Graph Neural Network for Fraud Detection via Spatial-temporal Attention

Dawei Cheng, Xiaoyang Wang, Ying Zhang and Liqing Zhang

Abstract—Card fraud is an important issue and incurs a considerable cost for both cardholders and issuing banks. Contemporary methods apply machine learning-based approaches to detect fraudulent behavior from transaction records. But manually generating features needs domain knowledge and may lay behind the modus operandi of fraud, which means we need to automatically focus on the most relevant fraudulent behavior patterns in the online detection system. Therefore, in this work, we propose a spatial-temporal attention-based graph network (STAGN) for credit card fraud detection. In particular, we learn the temporal and location-based transaction graph features by a graph neural network firstly. Afterwards, we employ the spatial-temporal attention on top of learned tensor representations, which are then fed into a 3D convolution network. The attentional weights are jointly learned in an end-to-end manner with 3D convolution and detection networks. After that, we conduct extensive experiments on the real-word card transaction dataset. The result shows that STAGN performs better than other state-of-the-art baselines in both AUC and precision-recall curves. Moreover, we conduct empirical studies with domain experts on the proposed method for fraud detection and knowledge discovery; the result demonstrates its superiority in detecting suspicious transactions, mining spatial and temporal fraud hotspots, and uncover fraud patterns. The effectiveness of the proposed method in other user behavior-based tasks is also demonstrated. Finally, in order to tackle the challenges of big data, we integrate our proposed STAGN into the fraud detection system as the predictive model and present the implementation detail of each module in the system.

Index Terms—Fraud Detection, Spatial-temporal Attention, Graph Neural Network.

1 Introduction

Card fraud is a general term for the unauthorized use of funds in a transaction typically through a credit or a debit card [1]. Global card fraud losses amounted to over 25 billion US dollars in 2018 and are forecast to continue to increase [2]. This huge amount of losses has increased the importance of fraud-fighting. Figure 1 shows a typical fraud detection framework deployed in a commercial system. The card alliance or banks, such as VISA, MasterCard, or Citibank, assess each transaction with an online predictive model once it has passed card checking. Unlike a simple rule checking system, which focuses on card blacklists, budget checking, fraud rules, etc., the predictive model is designed to detect fraud patterns automatically and produces a fraud risk score. Investigators can thereby focus on the high-risk transactions effectively and feedback the analysis results to the predictive model for model updating.

As attacking strategies from potential fraudsters change, it is essential that a well-behaved system can adapt to the evolving strategy [3], [4]. We summarize the following two major observations from real-world fraud transactions: 1). *Temporal aggregation.* Fraudsters are subject to the limited time of the activities. As the cardholder will freeze the card as soon as possible once suspicious transactions have been detected, fraudsters are required to reach the credit limit in a short time. That means the behaviors of the fraud transaction would be exposed in a limited time. 2). *Spatial aggregation.* Fraudsters are subjected to cost on the devices and merchants of transactions. That is, due to the economic constraints, fraudsters exploit few vulnerable merchants/devices more frequently, which leads to fraudulent transactions being hit from few particular locations compared to the many diverse locations issued by a regular card owner. It should be noted that in this circumstance, spatial aggregation involves both transaction locations and merchants, which requires the model to addresses this limitation by preserving the features from location-based merchants.

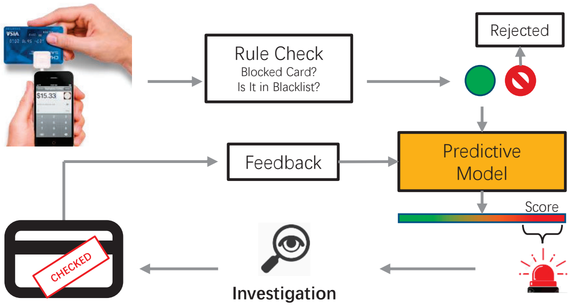


Fig. 1. The framework of credit card fraud detection.

Many existing models to deal with fraud transactions have been extensively studied [5], [6], [7]. They mainly include two folds: 1). *Rule-based methods* directly generate sophisticated rules by domain experts for identification; for example, the association rules method is proposed in [8] for mining frequent fraud rules. 2). *Machine learning-based methods* learn models by exploring large amounts of historical data. For example, the approach in [9] extracted features based on artificial neural network and built supervised classifiers to detect fraudulent transactions. The

- *Dawei Cheng and Liqing Zhang are with the MoE Key Lab of Artificial Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. E-mail: dawei.cheng@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn*
- *Xiaoyang Wang was with Zhejiang Gongshang University, Hangzhou, China. E-mail: xiaoyangw@zjgsu.edu.cn*
- *Ying Zhang was with University of Technology Sydney, Sydney, Australia. E-mail: ying.zhang@uts.edu.au*

Manuscript received April 19, 2015; revised August 26, 2015.

method in [10] advanced the usage of automatic feature engineering in a convolutional neural network (CNN). AdaBoost and majority voting are applied in [3] for fraud records detection. The work in [11] researched on this task by a sequence LSTM model. However, all these methods require manually constructing features before feeding into a classification model, which fails to automatically learn the joint impact on spatial and temporal patterns, as the spatial-temporal patterns have been observed as the main weaknesses of fraudsters, also reported in [12].

Recently developed graph neural network and attention mechanisms have shown the benefit of automatic feature learning [13], [14]. The superior performance of 3-dimensional (3D) convolution on spatial-temporal feature learning is also demonstrated in a wide range of prediction tasks [15]. In the credit card fraud detection task, it is important to jointly consider the "temporal aggregation" and "spatial aggregation" together and then drive them into a representative and deep classifier which well-suited for spatial-temporal feature learning. Attention networks using spatial-temporal information have been used in computer vision [16], [17]. But their techniques cannot be trivially extended to our problem. For instance, in the video-based tasks (e.g., classification and event detection), pixel-level spatial feature, and sequence-level temporal feature naturally fit the attention networks with spatial-temporal information. However, there is no straightforward spatial feature in the anti-fraud task. The users fraud pattern is embedded in location-based transaction graphs. Moreover, the data sparsity and frequent changes of fraud patterns make the use of spatial-temporal information on the attention network very challenging.

Therefore, in this paper, we present the STAGN model for credit card fraud detection, a novel deep learning-based method that jointly considers "temporal aggregation" and "spatial aggregation" in an attention network. Our proposed approach construct raw records into spatial-temporal based feature tensors, firstly, in which transaction patterns are learned from location-based graphs. We leverage a graph neural network to learn representations from the global transaction graphs. Then, we use an attention mechanism to infer the importance of different types of feature slices adaptively. To uncover the hidden fraud patterns, we introduce a 3D convolution layer to capture interdependent relationships between spatial-temporal patterns. In experiments, we show that the results of the proposed method significantly outperform the results from other state-of-the-art baselines.

In brief, the main contributions of this paper include:

- We present a novel attention-based 3D convolution neural network for card fraud detection by jointly capturing two weaknesses displayed by fraudsters, summarized as "temporal aggregation" and "spatial aggregation". To the best of our knowledge, this is the first time that a fraud detection problem has been addressed by spatial-temporal attention graph neural network approach with a 3D convolutional mechanism.
- We propose a graph neural network to learn representations from location-based transaction graphs, which are aggregated by temporal slices in a unified spatial-temporal attention framework. The results prove that graph representation considerably improves the accuracy of fraud detection.
- Our approach is extensively evaluated in a real-world credit card fraud post analysis system, hosted by a major commercial bank. The experimental results demonstrate the superiority of our proposed methods, which could detect more fraud transactions with relatively high precision compared with state-of-the-art baselines.
- We deploy our proposed STAGN into the bank's fraud detection system as predictive model. We carefully implement each module of the system to support the big data scenario. Experiment results show that our system can predict around 10,000 transactions per second, which already meets the efficiency requirement according to the industry standard in banks.

The rest of the paper is organized as follows: In Section 2, we review the related work in fraud detection. In Section 3, we formally define the problem in our task and introduce the framework of STAGN. We report the experimental results and case studies in Section 4 and present the model generalization in Section 5. We report system implementation and conclusion in Section 6 and 7.

2 Related work

We summarize the related work in three main areas: 1) location-based graph representative learning, 2) attentional convolutional neural networks, and 3) credit card fraud detection.

Location-based Graph Representative Learning

Graph representative learning [14], [18], also known as network embedding, aims to infer a low dimensional distributed vectors to represent nodes in a network, which has been widely applied in various tasks, such as node classification [19], node clustering [20], node visualization [21] and link prediction [22]. Recent studies of learning location-based graph representation are mainly in two directions: embedding with side information and advanced information preserved embedding [23]. Side information in the context of network embedding includes node contents and types of nodes and edges [24]. For example, TADW is proposed in [25] that takes the rich information (e.g., text. location codes) associated with nodes into account when they learn the low dimensional representations of nodes. Advanced information preserved embedding takes additional advanced information into account so as to solve some specific analytic tasks. Different from side information, the advanced information refers to the supervised or pseudo supervised information in a specific task. For example, the method in [26] learn graph-based point-of-interest (POI) embedding for a location-based recommendation system. Although initial efforts have been made using advanced network embedding methods that could learn representations of location-based graphs, there is little work focus on detecting frauds by preserving location-based graph representations.

