Detecting Telecommunication Frauds by Human-in-the-Loop Graph Neural Networks

Teng Ke, Yang Yang, Shiliang Pu, Xuan Yang, Quanjin Tao, Yifei Sun, Weihao Jiang, Hui Wang, Yingye Yu

Abstract—With the development of the telecommunication industry, telecom fraud becomes a fast-growing criminal activity in recent years, which significantly threatens the security of individual fortune and social wealth. Meanwhile, compared with other types of fraud, telecom frauds are more difficult to be identified. Less than 5% of telecom fraud cases are closed in the real world. In this paper, we aim to solve the challenging telecom problem of fraudster detection. And for that, we employ real-world telecommunication metadata offered by China Telecom to observe characteristics of telecom frauds. According to the observations, we find that both personal attributes (at node level) and call behaviors (at edge level) could provide useful and indispensable information. In addition, we find that some frauds may disguise themselves to reduce their dubiety, which makes the task even more challenging.
Inspired by our empirical observations, we propose a novel model, *graph neural networks for telecom fraud detection (GTF)*, to identify telecom frauds. More specifically, we design a dual attention module to fuse both node-level information and edge-level information. Existing GNNs estimate model parameters only to improve performance, but they fail to present an interpretable process that is necessary for telecom fraud detection. To improve the interpretability of our model, we further propose a *subgraph-level human-in-the-loop* based learning framework, where human annotators will guide the aggregation pattern of our model, enabling it to be closer to human intuitions. Experimental results demonstrate that our proposed model achieves the best performance in telecom fraudster detection tasks (e.g., at least +1.32 in terms of F1) compared with several state-of-the-art baselines. Moreover, we conduct both user studies and case studies to illustrate the clear improvement of the interpretability of our model.

Index Terms—Telecom fraud, Graph neural network, Human-in-the-loop.

1 Introduction

With the development of the telecom industry and the popularization of telecom equipment, telecom fraud becomes a fast-growing criminal activity in recent years, causing great damage to the global community. Millions of people around the world suffer greatly from telecom fraud. As reported in [1], the global economic losses caused by telecom fraud amount to US$32.7 billion annually and are on the rise. In 2021, China sees more than 2,700 telecom fraud cases every day, with a loss of nearly 140 million yuan[1]. In addition to huge economic losses, telecom fraud could also cause severe psychological damage to the victims and thus endanger their lives [2]. What makes it harder is the fact that, compared with other types of frauds, telecom fraudsters are more difficult to catch. According to real statistics[2], less than 5% of telecom fraud cases have been closed.

Although it has caused great damage to our society, few works have studied the problem of identifying telecom frauds from the view of machine learning methodologies. It is mainly due to the difficulty of capturing large-scale mobile data in the real world and the limitation of existing models. To narrow this gap, in this paper, we study a real-world dataset offered by China Telecom[3], which consists of around 9.6 million call logs spanning 30 days in Shanghai, China. According to these call logs, we construct a mobile communication network where nodes indicate users and edges denote calling relationships among users. As for modeling the constructed network, the rapidly developed graph neural networks (GNNs) offer the opportunity for effectively capturing structural features of network data by proposing a recursive neighborhood aggregation scheme. However, applying existing GNNs to the telecom fraud detection task is still facing many challenges.

Firstly, most existing GNN models are not able to explicitly and effectively fuse both node features and edge features in large-scale graphs. However, from the observation results on the real telecommunication network (Sec. 3), we find that both personal information (node features) and call information (edge features) are indispensable in the process of accurately identifying the telecom fraudsters. Thus, the first challenge is how to improve GNN to make it able to fully integrate node information and edge information.

