

Received April 21, 2021, accepted May 12, 2021, date of publication May 31, 2021, date of current version December 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3084924

Fraud Detection in Online Product Review Systems via Heterogeneous Graph Transformer

SONGKAI TANG^{ID}, (Graduate Student Member, IEEE),

LUHUA JIN, (Graduate Student Member, IEEE), AND FAN CHENG^{ID}, (Member, IEEE)

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. FAHGT shows a remarkable performance gain compared to several baselines on different datasets.

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 its effectiveness in learning node representations and discovering structure pattern, graph neural networks have been drawn attention in fraud detection domain including product review [2]–[5], mobile application distribution [6],

cybercrime 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 distinguish the unique semantic user pattern. Also, if a user is suspicious, then the other one should be more

The associate editor coordinating the review of this manuscript and approving it for publication was Szidónia Lefkovits^{ID}.

likely to be distrustful if they are connected by common-activity relation since fraudulent users tend to generate spam content 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 fraud detection performance of FAHGT on many real-world datasets. It is verified that FAHGT can considerably improve F1 score, KS and AUC over several baselines.

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.
- **Focus:** 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] learns the node representation by iterative aggregation in a sampled node tree. GAT [17] further proposes to learn in the spatial domain by computing different importance of neighbor nodes via the masked self-attention mechanism.

In recent years, lots of heterogeneous GNN based methods have been developed. HAN [18], HAHE [19], and DeepHGNN [20] transforms a heterogeneous graph into several homogeneous graphs based on handcrafted meta-paths, applies GNN separately on each graph, and aggre-

gates 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 spam detection graph model with iterative computation framework while NetWalk [25] extends fraud detection to dynamic networks. For industrial applications, [26] design graph-based system for suspicious users identification and [9] proposes embedding-based malicious account detector in Alibaba.com.

To apply neural network to 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 type-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 type-specific node embedding for aggregation.

Among the above works, few works [4]–[6] have acknowledged the camouflage behaviors and propose their solution, as stated in [5]. 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] is trained in an unsupervised manner and results in inferior performances. CAREGNN [5] cannot handle multiple node types and 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

A. HETEROGENEOUS GRAPH

Following [23], a heterogeneous graph is defined as $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where \mathcal{V} is the set of nodes, \mathcal{A} represents the set of node types and $\tau(v) : \mathcal{V} \rightarrow \mathcal{A}$ is the node type mapping function. Also, \mathcal{E} is the set of edges, \mathcal{R} represents the set of edge types and $\phi(e) : \mathcal{E} \rightarrow \mathcal{R}$ is the edge type mapping function.

B. 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 1 stands for **suspicious** users and 0 stands for **benign** users. The detector f is trained based on the node with label information and all node features 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:

$$H^l[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{\text{Aggregate}} \left(\text{Score}(s, e, t) \cdot \text{Feature}(s, e, t) \right) \quad (1)$$

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 node type. The following paragraphs provides definitions for basic operators in (1):

- **Score**: estimates the importance of each triplet (s, e, t) ;
- **Feature**: extracts the feature from the source node s ;
- **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 **Score**, uses the type aware mapping for calculating **Feature**, and uses weighted average operation for the **Aggregate** step. Our FAHGT model subsumes HGT model. Fig. 1 shows the overall architecture of FAHGT. FAHGT consists of three modules: (1) meta relation scoring; (2) feature node projection; (3) aggregation.

