Temporal-Aware Graph Neural Network for Credit Risk Prediction

Daixin Wang * Zhiqiang Zhang * Jun Zhou *† Peng Cui ‡ Jingli Fang * Quanhui Jia * Yanming Fang * Yuan Qi *

Abstract

Credit risk prediction is a fundamental problem for most financial institutions. Previous methods mainly adopt users' individual features on a single snapshot. However, users' individual features on financial platforms are usually too sparse to be informative. And previous methods ignore that the features, the behaviours and the credit risk of the users are all dynamic. To resolve the problems, we aim to model the credit risk prediction on dynamic graphs and propose a Temporal-Aware Graph Neural Network (**TemGNN**) to predict user credit risk. In detail, the model consists of three parts: i) a static model to extract the user's static factors regarding the credit risk. ii) a short-term graph encoder with special graph convolution modules for each snapshot to enrich the user's information through aggregating short-term temporal and structural information. iii) a long-term temporal model based on LSTM with interval-decayed attention to adaptively aggregate the long-term information from the static factors and interval-irregular dynamic snapshots. By combining the three parts together, our model is able to mine both the short- and long-term temporal-structural information. Experimentally, we use the users' authorized lending behaviours as the temporal graphs to do default prediction on Alipay. The results show that our model achieves the best performance among the state-of-the-art methods.

1 Introduction

Recently inclusive finance is a very innovative and critical topic [8, 23]. Inclusive finance aims at promoting the availability of many financial services like loan and insurance to the broadest users and enterprises. Therefore, it can help reduce poverty and inequality by helping most people manage financial risks to invest. Due to its great effect on reducing the inequality and promoting the economic growth, inclusive finance has been widely recognized by world leaders and policymakers and produced great social impact. To achieve inclusive finance and give financial services to a wide range of people at an affordable cost, how to predict user credit risk lays the foundation. An accurate risk prediction on the one hand can reduce the capital loss for the inclusive finance company, and on the other hand is crucial to preserve the financial stability of the whole society.

Credit risk prediction aims to predict whether a user will be involved in a default or other types of financial fraud in the future. Traditional methods are mainly the rule-based methods to predict these frauds [21, 13, 34]. However, the designed rules are difficult to fit the evolving and complex fraud patterns. Then following methods start to use the machine learning methods to mine the fraud patterns from the data. These methods mainly extract some statistical features from the user's own profile and the behaviour. However, in financial platforms most users have specific goals. It means that they often stay shortly online, making their individual features too sparse and not informative enough to model their own credit risk. Then some following graph-based methods start to build graphs to link the users and enrich the user information via their neighbors, leading to a better result [19, 15]. However, these graph-based methods only focus on modelling a single graph snapshot. Actually, users usually act on a series of time-ordered financial events such as lending money from different institutions at different times. And within each event, users can link to different entities. We aim to model the credit risk prediction on temporal graphs to further enrich the user information.

However, to predict the credit risk from the temporal graphs meets the following three challenges.

(1) **Time Interval Irregularity**: The time-ordered financial events of users are commonly acquired with very irregular intervals, ranging from days to years. For example, users may lend money monthly or even yearly. Such a time interval irregularity proposes a big challenge for traditional time-series model to aggregate the temporal information from different snapshots.

*Ant Financial Services Group. Email: {daixin.wdx, lingyao.zzq, jun.zhoujun, sophone.fjl, quanhui.jia, yanming.fym, yuan.qi }@antfin.com

†Corresponding Author

‡Tsinghua University, China. Email: cuip@tsinghua.edu.cn

(2) **Extract the static and temporal factors**: A user's credit risk is mostly stable for a period of time because it is affected by some relatively **static** factors like the user's occupation and education background. Simultaneously, in **short term**, user's status and behaviour may face some sudden changes like being fired and being sick. In **long term**, user's behaviour and financial risk have periodicity. How to simultaneously extract the static, the short- and long-term temporal factors from dynamic data is the second challenge.

(3) **Incorporate the structural and temporal information**: Relations between users and other entities can help enrich user's information and define the user's own credit risk [3]. But how to simultaneously incorporate structural and temporal information in a unified model is the third challenge.

To address the aforementioned challenges, we model the credit risk prediction on dynamic graphs and propose Temporal-Aware Graph Neural Network (**TemGNN**) to model the dynamic graphs. In detail, our model consists of three parts: (1) We propose a separate pathway to model the static factors regarding the user's credit risk. (2) We propose short-term graph encoders with special graph convolution modules for each snapshot. The encoder aims to enrich each user's information through aggregating the information from neighbors within short-term snapshots. (3) We design a time-series model to capture the long-term information from the representations learned from each graph encoder. Specifically, considering the time interval irregularity between snapshots, we design a time-decayed temporal attention mechanism when aggregating the long-term temporal information. By putting these three parts together, the proposed model is able to adaptively find a good combination of the static factors, short- and long-term temporal factors regarding the user's credit risk. Experimentally, we utilize user's lending behaviour as time-series events and in each event, we build a user-entity graph to correlate different users. The results demonstrate that our proposed model outperforms all the state-of-the-art methods.