Attentional Convolution Neural Network

Many recent works have shown the benefit of combining an attention mechanism in convolutional neural networks for a wide range of prediction tasks [13], [15], such as depth estimation [27], default prediction [28] or language understanding [29]. For instance, pervasive attention are employed on 2D convolutional neural networks for sequence-to-sequence prediction [30]. Attention-gated networks have been considered for integrating multi-scale information in [31]. In [32], an attention model is employed for combining multi-scale features in the context of semantic segmentation and object contour detection. Our approach develops from a similar intuition but further integrates an attention model in both spatial-temporal aspects before fading into a 3D convolutional neural network, which significantly improves the accuracy of the detection.

Credit Card Fraud Detection

Several machine learning techniques have been used in the literature to approach the credit card fraud detection problem. The author in [33] tried Bayesian belief networks (BBN) and artificial neural networks (ANN) on a real dataset obtained from Europay International. In [34] neural network-based model and decision tree classifier are compared, and the authors found that neural networks outperform decision trees. The authors in [10] prove that using a convolution model to extract spatial patterns can achieve higher accuracy compared with multi-layer perception network, SVMs, and decision trees. In [3], AdaBoost and majority voting are introduced to improve detection accuracy on fraud records. Afterwards, improved LSTM model is proposed in [11] from sequence classify perspective. These methods, however, feed manually generated features into a classification model directly, which ignores the joint feature learning on spatial and temporal patterns. As a result, they may not be appropriate for real-world large scale fraud detection systems with complex and unpredictable fraud patterns.

3 The Proposed Approaches

In this section, we first briefly present the preliminaries, including data analysis, to support our intuitions and the problem definition of our work. Then we introduce an overview of the spatial-temporal attention-based neural network framework. After that, we present the process of feature learning on location-based transaction graph, the spatial-temporal attention layer, the 3D convolution network (3D ConNet), and the detection layer, respectively. Lastly, we introduce the optimization strategy of the proposed methods.

3.1 Preliminaries

Definition 1. ***Transaction Record.*** A transaction record r can be defined as a tuple of attributes in a transaction payment process $r = \{u, t, l, m, a\}$, where u denotes the user, t and l are the time stamp and location of the transaction. m and a mean the merchant (receiver) and amount of this payment, respectively.

A fraud event d in this paper refers to a transaction which is not authorized by its cardholder. Note that a fraud event is a special type of transaction, which also preserves $\{u, t, l, m, a\}$ attributes.

Definition 2. ***Temporal Slices*** represent the feature generated in different time window. The value and diversity in temporal slices reflect its activeness and hence are related to the consuming behavior of the user.

Definition 3. ***Spatial Slices*** contain the location-based representation of each transaction record by aggregating features in different location spaces.

Figure 2 visualizes the statistical information of transaction records in different time windows and locations, where the left part shows fraud transactions, and the right part shows legitimate ones. The x-axis denotes feature ID, which contains statistics of the amount and frequency of given transactions. It is clear that in temporal analysis, fraud features (shown in Figure 2a) change abruptly across different slices, while legitimate records are much more stable (shown in Figure 2b). It confirms our assumption of "temporal aggregation". In the spatial analysis, we aggregate the features according to different location codes within a fixed time window (we set it to days here). Figures 2c to Figure 2d show the heat map of features in spatial slices. As we can see, fraud transactions are obviously located in only a small number of zones, which means fraudsters would use the card frequently under the constraint of locations or devices, while for the normal transactions, there are no noteworthy patterns for user consuming behavior in given time windows. The complete real-world fraud event data provided by our collaborating institution offers us the unique opportunity to tackle the problem of fraud detection. In conclusion, we now formalize our credit card fraud detection problem as follows:

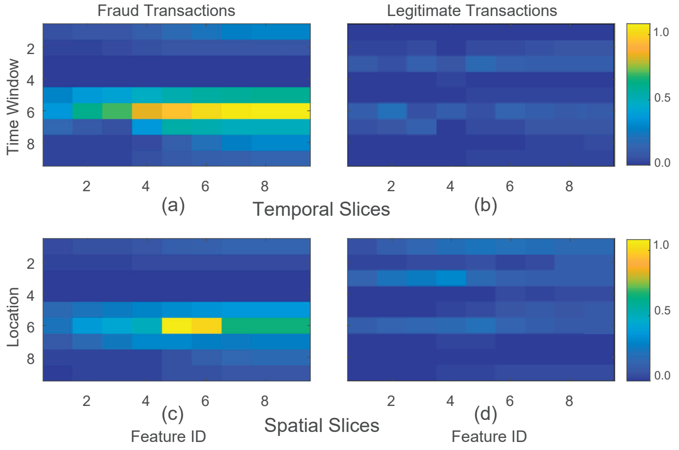


Fig. 2. Heat maps of spatial-temporal feature slices from both fraudulent and legitimate transactions.

Problem statement. *Given a set of transaction records* $\mathcal{R} = \{\mathcal{U}, \mathcal{T}, \mathcal{L}, \mathcal{M}, \mathcal{A}\}$*, a set of fraud events* $\mathcal{D}$*, which are a subset of the transaction collection* $\{\mathcal{D} \subset \mathcal{R}\}$*, and time window* t_i *&* t_{i+1}*, for each record within* $(t_i, t_{i+1}]$ *we want to infer the possibility of whether it is a fraud event, based on records from* t_1 *to* t_i*. The objective is to achieve high accuracy of fraud prediction, as well as to explore the fraud patterns of credit card transactions.*

3.2 Model Architecture and Feature Construction

Figure 3 shows the general network architecture of STAGN. The model takes transaction records as input and transfers them into high-order tensor representation in spatial, temporal, and feature orders. Then, we apply spatial-temporal attention and the 3D convolution layer to obtain a feature vector. Specifically, spatial-temporal attention helps to obtain information from tensor features by different weights, and the 3D convolution layer helps to model hidden patterns of transactions. Finally, we reshape the learned feature representation from tenors to vectors for the fraud detection task by a detection network. We will first introduce the feature construction process, then present each submodule of the proposed methods in the following subsections.

In feature engineering, we construct the representation of each transaction record into tensor format, denoted as $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, where N_1, N_2, and N_3 mean the dimensions of temporal, spatial and feature slices. That is, we construct a feature vector ($v_f \in \mathbb{R}^{N_3}$) for each spatial-temporal (time horizon, location) pair, which is extracted from transactions that fall into that pair space. The number of time horizon and location are N_1 and N_2, respectively. Thus, there are totally $N_1 \times N_2$ spatial-temporal pairs, where each pair contains a feature vector with N_3 dimensions. In our implementation, the feature vector is

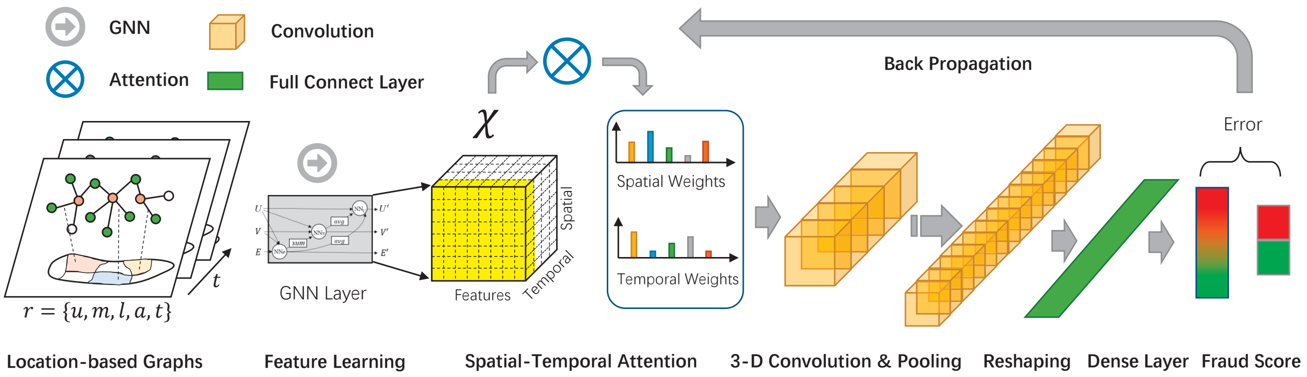


Fig. 3. The illustration of the proposed spatial-temporal attention-based neural network (STAGN) model. Raw transaction records are processed by location-based GNN layer, spatial-temporal attention, and multiple 3D ConvNet to learn high-level representations. Afterward, the learned representations are reshaped to vectors and fed into a detection network for fraud estimation. Attentional weights are jointly optimized in an end-to-end mechanism with 3D convolution and detection networks.

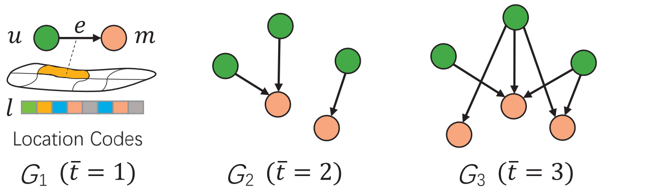
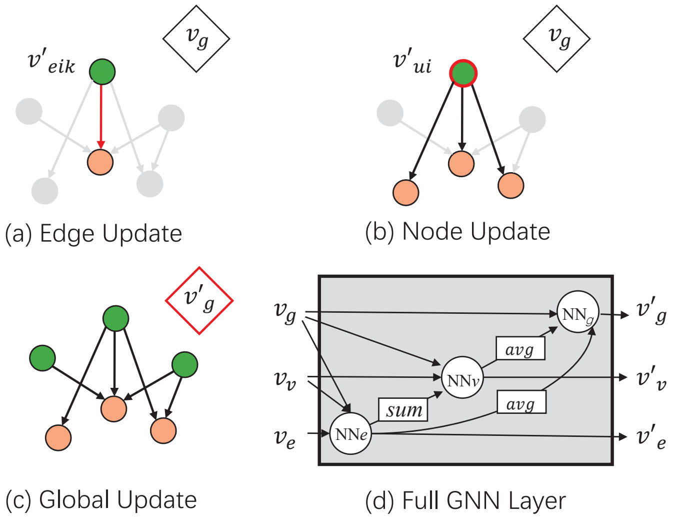


Fig. 4. A typical sequence of transaction graphs: G_1 denotes the graph of two transactions, where u denotes the user and m is the merchant. e denotes the edge between a merchant and a user. G_2 shows two users sharing the same merchant. G_3 gives a global view of a typical transaction graph at $\bar{t} = 3$, which involves multiple users and merchants.



