TGFRAUD: Temporal Fraud Detection via Graph and Feature Represntation

Yen-Wen Lu, Cheng-Te Li

Abstract—Currently, most research on fraud detection of online platforms is strive for applying reviews or rating networks on identifying fraudsters. However, due to the rising importance of personal privacy, rating or review information may be unavailable for researcher to analyze user's veracity. Therefore, we propose TGFRAUD, an end-to-end model based on GraphSage and self-attention mechanism to capture regular pattern for fraudsters on temporal networks. Experimental results on three real-world datasets indicates that TGFRAUD performs same or even better when competing with other rating-based models no matter on transductive or inductive settings. Meanwhile, TGFRAUD gives sufficient explainability on the decision of fraud identification via inspecting rating burstiness in a series of rating history.

Index Terms—GraphSage, Temporal Networks, Inductive Learning, Fraud Detection

1 Introduction

As the rise of e-commerce platform such as Amazon, Yelp, and Bitcoin Alpha, consumption on these platforms are more active nowadays. However, different from brick-and-motar stores, users' purchasing decision heavily replies on rating or reviews given by other users, so there is huge monetary motivation for fraudsters to give unfair ratings and reviews to mislead users. [1] [2] Therefore, it is crucial for online platforms to identify and eliminate fraudulent users to provide the public comfortable purchasing experiences. Recent studies mostly utlize rating and user-product bipartite network to fullfill this fraud detection task. Nevertheless, in some situation, it is not allowed to obtain rating score of users due to personal privacy. In order to solve this challenging task, the goal of this paper is to use timestamp of rating activity(i.e. rating scores are unavailable, which is a data insufficient condition) and user-product bipartite network to identify fraudulent users.

Previous study has shown the effectiveness of feature-based approaches and graph-based approaches for classifying fraudulent and benign users. For feature-based approaches, they distinghish user via historical rating scrores and timestamps [3] [4] [5]. From rating scrores perspective, rating pattern of frauds tends to be bimodal [6] [7], which means they give extremely high or low rating values in most cases. Figure 1 reveals discriminable rating pattern between two group of users: Fraudulent users' ratings tend to be extreme while benign users' are more neutral. From time perspective, fake ratings are more bursty and regular [6] [7]. (i.e. fraudulent users usually give huge amount of ratings in a short time period or in certain time.) As shown in Figure 2, time span of Fraudulent ratings distrbutes in seconds in a large proportion, while most benign ratings distrbute in days, which indicates rapid property of fraudulent ratings. Hence, previous studies most design fraud detection model based on these two perspectives. For graph-based approaches, recent studies exploit rating user-product bipartite networks, whose edge weights are assigned as rating values. these kind of approaches gain remarkable success due to its effective utilizing of neighbor information charateristics.

Since our goal is to achieve competent results compared to previous state of art models without rating score information, we design our model based on these three ideas: 1) Bustiness property has to be utterly used in this task. 2) To extract neighbor nodes information in a network, we design a GNN-based method to take advatage of network's nature. 3) Our design of GNN model does not only gain aggregated information from neighbor nodes, but also consider edges since edges possess more information than nodes in this task.

Hence, our model incorporates three modules: 1) a modified GraphSage-based module [8] [9] whose aggregator is engaged in collecting both edge and neighbor information. 2) a self attention based [10] module which addresses time-series data through offering attentions on bursty time intervals. 3) a hybrid module which copes with both embeddings from network and from time series.

To test its ability on fraud detection in rating score unobtainable condition, we conduct four experiments on three real-world network data. First, in transductive and inductive settings, we respectively compare performance with other models by means of using AUC as metric. Second, For proving explainability of our model, attention from self attention based module is extracted and visualized to find the relation with fraudulent bursty behvior. Third, ablation experiment is conducted to show indispensability of each module. Last, t-SNE dimension reduction method is exerted on to show clustering in different datasets. Results indicates that our model achieves the best performance in most transductive and inductive settings. Simultaneously, our model is able to provide proper explanation for this binary classification task as well.

Overall, we make the following contributions:

1) We propose a model which successfully detects fraud users under the rating score data unvailable condition.
2) Our model works well on both transductive and

inductive settings.

3) Our model reveals robust explanability via learning specific behavioral pattern of fraudulent users on temporal perspective.

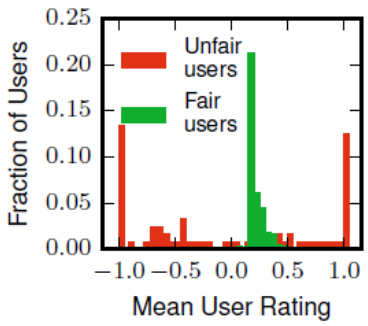


Fig. 1: Mean Rating chart. This chart is from REV2 [7] and displays each user's mean rating in OTC dataset.

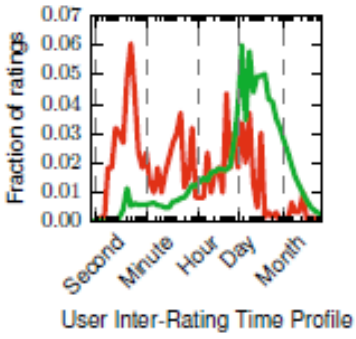


Fig. 2: Rating Burstiness Chart. This chart is from REV2 [7] and shows each user's rating temporal interval in OTC dataset.