B. META RELATION SCORING

The **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 **Score** via introducing a meta triplet specific attention mechanism, which cannot distinguish camouflage user behaviour. Our model generalizes **Score** by

injecting domain knowledge, leading to a more flexible and powerful architecture. The **Score** operator consists of h score heads and is defined as follows:

$$\text{Score}(s, e, t) = \underset{\forall s \in N(t)}{\text{Softmax}} \left(\parallel_{i \in \{1, 2, \dots, h\}} \mathbf{SH}^i(s, e, t) \right) \quad (2)$$

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

where \mathbf{SH}^i denotes the i -th score head and \parallel 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 (**SH**) are introduced here, i.e., heterogeneous mutual attention (**HA**) head and label aware similarity (**LS**) head:

1) HETEROGENEOUS MUTUAL ATTENTION

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

$$\mathbf{SH}_{\text{HA}}^i(s, e, t) = \left(K^i(s) P_{\phi(e)}^S Q^i(t)^T \right) \cdot \frac{\mu(\tau(s), \phi(e), \tau(t))}{\sqrt{d}} \quad (4)$$

$$K^i(s) = H^{l-1}[s] U_{\tau(s)}^i$$

$$Q^i(t) = H^{l-1}[t] V_{\tau(t)}^i$$

where the input $H^{l-1}[\cdot]$ is of dimension d , $U_{\tau(s)}^i \in \mathbb{R}^{h \times \frac{d}{h}}$ 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, $V_{\tau(t)}^i$ is operated on the representation of the target node. $P_{\phi(e)}^S \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ 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 **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 [5] and [29], we choose a single layer neural network for soft node label prediction and the L1 distance for measuring their label similarity. Specifically, if we have $\tau(s) = \tau(t) = \text{User}$, we set:

$$\mathbf{SH}_{\text{LS}}^i(s, \cdot, t) = 1 - \left\| \sigma \left(\text{MLP}_{\text{LS}}^i(\mathbf{X}[s]) \right) - \sigma \left(\text{MLP}_{\text{LS}}^i(\mathbf{X}[t]) \right) \right\|_1 \quad (5)$$

where σ denotes the sigmoid function. The input of MLP_{LS} is the original node features, and the output of 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 $\mathbf{SH}_{\text{LS}}^i(s, \cdot, t)$ to 0.5.