(a) Edge Update

(b) Node Update

(c) Global Update

(d) Full GNN Layer

Fig. 5. Illustration of the updates in a GNN layer. In (a-c) the red stroke indicates the element that is being updated, and the other colors indicate other elements which are involved in the update. (d) shows a full GNN layer updates, edge, and global output features based on the incoming vectors.

concatenated by: 1) transaction feature. Inspired by Fu's work [10], the transaction feature includes current amount, average amount, median amount, total amount, transaction times, the entropy of transaction times and amounts, etc. 2) Location-based graph feature, which will be described in Section 3.3. Finally, given feature tensor $\mathcal{X}$, we could extract the temporal slices $\mathcal{X}(\bar{t}, :, :), \bar{t} \in \{1, 2, \cdots, N_1\}$ denotes the time window and spatial slices $\mathcal{X}(:, \bar{s}, :), \bar{s} \in \{1, 2, \cdots, N_2\}$ denotes the location codes. Please note that the representation $\mathcal{X}$ is constructed for each transaction while it is extracted from the corresponding set of the global records.

3.3 Location-based Graph Neural Network

In order to preserve global information of card fraud events, we represent user consumptions of its card in different merchants as transaction graph with edges directed from user to merchant. For each time window $\bar{t}$, we construct the graph at its beginning time, denoted as $G_{\bar{t}}$. For the brevity of notations, we shorten it as G in a given time window. Then, the transaction graph is represented as $G(V = (U, M), E)$, where $U(G)$ the set of user nodes, $M(G)$ denotes the set of merchant nodes, $U(G) \cap M(G) = \varnothing, V(G) = U(G) \cup M(G)$, and $E(G) \subseteq U(G) \times M(G)$ denotes the edge sets. There are two types of nodes, where $u \in U(G)$ denotes user nodes and $m \in M(G)$ is merchant nodes. We employ v_u and v_m to represent the feature vector of user and merchant, respectively. If there is a transaction between user and merchant, we create an edge e between them. We introduce v_e to denote the feature vector of edge, where edge vector $v_e = [a, v_l]$ contains continuous transaction amount a and one-hot representation of transaction location l. The feature vector of a transaction graph G is denoted as v_g, which includes the number of users, merchants, the number and amount of total transactions, etc.

Figure 4 illustrates the structure of a transaction graph, where the user node is colored in green and the merchant node in orange. In practice, there may be multiple users sharing the same merchants (shown as G_2 in Figure 4) or one user purchases with a number of merchants (shown as G_3). As introduced in section 1, fraudsters are facing the limitation of "spatial aggregation", which means the fraud transactions usually share the same location or the same merchant in a short time. Therefore, the merchant behavior is embedded in location-based transaction graphs, and we then leverage graph neural network to learn its latent features.

A graph neural network (GNN) layer takes the transaction graph as inputs. Particularly, each transaction $r = \{u, t, l, m, a\}$ is constructed as an edge in graph G. Then, we initialize the node feature v_u and v_m with random value, the edge feature v_e as the concatenation of transaction amount and one-hot representation of location codes, and global graph feature v_g as statistical information of G. We introduce v_v as an uniformed notation of v_u and v_m. Afterwards, we update the representation of nodes and edges in order to preserve the global graph structure. Mathematically, we formulate the update function of edge feature as follows:

$$v'_{eik} = \mathrm{NN}_e(v_{eik}, I_{ui}, I_{mk}, v_g) \tag{1}$$

where i and k are the index of user and merchant node that are connected by the edge e. I_{ui} and I_{mk} denote the corresponding index of those nodes in the graph. NN_e is a shallow neural network contains two full connected layers with ReLU activation functions. The neural network NN_e maps across all edges to compute per-edge updates.

Afterwards, we utilize an aggregation function to reduce the edge updates to a single element which represents the aggregated information. In particular, we update the feature of each node by employing element-wise summation on edges that connected to that node.

$$\begin{aligned} v'_{ui} &= \mathrm{NN}_v\left(\sum\nolimits_{k=1}^{N_m}(v'_{eik}), v_{ui}, v_g\right) \\ v'_{mk} &= \mathrm{NN}_v\left(\sum\nolimits_{i=1}^{N_u}(v'_{eik}), v_{mk}, v_g\right) \end{aligned} \tag{2}$$

where N_u and N_m are the number of user and merchant nodes respectively. NN_v denotes another neural network to compute the attributes of nodes, which is also a multi-layer perception (MLP) that contains two full connected layers. Lastly, we update the embeddings of global graph as below:

$$v'_g = \mathrm{NN}_g(\overline{v}_u, \overline{v}_m, \overline{v}_e, v_g) \tag{3}$$

where

$$\overline{v}_u = \frac{1}{N_u}\sum\nolimits_{i}^{N_u} v'_{ui},\ \overline{v}_m = \frac{1}{N_m}\sum\nolimits_{k}^{N_m} v'_{mk} \tag{4}$$

and $\overline{v}_e = \frac{1}{N_e}\sum v'_e$ is the average of updated edge features. N_e denotes the number of edges and NN_g is the global update network implemented by another MLP.

Given the transaction graph, our proposed GNN layer proceeds from the edge to the node and to the global level. Figures 5a, b and c illustrate detail depiction of which graph elements are involved in each of these computations, and Figure 5d shows the full process of GNN layer with its update and aggregation functions. Algorithm 1 reports the detailed steps of computation. After obtaining the output of GNN layer, we concatenate the updated v'_u, v'_m, v'_e and v'_g as the feature of location-based transaction graph. It is then combined with transaction feature into the feature vector v_f, which is presented in Section 3.2. Please note that each graph feature is located in its corresponding temporal spatial pairs with time window $\overline{t}$ and location codes $\overline{s}$.

Similar to CNNs, the proposed location-based transaction graph learning is naturally polarizable, since the neural network layer NN_e and NN_v are shared over the edges and nodes, which can be computed in parallel. It means that in practice, GNN calculates like the batch dimension is typical mini-batch training regimes. In particular, we construct graphs according to different time windows that can be treated as disjoint components of a large graph to allow computing on these independent graphs together. Therefore, the proposed GNN layer can be accelerated by recently advanced batch parallelization on GPU devices.

Algorithm 1: **Steps of computation in a GNN layer.**

```
Input  : $G(V = (U, M), E)$: given the transaction graph,
         $v_g$: the feature of a given transaction graph,
         $v_e$: the feature of edges.
Output : $v'_u, v'_m, v'_e$ and $v'_g$.
$N_u = |U(G)|, N_m = |M(G)|, N_e = |E(G)|$ ;
Initialize $v_u, v_m, v_e$ and $v_g$;
for all edges do
    $v'_{eik} \leftarrow \mathrm{NN}_e(v_{eik}, I_{ui}, I_{mk}, v_g)$ ;
end for
for $i$ in 1 to $N_u$ do
    $v'_{ui} \leftarrow \mathrm{NN}_v\left(\sum_{k=1}^{N_m}(v'_{eik}), v_{ui}, v_g\right)$;
    $\overline{v}_u \leftarrow \frac{1}{N_u}\sum_i^{N_u} v'_{ui}$;
end for
for $k$ in 1 to $N_m$ do
    $v'_{mk} \leftarrow \mathrm{NN}_v\left(\sum_{i=1}^{N_u}(v'_{eik}), v_{mk}, v_g\right)$;
    $\overline{v}_m \leftarrow \frac{1}{N_m}\sum_k^{N_m} v'_{mk}$;
end for
$\overline{v}_e \leftarrow \frac{1}{N_e}\sum v'_e$ ;
$v'_g \leftarrow \mathrm{NN}_g(\overline{v}_u, \overline{v}_m, \overline{v}_e, v_g)$;
return $v'_u, v'_m, v'_e$ and $v'_g$
```

3.4 Spatial-temporal Attention Net

The attention network aims to perform proper credit assignment to the spatial and temporal slices according to their importance in the current transaction. It contains two self-attention layers targeting temporal and spatial slices, respectively.

3.4.1 Temporal Attention Layer

Formally, given the extracted feature tensor $\mathcal{X}$ as described above, the temporal attention layer represents the transaction by a weighted sum of the matrix representation of all the temporal slices. Mathematically, it takes the form as follows:

$$rept = \sum\nolimits_{\overline{t}=1}^{N_1} a_{1,\overline{t}}\mathcal{X}(\overline{t}, :, :) \tag{5}$$

$$a_{1,\overline{t}} = \frac{\exp\left((1-\lambda_1)\cdot \mathrm{NN}_{\overline{t}}(W_{\overline{t}}, \mathcal{X}(\overline{t}, :, :))\right)}{\sum_{\overline{t}=1}^{N_1}\exp\left((1-\lambda_1)\cdot \mathrm{NN}_{\overline{t}}(W_{\overline{t}}, \mathcal{X}(\overline{t}, :, :))\right)} \tag{6}$$

where: $a_{1,\overline{t}}$ is the weight for each temporal slice, and $\mathrm{NN}_{\overline{t}}(\cdot)$ is a fully connected layer with ReLU activation and parameters $W_{\overline{t}}$; $\lambda_1 \in [0, 1]$ is the temporal penalty factor to control the importance of temporal attention; $rept$ is the output of the temporal attention layer. It should be noted that we unfold matrices $\mathcal{X}(\overline{t}, :, :)$ to row vectors for computational convenience and reshape the output re into tensor format $rept \in \mathbb{R}^{N_1\times N_2\times N_3}$.

