IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Fraud Detection in Online Product Review Systems via Heterogeneous Graph Transformer

SONGKAI TANG[1] (Graduate Student Member, IEEE), LUHUA JIN[1] (Graduate Student Member, IEEE) , AND FAN CHENG[1] (Member, IEEE)

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Fan Cheng (chengfan@sjtu.edu.cn).

This work was supported in part by the NSFC under Grant 61701304, in part by the Shanghai Sailing Program under Grant 17YF1410100, and in part by the Program for Professor of Special Appointment, Eastern Scholar at Shanghai Institutions of Higher Learning.

ABSTRACT In online product review systems, users are allowed to submit reviews about their purchased items or services. However, fake reviews posted by fraudulent users often mislead consumers and bring losses to enterprises. Traditional fraud detection algorithm mainly utilizes rule-based methods, which is insufficient for the rich user interactions and graph-structured data. In recent years, graph-based methods have been proposed to handle this situation, but few prior works have noticed the camouflage fraudster's behavior and inconsistency heterogeneous nature. Existing methods have either not addressed these two problems or only partially, which results in poor performance. Alternatively, we propose a new model named Fraud Aware Heterogeneous Graph Transformer(FAHGT), to address camouflages and inconsistency problems in a unified manner. FAHGT adopts a type-aware feature mapping mechanism to handle heterogeneous graph data, then implementing various relation scoring methods to alleviate inconsistency and discover camouflage. Finally, the neighbors' features are aggregated together to build an informative representation. Experimental results on different types of real-world datasets demonstrate that FAHGT outperforms the state-of-the-art baselines.

INDEX TERMS Fraud Detection, Graph Neural Network, Data Mining

I. INTRODUCTION

Internet services have brought human beings with e-commerce, social networking, and entertainment platforms, which not only facilitate information exchange but also provide chances to fraudsters. Fraudsters disguise themselves as ordinary users to publish spam information [1] or collect user privacy, compromising the interest of both platforms and users. In addition, multiple entities on the Internet are connected with multiple relationships. Traditional machine learning algorithms cannot handle this complicated heterogeneous graph data well. The current approach is to model the data as a heterogeneous information network so that similarities in characteristics and structure of fraudsters can be discovered. Due to the effectiveness in learning the graph representation, graph neural networks (GNNs) have already been introduced into fraud detection areas including product review [2]–[5], mobile application distribution [6], cyber crime identification [7] and financial services [8], [9]. However, most existing GNN based solutions just directly apply homogeneous GNNs, ignoring the underlying heterogeneous graph nature and camouflage node behaviors. This problem has drawn great attention with many solutions proposed [4], [5], [10]. GraphConsis [4] found that there are three inconsistency problems in fraud detection and CAREGNN [5] further proposed two camouflage behaviors. These problems could be summarized as follows:

- **Camouflage:** Previous work showed that crowd workers could adjust their behavior to alleviate their suspicion via connecting to benign entities like connecting to highly reputable users, disguise fraudulent URLs with special characters [3], [6], or generate domain-independent fake reviews via generative language model [11] to conceal their suspicious activities.
- **Inconsistency:** Two users with distinct interests could be connected via reviewing a common product such as food or movies. Direct aggregation makes GNNs hardly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3084924, IEEE Access

distinguish the unique semantic user pattern. Also, if a user is suspicious, then the other one should be more likely to be distrustful if they are connected by common-activity relation since fraudulent users tend to post many fraudulent reviews in the same short period.

To address the above two problems, many methods have been proposed. GraphConsis addresses the inconsistency problem by computing the similarity score between node embeddings, which cannot distinguish nodes with different types. CAREGNN enhances GNN-based fraud detectors against camouflaged fraudsters by reinforcement learning-based neighbor selector and relation aware aggregator. Its performance still suffers from the heterogeneous graph. In this paper, we introduce the Fraud Aware Heterogeneous Graph Transformer(FAHGT), where we propose heterogeneous mutual attention to address the inconsistency problem and design a label-aware neighbor selector to solve the camouflage problem. Both are implemented in a unified manner called the “score head mechanism”. We demonstrate the effectiveness and efficiency of FAHGT on many real-world datasets. Experimental results suggest that FAHGT can significantly improve KS and AUC over state-of-the-art GNNs as well as GNN-based fraud detectors.

The advantages of FAHGT can be summarized as follows:

- **Heterogeneity:** FAHGT is able to handle heterogeneous graphs with multi-relation and multi-node type without designing meta-path manually.
- **Adaptability:** FAHGT attentively selects neighbors given a noise graph from real-world data. The selected neighbors are either informative for feature aggregation or risky for fraud detection.
- **Efficiency:** FAHGT admits a low computational complexity via a parallelizable multi-head mechanism in relation scoring and feature aggregation.
- **Flexibility:** FAHGT injects domain knowledge by introducing a flexible relation scoring mechanism. The score of a relation connecting two nodes not only comes from direct feature interaction but is also constrained by domain knowledge.

II. RELATED WORK

A. GRAPH NEURAL NETWORKS

The Graph Neural Network is a generalization of CNN to graphs [12]. The initial graph convolution idea in the spectral domain is inspired by the Fourier transformation in signal processing [13]. Then, ChebNet [14] and GCN [15] are proposed to improve efficiency by using approximation. For GNNs on spatial domain, GraphSAGE [16] samples a tree rooted at each node and computes the root’s hidden representation by hierarchically aggregating hidden node representations from the bottom to top. GAT [17] further proposes to learn in the spatial domain by computing different importance of neighbor nodes via the masked self-attention mechanism. All these methods are designed for homogeneous graphs. They cannot be directly applied to a heterogeneous graph with multiple types of entities and relations.