The main contributions of this paper include:

- We propose a novel temporal-aware graph neural network, which simultaneously aggregates the user's static factors, short-term and long-term temporal factors to predict credit risk.
- We design a special graph convolution module to capture short-term temporal and structural information.
- We propose an interval-decayed temporal attention mechanism to aggregate the long-term temporal information from the interval-irregular snapshots.
- We conduct extensive experiments on a real-world financial dataset. The results demonstrate that our proposed method achieves a substantial gain over baseline methods. The method is also deployed online to predict the user default label.

2 Related Work

2.1 Credit Risk Prediction Earlier works mainly design some rules to predict the credit risk [29, 2, 28]. However, manually defining rules is easily to be attacked and difficult to deal with the evolving fraud patterns. Then the learning-based methods start to learn the patterns automatically from the data. Previous learning-based works extract some individual statistic features from a single snapshot to represent each user and use different classifiers like neural network [10] and decision tree [22] to make the predictions. However, the behaviours of most users in financial scenarios are too sparse. Then following methods start to use the graph-based methods to enrich the user information through neighbor nodes [19, 15]. Some other methods [16, 7, 6] adopt the temporal information to enrich the user's information. However, all of these methods cannot well extract the relationships between the structural and temporal information, which is crucial for predicting a user's credit risk.

2.2 Graph-based Method Network embedding, aiming at learning a vectorized representation for each node, is effective to model the graph structure. Recently, graph neural network (GNN) and many of its variants like GCN [17], GraphSAGE [12] and GAT [27] are very popular to learn node representations from both the node attributes and the graph topology via a deep structure [30]. However, all of these methods only target on the static graph. To model the temporal and structural information simultaneously, some methods integrate convolution neural network (CNN) and time-series model like Recurrent neural network and LSTM to do traffic flow prediction [33, 32, 25]. However, CNN-based methods cannot process our data because different users have irregular neighbor structures. To address the limitations, following methods [9, 26, 4, 24] combine graph-based methods with time-series model. The graph-based methods can deal with the irregular neighbor structures and the time-series model is used to extract the temporal information. However, these methods disregard some specific properties of our problem of credit risk prediction. They do not consider the static factors and short-term temporal factors. Additionally, the time intervals between snapshots in financial applications are very irregular, which makes the aggregation of the long-term temporal information very challenging for the previous time-series methods.

3 Preliminaries

3.1 Problem Definition We first formally define the dynamic graphs as follows.

DEFINITION 1. *(Dynamic Graphs) Dynamic graphs are represented as a sequence of time-ordered graphs* $G_{t_1}, ..., G_{t_T}$*, where each graph snapshot is denoted as* $G_t = (V, E_t, X_t)(t \in \{t_1, ..., t_T\})$*. Let* V *denote all vertices,* E_t *denote the edges and* $X_t \in R^{n \times d}$ *denote the node features at time* t*, where* $n = |V|$*.*

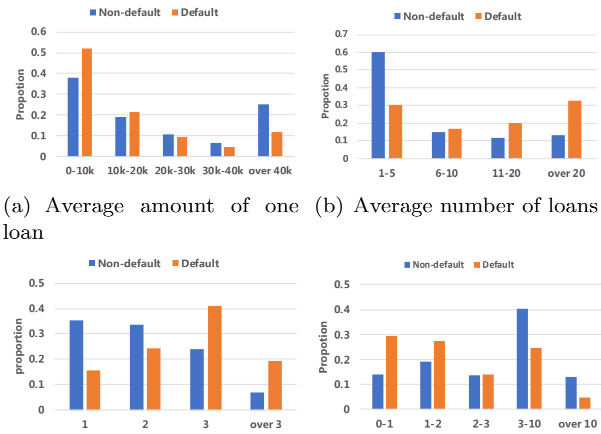
We aim to model the temporal and structural information via the evolution process of the dynamic graphs and drive these information into a representative model for spatial-temporal feature learning. Here we give the formal definition of our problem.

DEFINITION 2. *(Credit Risk Prediction in Dynamic Graphs) Given dynamic graphs* $\{G_{t_1}, ..., G_{t_T}\}$*, our problem aims to learn the temporal and structural information from the previous* T *graph snapshots to predict the user credit risk labels* $\hat{Y}_{t_T}$ *at time* T*, defined as*

$$\hat{Y}_{t_T} = \arg\max P(Y|G_{t_1}, ..., G_{t_T}).$$

3.2 Exploratory Data Analysis We sample a subset of Alipay users and use their authorized lending behaviour from the year of 2017 to 2019 to judge whether these users will be default. The "default" user means the user who borrows the money from the platform of Alipay cannot return the money on time. The lending behaviour data show when, where, who and how these users lend money and repay the money. If a user lends money from an institution, we build a link from the user to the institution. Then the users' time-ordered lending history can form the temporal graphs. The details about the data can be found in Section 5. With the data, we give some data analysis to support our intuitions that both the static features, the structural and the temporal information on lending behaviour are informative for credit risk prediction.