3.4.2 Spatial Attention Layer

Given the output from the temporal net $rept$, we then apply a spatial attention mechanism on the top of the temporal net. It is formulated as follows:

$$\mathcal{H}^a = \sum\nolimits_{\overline{s}=1}^{N_2} a_{2,\overline{s}} rept(:, \overline{s}, :) \tag{7}$$

$$a_{2,\overline{s}} = \frac{\exp\left((1-\lambda_2)\cdot \mathrm{NN}_{\overline{s}}(W_{\overline{s}}, rept(:, \overline{s}, :))\right)}{\sum_{\overline{s}=1}^{N_2}\exp\left((1-\lambda_2)\cdot \mathrm{NN}_{\overline{s}}(W_{\overline{s}}, rept(:, \overline{s}, :))\right)} \tag{8}$$

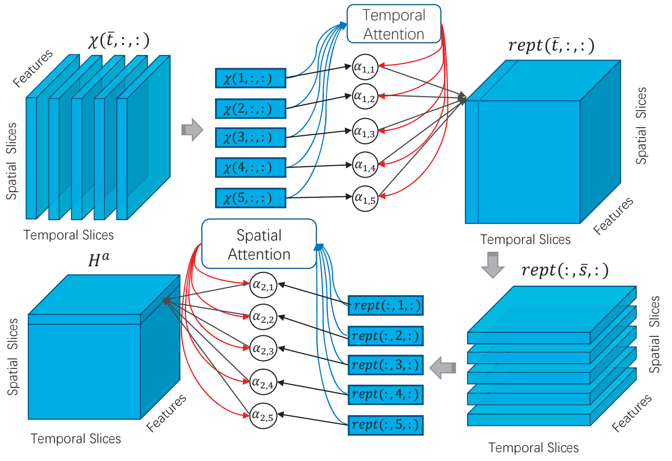


Fig. 6. Illustration of spatial-temporal attention neural networks.

where $W_{\bar{s}}$ is the weight of spatial network $\mathrm{NN}_{\bar{s}}$; $\mathcal{H}^a$ is the output of attention layers, we reshape it into tensor format with the same order as $\mathcal{X}$; $a_{2,\bar{s}}$ is the weight for each spatial slices; $\lambda_2 \in [0,1]$ is the spatial penalty factor to control the importance of spatial attention.

3.5 3D Convolutional Layers

For our mission, CNN is an attractive option for three main reasons. First, they can clearly exploit the spatial features of our problem. In particular, they can learn local spatial filters that are useful for classification tasks. In our case, we expect the filter at the input level to encode spatial structure. Second, by stacking multiple layers, the network can learn more complex features from spatial input spaces. Finally, the optimization of CNN could be learned by SGD based methods, which can be performed efficiently with commercial graphics hardware.

Compared to 2D convolution networks, 3D ConvNet is ideal for spatial-temporal learning of features. Due to 3D convolution and 3D pool operations, 3D ConvNet works temporally and spatially, whereas, in 2D ConvNet, it is only spatially executed. In general, the following equation represents a 3D convolution operation:

$$\mathcal{H}_j^c = \sum\nolimits_j \mathcal{H}^{c-1}(c_t - c_m, c_s - c_n, c_f - c_o)\mathcal{W}_i^c(c_m, c_n, c_o) \quad (9)$$

in which $\mathcal{W}_i^c$ is the 3D kernel in the c^{th} layer and i^{th} kernel which convolves over the feature $\mathcal{H}^{c-1}$. The first layer of $\mathcal{H}^{c-1}$ is the output of attention layer $\mathcal{H}^a$. c_t, c_s and c_f are the dimension of input tensor in temporal, spatial and feature slices, which are equal to N_1, N_2 and N_3 of the first convolution layer. $\mathcal{W}_i^c$ is the element-wise weight in the 3D convolution kernel (c_m, c_n, c_o). Thus, the output feature $\mathcal{H}^c$ is calculated by $\mathcal{H}^c = \sigma\left(\sum_j \mathcal{H}_j^c + b^c\right)$, which is obtained by different 3D convolution kernels. b^c is the bias parameter and σ denotes the sigmoid function.

Then we hierarchically build a deep 3D ConvNet model by stacking convolutional layers (represented as C) and pooling layers (represented as P), as shown in Figure 7. In particular, multiple 3D feature volumes are generated in the C layer. In the P layer, the maximum pool operation is also performed in 3D, that is, the feature volume is subsampled based on the cube neighborhood. In the fully connected layer, the 3D feature volume is flattened into a vector as the input of the detection layer.

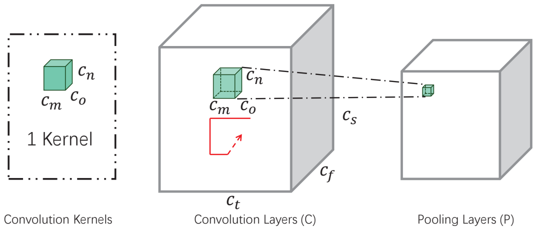


Fig. 7. The illustration of 3D convolution network, which is consisted of 3D convolution layer, with one or multiple convolution kernels, and pooling layer. The maximum pool operates in a cube manner and then outputs the flattened vector to downstream full connected layer.

3.6 Detection Layer and Optimization

The fraud detection task takes the transaction representation rep, which is the tensor flatted vector learned by attention and convolution networks, and aims to learn the probability of being a fraudulent trade. The loss function is the likelihood defined as follows:

$$\begin{aligned}\mathcal{L}(\theta) = -\frac{1}{N}\sum\nolimits_{i=1}^{N}&[y_i \log(\mathrm{detect}(rep_i:\theta)) \\ &+\lambda_3(1-y_i)\log(1-\mathrm{detect}(rep_i:\theta))]\end{aligned} \quad (10)$$

where rep_i denotes the representation of the $i-th$ transaction record, which is the output of 3D ConvNet, and λ_3 indicates the sample weight according to the biased distribution of fraud and legitimate records; y_i denotes the label of $i-th$ records, which is set to 1 if the record is fraud and 0 otherwise; $\mathrm{detect}(rep_i)$ is the detection function that maps rep_i to a real valued score, indicating the probability that whether the current transaction is fraudulent. We implement $\mathrm{detect}(rep_i:\theta)$ with two-layer ReLU and one-layer sigmoid network.

The proposed STAGN can be optimized through the standard SGD-based algorithms. In this paper, we used Adam Optimizer to learn the parameters. We set the initial learning rate to 0.001, and the batch size to 256 by default.

4 Experiments

4.1 Experiment Settings

4.1.1 Datasets

We collected fraud transactions from a major commercial bank, which comprises real-word credit card transaction records spanning twelve months, from Jan 1 to Dec 31, 2016. The ground truth labels are reported by consumers and confirmed by domain experts in the financial institution. If a trade is confirmed by financial experts as fraudulent, we label it as 1; otherwise, it is labeled as 0.

In data preprocessing, we first filter all fraudulent records as the positive dataset. As the number of users who have never been charged with unauthorized transactions is much larger than the number of affected users, we leverage downsampling of the negative (legitimate) part to alleviate the imbalanced problem. Please note that, in order to maintain user-level transaction patterns, we combine users who maintain multiple credit cards into one user ID, and then filtered out inactive users that have less than ten records within one month. After that, we adopt user-level downsampling of normal users instead of transaction-level sampling. Thus, for one user with multiple cards, which normally share credit limit, they are either all excluded or entirely included in the dataset.

Finally, the dataset contains 236,706 transaction records, by 1021 users, across 1160 location codes.

During implementation, we encode categorical data, such as user ID and location code, into one-hot representations. We round the time record from the millisecond level to a standard DataTime format (yyyy-MM-dd HH:mm:ss). For the amount attribute, like many other financial signals, it performs the distribution of long tail. We first cut off the outliers by the three-sigma rule [35] and then perform a log transform on the amount value.

4.1.2 Compared Methods

We employ the following state-of-the-art methods on our benchmark dataset to highlight the effectiveness of the proposed STAGN. In these experiments, the tasks are learned independently. These baselines include:

- *LR (Logistic Regression).* A linear logistical regression model widely used in financial systems.
- *GBDT [36].* This is a gradient boosting method to optimize the classification metric and effectively handle data in mixed types. We set the depth of tree to 3 and learning ratio to 0.01.
- *MLP [37].* A feed-forward network with three hidden layers which has 64 neurons and uses ReLU as the activation function.
- *Deep & Wide [38].* A mode that combines a deep neural network with a logistic regression model maintains the benefits of memorization and generation. On the deep side, the sizes of the three hidden layers are 128, 128, and 64.
- *CNN-max [10].* A convolution neural network model that applies a two-layer convolution and max pooling on the feature map, which has the same kernel size of 32. The output is then passed to a fully connected layer to produce the final fraud score.
- *AdaBM [3].* A hybrid method that uses AdaBoost and majority voting methods to detect fraud events.
- *LSTM-seq [11]* Phrase the fraud detection problem as a sequence classification task and employ LSTM networks to incorporate transaction sequences.
- *STAGN-notrans/nograph:* We exclude the manually engineered transaction feature/graph feature in STAGN, which means graph/transaction features are utilized alone.
- *STAN:* The spatial-temporal attention network [39]. It replaces graph feature with binned location categories.
- *STAGN-notemp/nospat/no3d:* Sub-models of the proposed STAGN, in which the temporal attention, spatial attention are not used, use the 2D convolution layer instead of our proposed 3D ConvNet.
- *STAGN-all:* The full proposed GNN-based spatial-temporal attention model in this paper.