In recent years, lots of heterogeneous GNN based methods have been developed. HAN [18], HAHE [19], and Deep-HGNN [20] transforms a heterogeneous graph into several homogeneous graphs based on handcrafted meta-paths, applies GNN separately on each graph, and aggregates the output representations by attention mechanism. GraphInception [21] constructs meta-paths between nodes with the same object type. HetGNN [22] first samples a fixed number of neighbors via random walk strategy. Then it applies a hierarchical aggregation mechanism for intra-type and inter-type aggregation. HGT [23] extends transformer architecture to heterogeneous graphs. They directly calculate attention scores for all the neighbors of a target node and perform aggregation accordingly without considering domain knowledge.

B. GRAPH BASED FRAUD DETECTION

Recently, many graph-based fraud detectors have proposed since suspicion between entities could be well captured. [24] firstly build a model learning structure information among reviewers, reviews, and stores while NetWalk [25] extends fraud detection to dynamic networks. For industrial applications, [26] design graph-based system for suspicious users identification and [9] presents detecting malicious accounts via graph embedding at the Alipay platform.

To deal with heterogeneous graph data, many GNN-based fraud detectors constructs graph without edge type information for applying homogeneous graph neural networks. Fdgars [2] and GraphConsis [4] ignores relation type information and constructs a single homogeneous graph for neighborhood information aggregation. GeniePath [27] further proposes to learn adaptive receptive fields and select neighbor nodes effectively.

For relation-aware graph fraud detectors, their main solution is to build multiple homogeneous graphs based on edge type information of the original graph then perform type-independent node level aggregation and graph level concatenation. GEM [9] learns weighting parameters for different homogeneous subgraph. Player2Vec [7] and SemiGNN [8] both adopt attention mechanism in feature aggregation and SemiGNN further leverages a structure loss to guarantee the node embeddings homophily. Some works directly aggregate heterogeneous information in the graph. For instance, under a user-review-item heterogeneous graph, GAS [3] learns a unique set of aggregators for different node types and updates the embeddings of each node type iteratively.

Among the above works, few works [4]–[6] have noticed the camouflage behaviors of fraudsters. All these works can only handle a multi-relation graph, where all nodes are considered to be the same type. ASA [6] creates static features for directly aggregating messages in each homogeneous graph. GraphConsis [4] suffers from inflexible filtering thresholds and unsupervised similarity measures. CAREGNN [5] cannot handle multiple node types and

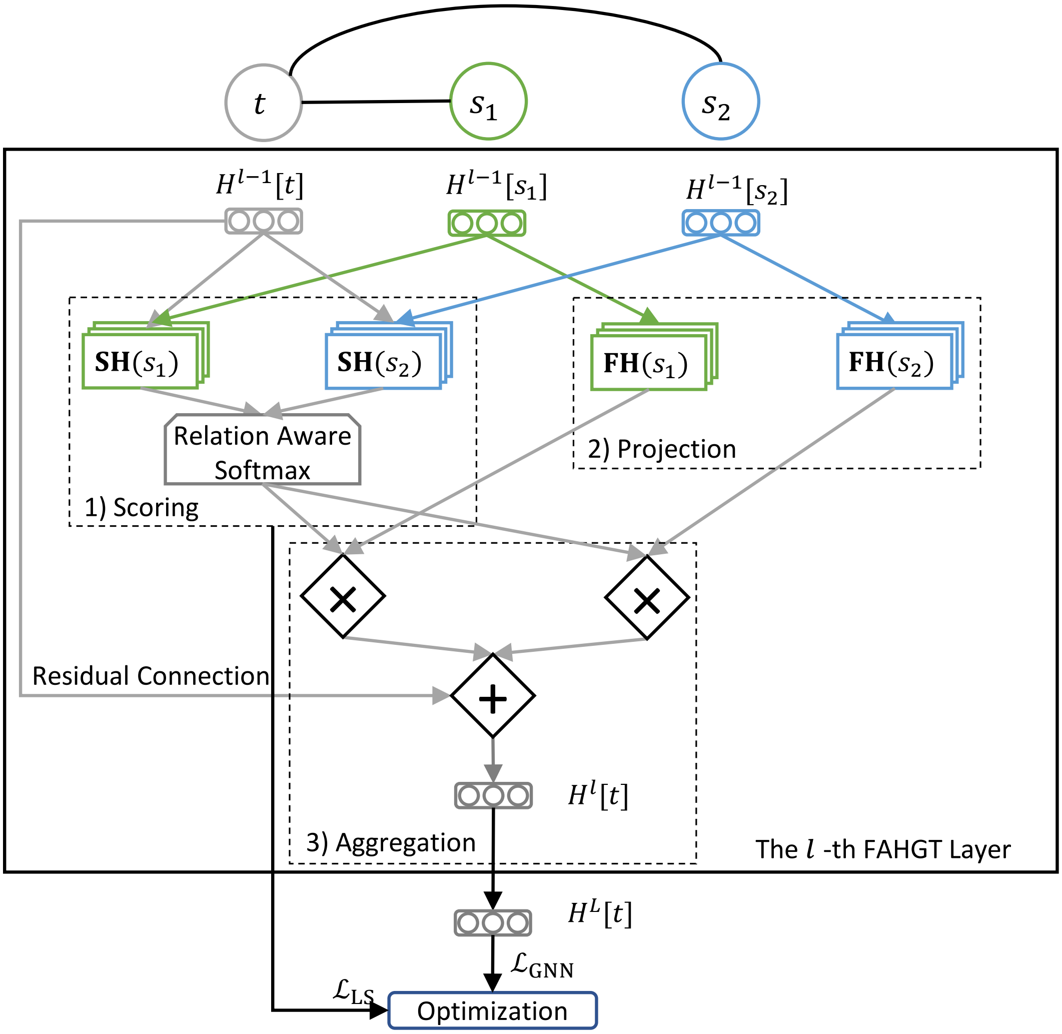


FIGURE 1. FAHGT Architecture. Different colors denote different node types. $H^{l-1}[\cdot]$ denotes the node feature learned of $(l-1)$-th layer. In each layer, the relation is scored by $\mathbf{SH}(\cdot)$ and the feature is projected by $\mathbf{FH}(\cdot)$. Then, projected neighbor node features with different types are aggregated into the l-th layer's representation $H^l[\cdot]$. The FAHGT layer can be stacked and the output of the final layer could be used for prediction.

requires a computational expensive reinforcement learning process. In addition, xFraud [28] takes node types into consideration, but their work does not uncover the camouflage behaviors of fraudsters.