Secondly, as we observed in Sec. 3, some telecom fraudsters may disguise themselves to reduce their dubiety. For example, their personal information may be similar to that of normal users, making it hard for GNNs to extract distinguishable features. Namely, a disguised fraudster may aggregate information from his/her normal neighbors rather than the fraudster's neighbors, leading to an erroneous prediction. Similar situations can be observed in the aspect of call information: users with the same identity (fraud-

- *Teng Ke, Yang Yang, Xuan Yang, Quanjin Tao and Yifei Sun are with the College of Computer Science and Technology, Zhejiang University, China. E-mail: 22021040, yangya, xuany, taoquanjin, yifeisun@zju.edu.cn. Yang Yang is the corresponding author.*
- *Shiliang Pu, Weihao Jiang, Hui Wang and Yingye Yu are with the Hikvision Research Institute, China. E-mail: pushiliang.hri, jiangweihao5, wanghui11, yuyingye@hikvision.com.*

1. http://tv.cctv.com/2021/04/10/VIDERVcybfE4v0VYW1USazzg210410.shtml
2. Reported in China News Service.
3. The largest fixed-line service and the third largest mobile telecommunication provider in China.

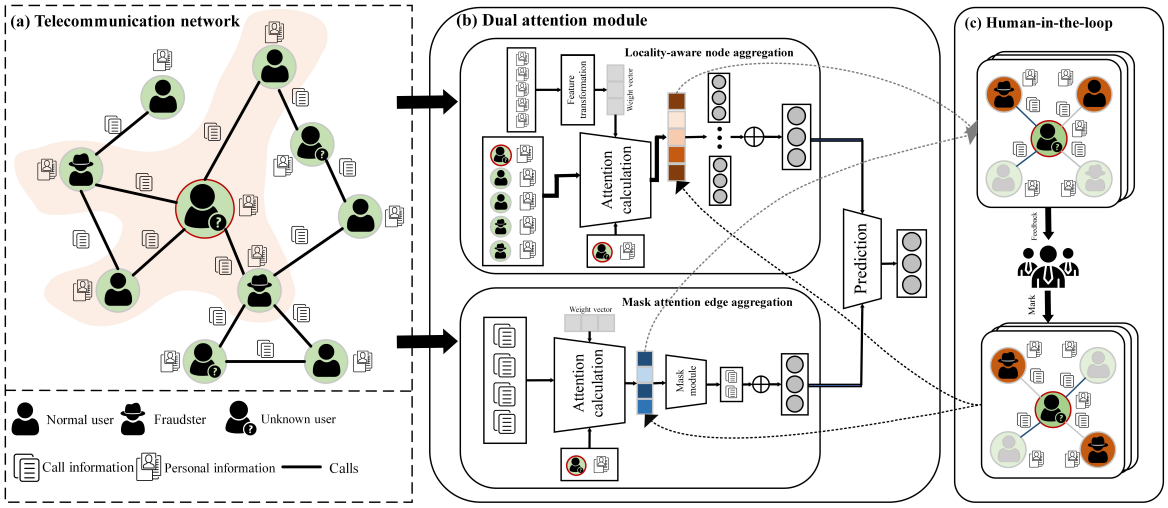


Fig. 1: The overall structure of the proposed approach, with (a) the input telecommunication network, (b) the dual attention module, and (c) the subgraph-level human-in-the-loop learning framework.

ster/normal) may have different calling behaviors. For example, the calls between a normal user and a fraud share the similar characteristic of low frequency with the calls between two normal users who are unfamiliar with each other. For a particular user, how to automatically identify his or her distinguishable features that lurk in the neighbor information and connected edges is challenging in this task.

Thirdly, telecom fraud detection requires *interpretability* serving as necessary evidence for a conviction. However, the traditional learning process of GNNs purely estimates model parameters that achieve the best performance but ignores the objective of being interpretable. Therefore, designing an appropriate learning framework to make the prediction results of GNNs interpretable is our third challenge.

To solve the above challenges, we propose a novel model, *graph neural network for telecom fraud detection (GTF)*, to detect telecom fraudsters in a large-scale telecom network. To fuse both user information and call information (Figure 1(a)), we explicitly leverage both node aggregation and edge aggregation and fuse the embedding from them to form the prediction (Figure 1(b)). More specifically, to encode different node contexts in node aggregation and avoid being misled by the disguised user information of fraudsters, we design a *locality-aware node aggregation* method to learn the attention coefficient of neighbors according to each ego-network respectively. As for the edge level information, we design a *mask attention edge aggregation* method to aggregate the representative edge information and prevent the disguised call records from affecting the aggregation process. Moreover, the existing GNNs estimate model parameters only to improve performance, which fails to present an interpretable process. In our scenario, we propose to describe how our model makes an inference to a user v's identity by providing the subgraph of v's ego-network, which consists of nodes and edges that provide effective information. However, without the guidance of domain experts, one can hardly select understandable subgraphs. To solve this, we further propose a *subgraph-level human-in-the-loop* learning framework (Figure 1(c)) to train our model and improve its interpretability. Extensive experimental results demonstrate a clear improvement has been achieved compared with several state-of-the-art baselines, and our contributions are summarized as follows:

(1) We conduct data analysis on the real-world telecommunication network provided by China Telecom, which discloses the differences between fraudsters and normal users from both node and edge level.
(2) We propose a graph neural network for telecom fraud detection (GTF), which includes locality-aware node aggregation and mask attention edge aggregation, to better extract the distinguishable features and to avoid the influence of the disguise of fraudsters.
(3) We introduce a subgraph-level human-in-the-loop framework to make the detection process interpretable, achieving a superior performance against all baseline methods.

2 Preliminaries

Problem definition. Let $V \in \mathbb{R}^n$ be a set of users, and $E \in \mathbb{R}^m$ be a set of calling relationships between users. $\mathcal{N}(i)$ means neighbors of the user v_i and $\overline{\mathcal{N}}$ means neighbors including himself. Each user $v_i \in V$ has personal information that is denoted as $x_i \in X$, and each calling relationship $e_i \in E$ has calling information that is denoted as $s_i \in S$. Meanwhile, each user in V has a label $y_i \in Y$ denoting whether he is a fraudster $(y_i = 1)$, a normal user $(y_i = 0)$, or an unknown user $(y_i = ?)$. A mobile network $G = (V, E)$ can be constructed by users V and relations E between users. In light of the above, we can define the problem addressed in this paper as follows:

Definition 2.1. *Telecom-fraud detection.* Given a mobile communication network $G = (V, E)$ and an identity vector Y with missing values. Our purpose is to infer the missing values in Y, i.e., to find fraudsters that are lurking among other users.

Data description. Our dataset consists of telecommunication records from September 1st to September 30th in 2016 provided by China Telecom, one of the major mobile service providers in China. Each record contains the anonymous

calling number, the anonymous called number, the starting time, the ending time, etc. Since a user is limited to having only one phone number of China Telecom, we regard a phone number as a user. Meanwhile, We also have access to some personal information such as gender, age, place of birth, etc. of all phone number owners. The ground truth for labeling a user as a fraudster or normal user is derived from Baidu [4] and Qihoo 360 [5] who collect abnormal phone numbers based on reports from users. Specifically, given a phone number, we can check whether the number is abnormal according to the services provided by Baidu and Qihoo 360. Because these services are obtained from a large number of user feedback, the ground truth has high confidence.

We build a directed graph (Chinatel) on these records and personal information. Each node represents a user, and each edge represents two users who have called at least once. The feature of each node represents the user's personal information that has been processed by feature engineering. Correspondingly, the feature of each edge is extracted from all call logs between the two users. The overall statistics of the dataset are summarized in Table 1. The features on the graph are based on the user's personal information and call information, so the graph can be generalized to a generic telecommunication network without the consideration of the mobile operator's influence to a certain extent.

Metric	Statistics
#(users)	290499
#(calling relationships)	1575701
#(call logs)	9599878
#(user information)	261
#(calling features)	37
fraudster rate	4.9%

TABLE 1: Overall statistics of the datasets.

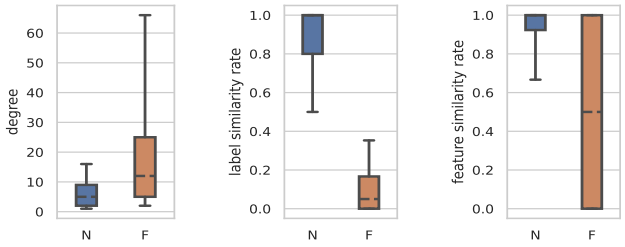
3 Empirical Observations

In this section, we observe the characteristics of fraudsters and normal users through the real-world telecommunication network from two levels: (1) node level; (2) edge level.