4.1.3 Parameter Settings and Evaluation Metrics

In this experiment, we apply the preferred parameters for each of the baseline methods as they were originally proposed. For STAGN, we employ two convolution layers; each of them is set to $4 \times 4 \times 4$ convolution kernel, followed by a max-pooling layer. Two fully connected layers are added on the top of 3D ConvNet, each of them consisting of 32 neurons. We set the temporal and spatial parameters (λ_1 and λ_2) by cross-validation. The sample weight λ_3 is set by the training distribution of the positive and negative samples. In GNN layer, NN_v shares the same parameter setting with NN_e, which includes two full connected layers, with 64 and 32 neurons respectively. NN_g is another two hidden layers MLP and both of their neuron sizes are set to 32.

TABLE 1
Result of the fraud detection experiment.

	AUC (Oct)	AUC (Nov)	AUC (Dec)
LR	0.7247	0.7163	0.7199
GBDT	0.7868	0.7949	0.7864
MLP	0.7803	0.8012	0.7891
Deep & Wide	0.8210	0.8197	0.8108
CNN-max	0.8352	0.8367	0.8267
AdaBM	0.8243	0.8249	0.8232
LSTM-seq	0.8368	0.8353	0.8290
STAGN-nograph	0.8435	0.8462	0.8509
STAGN-notrans	0.8413	0.8507	0.8596
STAN	0.8832	0.8789	0.8865
STAGN-notemp	0.8631	0.8604	0.8736
STAGN-nospat	0.8602	0.8531	0.8583
STAGN-no3d	0.8688	0.8629	0.8716
STAGN-all	**0.8973****	**0.8897****	**0.8983****

We compare different methods by precision, recall, and F-Score. The AUC (area under the ROC curve) is also reported in the fraud detection experiment.

4.2 Fraud Detection

We evaluated the performance of different models for the fraud detection task. Records of the first nine months (from January to September) were used as training data, and then we predicted the potential fraud transactions in the following three months (Oct, Nov, and Dec). According to the dataset distribution, we employ records spanning six months as the feature extraction window and the next month as the label window. Afterward, we slide the window each month for accuracy evaluation. In the experiment, we repeated the test 10 times and report the average AUC in Table 1. ** indicates that the improvements are statistically significant for $p < 0.01$ judged by the paired t-test.

The first seven lines of Table 1 contain the results of some of the latest baselines. GBDT and MLP perform better than LR considerably. It is probably because the capacity of the LR model is too low to address the complicated fraud patterns. In all baselines, CNN-max and LSTM-seq prove to be competitive, which improves over 5% enhancement compared with traditional machine learning methods (GBDT and MLP). This strong demonstrates the necessity of deep models for fraud detection and shows the effectiveness of deep neural networks in learning the latent fraud patterns. STAGN-nograph is close to STAGN-notrans, both perform better than conventional baselines, which proves that the graph feature could achieve comparable accuracy with manually engineered features. Lines 11-14 show the results of STAGN and some of its submodels. STAGN-notemp's performance is close to STAGN-no3d, better than CNN-max and LSTM-seq, the spatial-temporal attention layer proves to be effective. It should be noted that STAGN-nospat is considerably lower than the rest two sub-models, which proves the superior performance of spatial features learned from the graph neural networks. STAGN-all outperforms all the other models, including STAN in our previous work [39]. The result proves the contribution of improving manually binned location categories of STAN by the graph feature in STAGN.

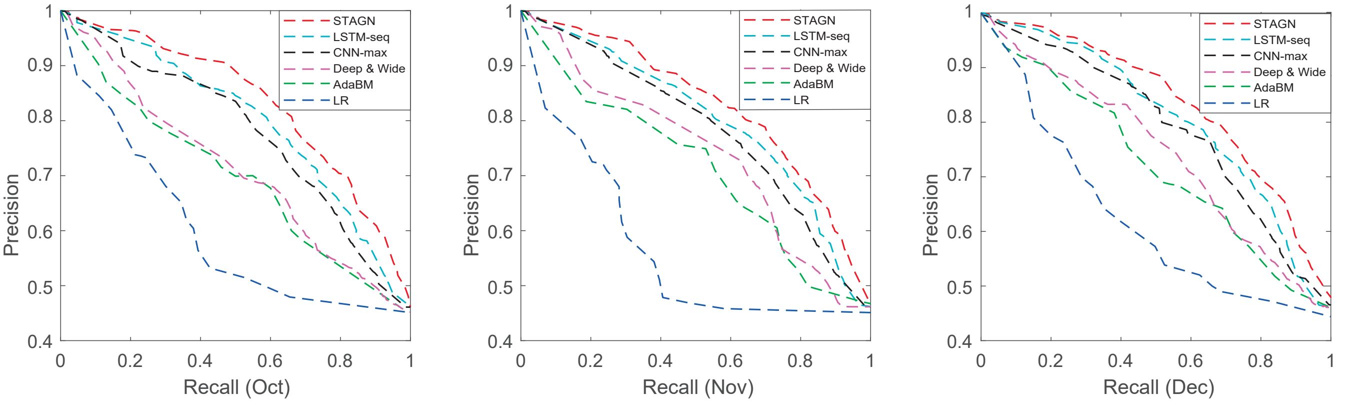


Fig. 8. Precision recall curve of STAGN compared with baseline methods. We employ the first nine months of the dataset as the training set and leave the next three months (Oct, Nov, and Dec) as the test set. As we can see, our proposed STAGN constantly performs better than other state-of-the-art baselines in all test time window, which demonstrates the effectiveness of fraud detection with spatial-temporal attention graph networks.

4.3 Precision-recall Curves

In Figure 8, we present the precision-recall curves for the lastest baselines. As shown, our proposed STAGN performs better than baselines with respect to the area under the precision-recall curves. The results of AdaBM and Deep & Wide are quite similar; both of them are much better than LR. Essentially, this might because fraud patterns in credit card transaction records are too complex for a simple linear model like LR to address. With the help of deep structures, LSTM-seq and CNN-max perform a slight promotion compared with AdaBM and Deep & Wide. In all baselines, LSTM-seq and CNN-max are shown to be the most competitive. The reason might be that they preserve the deep representation of raw features and explicitly makes use of the spatial features of our problem, while Deep & Wide and AdaBM are not optimized for local spatial and temporal patterns.

Our method, STAGN, consistently outperforms other state-of-the-art baselines. The reason is twofold: (1) STAGN deals with both spatial and temporal features and integrates them into an attention network, in which spatial features include not only the location-based features but also graph-based representations. While the CNN-max only deals with spatial ones that cannot address temporal patterns of transaction records. (2) STAGN uses a 3D convolutional network for tensor features instead of 2D convolution so that it can better capture hidden and intricate fraud patterns in spatial-temporal feature learning. Specifically, our methods work, comparable or even better, at the very beginning of the curve, compared to the state-of-the-art baselines. More importantly, our methods can accurately detect many more fraud transactions (high recall) with a relatively high precision, which is quite promising.

4.4 Parameter Sensitivity

In this section, we study the model generalization, which includes penalty parameters, the depth of hidden convolution layers, and their impact on our task.

We vary the temporal and spatial penalty parameters (λ_1 and λ_2) from 0 to 1 with a step of 0.02. As shown in Figure 9a, it can be easily found that the parameter has a great influence on model performance. Our model performs better by increasing λ from 0 to 0.1, and the AUC reaches the peak when $\lambda_1 = 0.1$ and $\lambda_2 = 0.15$. The performance is degraded if we keep on increasing the value of λ. The reason might be that varying λ could balance the model consider a proper spatial-temporal window. When we increase λ from 0 to 1, our proposed model could consider features in a different spatial-temporal range and reach a performance peak around $\lambda_1 = 0.1$ and $\lambda_2 = 0.15$.

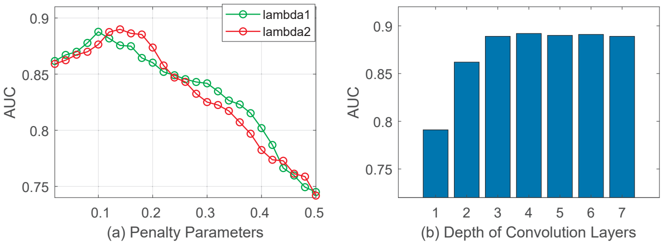


Fig. 9. The results of parameter sensitivity experiments. (a) presents the changes of AUC performance according to the value of temporal, spatial penalty parameters and (b) shows the test result on the different number of depths of convolution layers.

