are not very interpretable, our ablation study shows that the detection performance of our model is greatly improved by adding the current-historical attention module, which shows the effectiveness of the current-historical attention module.

F. Parameter Sensitivity

In order to evaluate the sensitivity of memory size on the proposed model, we set different memory sizes for comparison. As shown in Table VIII, when the memory size increases, the performance of the proposed model is improved, which means that it is effective to learn fraudulent behavioral patterns from historical transactional behaviors of users. The performance of the model does not continue to improve when the memory size is greater than 50, which demonstrates that it is feasible to set the memory size to $n/|U|$, where n is 5.12 million and $|U|$ is 107 192.

TABLE VIII
SENSITIVITY OF MEMORY SIZE

Memory	P_r	R_e	F_1	AUC
10	0.973	0.847	0.906	0.923
20	**0.982**	0.870	0.923	0.935
30	0.977	0.882	0.927	0.941
40	0.974	0.891	0.931	0.945
50	0.972	**0.901**	0.935	**0.950**
60	0.973	0.898	0.934	0.949
70	0.975	**0.901**	**0.936**	**0.950**
80	0.974	0.899	0.935	0.949

VII. Conclusion

In this article, we propose a new model to learn the transactional behavioral representations for each transaction record of users. A time-aware gate is augmented to LSTM to learn the behavioral changes brought by different time intervals. A current-historical attention mechanism is proposed and employed to build up the connections between current transactions and historical transactions of users to help the model capture the behavioral periodicity of users. In addition, an interaction module is designed to enable the model to learn more comprehensive and rational representations. The visualization of the learned transactional behavioral representations and the experiments on a real-world transaction dataset and a public one convincingly demonstrate the superiority of our model. Our future work intends to apply our model to other sequence classification tasks [72], [73] and handle how to learn behavioral representations of new users.

References

[1] G. Gianini, L. G. Fossi, C. Mio, O. Caelen, L. Brunie, and E. Damiani, "Managing a pool of rules for credit card fraud detection by a game theory based approach," *Future Gener. Comput. Syst.*, vol. 102, pp. 549–561, Jan. 2020.

[2] Y. Wu, Y. Xu, and J. Li, "Feature construction for fraudulent credit card cash-out detection," *Decis. Support Syst.*, vol. 127, Dec. 2019, Art. no. 113155.

[3] C. Whitrow, D. J. Hand, P. Juszczak, D. J. Weston, and N. M. Adams, "Transaction aggregation as a strategy for credit card fraud detection," *Data Mining Knowl. Discovery*, vol. 18, no. 1, pp. 30–55, 2009.

[4] F. Carcillo, Y.-A. L. Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Inf. Sci.*, vol. 557, pp. 317–331, May 2021.

[5] A. Langevin, T. Cody, S. Adams, and P. Beling, "Generative adversarial networks for data augmentation and transfer in credit card fraud detection," *J. Oper. Res. Soc.*, vol. 73, no. 1, pp. 153–180, 2022.

[6] A. A. Khine and H. W. Khin, "Credit card fraud detection using online boosting with extremely fast decision tree," in *Proc. IEEE Conf. Comput. Appl. (ICCA)*, Feb. 2020, pp. 1–4.

[7] X. Niu, L. Wang, and X. Yang, "A comparison study of credit card fraud detection: Supervised versus unsupervised," 2019, *arXiv:1904.10604*.

[8] D. Sánchez, M. A. Vila, L. Cerda, and J. M. Serrano, "Association rules applied to credit card fraud detection," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3630–3640, 2009.

[9] S. Han, K. Zhu, M. Zhou, and X. Cai, "Information-utilization-method-assisted multimodal multiobjective optimization and application to credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 856–869, Aug. 2021, doi: 10.1109/TCSS.2021.3061439.

[10] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, and Q. Kang, "Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection," *Neurocomputing*, vol. 407, pp. 50–62, Sep. 2020.

[11] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.

[12] 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," *Inf. Sci.*, vol. 557, pp. 302–316, May 2021.

[13] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Aug. 2018.

[14] F. Carcillo, A. D. Pozzolo, Y.-A. L. Borgne, O. Caelen, Y. Mazzer, and G. Bontempi, "SCARFF: A scalable framework for streaming credit card fraud detection with spark," *Inf. Fusion*, vol. 41, pp. 182–194, May 2018.

[15] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process.*, Krong Siem Reap, Cambodia, Dec. 2016, pp. 483–490.

[16] J. Jurgovsky *et al.*, "Sequence classification for credit-card fraud detection," *Expert Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.

[17] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio–temporal attention-based neural network for credit card fraud detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, New York, NY, USA, Feb. 2020, pp. 362–369.