2 Related Work

Existing studies in fraud detection can be generally categorized into two methods: 1) feature-based methods and 2) graph-based methods. As our method is extended from GraphSage [9], related work are focus on graph-based methods more.

2.1 Feature-Based Methods

Feature-based methods applies behavioral attributes to product user features, which can be derived from attributes such as review texts [11] [12] rating [3] and timestamps [4] [5]. For example, BIRDNEST [3] utilizes rating as input, implements Bayesian Inference for rating behavior and creates a likelihood-based metric for measuring a user's deviation from others. SpEagle [13], an iterative, propagation-based algorithm which is extended from FraudEagle [14] incorporates text, timestamp and rating as features to spot suspicious users and reviews.

2.2 Graph-Based Methods

Graph-based methods are able to be broadly classified into non graph neural networks(non-GNN) based and graph neural networks(GNN) based approaches. For non-GNN based approaches, Rev2 [7] formulates three metrics to iteratively evaluate fairness of a user, reliability of a rating and goodness of a product in a rating network. This algorithm achieves great success on real-world application. As recent rapid development of GNN, relevent application on network dataset is one of the research trend. GCN [15], GAT [16] and GraphSage [9] are the most widely-used GNN framework so far. Relational graph convolutional network (R-GCN) [17], signed graph convolutional network (SGCN) [18] and GCNEXT [19] modified from GCN take rating networks as input graph to complete fraud detection task. R-GCN first classifies edges as different types to build multiple subgraphs with same types of edges and then address subgraph with GCN to identify fraudsters. SGCN conducts balance theory [20] [21] to aggregate information from "friends nodes" which can be attained through signed edges. GCNEXT improves on SGCN via implementation on directed rating edges. For GAT, SemiGNN [22] uses hierarchical attentive graph neural network to apply semi supervised learning on financial fraud detection. CARE-GNN [23] utilizes attention aggregation for both intra-relation of node level and inter-relation of graph level. For GraphSage, GraphConsis [24] filters out inconsistent neighbors via consistency score to fulfill neighbor sampling and implements relation attenion aggregation to tackle relation inconsistency. xFraud [25] combines GNN and self-attention mechanism [10] to do fraudulent transaction detection task on heterogeneous transaction graph from real-world e-commerce platform.

From previous research, it can be seen that GNN-based methods on fraud detection are widely-used and effective. However, there is no research focus on temporal network to find fraud users. Therefore, we try to design a model to reach equal or better state of art permformance under this data insufficient condition.

3 Preliminaries

We first consider user-product interaction as a bipartite undirected and weighted graph $\mathfrak{G} = (U, T, P)$ where $u \in U$, and $p \in P$ represent user nodes and products nodes respectively and $t \in T$ denotes edges with timestamps as weights. In order to gain more information from timestamps, an undirected graph $\mathfrak{G} = (U, E, P)$ is presented, where $e \in E$ is a multi-dimentional vector instead of a weight:

$$e = [year,\ month,\ day,\ hour,\ timestamp \\ \Delta t_{user}, \Delta t_{product}] \tag{1}$$

where $year$, $month$, day, and $hour$ are derived from $timestamp$. For time span Δt, it is defined as $(timestamp_i - timestamp_{i-1})/3600$, where i is the index of time series of a user's rating history. For node initialization of U and P, we simply adopt the count of neighbor nodes. This undirected graph with multi-dimentional edge vector will be used for our GNN-based fraud detection model.

4 METHOD

In this section, we introduce our proposed model, which contains three modules shown in Figure 3. EdgeSage is for aggregating neighbor and edge level information. Temporal encoder is for capturing burstiness in temporal series rating history. Temporal-edge aggregator generates node embeddings possessing both neighbor and temporal series information. Each module will be described in details in the following subsections.

4.1 EdgeSage

Inspired by GRAPE [8], a GraphSage-based model originally tackling edge information to accomplish feature imputation, we modify it as our EdgeSage module since edges poccess more information than nodes for edges are multi-dimensional vectors based on our graph $\mathfrak{G}$ in section 3. At each EdgeSage layer l, the message passing function is as follows:

$$\hat{h}_u^l = \sigma(W_{mess}^l \cdot (\hat{h}_u^{l-1} || e_{uv}^{l-1}) + b_{mess}^l),\ \forall u \in \mathbb{N}(v) \tag{2}$$

where source node embedding $\hat{h}_u^{l-1}$ and edge embedding e_{uv}^{l-1} are concatenated first Then it multiplies a trainable weight W_{mess}^l and adds bias b_{mess}^l to generate intermediate source node embedding $\hat{h}_u^l$ afterwards. $\mathbb{N}(v)$ represents neighbors of target node. After message passing, updated source node $\hat{h}_u^l$ are aggregated to target nodes $\hat{h}_v^l$ as:

$$\begin{aligned} \hat{h}_v^l &= W_{AGG}^l \cdot (AGG^l(\hat{h}_u^l)) + b_{AGG} \\ h_v^l &= GRU(\hat{h}_v^l, h_v^{l-1}) \end{aligned} \tag{3}$$