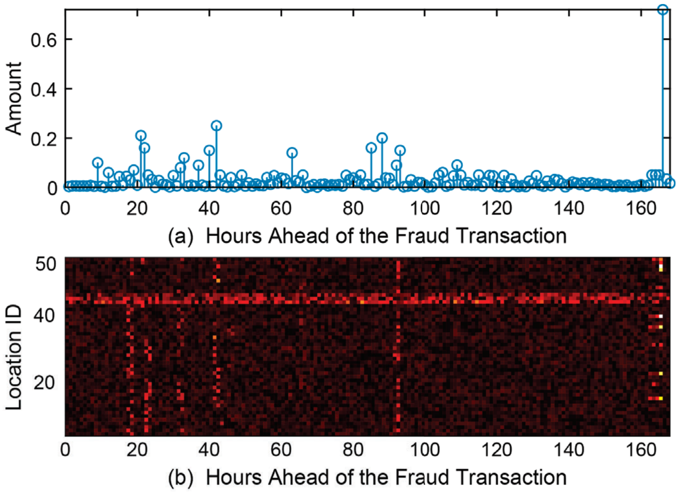
TABLE 2
The value of attention coefficients.

Temporal	Coefficients	Spatial	Coefficients
Seconds	0.2137	#13	0.1447
Minutes	0.0615	#21	0.1001
Hours	0.1533	#36	0.0158
Days	0.1057	#39	0.0192
Weeks	0.2930	#42	0.0374
Months	0.0104	#47	0.0182
Quarters	0.0006	#48	0.0091

Figure 9b shows the influence of the depth of hidden convolution layers on the AUC. With the deeper hidden convolution layers, the model tends to aggregate the temporal and spatial information from a neighborhood into a wider range. As we have seen, the AUC with a depth of 1 hidden layer does not work well because the information we have is mixed. Our model needs to ”swap” information in terms of temporal, spatial, and feature aspects, which requires a convolution of at least two hops to display.

4.5 Case Studies

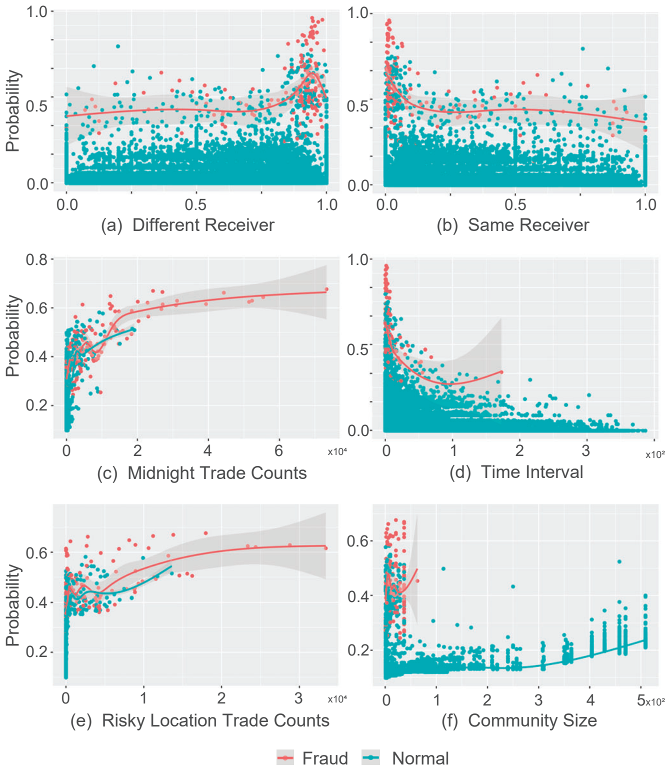
In this section, we employ empirical studies of the proposed STAGN in our collaborated bank and report the case and statistical results after six months of observation. During the process of case



(a) Hours Ahead of the Fraud Transaction

(b) Hours Ahead of the Fraud Transaction

Fig. 10. Case studies of attention weights. We randomly extract 1000 fraud transactions and backtracking records in one week before the fraud occurrence. (a) shows the hourly aggregated trading amount. (b) displays the heatmap of transaction locations in an hourly summary.



(a) Different Receiver

(b) Same Receiver

(c) Midnight Trade Counts

(d) Time Interval

(e) Risky Location Trade Counts

(f) Community Size

Fig. 11. Statistical results of the empirical study. Y-axis of all subgraphs denotes the predicted probability of whether a transaction is fraud by our proposed STAGN. (a) shows the ratio of recent transactions which are from the different receiver, while (b) reports the ratio of transactions that are from the same receiver. (c) displays the counts of midnight trades (0:00-4:00 in local time). (d) presents the time interval of two transactions. (e) shows the trade counts in risky locations and (f) displays the trade counts of different sizes of community in transaction graphs.

studies, we can retrieve the ground truth fraud label, which is confirmed by either the risk control manager from banks or the user reports. The spatial hotspot zones are learned by the predictive model (STAGN) and then investigated by the account managers from corresponding branches.

4.5.1 Fraud Hotspot Discovery

Table 2 shows the learned coefficients of spatial and temporal attention layers, in which “Weeks”, “Seconds” and “Hours” weights are noticeable. This is because user behavior usually shows a periodic distribution every week, but the fraudulent trades are concentrated in an instant until exceeding the user’s credit limit (there could be more than 100 transactions in one second). This phenomenon is also reported by [40]. In spatial studies, we present the top seven attention weights in Table 2.

In order to uncover the fraud patterns from learned attention weights, we adopted an empirical study on infected accounts with our collaborating domain experts. We firstly randomly selected 1000 fraud transactions and then backtracked other records within one week before the fraud event in the infected account. Finally, we collected the records from infected users into hours and aggregated them by summarizing the spending amount and times of trade location ID, as shown in Figure 10. We get the following observations.

Temporally: on average, fraud transactions account for over 70% of a user’s credit limit, illustrated in hour 166 of the x-axis (fraud event time) of Figure 10a, which means an average of 70% loss for each fraud event. We notice that a small equal number of trades are usually issued in 1-3 hours (between hour 162-165 on the x-axis) before the event. Domain experts have demonstrated they are trade attempts by fraudsters, after analysis of the records. This small number of attempts is important for fraudsters: 1) if successful, the card will be transferred for a large number of fraud transactions; 2) if failed, the cardholder might not notice the tiny amount of failed trade attempts. We also observe that the number of trades in hours 140-160 is low compared with hours 0-140, which means the card user may have missed the card one whole day before the fraud event.

Spatially: users obviously have a location propensity, as shown in Figure 10b, where the brighter color (red) means a higher frequency. We observe the two most popular trade merchants are located in ID #42 and #43, which are two popular online payment systems. It should be noted that fraud transactions are concentrated in limited locations, such as #13, #21, etc., which are generally different from user’s historically frequent trading locations. This study confirms our intuition of spatial aggregation and learned spatial attention coefficients.

4.5.2 Fraud Pattern Discovery

So far, based on the intuitions and observations from real-world datasets, we summarize that the fraud events are influenced by temporal and spatial limitations, in which spatial restriction includes both location and graph-based features. With support from the collaborated bank’s domain experts, we could get the unique opportunity to validate the effectiveness of the above intuitions in empirical studies. We show the statistical results of empirical study in Figure 11, which includes the temporal and spatial distributions from both the fraud and normal transactions.

As we can see, y-axis is the predicted fraud probability. The red dots denote the ground truth of confirmed fraud transactions, while the blue one represents normal transactions. In all subgraphs of Figure 11, we aim to learn the predictive model to better address clear discrimination between fraud and normal distributions. In Figure 11a and 11b, we extract the previous records within one day in user perspective for each transaction, and then calculate the ratio of the trade that are target to a different receiver or the same receiver. It is obvious that normal trades are uniformly distributed in different values of the ratio. But the fraud transactions show partially dense distribution. In particular, fraudsters are likely to trade the card with different receivers in limited times. As displayed in Figure 11a, frauds with a high ratio of different receivers are significantly larger than normal ones. The result proves that fraudulent behavior is complicated, and the swallow model would be suboptimal. Thus, we addressed the interdependence between location and time present in fraudulent behaviour with our spatial-temporal attention.

We summarize the temporal results with record time and interval time in Figure 11c and 11d. From naive intuition, the number of trades in midnight (from 0:00 to 4:00 in local time) should be rarely smaller than daytime, both of them are the temporal features in STAGN. However, due to the temporal limitations, fraudsters tend to swipe all the card balances as soon as possible, no matter at midnight or daytime. These behavior patterns are uncovered in Figure 11c, in which the midnight trade counts of fraud records are significantly larger than normal ones. As described above, fraudsters tend to swipe the card as soon as possible so that the time interval of fraud is less than normal transactions, which are demonstrated in Figure 11d. The results demonstrate the necessity of both record and interval time in temporal features and prove the essential of temporal attention in predictive models.

Figure 11e and 11f uncover the spatial fraud patterns during empirical studies. In the process of fraud hotspot discovery, we detect a set of potential risky locations by the proposed spatial attention mechanism. The risky location candidates are then sent for investigation by account managers in corresponding branches. We collect the labeled truth risky zones as risky locations and then count the number of trades which are located in the collected zones. As shown in Figure 11e, fraud counts are considerably larger than the number of normal consumptions. We then study the graph-based patterns, which is the other part of spatial attention. Recall the transaction graphs described in Section 3.3, we employ edge betweenness community detection [41] (implemented in igraph[1]) on the user-merchant transaction graphs, without losing generalization, a similar distribution is also observed with other community detection algorithms. It is obvious that regular trade locates in both small and large communities in a near-uniform distribution. But, the fraud transactions are restricted in a small community, which strongly echo the observation of spatial limitations. Thus, we uncover the location and graph-based spatial fraud patterns and prove the necessity of each component in spatial learning.