3.1 Node Level

We first study the characteristics of fraudsters that distinguish them from normal users at the node level. We extract the node level information of a user based on his ego-network; the subgraph consists of a user and all his neighbors. we analyze fraudsters by considering user degree, user neighbor label similarity, and user neighbor feature similarity.

User degree. User degree reflects how many users a user talks to. To illustrate the difference between fraudsters and normal users in the distribution of degree, we use Box-plot to separately exhibit the 0.1, 0.25, 0.5, 0.75, and 0.9 quantiles of user degree. As expected, Figure 2(a) shows that the degree distribution of fraudsters is higher than that of normal users. According to the calculation, the average degree of fraudsters is 36.28, which is 4 times that of normal users. Moreover, different from normal users, fraudsters' degree distribution is not centralized. These results are consistent with our recognition that fraudsters tend to call more people to cheat for money while normal users mostly talk to a few people they know.



(a) (b) (c)

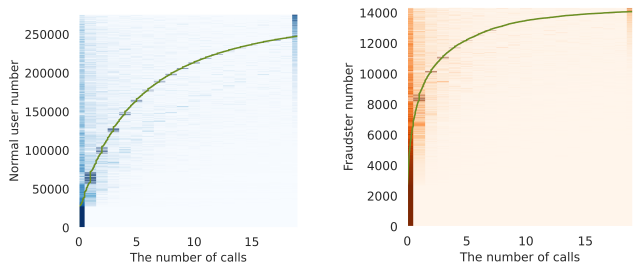
Fig. 2: Node level observation. We use Box-plot to separately exhibit the quantiles of 0.1, 0.25, 0.5, 0.75, and 0.9. (a) represents the degree distribution of normal users (N) and fraudsters (F). (b) represents the distribution of neighbor label similarity rate of normal users and fraudsters. (c) compares the distribution of neighbor feature similarity rate between normal users and fraudsters.

User neighbor label similarity. Social homophily suggests that people tend to develop connections with those who are similar to themselves [3]. To evaluate the neighbor homophily of both fraudsters and normal users, We define the neighbor label similarity rate $\frac{|\{j:j\in\mathcal{N}(i)\wedge y_i=y_j\}|}{|\mathcal{N}(i)|}$, which is the fraction of neighbors that share the same label. Similarly, we use Box-plot to exhibit the distribution of label similarity rate for normal users and fraudsters respectively. According to Figure 2(b), we observe that the rate of fraudsters is much smaller than that of normal users. The median value of the rate is close to 0 for fraudsters but close to 1 for normal users. The result that normal users always have normal neighbors reflects the social homophily phenomenon. However, contrary to this phenomenon, most fraudsters' neighbors are also normal users. The reason is fraudsters, the abnormal nodes in the telecommunication network, seek to scam normal users.

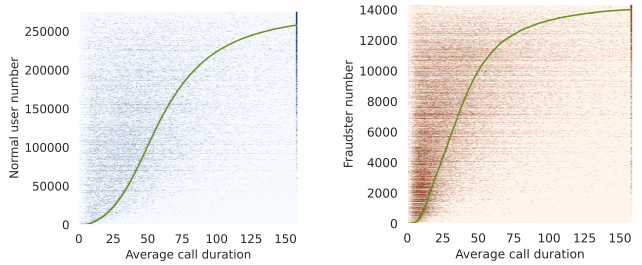
User neighbor feature similarity. To find out whether users with the same identity have similar attributes, we analyze the neighbor feature similarity in the ego-network. Here, we use cosine similarity to measure the similarity of two users. Then, for each user, we get its neighbor cosine similarity distribution. To reflect the distribution numerically, we define neighbor feature similarity rate as $\frac{|\{j:j\in\mathcal{N}(i)\wedge y_i=y_j\wedge Rank(cos(x_i,x_j))<N\}|}{|\{j:j\in\mathcal{N}(i)\}|}$, the fraction of the number of same-label user neighbors whose cosine similarity values are ranked in the top N among all neighbors, where N is the number of same-label user neighbors. Similarly, we use Box-plot to exhibit the distribution of neighbor feature similarity rate for normal users and fraudsters respectively. It is noted that a larger rate means that neighbors with the same label are more similar to the user among all neighbors. As Figure 2(c) shows, for normal users, the distribution of neighbor feature similarity rate is close to 1, that is, neighbors of normal users are similar to them. For fraudsters, the distribution is like a standard uniform distribution, that is, the similarity rate of different fraudsters varies greatly. Some fraudsters are similar to their

4. http://www.baidu.com, one of the largest AI and Internet companies in the world.