Detail of GRU is formalized as,

$$\begin{aligned} z_v^l &= \sigma(W_z \cdot h_v^{l-1} + U_z \cdot \hat{h}_v^l + \beta_z) \\ r_v^l &= \sigma(W_r \cdot h_v^{l-1} + U_r \cdot \hat{h}_v^l + \beta_r) \\ \tilde{h}_v^l &= tanh(W_h \cdot h_v^{l-1} + U_h(r_i^t \odot \hat{h}_v^l) + \beta_h) \\ h_v^l &= \tilde{h}_i^t \odot z_i^t + (1 - z_i^t) \end{aligned} \tag{4}$$

where AGG^l is the aggregation function, W_{AGG}^l and b_{AGG}^l are trainable weight and bias and GRU is Gated Recurrent Unit function, in which W_z, W_r, W_h, U_z, U_r, $U_h \in \mathbb{R}^{1 \cdot d}$ are trainable weights. b_z, b_r, $b_h \in \mathbb{R}^{1 \cdot d}$ are trainable bias. z_l^t is update gate and r_v^l is reset gate. Once AGG^l aggregates information of neighbors and corresponding edges, GRU updates target nodes h_v^l through update gate and reset gate. In addition, edge embedding e_{uv}^l are updated via concatenation of source node, edge and target node as follows:

$$e_{uv}^l = \sigma(W_{edge}^l \cdot (h_u^{l-1} || e_{uv}^{l-1} || h_v^{l-1}) + b_{edge}^l) \tag{5}$$

where W_{edge} and b_{edge} is trainable parameters and $e_{uv}^l \in \mathbb{R}^{1 \cdot h}$.

4.2 Temporal Encoder

REV2 [7] indicates that fraudulent behavior is bursty, which means that they give large amount of rating in a short time period. Therefore, we propose temporal encoder, a self attention module [10] to learn embedding of rating activites through adapted attention on time span series. For a user $u \in U$, temporal gap series is denoted as $\Delta T \in \mathbb{R}^N$. ΔT is utilized as input of Temporal Encoder as following:

$$\begin{aligned} &Q_i = ReLU(W_i^Q \cdot tanh(\Delta T_i^{-1})),\ \forall Q_i \in R^{N \cdot d}, \\ &K_i = ReLU(W_i^K \cdot tanh(\Delta T_i^{-1}),\ \forall K_i \in R^{N \cdot d}, \\ &V_i = ReLU(W_i^V \cdot \Delta T_i),\ \forall V_i \in R^{N \cdot d}, \\ &\Delta \hat{T}_i = Normalized(Attention(Q_i, K_i, V_i)),\ \forall \Delta \hat{T}_i \in \mathbb{R}^{N \cdot d}, \\ &Attention(Q_i, K_i, V_i) = softmax(\frac{Q_i K_i^\top}{\sqrt{d}}) V_i \end{aligned} \tag{6}$$

where query Q_i and key K_i are for the purpose of gaining much attention on rating activity with shorter time span, so ΔT is inverted before into activation function $tanh$ and multiplying trainable weight W_i^Q and W_i^K. N is the count of rating. Finally, we take the average of temporal differences series embedding $\Delta \hat{T}_i$ to obtaining a temporal burst embedding $\Delta \tilde{T}_i$:

$$\Delta \tilde{T}_i = mean(\Delta \hat{T}_i), \forall \Delta \tilde{T}_i, \in \mathbb{R}^d \tag{7}$$

4.3 Temporal-Edge aggregator

From EdgeSage and Temporal Encoder, we individually obtain graph node embedding gathering information from neighbors and temporal burst embedding from users' own rating history. Nevertheless, we would like to generate embedding possessing both graph and temporal information. For this purpose, we propose Temporal-Edge Aggregator, a module employing edge embedding from EdgeSage and inverse time span from Temporal Encoder to produce hybrid embedding, which is as follows:

$$\begin{aligned} &\bar{T}_i = ReLU(W_{temp}(tanh(\Delta T_i^{-1}))), \forall \bar{T}_i \in \mathbb{R}^{N \cdot d}, \\ &\bar{E}_u = ReLU(W_{edge}(E_u)),\ \forall e_{uv} \in E_u \ and\ \bar{E}_u \in \mathbb{R}^{N \cdot d}, \\ &\bar{te} = (mean(\bar{T}_i \,||\, \bar{E}_u),\ \forall i = u),\ \bar{te} \in \mathbb{R}^{2d}, \\ &\hat{te} = Normalized(sigmoid(W_{te}(\bar{te})))/N,\ \forall \hat{te} \in \mathbb{R}^k \end{aligned} \tag{8}$$

where $\bar{T}_i$ and $\bar{E}_u$ are updated embedding from temporal burst embedding $\bar{T}$ and edge embedding e_{uv}, E_u indicates all edge embeddings of one user u. We produce hybrid embedding through concatenating $\bar{T}_i$ and $\bar{E}_u$, averaging over N concatenated vectors and finally conducting normalization.

4.4 Final Prediction

This layer is for fusing embedding from EdgeSage , temporal encoder and temporal-edge aggregator and generate probability distribution of binary classification:

$$\begin{aligned} O_u &= (\tilde{T} \,||\, h_v \,||\, \hat{te}) \\ \hat{Y} &= softmax(MLP(O_u)) \end{aligned} \tag{9}$$

where O_u is final embedding for a user by concatenating three embedding from three modules, $\hat{Y} \in \mathbb{R}^2$ is the final prediction generated from an MLP layer. Our model is trained to minimize the cross-entropy error via loss function as,

$$Loss = -(\ ylog(\hat{y}_0) + (1 - y)log(\hat{y}_1)\ )$$

where y is labeled veracity and $\hat{y}_0,\ \hat{y}_1 \in [0, 1]$.