In Figure 1, we report the distributions of four features on the default and non-default users. The results show that there is an evident distinction on the distributions of the four features between two groups of users. Firstly we find that the average amount of each loan for default users is much smaller than the normal users. The reason is that default users usually show higher credit risk, which makes them difficult to lend too much money from a loan institution. Therefore, if default users need more money, they have to lend more number of loans from more types of loan institutions

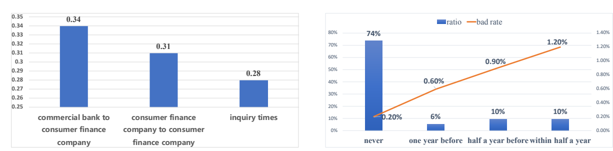


(a) Average amount of one loan (b) Average number of loans

(c) Average number of institution type (d) Average month of lending interval

Figure 1: Distributions on the (a) average amount of one loan (b) average number of loans (c) average number of lending institution types and (d) average month of lending interval for each user.

with smaller lending interval, which is in accordance with the result of Figure 1(b-d). It indicates that the structural information like the relations between users and lending institutions and the temporal patterns like the lending interval are discriminative.



(a) (b)

Figure 2: (a) Feature importance (IV value) on one static feature and two temporal features. (b) The relationship between the default rate and the most recent time that the user's lending institution type changes from commercial bank to consumer finance company.

In Figure 2, we give some deeper analysis on the temporal patterns. In detail, in Figure 2(a), we report the information value [14], a metric to measure the feature importance, of three features: (i) inquiry times (ii) the number of times that the user's lending institution type changes from commercial bank to consumer finance company (iii) the number of times changeing from consumer finance company to consumer finance company between two successive lending events. Note that feature (i) of inquiry times is the most significant static feature to predict the credit risk in our scenario. The

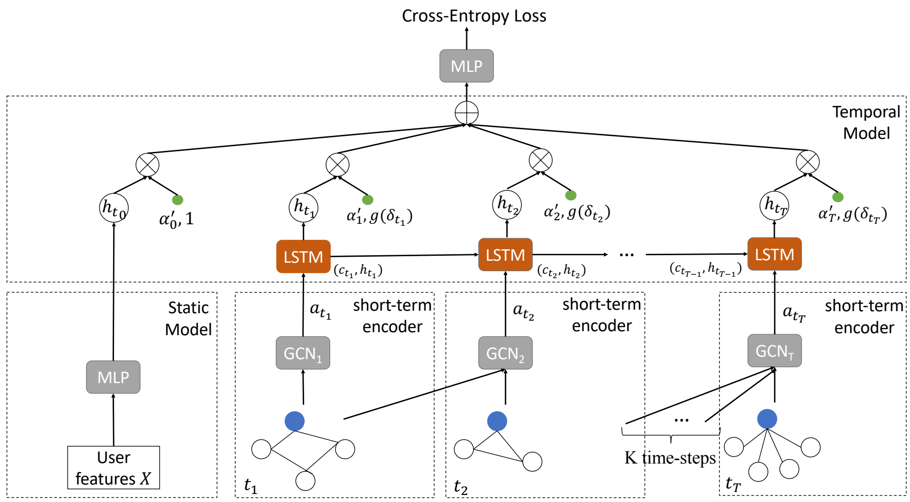


Figure 3: The framework of the proposed model **TemGNN**.

rest of two features are temporal features obtained from the user's lending history. We find that these two temporal features are even more important than inquiry times. Furthermore, in Figure 2(b), we find that the more recent the user changes the type of lending institution from commercial bank to consumer finance company in two successive lending events, the larger the bad rate of users is. These two results further demonstrate that temporal patterns are very important to indicate a user's credit risk.

In summary, we find that default and non-default users have an evident difference on many aspects of lending behaviour, including the static features, the temporal and structural features. Then we introduce our proposed model to learn from the dynamic graphs.

4 Methodology

4.1 Overall Framework We propose a novel Temporal-Aware Graph Neural Network named **TemGNN**, to predict the credit risk label. The model consists of three components: a separate pathway to model the static factor in terms of the credit risk, the short-term graph encoders for each snapshot to capture the short-term temporal and structural information, the time-series model to ensemble the long-term information of dynamic graphs. Finally, **TemGNN** will predict the risk label for each user. The overall framework is shown in Figure 3.