The designed **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

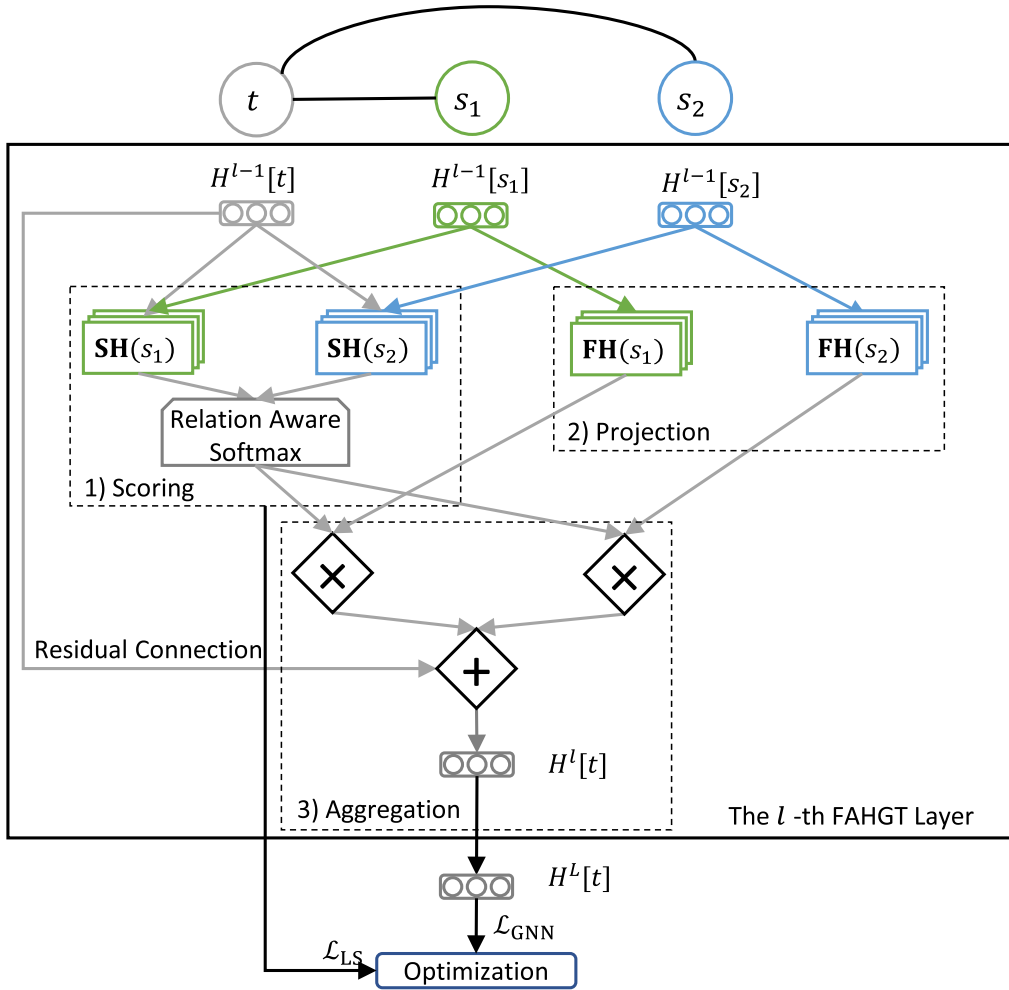


FIGURE 1. FAHGT Architecture. Different colors denote different node types. $H^{l-1}[\cdot]$ denotes the node feature learned of $(l-1)$ -th layer. In each layer, the relation is scored by $\text{SH}(\cdot)$ and the feature is projected by $\text{FH}(\cdot)$. Then, projected neighbor node features with 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.

of score heads can be used interchangeably. For example, three HA heads and one LS head can be used in one model. If only 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

Following [23], for the triplet (s, e, t) , we calculate its multi-head **Feature** by considering its type information:

$$\mathbf{Feature}(s, e, t) = \parallel_{i:i \in \{1, 2, \dots, h\}} \mathbf{FH}^i(s, e, t) \quad (6)$$

$$\mathbf{FH}^i(s, e, t) = H^{l-1}[s] W_{\tau(s)}^i P_{\phi(e)}^F \quad (7)$$

where the input $H^{l-1}[\cdot]$ is of dimension d . $W_{\tau(s)}^i \in \mathbb{R}^{d \times \frac{d}{h}}$ projects the representation of source node s into $\tau(s)$ -type subspace, followed by projection of $P_{\phi(e)}^F \in \mathbb{R}^{\frac{d}{h} \times \frac{d}{h}}$ for incorporating the edge dependency. The final step is to concatenate all h feature heads to get the **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. Following [23], we aggregate node feature with relation score, and get the aggregated representation $\tilde{H}^l[t]$ as:

$$\tilde{H}^l[t] = \bigoplus_{v \in N(t)} \left(\mathbf{Score}(s, e, t) \cdot \mathbf{Feature}(s, e, t) \right) \quad (8)$$

The representation $\tilde{H}^l[t]$ 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, following [23]:

$$H^l[t] = \sigma(\tilde{H}^l[t]) A_{\tau(t)} + H^{l-1}[t]. \quad (9)$$

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. The loss of node label prediction and soft label similarity can be

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

defined as two cross-entropy function, following [5]:

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

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

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

where $\mathcal{V}_{\text{labeled}} \subset \mathcal{V}$ is the set of labeled nodes, $\|\Theta\|_2$ is the L_2 -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 this section, we aim to present four research findings:

- Heterogeneous graph construction from real world fraud data;
- Performance evaluation among popular general graph neural networks and fraud detectors;
- Hyper-parameter sensitivity study and model efficiency evaluation;
- Visual evidence of model's risk analyzing ability.