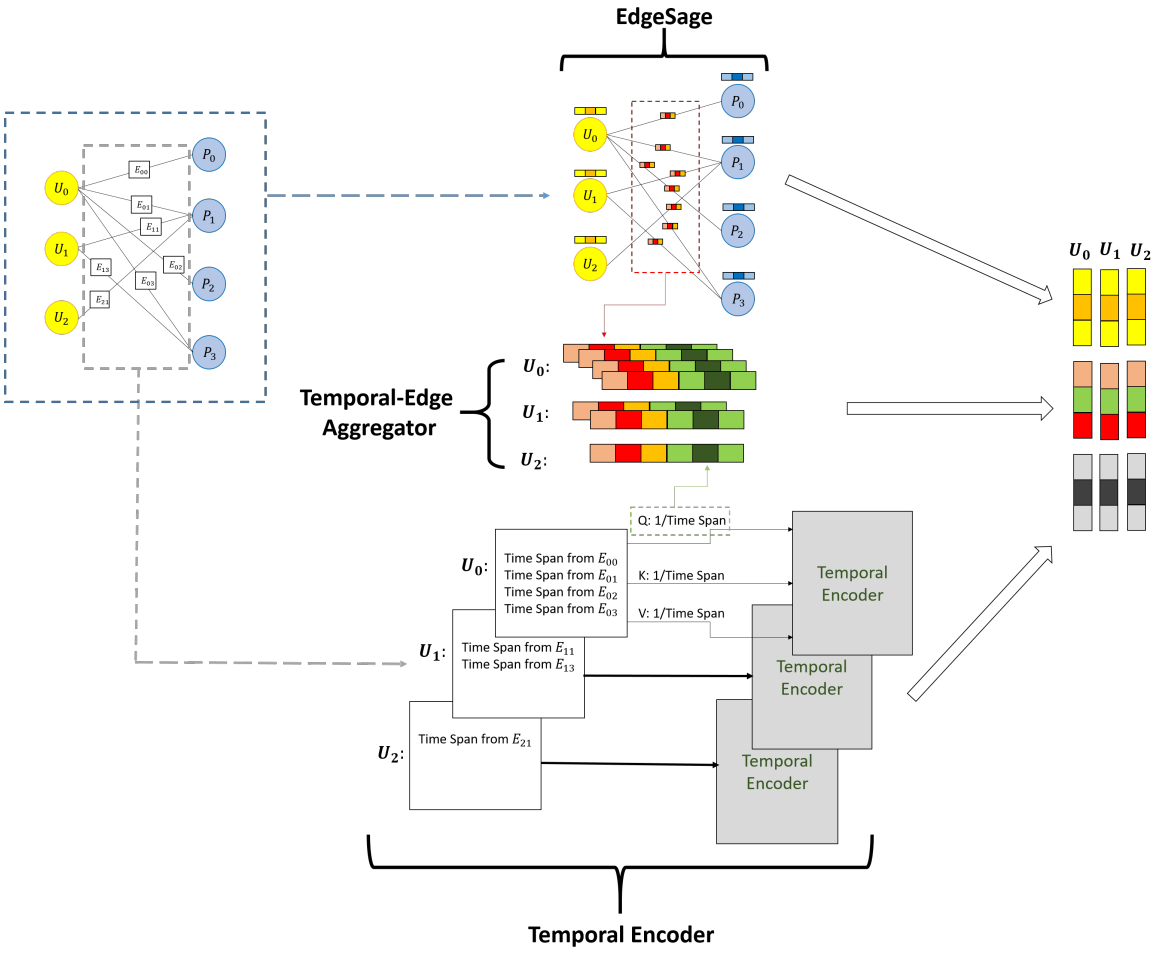


Fig. 3: Model

5 Experiments

In this section, we ask three questions and conduct corresponding experiments to show our models' contribution:

1) **Can our model achieve equal or better performance compared to state of art models on both transductive and inductive settings?**
2) **How is the effectiveness of each modules for final fraud detection?**
3) **Is our model able to provide explainability on identifying fraudulent users only based on temporal information?**

Dataset	OTC	ALPHA	AMAZON
Users	4814	5858	35592
Products	3286	3754	24186
Benign Users	136	138	2358
Fraudulent Users	180	102	241
Edges	35592	24186	560804

TABLE 1: Dateset for Experiment

5.1 Datasets

The datasets we adopt are three well-known datasets from Rev2 [7]: Bitcoin OTC, Bitcoin Alpha and Amazon. Detail of three datasets is shown in Table 1.

- **Bitcoin OTC** is a user-to-user trust network of Bitcoin users on OTC platform. Nodes are splitting into two types: 'rater' with outgoining edges and 'product' with all incoming edges. Edges contains information of ten-point scale rating score and timestamp. For ground truth label, benign users are defined as platform's founder and the users he rated highly positively (≥ 0.5) and fraudulent users are those whome benign users uniformly rated negatively (≤ 0.5).
- **Bitcoin Alpha** is similar to **Bitcoin OTC** but it is the Bitcoin network on Alpha platform.
- **Amazon** is a user-to-product rating network. Edges contains information of five-point scale rating score and timestamp. Ground truth label is defined by helpfulness votes. If the fraction of helpful-to-total votes is ≥ 0.75, user is labeled as benign and fraudulent if ≤ 0.25.