4.1.1 Static Feature Learning Model As we claimed before, in most times the credit risk of users is stable and will not change too much because the risk is greatly affected by some user static factors like the consuming habits, the education background and the amount of family asset. Therefore, it is important to model the static factors. Unlike temporal factors, static factors remain unchanged for all the snapshots. Considering this, we propose a separate model to learn the static factors first. In detail, we extract some relatively stable raw features of users, like the amount of fixed assets, the industry and the number of family numbers, denoted as $\mathbf{x}$. We use a Multi-layer Perceptron (MLP) with ReLU to project the attribute vector into the high-level space termed as $\mathbf{h}_{t_0} = \mathrm{MLP}(\mathbf{x})$. In this way, the model can learn latent factors from the static features.

4.1.2 Short-term graph encoder In each snapshot, we build a graph whose edges denote the lending relations between the user and lending institution. The relations can help to enrich the user information via his neighbor nodes to give more hints on his credit risk. Therefore, we hope to use the graph convolution to aggregate the neighbor information.

Unlike previous GCN models only aggregating the neighbor information of the current time, we assume that the node's current representations are also greatly affected by the recent information of his neighbors. For example, Figure 2(a) shows that the user's current status is greatly affected by both the lending institution type of the current snapshot and the previous snapshots. Considering this, we propose a short-term graph encoder with special graph convolution modules to learn short-term temporal and structural information. In detail, our designed graph convolution module operates on a small historical window (K time steps and we set K as 3 in this paper) of temporal graphs. For each node, its neighbor embedding is obtained by aggregating its neighbors' information at the current time step and the previous K-1 time steps. The aggregated neighbor embeddings of node u at time-step t are defined as:

$$\hat{\mathbf{h}}_{\mathcal{N}_t(u)} = \sum_{t' \in \mathrm{Prev}_K(t)} \sum_{v \in \mathcal{N}_{t'}(u)} \alpha(u_t, v_{t'}) \hat{\mathbf{h}}_{v_{t'}}, \tag{4.1}$$

where $\mathrm{Prev}_K(t)$ denotes the current and the previous K-1 snapshots of the time-step t and we use zero-padding if it exceeds the bound, $\hat{\mathbf{h}}_{u_t}$ is the node u's input features at time-step t, $\mathcal{N}_t(u)$ is defined as the node u's neighbors at time t and $\alpha(u, v)$ is the attention coefficient between two nodes. Considering that different entities at different times should have different importance to characterize the user's credit risk, we define structural attention by matching the embedding of the node u and node v as follows:

$$s(u_t, v_{t'}) = \mathbf{a}^T \cdot W_{\mathrm{spatial}} \cdot (\hat{\mathbf{h}}_{u_t} || \hat{\mathbf{h}}_{v_{t'}}), \tag{4.2}$$

where $||$ denotes the concatenation, $\cdot^T$ denotes the transposition, W_{spatial} denotes a trainable mapping matrix and $\mathbf{a}$ maps the unified representations into an attention score.

To make the attention score comparable across different nodes, we normalize the attention score by using softmax function:

$$\alpha(u_t, v_{t'}) = \frac{\exp\left(s(u_t, v_{t'})\right)}{\sum_{\hat{t} \in Pre_K(t)} \sum_{j \in \mathcal{N}_t(u)} \exp\left(s(u_t, j_{\hat{t}})\right)}. \tag{4.3}$$

Finally, we combine the user representation $\hat{\mathbf{h}}_{u_t}$ and the neighborhood representation $\hat{\mathbf{h}}_{\mathcal{N}_t(u)}$ to update the user representation in the next layer as $h^{(1)}_{u_t}$:

$$\hat{\mathbf{h}}^{(1)}_{u_t} = \mathrm{ReLU}(W^{(1)}_{\mathrm{GNN}}(\hat{\mathbf{h}}_{u_t} \| \hat{\mathbf{h}}_{N_t(u)})) \tag{4.4}$$

where $W^{(1)}_{\mathrm{GNN}}$ is the trainable weight matrix.

We can stack multiple such graph convolution modules to form the hierarchical model. And finally we get:

$$\mathbf{a}_{u_t} = \hat{\mathbf{h}}^{(L)}_{u_t} \tag{4.5}$$

where L denotes the number of layers for the GNN.

In brief, the proposed GNN model can learn user embedding by aggregating the neighbor information and the short-term temporal information while considering the structural importance.

4.1.3 Temporal Model As we discussed before, since the user's behaviour and the credit risk have both periodicity and mutability, it is important to capture both the short-term and long-term temporal information. To achieve this, we collect T snapshots of dynamic graphs ($T > K$) and propose a time-series model based on LSTM to learn long-term temporal patterns from the dynamic graphs.