[18] J. Guo, G. Liu, Y. Zuo, and J. Wu, "Learning sequential behavior representations for fraud detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Singapore, Nov. 2018, pp. 127–136.

[19] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*.

[20] R. Cao, G. Liu, Y. Xie, and C. Jiang, "Two-level attention model of representation learning for fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1291–1301, Dec. 2021.

[21] D. Li, J. Liu, Z. Yang, L. Sun, and Z. Wang, "Speech emotion recognition using recurrent neural networks with directional self-attention," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114683.

[22] W. Huang, P. Zhang, Y. Chen, M. Zhou, Y. Al-Turki, and A. Abusorrah, "QoS prediction model of cloud services based on deep learning," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 564–566, Mar. 2022.

[23] S. Yang *et al.*, "On the localness modeling for the self-attention based end-to-end speech synthesis," *Neural Netw.*, vol. 125, pp. 121–130, May 2020.

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

[25] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[26] J. Collins, J. Sohl-Dickstein, and D. Sussillo, "Capacity and trainability in recurrent neural networks," 2016, *arXiv:1611.09913*.

[27] Y. Zhu *et al.*, "What to do next: Modeling user behaviors by time-LSTM," in *Proc. IJCAI*, vol. 17, 2017, pp. 3602–3608.

[28] G. Liu, J. Guo, Y. Zuo, J. Wu, and R.-Y. Guo, "Fraud detection via behavioral sequence embedding," *Knowl. Inf. Syst.*, vol. 62, pp. 2685–2708, Jan. 2020.

[29] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware LSTM networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 65–74.

[30] L. Li, Z. Liu, C. Chen, Y.-L. Zhang, J. Zhou, and X. Li, "A time attention based fraud transaction detection framework," 2019, *arXiv:1912.11760*.

[31] A. G. C. D. Sá, A. C. M. Pereira, and G. L. Pappa, "A customized classification algorithm for credit card fraud detection," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 21–29, Jun. 2018.

[32] E. Kim *et al.*, "Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning," *Expert Syst. Appl.*, vol. 128, pp. 214–224, Aug. 2019.

[33] A. Salazar, G. Safont, A. Rodriguez, and L. Vergara, "Combination of multiple detectors for credit card fraud detection," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Limassol, Cyprus, Dec. 2016, pp. 138–143.

[34] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," 2010, *arXiv:1009.6119*.

[35] R. Bolton and D. Hand, "Statistical fraud detection: A review," *Stat. Sci.*, vol. 17, no. 3, pp. 235–249, 2002.

[36] Z. Li, G. Liu, and C. Jiang, "Deep representation learning with full center loss for credit card fraud detection," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 569–579, Apr. 2020.

[37] A. A. Khine and H. W. Khin, "Credit card fraud detection using online boosting with extremely fast decision tree," in *Proc. IEEE Conf. Comput. Appl. (ICCA)*, Feb. 2020, pp. 1–4.

[38] F. Ugo, D. S. Alfredo, P. Francesca, Z. Paolo, and P. Francesco, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.

[39] A. C. Bahnsen, A. Stojanovic, D. Aouada, and B. Ottersten, "Cost sensitive credit card fraud detection using Bayes minimum risk," in *Proc. 12th Int. Conf. Mach. Learn. Appl.*, Miami, FL, USA, Dec. 2013, pp. 333–338.

[40] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decis. Support Syst.*, vol. 50, no. 3, pp. 602–613, 2011.

[41] A. D. Pozzolo, O. Caelen, Y.-A. L. Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, 2014.

[42] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5916–5923, 2013.

[43] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–6.

[44] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Charlottesville, VA, USA, Apr. 2018, pp. 129–134.

[45] Y. Lucas *et al.*, "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs," *Future Gener. Comput. Syst.*, vol. 102, pp. 393–402, Jan. 2020.

[46] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decis. Support Syst.*, vol. 50, no. 3, pp. 559–569, 2011.

[47] A. Salazar, G. Safont, and L. Vergara, "Surrogate techniques for testing fraud detection algorithms in credit card operations," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Rome, Italy, Oct. 2014, pp. 1–6, doi: 10.1109/CCST.2014.6986987.

[48] E. M. Carneiro, L. A. V. Dias, A. M. Da Cunha, and L. F. S. Mialaret, "Cluster analysis and artificial neural networks: A case study in credit card fraud detection," in *Proc. 12th Int. Conf. Inf. Technol.-New Generations*, Las Vegas, NV, USA, Apr. 2015, pp. 122–126, doi: 10.1109/ITNG.2015.25.

[49] L. I. Kuncheva and W. J. Faithfull, "PCA feature extraction for change detection in multidimensional unlabeled data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 69–80, Jan. 2014.