Note that 1) all rating values in datasets' network are between -1 and 1. 2) we do not use Epinion dataset since its edges do not incorporate timestamp information.

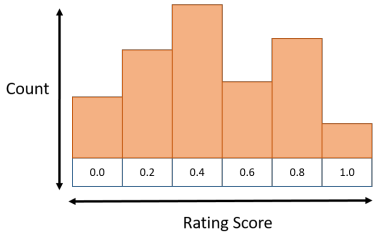
5.2 Hyperparameter Settings

For both dataset, number of training epochs is set as 400. We use Adam optimizer and set learning rate as 0.001. For each dimension of embedding, we assign dimension of node embedding $\hat{h}_u^l$ as 16, dimension of edge embedding e_{uv}^l as 8, dimension of temporal burst embedding $\Delta\hat{T}_i$ as 16 and dimension of hybrid embedding $\hat{te}$ as 16. For temporal encoder, each user's time span series is set as 150.

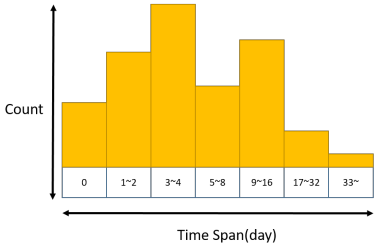
5.3 Performance Comparison

5.3.1 Baselines

In this section, we compare our model with REV2 and other four GNN based model. To show the effectiveness of model on temporal information, we address 1) rating and 2) timestamp as input respectively for other state of art models. 1) When rating as input, initial node embedding, which is shown as figure 4(a), is assigned as a vector whose entry is the count of certain rating value. For example, node embedding for AMAZON is a five dimentional vector as AMAZON is five-point scale rating and five dimentional vector for OTC and ALPHA as these two are ten-point scale rating. Edges weights are assigned as rating values. 2) As for timestamp as input, we follow the setting from SpEagle [13], which separates time span of into five groups: [0 day, $1-2$ days, $3-4$ days, $5-8$ days, $9-16$ days, $17-32$ days, $33-$ days], which is shown as figure 4(b). Node embedding is a seven dimensional vector whose features are the count of time span groups.



(a) Node Initialization by Rating Score



(b) Node Initialization by Time Span

Fig. 4: Node initialization. We show five-point scale rating score and time span as visualization example of node initialization

- **REV2** [7] employs rating to compute product quality, user fairness and rating reliability first and then identify whether the user is fraudulent or not based on these three scores.
- **SGCN** [18] is a GCN-based model which applies balance theory on GNN to achieve link sign prediction. Therefore, We add one final MLP embedding to node to fulfill binary classification task. For (1) rating as input, we simply set up positive and positive edges as rating > 0 or ≤ 0 for ten-point scale and > 0.4 or ≤ 0.4 for five-point scale. For (2) timestamp as input, we set up positive and positive edges as time gap > 4 or ≤ 4 days.
- **RGCN** [17] is a GCN-based model for networks with different types of edges. Hence, we adop two ways to catorgoize edges. For (1) rating as input, Edges are split into two classes(rating > 0 or ≤ 0 for ten-point scale or > 0.4 or ≤ 0.4 for five-point scale). For (2) timestamp as input, edges are split in two classes(time gap > 4 or ≤ 4 days) as well.
- **GAT** [16] is a spacial-based GNN model whose nodes embedding are obtained via attending over their neighborhood. different nodes in a neighborhood
- **GraphSAGE** [9] is an inductive method that implements an node aggregator to leverage neighbor node attribute information.

5.3.2 Comparison Analysis on Transductive Setting

We adopt AUC as metrics and compute average AUC score of stratified tenfold cross-validation for rating and temporal as input repectively. First we analyze result when rating as input. In OTC and ALPHA, almost all models outperforms REV2 (0.840) except RGCN in ALPHA gets slightly lower AUC(0.839). In AMAZON, probably becuase of sparsity of adjacency matrix, only RGCN performs better than REV2. Nevertheless, apart from GAT, REV2 only wins other GNN-based models by a small margin(0.16 over SGCN and 0.03 over GraphSage) in AMAZON, which indicates that GNN-based model with rating as edges is effective in fraud detection with bipartite network. Second, when timestamp as input, all performances of GNN-based model regress drastically comparing performances in rating as input, which may because these models is not able to use time difference as edges directly. As further analzing the extend of regression, GAT drops the most while GraphSage drops the least. The possible reason is that GAT generates edge weights from nodes' similarity rather than directly considering time span, so it may put unsuitable attention on neighbor nodes. As for GraphSage, since it directly trains an aggregator to integrate all neighbor information, it loses less neighbor information. From this perspective, our model takes the advantages of GraphSage and directly utilizes time span, so it reaches state of art performance in OTC and ALPHA even comparing with models using ratings. In AMAZON, though not performing as well as models using ratings, it's AUC is 5.8% higher than GraphSage's which is the second highest when employing time span as input.