In detail, with the T snapshots of dynamic graph, we can learn the user embeddings $\mathbf{a}_{t_1}, ..., \mathbf{a}_{t_T}$[1] by using the proposed short-term graph encoders for each snapshot. User embedding of each snapshot characterizes the short-term temporal and structural information. We further propose a temporal model based on LSTM to model the long-term temporal information.[2]

$$(\mathbf{h}_{t_i}, \mathbf{c}_{t_i}) = \mathrm{LSTM}_i(\mathbf{a}_{t_i}, \mathbf{h}_{t_{i-1}}, \mathbf{c}_{t_{i-1}}), i \in \{1, ..., T\}$$

The t-step cell of the LSTM uses $\mathbf{a}_{t_i}$, $\mathbf{h}_{t_{i-1}}$ and $\mathbf{c}_{t_{i-1}}$ as the input and gives $\mathbf{h}_{t_i}$ and $\mathbf{c}_{t_i}$ as the output. $\mathbf{h}_{t_i}$ theoretically can aggregate the temporal information until the current time-step. Therefore, previous works mainly use the output embedding of the last LSTM cell $\mathbf{h}_{t_T}$ for the subsequent task. However, due to the vanishing gradient problem [1], the learning of $\mathbf{h}_{t_T}$ is difficult to capture the remote information. Actually, credit risk is a complex problem that both the recent and the remote events affect the credit risk. For example, banks usually need a rather long time financial records to review a person's risk. Therefore, if we only utilize the output of the most recent cell, it is difficult to capture the long-term information to give an accurate prediction result. We hope to aggregate the representations of long steps.

To aggregate the long-term information from multiple cells and the user's static embedding, we propose a temporal attention. The attention score for each cell is defined as:

$$s'(\mathbf{h}_{t_i}) = \tanh(\mathbf{h}_{t_i}) \cdot \mathbf{w}_{\mathrm{tem}}, i \in \{0, 1, ..., T\} \tag{4.6}$$

where $\mathbf{w}_{\mathrm{tem}} \in R^{d \times 1}$ denotes the learnable weight and d is the embedding dimension.

With the attention score, we use the softmax to get the attention coefficients for each time-step:

$$\alpha'(t) = \frac{\exp\left(s'(\mathbf{h}_t)\right)}{\sum_{i \in \{0,1,...,T\}} \exp\left(s'(\mathbf{h}_{t_i})\right)}. \tag{4.7}$$

Then we can perform weighted aggregation on the embedding of each time-step to get the final user representation. However, such an aggregation method still assumes that the intervals between time steps are uniformly distributed. Actually, in our problem, the intervals between two consecutive lending events usually range from days to years. And intuitively the events which have a larger interval should have less impact on the subsequent events. Then to consider such interval irregularity, we further define a monotonic decreasing function of intervals as follows:

$$g(\delta_t) = 1/\log(e + \delta_t), \tag{4.8}$$

[1] For simplicity, we omit the superscript of user index u here.
[2] We omit the detailed calculation process of LSTM here.

where δ_t is the time interval between the event t to now.

Then, user embedding is defined as the sum over all the time-steps using the interval-decayed temporal attention:

$$\mathbf{h} = \sum_{i=0}^{T} (g(\delta_{t_i}) \cdot \alpha'(t_i) \cdot \mathbf{h}_{t_i}). \tag{4.9}$$

From Eqn. 4.9, instead of only utilizing the state of the current time-step, we collect several previous memories and regularize these memories by the interval decayed attention mechanism. Then, we use an MLP to map the user embedding $\mathbf{h}_u$ into a risk score $y_u = \text{MLP}(\mathbf{h}_u)$. Accordingly we can define the cross-entropy loss as the classification error:

$$\mathcal{L} = -\frac{1}{|V|} \sum_{u \in V} y'_u \log(y_u) + (1 - y'_u) \log(1 - y_u), \tag{4.10}$$

where $y'_u \in \{0, 1\}$ is the ground-truth for the user u.

We optimize the whole model using Stochastic Gradient Descent. Then the loss is back-propagated over the overall framework to update all the parameters.

In summary, the proposed **TemGNN** can find a better combination of short- and long-term temporal-structural information and the static factors to predict the user credit risk.

4.2 Discussions (1) Generalizability: Although **TemGNN** is for credit risk prediction in this paper, it is general for the dynamic graphs. We do not pre-assume any distributions on the time interval between time steps. Furthermore, our model can find a better way to extract both the static and long- and short-term dynamic factors on the optimized target. Additionally, our method is inductive which can deal with the scenarios of the newly coming nodes and edges.

(2) Complexity: The time complexity of the model is $O(ITK \sum_{t=1}^{T} |E_t|D)$, where I denotes the number of epochs and D denotes the product of the dimension of two consecutive layers. Therefore, the time complexity is linear to the number of edges, which demonstrates the scalability of the model. Experimentally, our model is also available for the tens of million scenarios.

5 Experiments

5.1 Dataset We sample about 5 million Alipay users and use their lending behaviour from Jan 1st 2017 to Dec 31th 2019. The behaviour data shows the information of each loan, like the amount, the balance, the status, the purpose of the loan, the time to borrow the loan and so on. Each lending event is regarded as a snapshot. Within each snapshot, we build a graph that each edge denoting that the user lends the money from the institution. Each node has the node features about the loan information. By collecting multiple lending events of users and ordering the events by the lending time, we can build the dynamic graphs. Furthermore, users also have some stable profiles like marriage status, industry and some other financial-related attributes in our platform. These features are regarded as the static features. Note that all above data are definitely authorized by the users since these users hope to apply for loan in our platform and they should provide their lending history and personal profiles.