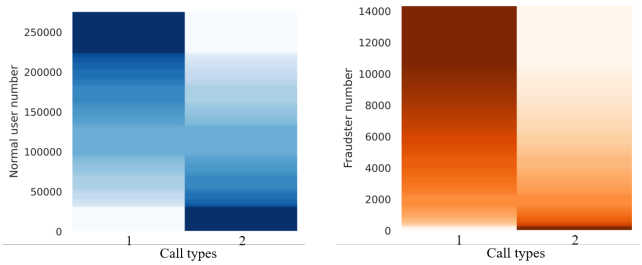
5. http://www.360.com, a Chinese internet security company.



(a) distribution of the number of calls.



(b) distribution of average call duration.



(c) distribution of call types.

Fig. 3: The observation of edge level in Chinatel. We analyze three types of attributes in call information. The blue distribution represents normal users, and the orange distribution represents fraudsters. Deep color means high proportion. The green curve reflects the average value of each user in this attribute. (a) shows the distribution of the number of calls; (b) shows the distribution of average call duration; (c) shows the distribution of call types.

fraudster neighbors, but some fraudsters are more similar to their normal user neighbors. Normal users reasonably have a high neighbor feature similarity rate because they communicate with acquaintances. As for fraudsters, they can disguise their personal information as normal users, which may lead to the uniform distribution of neighbor feature similarity rates.

3.2 Edge Level

To study whether edge-level information (call information) is helpful to determine the identity of users, we analyze three types of calling behavior attributes: the number of calls, average call duration, and call types. These observations are shown by the distribution diagram in Figure 3, where the x-axis represents the attribute value and the y-axis represents the user's index. To better display the distribution, we order users according to the mean value of their attributes.

The number of calls. We count the number of calls made by each user. Since the high number of calls is sparse, we set a threshold, values above the threshold are regarded in the same interval (>threshold). We set the threshold to 20. As Figure 3(a) shows, the distribution of fraudsters is different from that of normal users. For fraudsters, the number of calls is clustered in an interval of less than 5. For normal users, values are widely distributed. The difference can also be seen in the average number of calls of each user, with the average value of fraudsters and normal users exhibiting 3.59 and 9.29, respectively. This observation result is intuitive, from which we can find fraudsters reasonably do not have multiple calls with a single user while normal users tend to communicate with certain acquaintances repeatedly.

Average call duration. We make statistics on the average call duration between two users. Similar to the analysis of the number of calls, we set a threshold to limit the discrete long call duration within the interval (>threshold). The threshold is set to 160. As Figure 3(b) shows, the distribution of average call duration is different between fraudsters and normal users. The average call duration of fraudsters is distributed in the low interval (<25), while the duration of normal users is in a higher interval (25-75). The average call duration of all fraudsters and all normal users is 50.55 and 77.40 respectively, which further illustrates the difference between the average call duration of different subjects. This is consistent with the phenomenon in real-world that the calls from fraudsters are mostly hung up by normal users at the beginning stage.

Call types. We analyze whether fraudsters and normal users have different calling habits. To do this, we define two types of calls: 1) calls that only happen in a continuous short time (3 days), 2) calls that exist in a long-term period. As expected, Figure 3(c) demonstrates that fraudsters have different calling habits compared with normal users. Most fraudsters have the call habit of type 1. This is consistent with our insights that fraudsters prefer to make multiple calls in a short period. However, normal users are more inclined to be categorized as the second call type because normal users mostly keep in contact with someone who they are familiar with for a long time. Note that a certain number of normal users present some calls belonging to type 1. This can be explained by the fact that some normal users, who do not like social activities, will also be likely to make short-time calls within 30 days.