Our model could handle heterogeneous graphs with multi-node types and overcome those shortcomings by using an efficient score head attention architecture.

III. PROBLEM DEFINITION

Heterogeneous Graph. A heterogeneous graph is defined as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges, $\mathcal{A}$ represents the set of types of nodes and $\mathcal{E}$ represents the set of types of edges. Each node $v \in \mathcal{V}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping functions $\tau(v) : \mathcal{V} \to \mathcal{A}$ and $\phi(e) : \mathcal{E} \to \mathcal{R}$, respectively.

Graph-based Fraud Detection. Given a set of nodes $\mathcal{V}$, the node feature matrix $\mathbf{X}$ and its corresponding graph G, our aim is to justify the node's suspicious Y by finding an optimal detector f such that $Y = f(\mathbf{X}, G)$. $Y \in \{0, 1\}$, where 0 represents **benign** and 1 represents **suspicious**. The detector f is trained based on the labeled node information in a semi-supervised manner. For example, the node could be an account in a transaction system or a user in a social network. The edges could be transactions between accounts or contacts between users. The suspicious label could be determined by whether a user has posted spam content.

IV. OUR MODEL

A. OVERVIEW

In general, the Fraud Aware Heterogeneous Graph Transformer(FAHGT) resorts to the aggregation-based GNN layer [23] defined as follows:

$$H^l[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\mathbf{Aggregate}} \Big(\mathbf{Score}(s,e,t) \cdot \mathbf{Feature}(s,e,t)\Big) \tag{1}$$

where t is the target node and $N(t)$ is the set of neighbors of t. All its neighbors $s \in N(t)$ might belong to different distributions. There are three basic operators in (1):

- $\mathbf{Score}$: estimates the importance of each triplet (s, e, t);
- $\mathbf{Feature}$: extracts the feature from the source node s;
- $\mathbf{Aggregate}$: can be functions of sum, mean, max or concatenation operation, which aggregates the neighborhood feature by the score.

For example, Heterogeneous Graph Transformer (HGT) [23] adopts a meta triplet specific attention mechanism as $\mathbf{Score}$, uses the type aware mapping for calculating $\mathbf{Feature}$, and leverages the average followed by a nonlinear activation for the $\mathbf{Aggregate}$ step. Our FAHGT model subsumes HGT model. Fig. 1 shows the overall architecture of FAHGT. Given a sampled heterogeneous sub-graph with t as the target node, s_1 and s_2 as the source nodes, the FAHGT model takes its edges $e_1 = (s_1, t)$ and $e_2 = (s_2, t)$ as input to learn the representation for each node. FAHGT consists of three modules: (1) meta relation scoring; (2) feature node projection; (3) aggregation.

B. META RELATION SCORING

The $\mathbf{Score}$ operator evaluates the importance of each **meta relations**, i.e., the $\langle \tau(s), \phi(e), \tau(t) \rangle$ triplets. The original HGT model implemented $\mathbf{Score}$ via introducing a meta triplet specific attention mechanism, which cannot distinguish camouflage user behaviour. Our model generalizes $\mathbf{Score}$ by injecting domain knowledge, leading to a more flexible and powerful architecture. The $\mathbf{Score}$ operator consists of h score heads and is defined as follows:

$$\mathbf{Score}(s,e,t) = \underset{\forall s \in N(t)}{\mathrm{Softmax}}\Big(\underset{i: i \in \{1,2,\ldots,h\}}{\Big\|} \mathbf{SH}^i(s,e,t) \Big) \tag{2}$$

$$\sum_{\forall s \in N(t)} \mathbf{SH}^i(s,e,t) \; = 1, 1 \leq i \leq h \tag{3}$$

where $\mathbf{SH}^i$ denotes the i-th score head and $\|$ is the concatenate operation. The Softmax function is conducted for each i-th score head over neighbor nodes for score normalization. Two types of scoring heads ($\mathbf{SH}$) are introduced here, i.e., heterogeneous mutual attention ($\mathbf{HA}$) head and label aware similarity ($\mathbf{LS}$) head:

1) Heterogeneous Mutual Attention

Heterogeneous mutual attention is adopted from HGT [23]. The i-th $\mathbf{HA}$ head is obtained from the triplet (s, e, t) as follows:

$$\mathbf{SH}^i_{\mathbf{HA}}(s,e,t) = \Big(K^i(s) \; P^S_{\phi(e)} \; Q^i(t)^T \Big) \cdot \frac{\mu_{\langle \tau(s), \phi(e), \tau(t) \rangle}}{\sqrt{d}} \tag{4}$$

$$K^i(s) = H^{l-1}[s] U^i_{\tau(s)}$$
$$Q^i(t) = H^{l-1}[t] V^i_{\tau(t)}$$

where the input $H^{l-1}[\cdot]$ is of dimension d, $U^i_{\tau(s)} \in \mathbb{R}^{h \times \frac{d}{h}}$ projects the representation of $\tau(s)$-type source node s into the i-th vector $K^i(s) \in \mathbb{R}^{\frac{d}{h}}$. Similarly, $V^i_{\tau(t)}$ is operated on the representation of the target node. $P^S_{\phi(e)} \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ is to be learned to generate different representations for different edge type $\phi(e)$. $\mu \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{R}| \times |\mathcal{A}|}$ serves as the global importance of each meta relation triplet $\langle \tau(s), \phi(e), \tau(t) \rangle$.

The designed $\mathbf{HA}$ relieves the aggregation-based GNN model from inconsistency problem via utilizing the specific type information of nodes and edges.