To conduct the experiment, we collect the users' default labels every month during the two years. About 2% of the users are default in the dataset. Our task is to predict the default labels via the dynamic graph. The graphs consists of 5 million nodes and 30 million edges. We use 70% of samples as the training set, 10% as the validation set and the rest as the test set. In each snapshot, the node in each snapshot is represented by a 138-dimension vector. And the node's static features have the dimension of 415.

5.2 Baselines We compare our method with the following baselines:

(1) Xgboost (AVG/CUR) [5]: Xgboost concats the user features and static features as the input. Here AVG means to average the user features of recent T snapshots. CUR means only using the user features of the most recent snapshot to make predictions.

(2) GraphSAGE/GAT (AVG/CUR) [27]: GraphSAGE and GAT are both a kind of GNN. GAT will do neighbor attention when aggregating the neighbors. AVG here means to average the user features and aggregate the graph topology on the recent T snapshots.

(3) LSTM [11]: LSTM models the time-ordered temporal sequence of the user features. The performance of GRU is similar with LSTM. Therefore, we only report the result of LSTM here.

(4) ST-GCN [31]: The model consists of both the temporal convolution and the structural convolution to deal with the dynamic graphs.

(5) GCN-LSTM [18]: It is a popular and recent method to learn from temporal graphs, which uses GCN to learn the embedding of each snapshot and LSTM to learn the temporal information. The output embedding of the last LSTM unit is used to predict the target.

(6) **TemGNN**$_{mix}$: A simple version of our method which removes the static model and moves the static features into the user features of each snapshot.

In summary, Xgboost only models the user's individual features. GraphSAGE and GAT models structural information. LSTM only models the temporal information. The rest methods all model both the structural and temporal information together.

Methods	T=5		T=10		T=20	
	KS	AUC	KS	AUC	KS	AUC
XGboost (AVG)	0.410	0.767	0.390	0.760	0.377	0.752
XGboost (CUR)	0.406	0.766	0.406	0.766	0.406	0.766
GraphSAGE (AVG)	0.439	0.783	0.442	0.785	0.442	0.785
GraphSAGE (CUR)	0.422	0.768	0.422	0.768	0.422	0.768
GAT (AVG)	0.442	0.785	0.444	0.786	0.446	0.786
GAT (CUR)	0.427	0.770	0.427	0.770	0.427	0.770
LSTM	0.444	0.786	0.450	0.790	0.451	0.790
ST-GCN	0.462	0.790	0.464	0.793	0.464	0.793
GCN-LSTM	0.458	0.789	0.465	0.793	0.467	0.794
TemGNN$_{mix}$	0.486	0.807	0.492	0.811	0.496	0.813
TemGNN	**0.494**	**0.812**	**0.501**	**0.816**	**0.506**	**0.819**

Table 1: The performance of different methods on the task of default prediction using the user's lending behaviours. We use different number of T to report the AUC and KS.

5.3 Experiment Setting For Xgboost, we use 500 trees to train the tree-based model. For GraphSAGE, GAT and the GNN components in GCN-LSTM and **TemGNN**$_{mix}$, we use two layers of graph convolution module with 128 and 32 units. And we use the max-pooling for GraphSAGE. For LSTM and the LSTM component in GCN-LSTM and **TemGNN**$_{mix}$, we use a one-layer LSTM with 64 output units. And the MLP structure on top of LSTM is 32-16. For ST-GCN, the temporal kernel size is set to 6. For **TemGNN**, the MLP structure for the static model is set to 128-64. We use 3 epochs for **TemGNN** to learn the whole model. We use the Adam optimizer for mini-batch training and the learning rate is set to 0.0001. We run all the experiments for 3 times and report the averaged results. We use KS (Kolmogorov-Smirnov statistics [20]) and AUC as the evaluation metrics. KS is a very popular metric used in financial areas to denote the risk differentiation ability of the model.

5.4 Overall Results We first change the sequence length T from 5 to 20 and report the AUC and KS of different methods in Table 1.

Among all the baselines, GCN-LSTM and ST-GCN are competitive, demonstrating the necessity of incorporating the structural and temporal information to predict the credit risk from lending behaviours. Additionally, our method of **TemGNN** and **TemGNN**$_{mix}$ outperform GCN-LSTM and ST-GCN. Specifically, our method improves KS by 10% and improves AUC by 5% compared with GCN-LSTM. The reason for smaller improvement of AUC is that the basic AUC value is large. When translated into the remaining space, the AUC is increased by 15%. The performance improvement demonstrates the effectiveness of the proposed model to capture the short-term and long-term temporal information with irregular intervals. Furthermore, we have the following observations and analysis.