A. EXPERIMENTAL SETUP

1) DATASET

The Amazon review dataset [30] is used for training and evaluating models. We select three categories from dataset: Baby (BB), Music Instruments (MI), and Automotive (AM). Similar to [5], we consider users received at least 20 votes as labeled. Fraudulent users have less than 20% helpful votes

and benign users have more than 80% helpful votes. We also adopt the node feature transformation and edge construction from [5]. 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 one or more identical rating in 7 days.
- **U-S-U**: Two users with top 5% review text TF-IDF score.

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-I (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]

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

handles heterogeneous graph but lacks the ability to discover camflouge behavior.

3) EXPERIMENTAL SETTING

Table 1 indicates that the amount of fraudsters is rare 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 code¹ provided by the authors. All models are running on an Ubuntu server with 4 NVIDIA 1080 Ti GPUs.

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 consider the same importance for each class. A high the F1 score indicates a good performance of a classifier.
- KS: The Kolmogorov-Smirnov Test measures the similarity between predicted and observed data. A high KS score indicates a strong risk measuring ability.
- AUC: The Area Under the Curve (AUC) measures the performance of model at distinguishing between the positive and negative instances.

¹<https://github.com/YingtongDou/CARE-GNN>

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. The performances of four homogeneous GNN baselines from GCN to GeniePath are comparable to multi-relation methods. This result implies that previous state-of-the-art graph-based solution are not applicable for heterogeneous graph data. Direct node features aggregation with types information may introduce noises, resulting in degraded performance. FAHGT and its variants aggregate information from the nodes with type-aware and label-aware scoring, which could discover camouflage behaviors by unrelated neighbor filtering and extract heterogeneous graph information.

2) MODEL VARIANTS

FAHGT and its variants with different score head combinations is also reported in Table 4. FAHGT-I(LAGCN) only uses the label head and FAHGT-h(HGT) only uses the attention head.

The experiment result shows that our model outperforms most of 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-I(LAGCN) model. In addition, on the Baby dataset, we observe that GraphSAGE shows comparable performance with FAHGT-I(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-I(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

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,

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

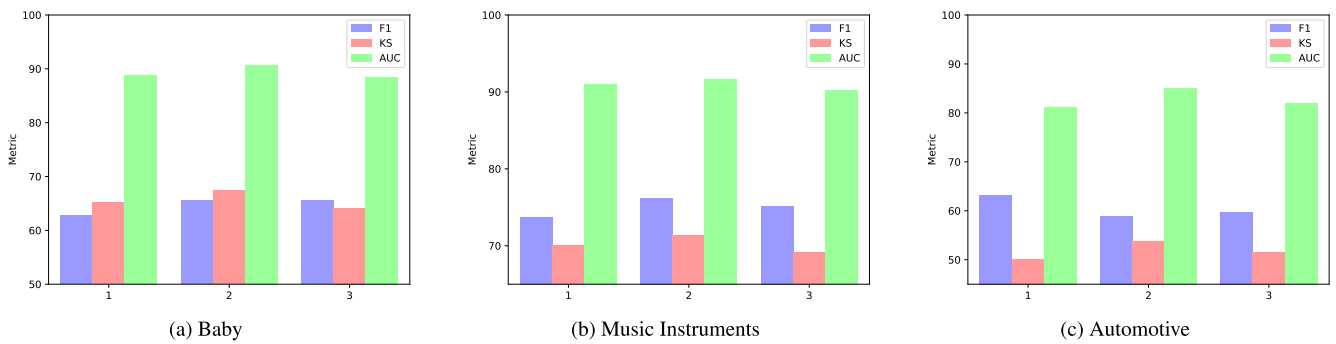


FIGURE 2. Performance comparison under different layers.