2) Label Aware Scoring

Inspired by LAGCN [29], we employ a one-layer MLP as the node label predictor for user node and use the l_1 distance between the prediction results of source node s and target node t as their similarity measure. Specifically, if we have $\tau(s) = \tau(t) =$ User, we set:

$$\begin{aligned} &\mathbf{SH}^i_{\mathbf{LS}}(s, \cdot, t) \\ &= 1 - \left\| \sigma \left(\mathbf{MLP}^i_{\mathbf{LS}}(\mathbf{X}[s]) \right) - \sigma \left(\mathbf{MLP}^i_{\mathbf{LS}}(\mathbf{X}[t]) \right) \right\|_1 \end{aligned} \tag{5}$$

where σ denotes the sigmoid function. The input of $\mathbf{MLP_{LS}}$ is the original node features, and the output of $\mathbf{MLP_{LS}}$ is passed into sigmoid function σ to generate the positive label probability. For nodes whose fraud label is undefined like product node, we manually set the score $\mathbf{SH}^i_{\mathbf{LS}}(s, \cdot, t)$ to 0.5.

The designed $\mathbf{LS}$ actively selects optimal neighbors by considering its fraud behavior. An edge connecting two suspicious user nodes will be assigned a higher score in aggregation, which enables fraud-aware node embedding generation.

Finally, we gather h score heads together from its neighbors $N(t)$ for each triplet. It should be noted that different types of score heads can be used interchangeably. For example, three $\mathbf{HA}$ heads and one $\mathbf{LS}$ head can be used in one model. If only $\mathbf{HA}$ heads are used, the FAHGT model reduces to the HGT model. In this manner, feature-based attention and label-based similarity could cooperate in a unified framework.

C. FEATURE NODE PROJECTION

Similar to the score process, for the triplet (s, e, t), we calculate its multi-head $\mathbf{Feature}$ by considering its type information:

$$\mathbf{Feature}(s,e,t) = \underset{i: i \in \{1,2,\ldots,h\}}{\Big\|} \mathbf{FH}^i(s,e,t) \tag{6}$$

$$\mathbf{FH}^i(s,e,t) = H^{l-1}[s] W^i_{\tau(s)} P^F_{\phi(e)} \tag{7}$$

where the input $H^{l-1}[\cdot]$ is of dimension d. $W^i_{\tau(s)} \in \mathbb{R}^{d \times \frac{d}{h}}$ projects the representation of source node s into $\tau(s)$-type subspace, followed by projection of $P^F_{\phi(e)} \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ for incorporating the edge dependency. The final step is to concatenate all h feature heads to get the $\mathbf{Feature}(s, e, t)$.

D. AGGREGATION AND OPTIMIZATION

After meta relation scoring and feature node projection, we perform message passing from the source nodes to the target node. We compute a weighted average of source nodes feature with relation score, and get the aggregated representation $\widetilde{H}^l[t]$ as:

$$\widetilde{H}^l[t] = \bigoplus_{\forall s \in N(t)} \Big(\mathbf{Score}(s,e,t) \cdot \mathbf{Feature}(s,e,t) \Big) \tag{8}$$

The representation $\widetilde{H}^l[t]$ of target node t is then projected back to its origin $\tau(t)$-type feature subspace via a linear projection $A_{\tau(t)}$ and residual connection:

$$H^l[t] = \sigma\big(\widetilde{H}^l[t]\big)\mathrm{A}_{\tau(t)} + H^{l-1}[t]. \tag{9}$$

In this way, we get the l-th layer's output $H^l[t]$ for the target node t.

The FAHGT layer can be stacked many times for incorporating distant node information. For each node v, its final embedding $\mathbf{z}[v] = H^L[v]$ can be used for prediction task. We can define the loss of GNN and label similarity as a cross-entropy loss function:

$$\mathcal{L}_{\text{GNN}} = \sum_{v \in \mathcal{V}_{\text{labeled}}} -\log\left(y_v \cdot \sigma(\text{MLP}_{\text{GNN}}(\mathbf{z}[v]))\right) \tag{10}$$

$$\mathcal{L}_{\text{LS}} = \sum_{v \in \mathcal{V}_{\text{labeled}}} -\log(y_v \cdot \sigma(\text{MLP}_{\text{LS}}(\mathbf{X}[v]))) \tag{11}$$

$$\mathcal{L}_{\text{FAHGT}} = \mathcal{L}_{\text{GNN}} + \lambda_{\text{LS}}\mathcal{L}_{\text{LS}} + \lambda_{\text{Reg}}||\Theta||_2 \tag{12}$$

where $\mathcal{V}_{labeled} \subset \mathcal{V}$ is the set of labeled nodes, $||\Theta||_2$ is the $L2$-norm of all model parameters, λ_{LS} and λ_{Reg} are weighting parameters. The whole model is trained in a semi-supervised manner.

V. EXPERIMENTS

In the experiment section, we mainly present:

- Heterogeneous graph construction from real world fraud data;
- Performance comparison over popular baselines and FAHGT variants;
- Hyper-parameter sensitivity study and model efficiency evaluation;
- Visual evidence of model's risk analyzing ability.

A. EXPERIMENTAL SETUP

1) Dataset.

We use the Amazon review dataset [30] to conduct the experiment. The Amazon dataset includes product reviews under 24 categories and we select three of them: Baby (BB), Music Instruments (MI), and Automotive (AM). Other previous datasets like Epinions [26] only contain graph structures and compacted features, where heterogeneous graph information cannot be utilized. Similar to [31], we label users with more than 80% helpful votes as the benign user and users with less than 20% helpful votes as the fraudulent user. The user and product are regarded as a node in the graph. We take 24 handcrafted features as the user node features and 50 handcrafted features as the product node feature described in Table 2 and 3. The edges between nodes are built from both underlying distribution of data and domain knowledge:

- **U-P**: User with its reviewed product.
- **U-A-U**: Two users with at least one identical rating within one week.
- **U-S-U**: Two users with top 5% mutual review text TF-IDF similarities.

Table 1 shows the dataset statistics.

2) Baselines.