[50] J. Abreu, L. Fred, D. Macêdo, and C. Zanchettin, "Hierarchical attentional hybrid neural networks for document classification," 2019, *arXiv:1901.06610*.

[51] K. Song, X. Tan, F. Peng, and J. Lu, "Hybrid self-attention network for machine translation," 2018, *arXiv:1811.00253*.

[52] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li, "Hierarchical recurrent neural network for document modeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, Sep. 2015, pp. 899–907.

[53] H. Lu, Z. Ge, Y. Song, D. Jiang, T. Zhou, and J. Qin, "A temporal-aware LSTM enhanced by loss-switch mechanism for traffic flow forecasting," *Neurocomputing*, vol. 427, pp. 169–178, Feb. 2021.

[54] F. M. Shakiba and M. Zhou, "Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 10856–10867, Nov. 2021, doi: 10.1109/TIE.2020.3034856.

[55] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12275–12284.

[56] P. Liu, Y. Zhou, D. Peng, and D. Wu, "Global-attention-based neural networks for vision language intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 7, pp. 1243–1252, Jul. 2021.

[57] P. Liu, Y. Zhou, D. Peng, and D. Wu, "Global-attention-based neural networks for vision language intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 7, pp. 1243–1252, Jul. 2021.

[58] Z. Huang, X. Xu, H. Zhu, and M. Zhou, "An efficient group recommendation model with multiattention-based neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4461–4474, Nov. 2020.

[59] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[60] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020.

[61] P.-T. D. Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, Feb. 2005.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[63] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[64] G. Zhang *et al.*, "FRAUDRE: Fraud detection dual-resistant to graph inconsistency and imbalance," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Auckland, New Zealand, Dec. 2021, pp. 867–876, doi: 10.1109/ICDM51629.2021.00098.

[65] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[66] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[67] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.

[68] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

[69] S. Serrano and N. A. Smith, "Is attention interpretable?" 2019, *arXiv:1906.03731*.

[70] S. Jain and B. C. Wallace, "Attention is not explanation," 2019, *arXiv:1902.10186*.

[71] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," 2019, *arXiv:1908.04626*.

[72] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep learning for EMG-based human-machine interaction: A review," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 3, pp. 512–533, Mar. 2021.

[73] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, "Multi-user activity recognition: Challenges and opportunities," *Inf. Fusion*, vol. 63, pp. 121–135, 2020.

Yu Xie received the B.E. degree from the School of Computer, Qingdao University, Qingdao, Shandong, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China.

His current research interests include credit card fraud detection, machine learning, deep learning, big data, ensemble learning, and natural language processing.

Guanjun Liu (Senior Member, IEEE) received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011.

He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013, and a Post-Doctoral Research Fellow with the Humboldt University of Berlin, Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently a Professor with the Department of Computer Science, Tongji University.

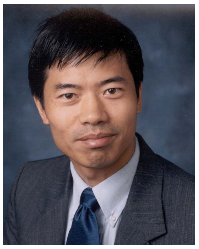
Chungang Yan received the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China, in 2006.

She is currently a Professor with the Department of Computer Science and Technology, Tongji University. Her current research interests include Petri net, formal method, credibility computing, intelligence computing, and service-oriented computing.

Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995.

He is currently the Leader of the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai, China. He is an Honorary Professor with Brunel University London, Uxbridge, U.K.

Dr. Jiang is an IET Fellow. He was a recipient of one international prize and seven prizes in the field of science and technology.



MengChu Zhou (Fellow of IEEE) received the B.S. degree from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor. He is also affiliated with Zhejiang Gongshang University, Hangzhou, China. He has over 1000 publications, including 14 books, more than 700 journal articles including over 600 IEEE TRANSACTIONS articles, 31 patents, and 30 book chapters. His research interests include automation, Petri nets, artificial intelligence, the Internet of Things, edge/cloud computing, and big data analytics.

Dr. Zhou is a Life Member of the Chinese Association for Science and Technology, USA, and served as its President in 1999. He is also a fellow of the International Federation of Automatic Control (IFAC), the American Association for the Advancement of Science (AAAS), the Chinese Association of Automation (CAA), and the National Academy of Inventors (NAI).

Maozhen Li received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1997.

He is currently a Professor with the Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, U.K. He is also a Visiting Professor with Tongji University, Shanghai, China. His main research interests include high-performance computing, big data analytics, and intelligent systems with applications to smart grids, smart manufacturing, and smart cities. He has over 180 research publications in these areas, including four books.

Dr. Li is a fellow of the British Computer Society (BCS) and the Institute of Engineering and Technology (IET). He has served at over 30 IEEE conferences and is on the editorial board of a number of journals.