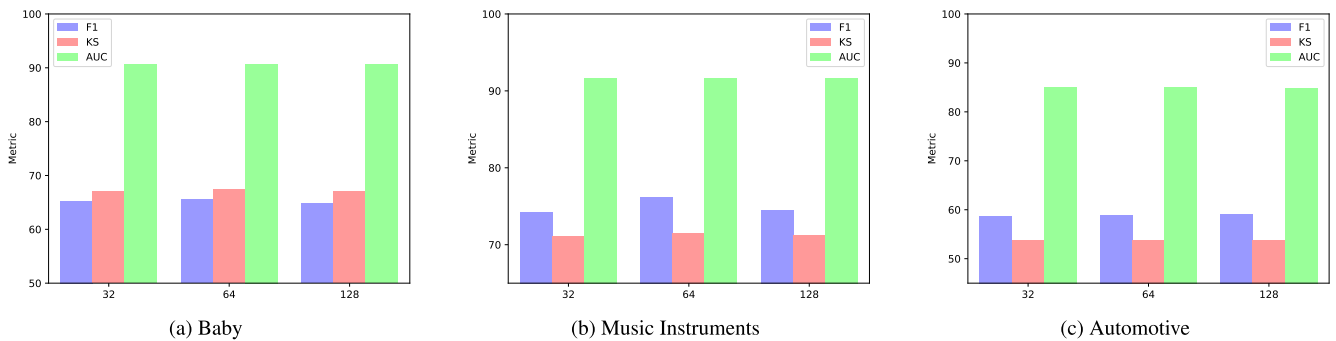


FIGURE 3. Performance comparison under different embedding size.

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. The increasing training fraction of data show little improvement in performance, which is consistent with observation in [5].

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

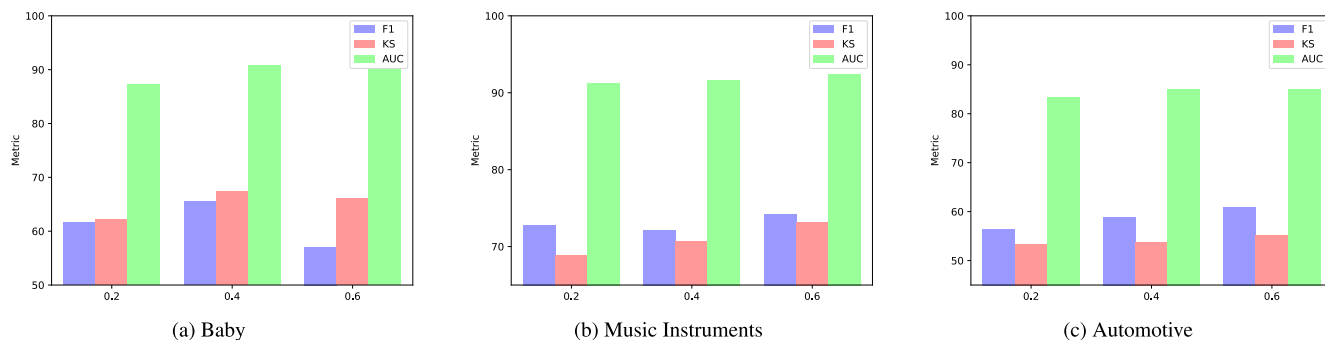


FIGURE 4. Performance comparison under different training fraction.