5 MODEL GENERALIZATION

Our proposed STAGN has shown its superiority in card fraud detection. But how its performance on other tasks, or is the model restricted to fraud detection tasks. We answer the above question in this section. Theoretically, STAGN can also improve other user behavior-based learning tasks because it is designed to extract spatial-temporal patterns from user historical records, especially for graph-structured behaviors. For example, 1) in *anti-money laundering* tasks, user transactions in time series could also be constructed into sequential graphs, and then employ graph neural network to learn spatial features and spatial-temporal attention over 3D convolution layer to address the hidden patterns. 2) *Location-based recommendation*, graphs could be generated by user historical behavior, in which user and item are denoted as nodes. Then, embed the location information into edges in unified graph neural networks. We could align the features with naturally sequential behaviors into spatial-temporal attention mechanisms and then train the network by supervised labels. 3) *Guaranteed-loan default prediction*, in which we could generate edges between two guaranteed companies and embed company locations in the graph. Loan historical behavior is learned in temporal in the proposed STAGN. We take the guaranteed-loan default prediction as an example and report the experiment results in the rest of this section.

Guaranteed loans (also known as guarantee circles) are a widespread economic phenomenon in Asia countries. In order to obtain loans from banks, groups of small and medium enterprises (SMEs) back each other to enhance their financial security [28]. It is difficult for small and medium enterprises (SMEs) to meet the banks lending criteria, which are designed for big companies. There is something of a blank area for setting the criteria for SMEs due to their lack of security. However, they are permitted to offer other corporations as an endorsement. Usually, banks need to collect as much fine-grained information as possible to make the decision, including transaction information, customer information, asset information such as mortgage status, and loan approval history. There is often more than one guarantor per loan transaction, and a single guarantor may make several loan transactions in a period. Upon approval of the loan, the SMEs usually obtain the full loan amount immediately and start making repayments by regular installments until the loan contract ends. If the borrower fails to repay in time, its guarantor(s) has a legal obligation to take the debt and continue to repay the loans.

We collect data from a major financial institution in East Asia, from 01/01/2013 to 31/12/2016. It includes 112872 nodes(SMEs), with 124957 edges (guarantee relationships). We construct the location-based graphs and spatial-temporal features similar to card fraud detection tasks, based on nine data tables: customer profile, loan account information, repayment status, guarantee profile, customer credit, loan contract, guarantee relationship, guarantee contract, and default status. We observed that most of the loans are repaid in month. Hence, we aggregate the behavior feature with one-month time window and mark the delinquency loans as target labels in month. The following state-of-the-art methods are used to highlight the effectiveness of STAGN on the loan dataset, including crDNN [42], INDDP [43] and DDPF [44]. The model parameters are set as their default recommendations, and the performance of the proposed prediction model is evaluated by Precision@k and MAP (Mean Average Precision).

Precision@k is used to evaluate the performance of default prediction, which means the predicted precision of top k SMEs. The formula of Precision@k is:

$$Precision@k = \frac{|\{i | i \in \mathbf{V}_p \cap \mathbf{V}_o\}|}{|\mathbf{V}_p|} \tag{11}$$

where $\mathbf{V}_p$ is the set of predicted top k defaulted SMEs, $\mathbf{V}_o$ is the set of observed default SMEs and $|\cdot|$ represents the size of the set.

Mean Average Precision(MAP) is used to evaluate the performance of default diffusion, which measures the rank accuracy of predicted SMEs list. The formula of MAP@k is:

$$\begin{aligned} AP@k(i) =& \frac{\sum_{j=1}^{k} Precision@j(i) \cdot \delta_i(j)}{\sum_{j=1}^{k} \delta_i(j)} \\ MAP@k =& \frac{\sum_{v_i \in V} AP@k(i)}{|V|} \end{aligned} \tag{12}$$

[1] https://github.com/igraph/igraph

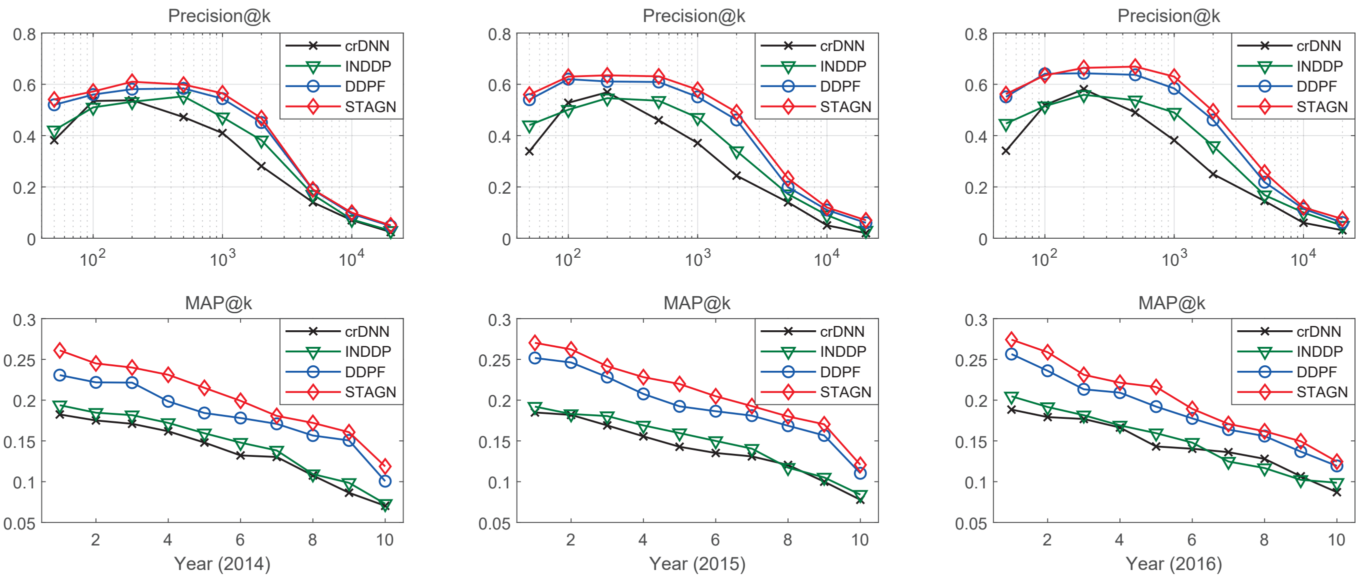


Fig. 12. Loan default prediction results. Upper part shows the Precision@k of delinquent predictions and low part displays the MAP@k in default diffusion prediction tasks.

where $Precision@j(i)$ is the Precision@j for SME v_i, and the $\delta_i(j)$ indicates the v_j is diffused by delinquent SME v_i.

Figure 12 presents the result in loan delinquency prediction. The records with the year 2012 are employed as training data, and then we predict the default probability of SMEs in month over the following three years. We conduct the experiment 10 times and report the average result here. The upper part shows the precision@k of default prediction with different k. The STAGN model outperforms the other three baselines considerably. The experimental results demonstrate the effectiveness of feature learning of the proposed spatial-temporal attention based prediction model. The superior performance is more significant in default diffusion experiments. We first select delinquent SMEs in the current time window and then predict its guarantors' default probability in the next time window. The MAP@k results are shown in the lower parts of Figure 12. Of all baselines, DDPF is shown to be competitive, proves the effects of temporal and graph-based patterns. STAGN performs much better than three baselines. This indicates that our methods can effectively capture the latent spatial and temporal behavior patterns in default diffusion prediction. The experimental results strongly prove that our proposed STAGN could benefit user-behavior based classification tasks by effectively learning complex temporal and spatial patterns from large scale datasets.

6 Tools, Implementation and Online Deployment

In this section, we describe the system implementation of the card fraud detection framework. So far, the detail of the STAGN and its performance is presented. But how to implement it into an industry level system is still challenging in two folds: (1) The online environment needs high-quality parallel detection capacity, which requires the system can predict massive real-word transactions in a short time. (2) The predictive model needs to be trained as quickly as possible, which can be substantially improved by new labeled data. At the same time, the retraining model should not block the online system.

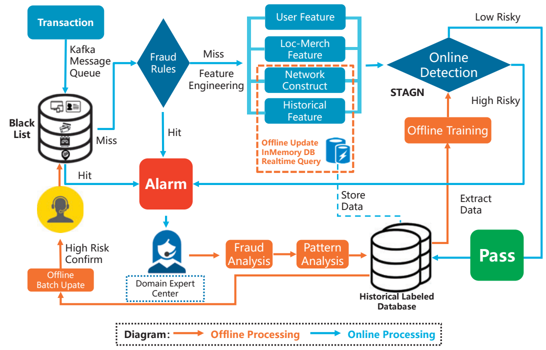


Fig. 13. The implementation of our proposed methods in an online and realtime fraud detection system.

In terms of the function scope, anti-fraud includes application anti-fraud and transaction anti-fraud, in which transaction anti-fraud consists of in-process detection and post-process detection. The in-process system mainly consists of rules and blacklists, and it will block the transaction once detected as high-risk of fraud. While the post-process detection is parallel to transaction flow, it alarms identified fraud candidates to domain experts for further judgment. Typically, the post-process system is implemented by a predictive model, like STAGN in our work, which is mainly used to find potential fraud candidates for domain experts to judge and update (or retrain) the predictive model. Normally, in-process and post-process detection systems are deployed simultaneously in commercial banks. One of the important impacts of the post-process predictive model is that new alarmed frauds will be used to update in the in-process system, which means the confirmed frauds from the predictive model are utilized to generate rules and blacklists in in-process detection. In particular, as shown in Figure 13, transactions are processed in a distributed message queue for real-time detection. They will be evaluated by the blacklist and fraud rules (called in-process detection) first and will be directly blocked if there is a hit on any blacklists or fraud rules. If not, we then construct spatial and temporal features for an online predictive model (STAGN here, which called post-process detection). In the process, we store the hit historical data within an in-memory database to support large-scale detection. The high risky transactions are sent for confirmation by domain experts and feedback the result to the historical database. Afterward, we retrain the model offline in a batch manner, so that the updated model can learn from the latest feedback of experts. The newly detected frauds are also utilized to enhance the blacklist and fraud rules.