5.3.3 Comparison Analysis on Inductive Setting

For inductive setting, we proportionally mask out both edges and corresponding nodes happening late as training data, while in the testing phase, the evaluation was conducted with complete network. Figure 5 shows the inductive learning on three dataset. First, GraphSage performs stable in both rating and temporal inputs, which indicates

Dataset	OTC		ALPHA		AMAZON	
	Rating	Temporal	Rating	Temporal	Rating	Temporal
REV2	0.895	NA	0.840	NA	0.854	NA
SGCN	0.957	0.933	0.882	0.847	0.840	0.649
RGCN	0.912	0.86	0.839	0.802	**0.865**	0.625
GAT	0.960	0.736	0.858	0.767	0.796	0.544
GraphSage	0.942	0.941	0.868	0.864	0.851	0.680
Ours	NA	**0.967**	NA	**0.902**	NA	0.738

TABLE 2: Average AUC Scores of Tenfold Cross-Validation for Rating and Temporal Input

its inductive learning ability. Second, SGCN performs the worst in rating as inputs, while it gets second best AUC score in OTC and ALPHA in transductive learning, which may indicate that SGCN is more fit in complete network. As for GAT, it performs the worst in Temporal inputs in both settings, which shows that it does not capable of using temporal information to fulfill this fraud detection task. As our model, when rating as input, our model's performance keeps in the top two places in OTC and ALPHA. When time as input, our model reveals best performance in OTC and AMAZON. As for ALPHA, our model's AUC score is only lower than GraphSage in 70% percentage of training data while our model keeps best performance in 60%, 80% and 90%, Therefore, the experiment indicates the effectiveness of our model on inductive setting.

5.3.4 Ablation Study

Table 3 reveals the influence of each components of our model. First, comparing **EdgeSage only** and **EdgeSage with GRU**, GRU module greatly improves AUC score in all three dataset. Possible reason is that GRU gate does not only keep residual information from last layer of EdgeSage, but also filter out noise through updating gate vector and reset gate vector. Second, **Temporal Encoder Only** proves that time difference highly related to fraudulent behavior. Further explanation of this relation will be introduced thoroughly in the next section. Third, as comparing **EdgeSage and Temporal Encoder** with **All(Sum Aggregator)**, it reveals the effect of edge-temporal aggregator. In addition, we also compare different kinds of aggregator in **EdgeSage** and the result indicates that **Sum Aggregator** has the best performance.

5.3.5 Explainability Analysis

This section will discuss our model's Explainability on identifying fraudulent users. Table 4 shows that temporal encoder gives much attention weight on shorter time gap, which means that model puts much attention on short time span of user's rating history. In addition, Comparing attention weight of one fraudulent and benign user, we are able to find that fraudulent user rates more frequently in some intervals while rateing of benign user is more divergent. In order to prove that fraudulent users temporal behavior tends to be more bursting than benign users'. We respectively overlaps attention of all fraudulent users and benign users as shown in Figure 6. From Figure 6(a), temporal encoder generates more large attention weights

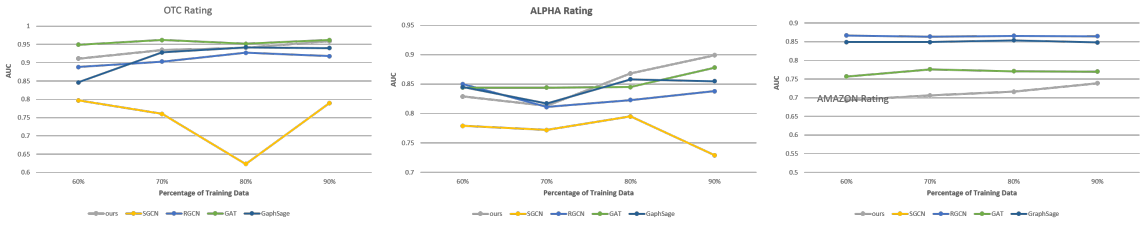
	OTC	ALPHA	AMAZON
EdgeSage only	0.554	0.58	0.601
EdgeSage with GRU	0.953	0.88	0.674
Temporal Encoder Only	0.925	0.82	0.674
EdgeSage and Temporal Encoder	0.966	0.899	0.734
All (Sum Aggregator)	0.967	0.902	0.738
All (Mean Aggregator)	0.960	0.884	0.719
All (Max Aggregator)	0.956	0.904	0.691

TABLE 3: Ablation Study

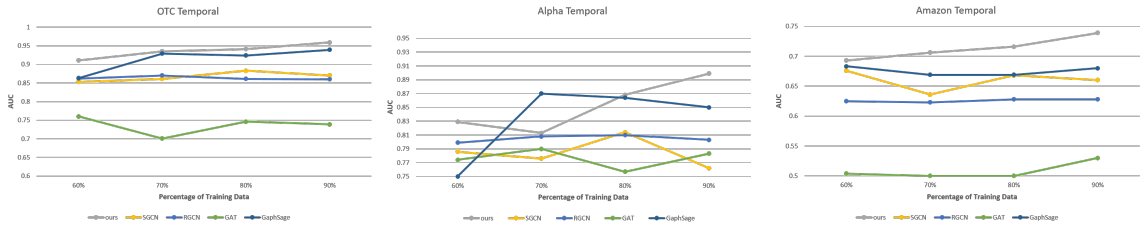
comparing to Figure 6(b). This infers that fraudulent users tend to give more ratings and most ratings are converged in short time intervals. Our model has the ability to capture such rating pattern and provide proper explanation.