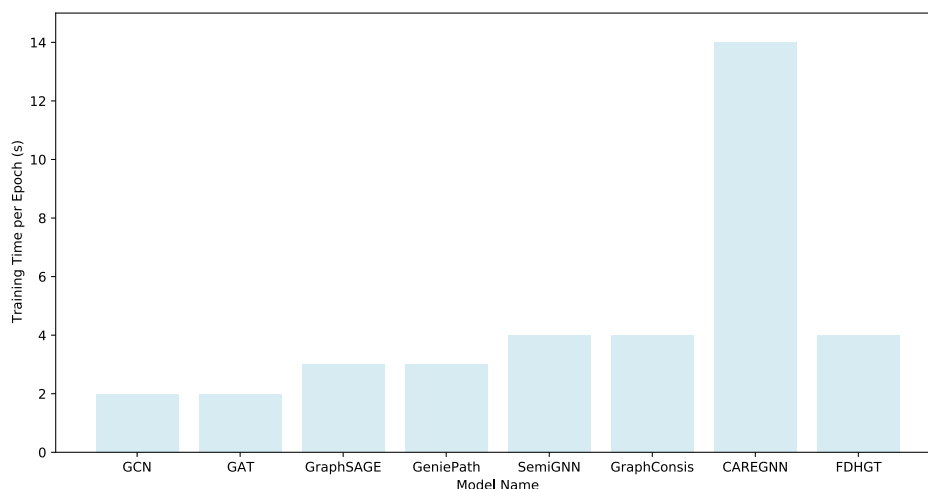


FIGURE 5. Training time per epoch for each model.

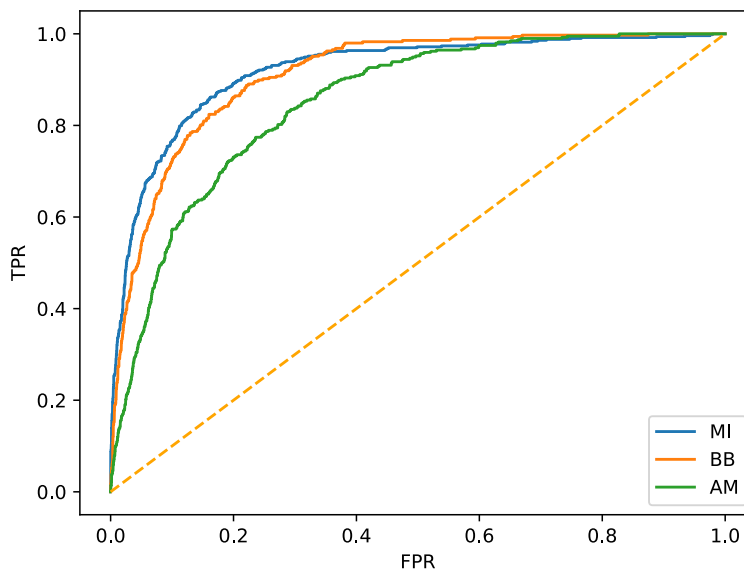


FIGURE 6. ROC-AUC curve on three datasets.

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

We present 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.

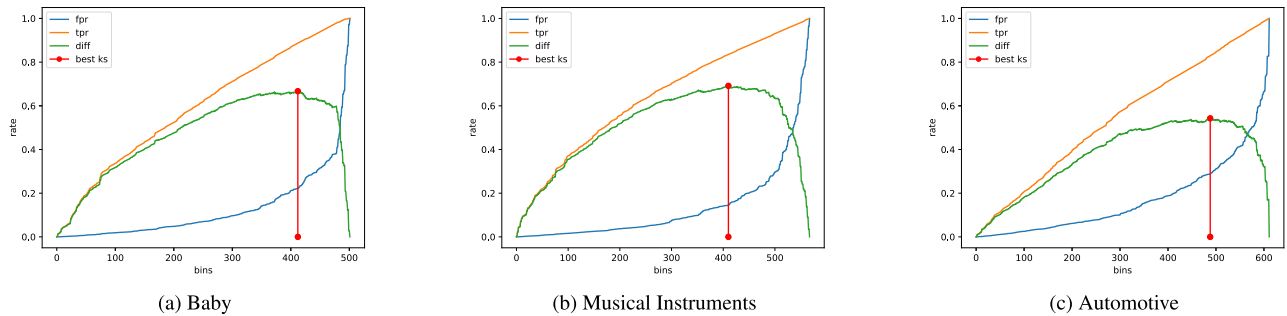


FIGURE 7. KS curve on three datasets.