During implementation, our hardware system includes two servers with the same configuration. Each server contains two CPUs with Intel Xeon E5-2680 v3, four pieces of GPU with Telsa P100 and 512G memory. The software system is hosted by CentOS 7.2, Java 8 and Python 3.6. For each component, we employ Kafka [45] as the distributed message queue, Redis [46] as the in-memory database and Drools [47] on Apache Flink as the rule and streaming engine. For graph visualization we use the open-source software package Gephi [48] and layout ForceAtlas2 [49]. We utilize Apache Spark GraphX [50] and igraph [51] for large-scale graph analysis. The training model and system implementation are written in Python, Java, and Scala [52]. It is reported that STAGN can predict around 10,000 online transactions per second. This already meets the efficiency requirement according to the industry standard in banks. When deployed to the industry-level system, STAGN takes only 30 minutes to train the model on 2 billion records during the training process. Note that the training of the updated model and the prediction of the current learned model can be done independently. Thus, there is no block for the system.

Due to the limitation of industry support, there are rarely systematic studies on credit card fraud prediction in academic literature. We fill this gap by studying real-world fraud transactions. Our work uncovers fraudsters' weakness and protects innocent victims from potential fraud loss, which may inspire more research work in the literature.

7 Conclusion

In this paper, we present a novel attentional 3D convolution neural network for credit card fraud detection. In particular, we summarize the weakness of fraudsters, called "temporal aggregation" and "spatial aggregation", and propose a 3D convolutional neural network approach based on a spatial-temporal attention mechanism. The spatial features are learned by a graph neural network from location-based transaction graphs. This is the first work in which attentional 3D ConNet has ever been employed for the credit card fraud detection problem. Our methods achieve promising AUC and precision-recall curves compared with other state-of-the-art baseline methods. Furthermore, we explore to uncover fraud patterns by the observation of learned attention weights in case studies. The proposed method is extensively evaluated in an online transaction post-analysis system. The result demonstrates that our methods can effectively detect fraudulent transactions.

To avoid fraud, we have considered the transaction records to address "temporal aggregation" and "spatial aggregation" weaknesses. However, fraudsters also leave abundant actions in online systems, such as applying information, making transaction actions by mobile applications, which could assist both the predictive model and domain experts. Besides, our current model is deployed to alert domain experts to fraud candidates, which are manually processed by agents in the follow-up actions; it does not block the transaction in real-time. Thus, in the future, we plan to investigate integrating online e-commerce behavior in a fraud detection model and build a real-time in-process fraud detection system.

Acknowledgments

The work is supported by the National Key R&D Program of China (2018AAA0100704) and the China Postdoctoral Science Foundation (2019M651499). Xiaoyang Wang is supported by the National Natural Science Foundation of China (61802345). Ying Zhang is supported by ARC DP180103096 and FT170100128.

References

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

[2] D. Wang, B. Chen, and J. Chen, "Credit card fraud detection strategies with consumer incentives," *Omega*, vol. 88, pp. 179–195, 2019.

[3] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using adaboost and majority voting," *IEEE access*, vol. 6, pp. 14 277–14 284, 2018.

[4] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan, "Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3637–3647, 2018.

[5] R. Patidar, L. Sharma *et al.*, "Credit card fraud detection using neural network," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 1, no. 32-38, 2011.

[6] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.

[7] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," *Decision Support Systems*, vol. 95, pp. 91–101, 2017.

[8] K. Seeja and M. Zareapoor, "Fraudminer: A novel credit card fraud detection model based on frequent itemset mining," *The Scientific World Journal*, vol. 2014, 2014.

[9] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, 2017.

[10] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *International Conference on Neural Information Processing*. Springer, 2016, pp. 483–490.

[11] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, "Sequence classification for credit-card fraud detection," *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.

[12] J. A. Gómez, J. Arévalo, R. Paredes, and J. Nin, "End-to-end neural network architecture for fraud scoring in card payments," *Pattern Recognition Letters*, vol. 105, pp. 175–181, 2018.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[14] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2018.

[15] M. Allamanis, H. Peng, and C. Sutton, "A convolutional attention network for extreme summarization of source code," in *International Conference on Machine Learning*, 2016, pp. 2091–2100.

[16] F. Baradel, C. Wolf, and J. Mille, "Human action recognition: Pose-based attention draws focus to hands," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 604–613.

[17] D. Li, T. Yao, L.-Y. Duan, T. Mei, and Y. Rui, "Unified spatio-temporal attention networks for action recognition in videos," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 416–428, 2018.

[18] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 855–864.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[20] J. Fernández-Gracia and J.-P. Onnela, "Flexible model of network embedding," *Scientific reports*, vol. 9, no. 1, pp. 1–7, 2019.

[21] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016, pp. 1225–1234.

[22] L. Liao, X. He, H. Zhang, and T.-S. Chua, "Attributed social network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2257–2270, 2018.

[23] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[24] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 731–739.

[25] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[26] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 15–24.

[27] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3917–3925.

[28] D. Cheng, Y. Tu, Z. Ma, Z. Niu, and L. Zhang, "Risk assessment for networked-guarantee loans using high-order graph attention representation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 5822–5828.

[29] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[30] M. Elbayad, L. Besacier, and J. Verbeek, "Pervasive attention: 2d convolutional neural networks for sequence-to-sequence prediction," *arXiv preprint arXiv:1808.03867*, 2018.

[31] D. Xu, W. Ouyang, X. Alameda-Pineda, E. Ricci, X. Wang, and N. Sebe, "Learning deep structured multi-scale features using attention-gated crfs for contour prediction," in *Advances in Neural Information Processing Systems*, 2017, pp. 3961–3970.

[32] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[33] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, 2002, pp. 261–270.

[34] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[35] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[36] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146–3154.

[37] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2015.

[38] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, 2016, pp. 7–10.

[39] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 362–369.

[40] M. R. Lepoivre, C. O. Avanzini, G. Bignon, L. Legendre, and A. K. Piwele, "Credit card fraud detection with unsupervised algorithms," *Journal of Advances in Information Technology*, vol. 7, no. 1, 2016.

[41] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[42] F. Tan, X. Hou, J. Zhang, Z. Wei, and Z. Yan, "A deep learning approach to competing risks representation in peer-to-peer lending," *IEEE transactions on neural networks and learning systems*, 2018.

[43] D. Cheng, Z. Niu, Y. Tu, and L. Zhang, "Prediction defaults for networked-guarantee loans," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 361–366.

[44] D. Cheng, Y. Zhang, F. Yang, Y. Tu, Z. Niu, and L. Zhang, "A dynamic default prediction framework for networked-guarantee loans," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2019, pp. 2547–2555.

[45] J. Kreps, N. Narkhede, J. Rao *et al.*, "Kafka: A distributed messaging system for log processing," in *Proceedings of the NetDB*, 2011, pp. 1–7.

[46] A. Patel, "Sales transaction system using redis database," 2015.

[47] E. E. Thu and N. Nwe, "Transforming model oriented program into android source code based on drools rule engine," *Journal of Computer and Communications*, vol. 5, no. 03, p. 49, 2017.

[48] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: an open source software for exploring and manipulating networks," in *Third international AAAI conference on weblogs and social media*, 2009.

[49] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PloS one*, vol. 9, no. 6, p. e98679, 2014.

[50] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica, "Graphx: Graph processing in a distributed dataflow framework," in *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, 2014, pp. 599–613.

[51] W.-S. Han, J. Lee, M.-D. Pham, and J. X. Yu, "igraph: a framework for comparisons of disk-based graph indexing techniques," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 449–459, 2010.

[52] J. S. Andersen and O. Zukunft, "Evaluating the scaling of graph-algorithms for big data using graphx," in *2016 2nd International Conference on Open and Big Data (OBD)*. IEEE, 2016, pp. 1–8.

Dawei Cheng is PostDoc researcher at MoE key lab of artificial intelligence, department of computer science and engineering, Shanghai Jiao Tong University. He received the Ph.D. Degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2018. His research fields include data mining, machine learning, and knowledge discovery. Before that, Dawei was a senior software engineer in Intel Asia Pacific Research and Development Center.

Xiaoyang Wang received the BSc and MSc degrees in computer science from Northeastern University, China, in 2010 and 2012, respectively, and the PhD degree from the University of New South Wales, Australia, in 2016. He is a professor in Zhejiang Gongshang University, Hangzhou, China. His research interest includes query processing on massive spaital data stream.

Ying Zhang is a Professor and ARC Future Fellow (2017-2021) at CAI, the University of Technology, Sydney (UTS). He received his BSc and MSc degrees in Computer Science from Peking University, and PhD in Computer Science from the University of New South Wales. His research interests include query processing on data stream, uncertain data and graphs. He was an ARC APD (2011-2013) and ARC DECRA (2014-2016) holder.

Liqing Zhang received the Ph.D. degree from Sun Yet-sen University, Guangzhou, China, in 1988. He was a Research Scientist with the RIKEN Brain Science Institute, Japan, from 1997 to 2002. He is now a full Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include visual cognitive computing, uncertainty reasoning, statistical machine learning and inference.