5.3.6 t-SNE Visualization

This section investigates the effectiveness of the feature representation before MLP layer. We adopt t-SNE to apply dimension reduction on observing clustering of feature representation from each model on 2D space. Note that these clusterings corresponds to fraudulent and benign labels and we visualize embedding from timestamp as input only. As looking into Fig 7, Fig 7(a) and Fig 7(b) display discernible clustering in the projected 2D space while only SGCN on OTC Fig 7(d) and GAT on OTC Fig 7(j) displays comparable clustering, other models' t-SNE dimension reduction display does not provide clear clustering group, which prove our model's discriminative ability. In addition, this projection corresponds to the result of AUC score since our model performs the worst on AMAZON while clustering on AMAZON is the most ambiguous as well.



(a) OTC Rating (b) ALPHA Rating (c) AMAZON Rating



(d) OTC Temp (e) ALPHA Temp (f) AMAZON Temp

Fig. 5: Inductive Result on both rating and temporal inputs

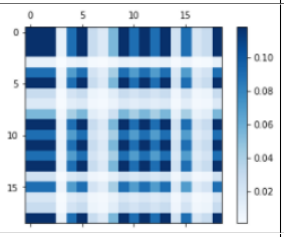
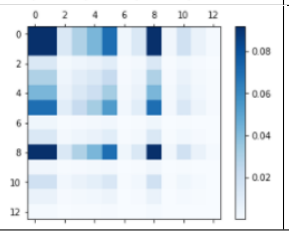
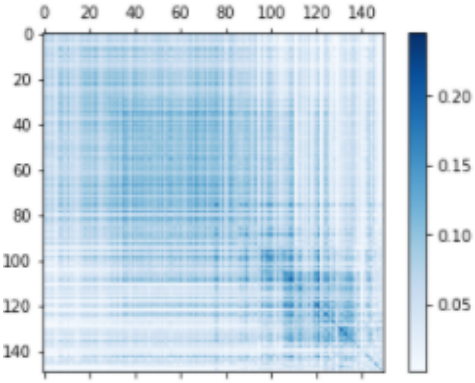
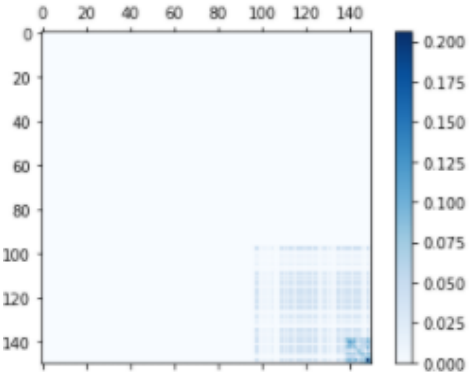
User	Attention Weight	Time Gap History (hour)
Fraudulent		[0, 0, 0, 12, 1, 0, 4, 7, 2, 0, 0, 0, 0, 0, 5, 1, 6, 4, 0]
Benign		[0, 0, 7, 3, 2, 1, 27, 7, 0, 28, 5, 15, 68]

TABLE 4: Attention Weight and Corresponding Time Gap from ALPHA



(a) Attention Overlapping of all Fraudulent Users



(b) Attention Overlapping of all Benign Users

Fig. 6: Attention Visualization Overlapping for ALPHA

6 Conclusion

In conclusion, our work TGFRAUD effectively implements GraphSage extended module to capture useful user and edge information, exploits self-attention module to find bursty behavioral pattern of fraudulent users and incorporates both module properly to achieve fraud detection task on temporal network. Experiments reveals brilliant results in transductive and stable performacne in inductive setting. In addition, our model also provide explainable process during fraudulent identification. In dimension reduction experiment, discernable clustering on 2D space show feature representation from TGFRAUD is indicative for this downstream task.

For our future work, we may adopt advanced approaches to filter out some nodes or since these nodes can be regard as noise to undermine final prediction. Hence, we will be focus on node sampling to further improve our model.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