Effects of the structural information: We find that on all the evaluations **TemGNN**, ST-GCN and GCN-LSTM outperforms LSTM. Meanwhile, GAT and GraphSAGE outperforms XGboost. It demonstrates that modelling the relations between the user and the lending institutions can enrich the user information and thus helps to define a user's credit risk better. Additionally, the result that GAT outperforms GraphSAGE demonstrates that it is essential to do structural attention when aggregating the neighbor nodes.

Effects of the temporal information: We find that **TemGNN** outperforms GAT and GraphSAGE, meanwhile LSTM outperforms XGboost on all evaluations. It demonstrates the necessity to model the temporal information.

But we find that when the sequence length increases, which means that we can obtain more temporal information, the performance of different methods does not necessarily increase: the performance of XGboost (AVG) quickly drops while the time-series model can improve. It indicates that it is important to use a sophisticated model to capture the temporal information. Furthermore, when the sequence length increases, the improvement of **TemGNN** is much larger than that of LSTM. It demonstrates that the proposed temporal attention mechanism with the interval decay can well extract more useful information from time-ordered events, even for the very remote lending events.

5.5 Further Discussions Now we compare **TemGNN** with some of its variants by removing different components to give further discussions.

Effects of the special graph convolution module: To capture the short-term temporal and structural information, we propose a special graph convolution module in this paper, which will gather the information from the neighbors of both the current snapshot and a few recent snapshots. To see how the module affects the performance, we change the value of K and report the performance. Note that when $K = 1$, it is the basic graph convolution operation which only aggregates the neighbors' information of the current snapshot. The result is shown in Figure 4.

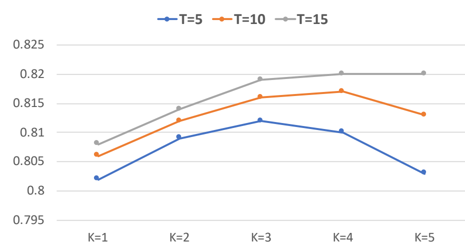


Figure 4: The AUC of **TemGNN** when changing the value of K.

We find that when K increases at the early time, the AUC on all the evaluations will increase. It demonstrates the necessity to model the short-term temporal information from neighbors when aggregating the neighbor nodes. However, when K continuously increases, the performance will decrease. The reason may be that simply aggregating the long-term temporal snapshots with irregular intervals will be detrimental.

Effects of the interval-decayed temporal attention: To aggregate the long-term temporal information from time-ordered lending events, we propose interval-decayed temporal attention. Here we give some analysis regarding this component. We use **TemGNN**$_{last}$ to denote our method which only utilizes the last unit of the LSTM to make prediction. We use **TemGNN**$_{nodecay}$ to denote our method without interval-decayed function in Eq. 4.8. We use **TemGNN**$_{notem}$ to denote our method without the full interval-decayed temporal attention. Due to the limit of space, we only report the AUC and KS for the sequence length of 10.

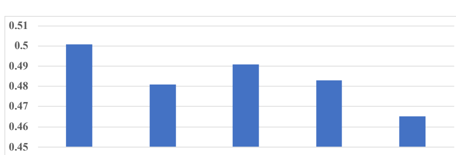


Figure 5: The KS of different variants of our proposed method on default prediction when $T = 10$.

From Figure 5, we find that **TemGNN** outperforms **TemGNN**$_{nodecay}$ and **TemGNN**$_{notem}$. It demonstrates the effectiveness of the designed interval-decayed attention mechanism. Specifically, the result that **TemGNN** outperforms **TemGNN**$_{last}$ demonstrates that each unit of the GNN-embedded LSTM contains useful short-term temporal-structural information. And the proposed interval-decayed temporal attention is a sophisticated way to combine the short-term and long-term information. Interestingly, although **TemGNN**$_{notem}$ also utilizes all the snapshots of information, its performance is almost the same with the performance of **TemGNN**$_{last}$. The reason is that simply concatenating the output embedding of each LSTM unit is not a good way to preserve the long-term temporal information. Furthermore, we find that **TemGNN**$_{last}$ achieves a substantial gain over GCN-LSTM. The main difference between these two methods is that **TemGNN**$_{last}$ uses our designed graph convolution modules to aggregate the neighbors' embeddings. Therefore, this result demonstrates that aggregating the temporal-near neighbor nodes is important to enrich the structural information.

6 Conclusion

In this paper, we propose a temporal-aware graph neural network to predict the user's credit risk. The problem is modelled on several time-ordered graph snapshots. We first propose a separate model to extract the static information. In each snapshot we propose a short-term graph encoder to capture the short-term temporal and structural information. We further propose an LSTM with interval-decayed attention to assemble the long-term temporal information and the static factors. We conduct the experiments by using the lending behaviours to predict the user's default behaviour. The results demonstrate the effectiveness of **TemGNN** compared with all the baselines.

The future work will focus on heterogeneous entities to form the heterogeneous network, which may further enrich user's information and achieve better results.