3.3 Conclusion

We summarize the results of the data analysis. At the node level, we conclude that fraudsters tend to have a large degree but a small proportion of fraudster neighbors, meaning that most fraudsters have a large number of neighbors but several of which are fraudster neighbors. Besides, the correlations of personal information between fraudsters and their neighbors are irregular, where the distribution of fraudsters' neighbors' feature similarity rate is like a standard uniform distribution. Meanwhile, normal users have smaller degrees and they follow the social homophily phenomenon because normal users mostly contact normal users with similar personal information. At the edge level, we find that the calling behaviors of fraudsters are different from that of normal users in several aspects, that is, fraudsters make calls less frequently and with a shorter duration and a short-term

period compared with normal users. These are important characteristics to identify fraudsters.

4 Our Approach

In this section, we integrate the insights gained from empirical observations (Sec. 3) into our proposed model, *graph neural networks for telecom fraud detection (GTF)*.

Overview. Motivated by Sec. 3, we adaptively encode the distinguishable features from both node and edge information into the node embedding by a *dual attention module*, while adjusting the aggregation weights by a *subgraph-level human-in-the-loop* framework to make the fraud detection process interpretable.

More specifically, the empirical observations demonstrate that both user information (node features) and call information (edge features) contain the distinguishable yet implicit features that are useful to identify the fraudsters. In order to integrate the dual-source information so as to encode the abnormal patterns of fraudsters, we propose a dual attention module to generate a basic prediction by adaptively aggregating the representation of a target node's neighbors and connected edges. For this purpose, the module utilizes a *locality-aware node aggregation* and a *mask attention edge aggregation* to encode the node and edge information respectively, and fuse them together to produce the prediction. At a certain interval of model training, we sample some ego-networks with high uncertainties from the prediction of the dual attention module as a target requiring manual guidance. For a sampled ego-network, each node (and edge) connected to the center node is marked with a binary symbol to indicate if the node (edge) is assigned by a high aggregation weight. Domain experts are then invited to review these ego-networks, and conveniently adjust the aggregation patterns by flipping the binary symbols. Annotations provided by experts will be fed into the model, which continues to optimize the aggregation weights accordingly. After that, the prediction process is expected to be more interpretable and close to human intuition as the aggregation pattern is partially supervised by domain experts. We illustrate the pipeline of the proposed framework in Fig. 1.

4.1 Dual Attention Module

Node aggregation. From the observation results in Sec. 3.1, we find that a fraudster only has a small proportion of the neighbors who are also fraudsters, i.e., his partners. After the aggregation process of GNNs [4], a fraudster's personal information could be over-smoothed and thus be similar to that of normal users, leading to poor performance. Although the attention-based models such as [5], [6] can aggregate neighbors with adaptive weights, the weights are assigned according to the global pattern of every neighborhood. Besides, as observed in Sec. 3.1, the personal information of the fraudsters may be similar to that of normal users. These disguised fraudsters make the global attention mechanism more difficult to assign suitable attention coefficients to neighbors. Thus, how to design a node aggregation method with the ability to selectively aggregate the distinguishable features for each user (including fraudster and normal user) is the key for us.

To address this challenge, we design a *locality-aware node aggregation* to adaptively aggregate each specific neighborhood so as to encode the local context into the node embedding. To adaptively aggregate different neighbors for each node, we generate a unique weight vector specifically for each ego-network, where we encode local context into the weight vector as follows:

$$a_i^{(l)} = MLP(\frac{1}{|\overline{\mathcal{N}}(i)|}\sum\nolimits_{j\in\overline{\mathcal{N}}(i)} h_j^{(l-1)}) \tag{1}$$

where $h_j^{(l-1)}$ is the node embedding of node v_j in layer $l-1$, $\overline{\mathcal{N}}(i)$ is the neighbors of node v_i including itself, $MLP(\cdot)$ is the multiple layer perception, and $a_i^{(l)}$ is the weight vector of node v_i in layer l. With respect to the different local contexts of nodes, we can obtain the exclusive weight vectors. Moreover, we use different weight vectors to calculate the attention coefficients as follows:

$$\alpha_{i,j}^{(l)} = LeakyReLU(a_i^{(l)^T}[\omega^{(l)}h_i^{(l-1)}||\omega^{(l)}h_j^{(l-1)}]) \tag{2}$$

where $a_i^{(l)}$ is the weight vector of node v_i in layer l which we obtain from Eq. (1), $\omega^{(l)}$ is the l-layer weight parameter matrix, h_i^{l-1} and h_j^{l-1} are node embedding in layer $l-1$, $LeakyReLU(\cdot)$ denotes activation function, and $\alpha_{i,j}^{(l)}$ is the attention coefficient between node v_i and v_j. To get the aggregation weight, we use the softmax function to normalize the attention coefficient as follows:

$$\alpha_{i,j}^{(l)} = \frac{exp(\alpha_{i,j}^{(l)})}{\sum_{k\in\mathcal{N}(i)\cup i} exp(\alpha_{i,k}^{(l)})} \tag{3}$$

Then, we aggregate node information of neighbors with the aforementioned aggregation weights as follows:

$$h_i^{(l)} = \sum\nolimits_{j\in\overline{\mathcal{N}}(i)} \alpha_{i,j}^{(l)}\omega^{(l)}h_j^{(l-1)} \tag{4}$$

Since the target node may be dissimilar to its neighborhood, we combine node embedding and aggregation result of neighbors by concatenation to avoid the over-smoothing problem as follows:

$$h_i^{(l)} = concat(h_i^{(l-1)}, h_i^{(l)}) \tag{5}$$

Edge aggregation. On the other hand, according to the previous edge level analysis in Sec. 3.2, the fraudsters could also disguise themselves by imitating the calling behaviors of the normal users. If these similar behaviors are taken into account, the differences between them will be blurred. In our task, it is essential to find the most representative calling behaviors of each user to detect its identity effectively.

To catch this representative information in the call behaviors, we design a *mask attention edge aggregation*. Generally, we first calculate the attention weights of different calling behaviors for each user. We then select several representative calling behaviors with high coefficients to encode the call information into edge embedding. Thus, some calls made by fraudsters to disguise their calling behaviors as the normal ones can be masked (discarded) and our model can avoid aggregating the misleading features brought by these disguised fraudsters to extract the distinguishable features.

More specifically, to get an initialized edge embedding, we first pretrain the call logs along edges as time series sequences by a CPC model [7] before aggregating edge information. To avoid the influence of the disguised calling behaviors on feature aggregation, we first need to identify the disguised calling behaviors. We propose that disguised abnormal calling behaviors can be detected by referring to the overall calling patterns in the ego-network of each user. Specifically, we get the embedding $\tilde{s}_i$ of the overall calling pattern by aggregating the user's edge features to capture the interaction information in calling behaviors as follows:

$$\tilde{s}_i = \frac{1}{|\mathcal{E}(i)|} \sum\nolimits_{j \in \mathcal{E}(i)} s_j \tag{6}$$

where s_j is the edge embedding of edge e_j, and $\mathcal{E}(i)$ is the set of neighbors of the node v_i. Based on node features and the embedding of the overall calling pattern, we calculate the important coefficients of the edge embedding as follows:

$$\tilde{\alpha}_{i,j} = LeakyReLU(a^T[\omega_n x_i || \omega_e \tilde{s}_i || \omega_e s_j]) \tag{7}$$

where a is a shared learnable attention weight vector, ω_n is the weight parameter matrix of node features and ω_e is the weight parameter matrix of edge embedding. A representative edge embedding s_j of an edge connecting with the node v_i gets a high important coefficient $\tilde{\alpha}_{i,j}$. The lower the important coefficient, the more likely this edge is to be a disguised calling behavior. Therefore, we rank the important coefficient $\tilde{\alpha}_{i,j}$ of each node in descending order as r_i. We select the first k edges with r_i for each node v_i, that is, we get a set $\Omega(i) = \{j_k : \tilde{\alpha}_{i,j_k} \in Topk(r_i)\}$ for each node. The other edges are masked due to the likelihood of being camouflaged. Moreover, we normalize the selected important coefficients of each node as the attention mechanism:

$$\alpha_{i,j} = \frac{exp(\tilde{\alpha}_{i,j})}{\sum_{k \in \Omega(i)} exp(\tilde{\alpha}_{i,k})} \tag{8}$$