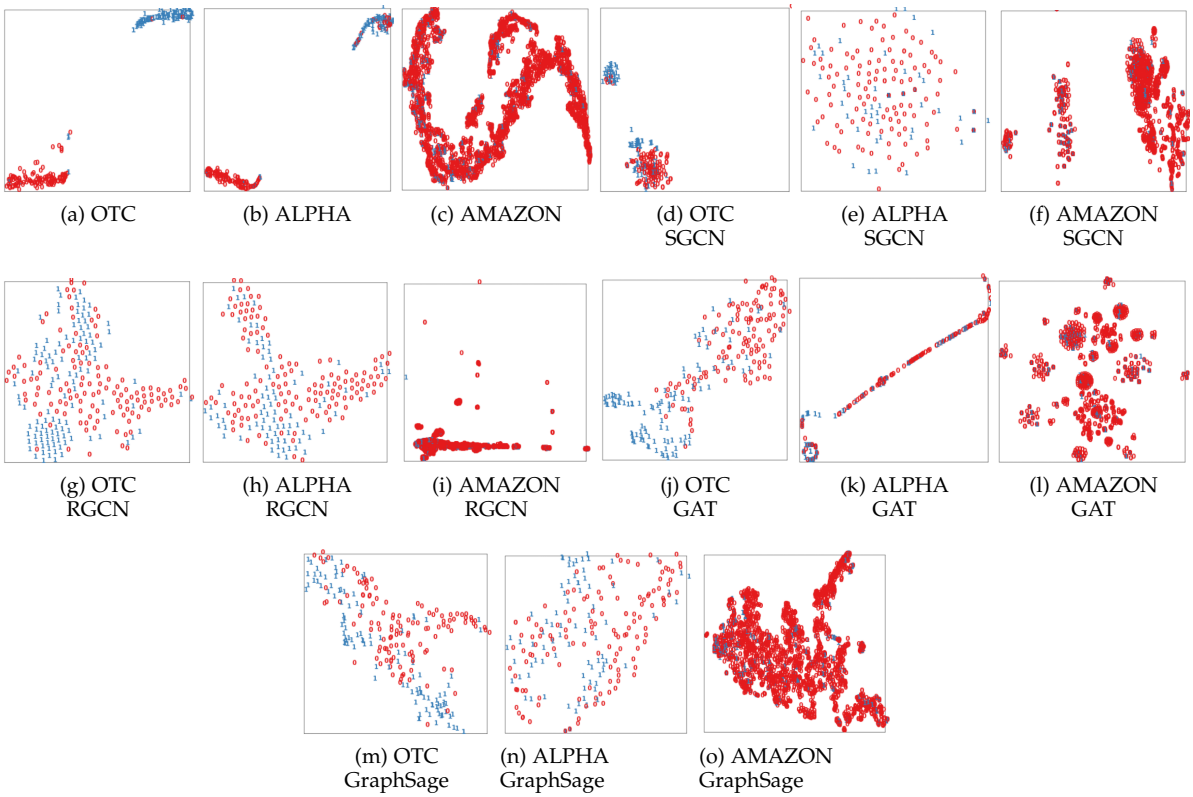


Fig. 7: t-SNE Visualization for timestamp as input only

ACKNOWLEDGMENTS

The authors would like to thank...

REFERENCES

[1] B. Viswanath, M. A. Bashir, M. B. Zafar, S. Bouget, S. Guha, K. P. Gummadi, A. Kate, and A. Mislove, "Strength in numbers: Robust tamper detection in crowd computations," in *Proceedings of the 2015 ACM on Conference on Online Social Networks*, ser. COSN '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 113–124. [Online]. Available: https://doi.org/10.1145/2817946.2817964

[2] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove, "Towards detecting anomalous user behavior in online social networks," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 223–238. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/viswanath

[3] B. Hooi, N. Shah, A. Beutel, S. Gunnemann, L. Akoglu, M. Kumar, D. Makhija, and C. Faloutsos, "Birdnest: Bayesian inference for ratings-fraud detection," 2016.

[4] S. Xie, G. Wang, S. Lin, and P. Yu, "Review spam detection via temporal pattern discovery," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2012.

[5] A. J. Minnich, N. Chavoshi, A. Mueen, S. Luan, and M. Faloutsos, "Trueview: Harnessing the power of multiple review sites," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015, p. 787–797. [Online]. Available: https://doi.org/10.1145/2736277.2741655

[6] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao, "Bimodal distribution and co-bursting in review spam detection," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 1063–1072. [Online]. Available: https://doi.org/10.1145/3038912.3052582

[7] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "Rev2: Fraudulent user prediction in rating platforms," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, ser. WSDM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 333–341. [Online]. Available: https://doi.org/10.1145/3159652.3159729

[8] J. You, X. Ma, D. Y. Ding, M. Kochenderfer, and J. Leskovec, "Handling missing data with graph representation learning," 2020.

[9] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2018.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[11] A. Fayazi, K. Lee, J. Caverlee, and A. Squicciarini, "Uncovering crowdsourced manipulation of online reviews," 08 2015, pp. 233–242.

[12] V. Sandulescu and M. Ester, "Detecting singleton review spammers using semantic similarity," *Proceedings of the 24th International Conference on World Wide Web*, May 2015. [Online]. Available: http://dx.doi.org/10.1145/2740908.2742570

[13] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 985–994. [Online]. Available: https://doi.org/10.1145/2783258.2783370

[14] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, pp. 2–11, 01 2013.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.

[17] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," 2017.

[18] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional network," 2018.

[19] W. Kudo, M. Nishiguchi, and F. Toriumi, "Gcnext: graph convolutional network with expanded balance theory for fraudulent user detection," *Social Network Analysis and Mining*, vol. 10, 12 2020.

[20] F. Heider, "Attitudes and cognitive organization," *The Journal of Psychology*, vol. 21, no. 1, pp. 107–112, 1946, pMID: 21010780. [Online]. Available: https://doi.org/10.1080/00223980.1946.9917275

[21] "Heider's theory of balance:," *Human Relations*, vol. 21, no. 2, pp. 177–210, 1968. [Online]. Available: https://doi.org/10.1177/001872676802100205

[22] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 598–607.

[23] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," *Proceedings of the 29th ACM International Conference on Information Knowledge Management*, Oct 2020. [Online]. Available: http://dx.doi.org/10.1145/3340531.3411903

[24] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, "Alleviating the inconsistency problem of applying graph neural network to fraud detection," in *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[25] S. X. Rao, S. Zhang, Z. Han, Z. Zhang, W. Min, Z. Chen, Y. Shan, Y. Zhao, and C. Zhang, "xfraud: Explainable fraud transaction detection on heterogeneous graphs," 2020.