7 Acknowledgement

We would like to thank the support of many colleagues from AI lab and the Financial Risk Management in Ant Financial Services Group. Thanks for the support of the China Postdoctoral Science Foundation.

References

[1] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, *Patient subtyping via time-aware lstm networks*, in SIGKDD, ACM, 2017, pp. 65–74.

[2] L. Bermúdez, J. Pérez, M. Ayuso, E. Gómez, and F. Vázquez, *A bayesian dichotomous model with asymmetric link for fraud in insurance*, Insurance: Mathematics and Economics, 42 (2008), pp. 779–786.

[3] A. T. Carswell and D. C. Bachtel, *Mortgage fraud: A risk factor analysis of affected communities*, Crime, law and social change, (2009).

[4] J. Chen, X. Xu, Y. Wu, and H. Zheng, *Gc-lstm: Graph convolution embedded lstm for dynamic link prediction*, arXiv preprint arXiv:1812.04206, (2018).

[5] T. Chen and C. Guestrin, *Xgboost: A scalable tree boosting system*, in SIGKDD, ACM, 2016, pp. 785–794.

[6] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, *Risk guarantee prediction in networked-loans*.

[7] D. Cheng, Y. Zhang, F. Yang, Y. Tu, Z. Niu, and L. Zhang, *A dynamic default prediction framework for networked-guarantee loans*, in CIKM, 2019.

[8] A. Demirguc-Kunt, L. Klapper, and D. Singer, *Financial inclusion and inclusive growth: A review of recent empirical evidence*, The World Bank, 2017.

[9] L. Du, Y. Wang, G. Song, Z. Lu, and J. Wang, *Dynamic network embedding: An extended approach for skip-gram based network embedding.*, in IJCAI, 2018, pp. 2086–2092.

[10] 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, (2019).

[11] F. A. Gers, J. Schmidhuber, and F. Cummins, *Learning to forget: Continual prediction with lstm*, (1999).

[12] W. Hamilton, Z. Ying, and J. Leskovec, *Inductive representation learning on large graphs*, in NIPS, 2017, pp. 1024–1034.

[13] B. Hoogs, T. Kiehl, C. Lacomb, and D. Senturk, *A genetic algorithm approach to detecting temporal patterns indicative of financial statement fraud*, Intelligent Systems in Accounting, Finance & Management: International Journal, 15 (2007), pp. 41–56.

[14] R. A. Howard, *Information value theory*, IEEE Transactions on systems science and cybernetics, (1966).

[15] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi, *Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism*, AAAI, 2019.

[16] 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, 100 (2018), pp. 234–245.

[17] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, arXiv preprint arXiv:1609.02907, (2016).

[18] Y. Liu, X. Shi, L. Pierce, and X. Ren, *Characterizing and forecasting user engagement with in-app action graph: A case study of snapchat*, (2019).

[19] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, *Heterogeneous graph neural networks for malicious account detection*, in CIKM, ACM, 2018.

[20] F. J. Massey Jr, *The kolmogorov-smirnov test for goodness of fit*, Journal of the American statistical Association, 46 (1951), pp. 68–78.

[21] J. T. Quah and M. Sriganesh, *Real-time credit card fraud detection using computational intelligence*, Expert systems with applications, (2008).

[22] 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, 6 (2018), pp. 14277–14284.

[23] M. Sarma and J. Pais, *Financial inclusion and development*, Journal of international development, 23 (2011), pp. 613–628.

[24] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, *An attention enhanced graph convolutional lstm network for skeleton-based action recognition*, in CVPR, 2019, pp. 1227–1236.

[25] C. Song, Y. Lin, S. Guo, and H. Wan, *Spatial-temporal sychronous graph convolutional networks: A new framework for spatial-temporal network data forecasting*, 2020.

[26] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, *Dyrep: Learning representations over dynamic graphs*, (2018).

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, *Graph attention networks*, arXiv preprint arXiv:1710.10903, (2017).

[28] S. Viaene, M. Ayuso, M. Guillen, D. Van Gheel, and G. Dedene, *Strategies for detecting fraudulent claims in the automobile insurance industry*, European Journal of Operational Research, (2007).

[29] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams, *Transaction aggregation as a strategy for credit card fraud detection*, Data mining and knowledge discovery, 18 (2009), pp. 30–55.

[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, *A comprehensive survey on graph neural networks*, arXiv preprint arXiv:1901.00596, (2019).

[31] S. Yan, Y. Xiong, and D. Lin, *Spatial temporal graph convolutional networks for skeleton-based action recognition*, in AAAI, 2018.

[32] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, *Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction*, in AAAI, 2019.

[33] B. Yu, H. Yin, and Z. Zhu, *Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting*, arXiv preprint arXiv:1709.04875, (2017).

[34] D. Yue, X. Wu, Y. Wang, Y. Li, and C.-H. Chu, *A review of data mining-based financial fraud detection research*, in 2007 International Conference on Wireless Communications, Networking and Mobile Computing, Ieee, 2007, pp. 5519–5522.