The performance of FAHGT in fraud detection is verified by comparing it with various GNN baselines and popular graph based fraud detectors. For general graph neural networks, we select GCN [15], GAT [17], GraphSAGE [16], GeniePath [16]. For popular GNN-based fraud detectors, we select SemiGNN [8], GraphConsis [4] and CAREGNN [5]. GCN, GAT, GraphSAGE, and GeniePath are run on homogeneous graphs. SemiGNN, GraphConsis, and GeniePath consider edge types in their approaches, and node types are not considered but equally treated. We also include two variants of our model for ablation study. FAHGT-l (LAGCN) [29] filters noise neighbor node effectively, but is not able to learn informative representation from a graph with different node type and edge type. FAHGT-h(HGT) [23] handles heterogeneous graph but lacks the ability to discover camflouge behavior.

3) Experimental Setting.

From Table 1, we can see that the fraction of fraudsters is relatively small compared to the whole users in all three datasets and the dense edge connections form a large-scale graph. To improve training efficiency, we sample a small batch of the labeled node with its k-hop subgraph. Under each batch, the number of positive instances and negative instances is kept equally.

We use 64 as the embedding size throughout all neural network baselines. All GNNs keep 2 layers of receptive fields and we use fixed neighborhood sample sizes of 25 and 10 following [16]. For model parameter optimization, we use a unified optimizer (Adam), training fraction (40%), learning rate (1e-3), training epochs (500), and L2 regularization weight ($\lambda_{Reg} = 0.001$) for all models. For CAREGNN, we set the RL action step size as 0.02. In our proposed FAHGT, we set the similarity loss weight (λ_{LS}) as 2. The sensitivity of the layer numbers, embedding size, and training fraction are studied in section V-C.

4) Implementation.

We implement all models in PyTorch 1.7, PyTorch Geometric 1.7, and Python 3.8. GCN, GAT, GeniePath, GraphSAGE, SemiGNN, and GraphConsis are implemented following the original paper. For the CAREGNN, we use the source code[1] provided by the authors. All models are running on 4 NVIDIA GTX 1080 Ti GPUs, 128GB RAM, i9 7900X Ubuntu server.

[1] https://github.com/YingtongDou/CARE-GNN

TABLE 1. Statistics of Dataset

Dataset	#nodes	#edges	#users(#fraudsters)	#products	#U-P	#U-A-U	#U-S-U
Musical Instruments	33058	4827387	11944(821)	21114	39837	1651733	3566479
Baby	31717	8694986	16227(579)	15490	57422	2850313	6582889
Automotive	48451	21651080	17581(656)	30870	8460732	15454580	106768

TABLE 2. Detail of User Node Feature

Feature Type	Feature Count	Feature Description
User	1	Length of username
	1	Number of rated products
Rating	5	Count of each review rating (from 1 to 5)
	5	Ratio of each review rating (from 1 to 5)
	1	Count of positive review rating (larger than 4)
	1	Ratio of positive review rating (larger than 4)
	1	Entropy of rating distribution of user's reviews
	4	Median, Max, Min, Mean of user's review rating
Time	1	Day gap
	1	Time entropy
	1	Same date indicator
Text	1	Feedback summary length
	1	Review text sentiment

TABLE 3. Detail of Product Node Feature

Feature Type	Feature Count	Feature Description
Text	25	PCA transformed TFIDF vector of product title
	25	PCA transformed TFIDF vector of product description

5) Evaluation Metric.

Since the imbalanced nature of all three datasets, and positive instances should be paid more attention in the fraud detection domain, we utilize Macro F1 (F1), Kolmogorov Smirnov Test (KS), and ROC-AUC (AUC) to evaluate the overall performance of all classifiers like previous works [5], [32].

- F1: Macro F1-score is used to evaluate the performance of a classifier with multiple binary labels or multiple classes. It will give the same importance to each label/-class. The Macro F1-score is defined as the mean of class-wise F1-scores. The higher the F1 score, the better the performance of a classifier.
- KS: The Kolmogorov-Smirnov Test is a very commonly used metric in the risk analysis domain, which measures the similarity between the cumulative empirical distributions between predicted and observed data. The higher the KS, the better the performance of the model at measuring risk.
- AUC: The Area Under the Curve (AUC) measures the ability of a classifier to distinguish between classes and summarizes the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

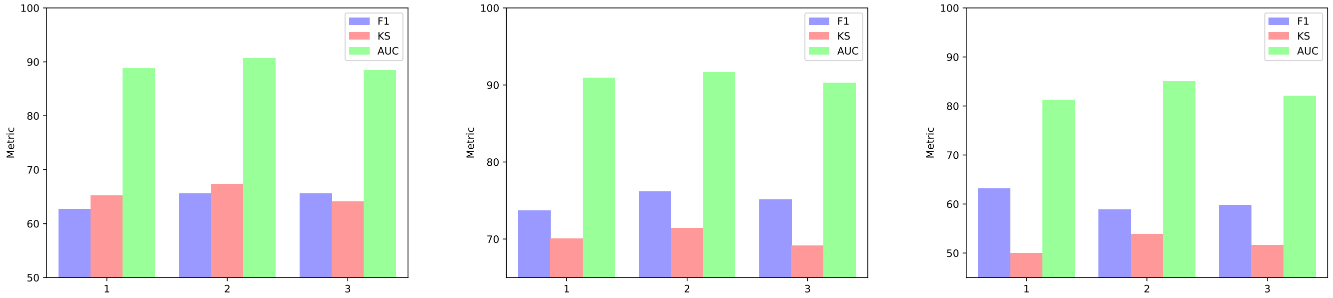
B. OVERALL EVALUATION

1) Expressive Power of Data Heterogeneity.