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 handling 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, “Frauddetector: A graph-mining-based framework for fraudulent phone call detection,” in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, L. Cao, C. Zhang, T. Joachims, G. I. Webb, D. D. Margineantu, and G. Williams, Eds., Aug. 2015, pp. 2157–2166, doi: 10.1145/2783258.2788623.
- [2] J. Wang, R. Wen, C. Wu, Y. Huang, and J. Xion, “FdGars: Fraudster detection via graph convolutional networks in online app review system,” in *Proc. Companion Proc. World Wide Web Conf.*, May 2019, pp. 310–316.
- [3] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, “Spam review detection with graph convolutional networks,” in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2703–2711.
- [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 *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1569–1572.
- [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 *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 315–324.
- [6] R. Wen, J. Wang, C. Wu, and J. Xiong, “ASA: Adversary situation awareness via heterogeneous graph convolutional networks,” in *Proc. Companion Web Conf.*, Apr. 2020, pp. 674–678.
- [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 *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 549–558.
- [8] D. Wang, Y. Qi, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, and S. Yang, “A semi-supervised graph attentive network for financial fraud detection,” in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 598–607.
- [9] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, “Heterogeneous graph neural networks for malicious account detection,” in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 2077–2085.
- [10] Y. Dou, G. Ma, P. S. Yu, and S. Xie, “Robust spammer detection by Nash reinforcement learning,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 924–933.
- [11] P. Kaghazgaran, M. Alfi, and J. Caverlee, “Wide-ranging review manipulation attacks: Model, empirical study, and countermeasures,” in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 981–990.
- [12] Z. Zhang, P. Cui, and W. Zhu, “Deep learning on graphs: A survey,” *IEEE Trans. Knowl. Data Eng.*, early access, Mar. 17, 2020, doi: 10.1109/TKDE.2020.2981333.
- [13] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” 2013, *arXiv:1312.6203*. [Online]. Available: <http://arxiv.org/abs/1312.6203>
- [14] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. NeurIPS*, 2016, pp. 3844–3852.
- [15] T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. ICLR*, 2017, pp. 1–14.
- [16] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proc. NeurIPS*, 2017, pp. 1024–1034.
- [17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” in *Proc. ICLR*, 2017, pp. 1–12.
- [18] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.
- [19] S. Zhou, J. Bu, X. Wang, J. Chen, and C. Wang, “HAHE: Hierarchical attentive heterogeneous information network embedding,” 2019, *arXiv:1902.01475*. [Online]. Available: <http://arxiv.org/abs/1902.01475>
- [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 *Proc. SIAM Int. Conf. Data Mining*, Philadelphia, PA, USA: 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 *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 399–408.
- [22] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in *Proc. KDD*, 2019, pp. 793–803.
- [23] Z. Hu, Y. Dong, K. Wang, and Y. Sun, “Heterogeneous graph transformer,” in *Proc. Web Conf.*, Apr. 2020, pp. 2704–2710.
- [24] G. Wang, S. Xie, B. Liu, and P. S. Yu, “Review graph based online store review spammer detection,” in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 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 *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2672–2681.
- [26] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. S. Subrahmanian, “REV2: Fraudulent user prediction in rating platforms,” in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 333–341.
- [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 *Proc. AAAI*, 2019, pp. 4424–4431.
- [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,” *CoRR*, vol. abs/2011.12193, 2020. [Online]. Available: <https://arxiv.org/abs/2011.12193>
- [29] H. Chen, Y. Xu, F. Huang, Z. Deng, W. Huang, S. Wang, P. He, and Z. Li, “Label aware graph convolutional network—not all edges deserve your attention,” 2019, *arXiv:1907.04707*. [Online]. Available: <http://arxiv.org/abs/1907.04707>

- [30] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 897–908.
- [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 *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 689–698.
- [32] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 985–994.



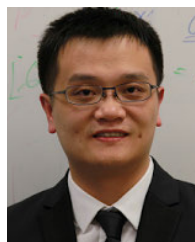
SONGKAI TANG (Graduate Student Member, IEEE) was born in Hunan, China, in 1998. He received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2019, where he is currently pursuing the M.S. degree in computer science and technology.

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, 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.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University.

...