Table 4 shows the performance of FAHGT and various baselines on three datasets. We observe that FAHGT outperforms other baselines under all of the metrics. The poor result of logistic regression (LR) indicates that graph structure and neighbor features both are useful in fraudster prediction. Among all GNN baselines in Table 4, GCN, GAT, GraphSAGE, and GeniePath run on the homogeneous graph where all relations are treated equally. Other baselines are evaluated in multi-relation graphs. The performances of homogeneous GNNs are comparable to multi-relation methods like GraphConsis and SemiGNN, which indicates previously designed graph-based fraud detectors are not suitable for heterogeneous graphs. The reason is that direct aggregation of node features with multiple types may introduce noises, resulting in the same poor performance of single-relation and multi-relation GNN. Better than CAREGNN, FAHGT and its variants aggregate information from the nodes with type-

TABLE 4. Overall Result

Dataset	Baby			Musical Instruments			Automotive		
Model	F1	KS	AUC	F1	KS	AUC	F1	KS	AUC
LR	50.84	61.41	86.72	73.20	67.05	89.18	48.14	47.92	80.35
GCN	55.20	60.97	87.13	74.60	68.68	91.00	52.67	51.31	81.90
GAT	57.48	63.73	88.25	74.61	70.11	91.16	56.90	49.95	81.43
GraphSAGE	58.50	64.67	89.10	71.63	68.97	91.14	61.55	50.62	83.11
GeniePath	60.58	64.14	89.16	73.83	70.54	90.80	59.93	49.14	80.82
SemiGNN	65.40	65.52	88.88	74.73	69.77	91.23	46.77	54.80	84.12
GraphConsis	60.64	64.37	88.58	74.07	68.58	90.85	56.41	52.98	84.19
CAREGNN	61.66	46.37	79.50	59.59	54.09	82.62	44.82	45.81	77.30
FAHGT-l	61.55	63.82	88.67	74.67	68.84	90.02	58.19	51.76	82.96
FAHGT-h	62.91	65.23	89.25	74.25	71.26	91.49	**61.85**	**55.48**	84.24
FAHGT	**65.58**	**67.37**	**90.69**	**76.14**	**71.41**	**91.61**	58.85	53.85	**85.09**



(a) Baby

(b) Music Instruments

(c) Automotive

FIGURE 2. Performance Comparison under Different Layers.

aware and label-aware scoring, which could better identify camouflage behaviors and filter unrelated nodes and handle heterogeneous graph information.

2) Model Variants.

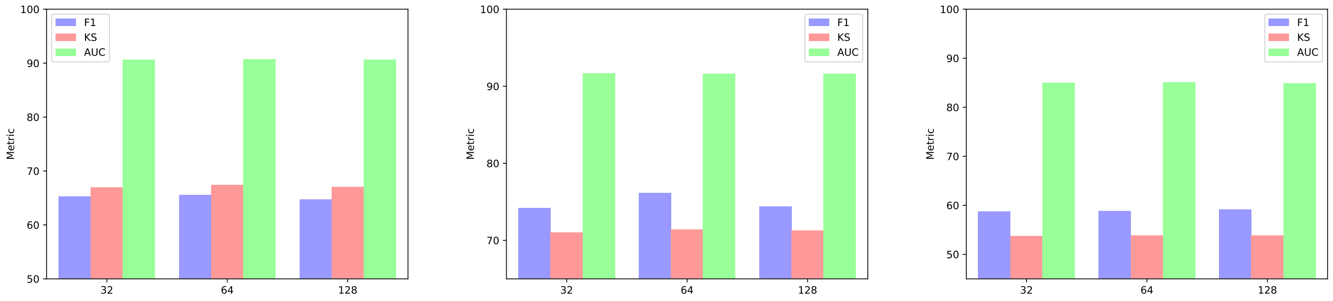
The last three rows of Table 4 show the performance of FAHGT and its variants with different score head combinations. FAHGT-l(LAGCN) only uses the label head and FAHGT-h(HGT) only uses the attention head.

It is observed that those three models have comparable performances with GCN and GAT. FAHGT-h(HGT) outperforms all three GNN-based fraud detectors in most metrics and datasets, which reveals that current graph fraud detection approaches suffer from inconsistency problems when dealing with heterogeneous graph data. While GraphConsis and CAREGNN also use similarity measure to discover node camouflage, both of them shows unstable performance comparing with our FAHGT-l(LAGCN) model. In addition, on the Baby dataset, we observe that GraphSAGE shows comparable performance with FAHGT-l(LAGCN) and FAHGT-h(HGT). The reason for this phenomenon may be that the number of U-S-U relations between users in the Baby dataset is relatively high, making it easier to find fraudulent users who tend to post similar reviews. In such a situation, the FAHGT-l(LAGCN) and FAHGT-h(HGT) both show relatively ordinary performance due to parameter overfitting.

That is, both attention and score head will become necessary in scoring meta relation. Combining both of them results in a better performance.

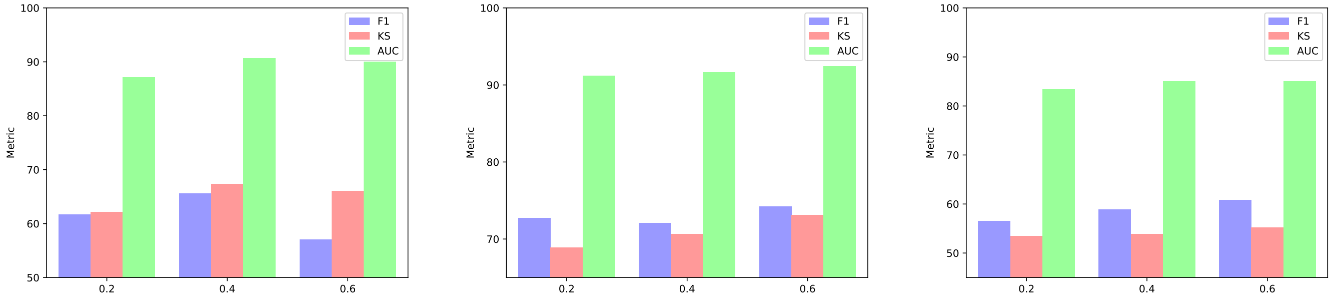
C. HYPER-PARAMETER SENSITIVITY

Fig. 2, 3 and 4 demonstrates the impact of model hyperparameters on the three dataset. From Fig. 2, we observe that two-layer FAHGT usually performs better than one-layer FAHGT, but three-layer FAHGT barely improves the performance. This may due to the over-smooth problem in a larger receptive field. Therefore, the two-layer model can achieve better classification performance with reduced training complexity. Fig. 3 presents FAHGT's performance under different embedding sizes. With larger embedding sizes, the regularization constraints on model parameters are slightly stronger and we do not find significant differences in terms of classification metrics. Fig. 4 demonstrates the performance of FAHGT under different training fractions. It is observed that there is little performance gain for GNNs when adding more training data. This observation demonstrates the advan-



(a) Baby (b) Music Instruments (c) Automotive

FIGURE 3. Performance Comparison under Different Embedding Size.



(a) Baby (b) Music Instruments (c) Automotive

FIGURE 4. Performance Comparison under Different Training Fraction.

tage of semi-supervised learning, where a small fraction of supervised signals is enough to optimize model parameters and generate informative node representation.

D. DISCUSSION

From Fig. 5, we can see that the training process of FAHGT takes only 4 seconds per epoch on average for each dataset, with an effective performance gain and comparable efficiency comparing to other baselines. The computational efficiency of FAHGT comes from the carefully designed scoring mechanism, which computes neighbor filtering and relation weighting in a parallelized manner.

We also provide details of model prediction via plotting ROC and KS curves. Fig. 6 shows ROC curves of FAHGT on different dataset. All the curves indicate that FAHGT can maintain stable performance for fraudster detection. Fig. 7a–7c report the KS curves of FAHGT on three datasets. Those figures demonstrate the power of our models of distinguishing suspicious users from benign users.

VI. CONCLUSION

In this paper, we propose FAHGT, a novel heterogeneous graph neural network for fraudulent user detection in online review systems. To handle inconsistent features, we adopt heterogeneous mutual attention for automatic meta path construction. To detect camouflage behaviors, we design the label aware scoring to filter noisy neighbors. Two neural modules are combined in a unified manner called "score head mechanism" and both contribute to edge weight computation in final feature aggregation. Experiment results on real-world business datasets validate the excellent effect on fraud detection of FAHGT. The hyper-parameter sensitivity and visual analysis further show the stability and efficiency of our model. In summary, FAHGT is capable of alleviating inconsistency and discover camouflage and thus achieves state-of-art performance in most scenarios. In the future, we plan to extend our model in handing dynamic graphs data and incorporate fraud detection into other areas, such as robust item recommendation in E-commerce or loan default prediction in financial services.

REFERENCES

[1] V. S. Tseng, J. Ying, C. Huang, Y. Kao, and K. Chen, "Fraudetector: A graph-mining-based framework for fraudulent phone call detection," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D. Margineantu, and G. Williams, Eds. ACM, 2015, pp. 2157–2166. [Online]. Available: https://doi.org/10.1145/2783258.2788623

[2] J. Wang, R. Wen, and C. Wu, "Fdgars: Fraudster detection via graph convolutional networks in online app review system," in WWW Workshops, 2019.

[3] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in CIKM, 2019.

[4] 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 SIGIR, 2020.

[5] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in CIKM, 2020.

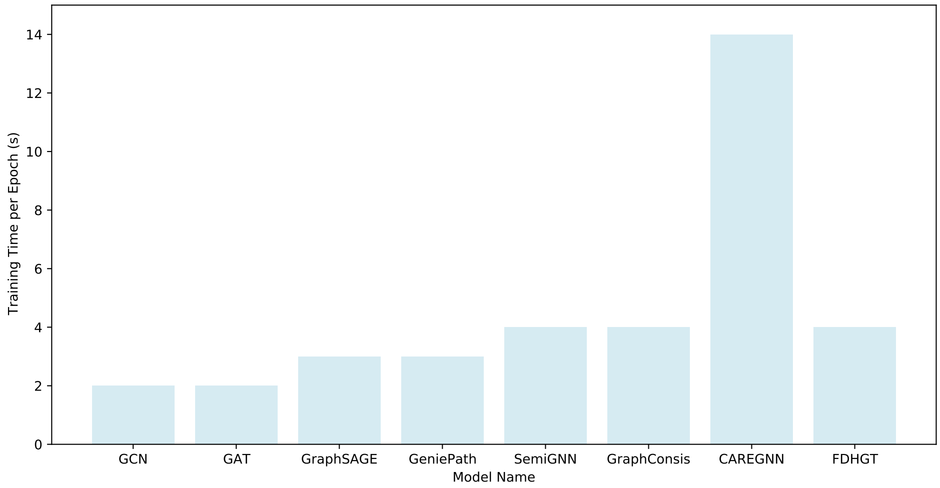


FIGURE 5. Training Time per Epoch for Each Model

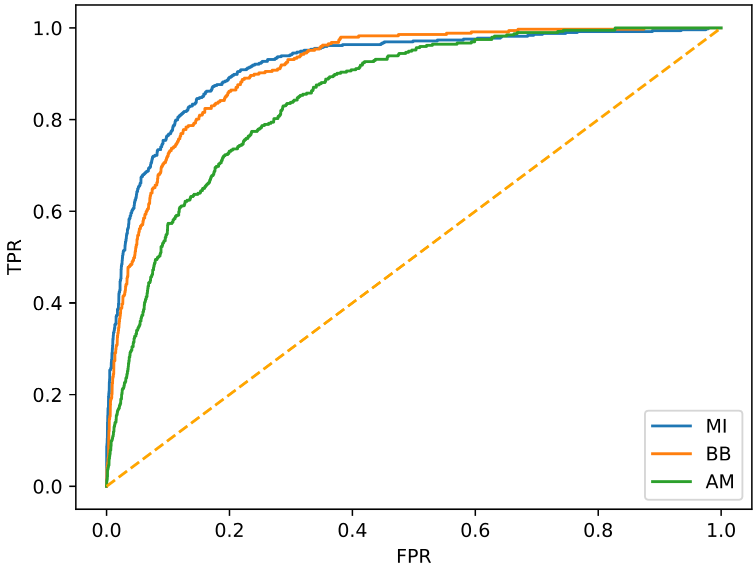
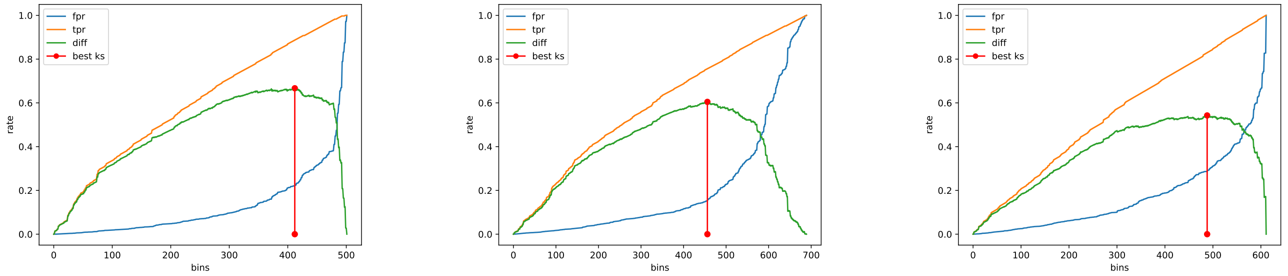


FIGURE 6. ROC-AUC Curve on Three Datasets



(a) Baby

(b) Musical Instruments

(c) Automotive

FIGURE 7. KS Curve on Three Datasets

[6] R. Wen, J. Wang, C. Wu, and J. Xiong, "Asa: Adversary situation awareness via heterogeneous graph convolutional networks," in WWW Workshops, 2020.

[7] Y. Zhang, Y. Fan, Y. Ye, L. Zhao, and C. Shi, "Key player identification in underground forums over attributed heterogeneous information network embedding framework," in CIKM, 2019.

[8] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, and J. Zhou, "A semi-supervised graph attentive network for fraud detection," in ICDM, 2019.

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

[10] Y. Dou, G. Ma, P. S. Yu, and S. Xie, "Robust spammer detection by nash reinforcement learning," in KDD, 2020.

[11] P. Kaghazgaran, M. Alfifi, and J. Caverlee, "Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures," in CIKM, 2019.

[12] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," TKDE, 2020.

[13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," arXiv preprint arXiv:1312.6203, 2013.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in NeurIPS, 2016, pp. 3844–3852.

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

[16] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in NeurIPS, 2017.

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in ICLR, 2017.

[18] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in WWW, 2019, pp. 2022–2032.

[19] S. Zhou, J. Bu, X. Wang, J. Chen, and C. Wang, "Hahe: Hierarchical attentive heterogeneous information network embedding," arXiv preprint arXiv:1902.01475, 2019.

[20] S. Wang, Z. Chen, D. Li, Z. Li, L.-A. Tang, J. Ni, J. Rhee, H. Chen, and P. S. Yu, "Attentional heterogeneous graph neural network: Application to program reidentification," in Proceedings of the 2019 SIAM International Conference on Data Mining. SIAM, 2019, pp. 693–701.

[21] Y. Zhang, Y. Xiong, X. Kong, S. Li, J. Mi, and Y. Zhu, "Deep collective classification in heterogeneous information networks," in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 399–408.

[22] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in KDD, 2019, pp. 793–803.

[23] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in WWW, 2020, pp. 2704–2710.

[24] G. Wang, S. Xie, B. Liu, and S. Y. Philip, "Review graph based online store review spammer detection," in ICDM. IEEE, 2011, pp. 1242–1247.

[25] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in KDD, 2018, pp. 2672–2681.

[26] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "Rev2: Fraudulent user prediction in rating platforms," in WSDM, 2018.

[27] Z. Liu, C. Chen, L. Li, J. Zhou, X. Li, L. Song, and Y. Qi, "Geniepath: Graph neural networks with adaptive receptive paths," in AAAI, 2019.

[28] 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.

[29] H. Chen, L. Wang, S. Wang, D. Luo, W. Huang, and Z. Li, "Label aware graph convolutional network–not all edges deserve your attention," arXiv preprint arXiv:1907.04707, 2019.

[30] J. McAuley and J. Leskovec, "From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews," in WWW, 2013.

[31] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui, "Gcn-based user representation learning for unifying robust recommendation and fraudster detection," in SIGIR, 2020.

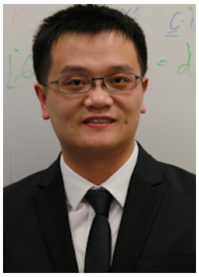
[32] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in KDD, 2015.

SONGKAI TANG (Graduate Student Member, IEEE) was born in Hunan, China, in 1998. He received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2019, where he is currently pursuing the M.S. degree in computer science and technology.

In 2018, he was a Data Engineer Intern with the Computing Advertising Department, CooTek, Shanghai. From 2019 to 2020, he was an Algorithm Engineer Intern with the Anti-Fraud Department, 360 DigiTech Inc., Shanghai. His research interests include graph neural networks, fraud detection, and recommender systems.

LUHUA JIN (Graduate Student Member, IEEE) was born in Shanghai, China, in 1996. He received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2019, where he is currently pursuing the M.S. degree in computer technology.

From 2019 to 2020, he was an Intern with the Algorithm Department, 360 DigiTech Inc., Shanghai. His research interests include natural language processing, tabular data processing, and normalization theory.



FAN CHENG (Member, IEEE) received the bachelor's degree in computer science and engineering from Shanghai Jiao Tong University, in 2007, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong, in 2012. From 2012 to 2014, he was a Postdoctoral Fellow with the Institute of Network Coding, The Chinese University of Hong Kong. Since 2015, he has been a Research Fellow with the Department of Electrical and Computer Engineering, NUS, Singapore.