Finally, we aggregate selected edge embedding according to the attention score $\alpha_{i,j}$ as follows:

$$z_i = \sum\nolimits_{j \in \Omega(i)} \alpha_{i,j} \omega_e e_j \tag{9}$$

Putting it all together. Given our telecommunication network $G = (V, E, X, S)$ as input, we obtain user node level embedding H^L and user edge level embedding Z after passing a L-layer locality-aware node aggregation and mask attention edge aggregation respectively. To make full use of two aspects of information, we concatenate these two embedding for final prediction as follows:

$$o_i = concat(h_i^L, z_i) \tag{10}$$

where o_i is the final embedding of user v_i. To identify fraudsters, we feed the embedding to a linear transformation layer and a softmax layer for classification as follows:

$$\hat{y}_i = softmax(\omega_f o_i + b_f) \tag{11}$$

where $\hat{y}_i$ is the predicted value of user v_i which indicates the probability of the user being a fraudster. We finally define our loss function as the cross-entropy loss with regularization:

$$\mathcal{L}(\theta) = -\sum\nolimits_{i \in \mathcal{D}} (y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)) + \lambda_1 \left\| \theta \right\|_2^2 \tag{12}$$

where y_i is the label of v_i, θ is the learnable parameter set of our model, λ_1 is the regularizer parameter, and $\mathcal{D}$ is the training set.

4.2 Subgraph-level Human-in-the-loop

Telecom fraud is a criminal activity that requires evidence for a conviction. Thus, the fraud detection task requires the interpretability of the prediction results. In order to present the detection process, one natural way is to show the subgraph aggregation decision of each node. Namely, the higher the aggregation weights, the more likely it is to choose the node or edge for aggregation and provide useful information. Thus, the aggregation weights of each subgraph are the key indicator for the interpretability of our model. However, although GNNs are able to achieve good performance with a large amount of data, the aggregation weights of each ego-network trained by GNNs may be confusing. In other words, plenty of combinations of all aggregation weights can lead to satisfactory performance, but few of them can be close to human intuition. To solve the above problem, we introduce a *subgraph-level human-in-the-loop framework* in the training process of our model.

Unlike most human-in-the-loop (hitl) methods, we feed the aggregation pattern of ego-network of the user predicted by the dual attention module to domain experts. To facilitate the domain knowledge of human experts, we discrete the aggregation weights so that experts only get information about whether or not to aggregate the specific node or edge. In this way, experts can guide the aggregation strategy of predicting the user identity of our module according to domain knowledge. After training, our module is able to leverage a more interpretable strategy to aggregate the subgraphs and predict. We next introduce the details.

During the training process, we sample some nodes (users) from the training set for annotation. The probability of a node being sampled is proportional to its information entropy in the prediction of the dual attention module, i.e., users with more uncertain identities in the model are favored. After sampling, we feed back the ego-networks of these fraudsters to several human annotators. For an ego-network, each node (and edge) connected to the center node is marked with a binary symbol to indicate if the aggregation weight of the node (edge) is higher than a predefined threshold. Annotators are then asked to *flip* some of the binary symbols that they think are incorrect. For instance, an edge indicates abnormal calls from a neighbor who is labeled as fraud tend to be informative. If the edge is marked as "unuseful" by the model, the annotator can provide her suggestion by simply flipping the edge's mark.

The ego-networks with adjusted marks are then sent back to our model for improving their interpretability. To make the aggregation pattern closer to human intuitions, we define a loss function by calculating the similarities between these marks M_{sample} and the aggregation weights P_{sample} obtained by the model as follows:

$$\mathcal{L}_{human} = 1 - \frac{1}{|V_{sample}|} \sum\nolimits_{i \in |V_{sample}|} \frac{M_i P_i}{||M_i|| ||P_i||} \tag{13}$$

where M_i is the marks of the ego-network of the i-th sampled user, and P_i is the aggregation weights of the ego-network of the i-th sampled user. Combing with the dual attention module, the loss function for the entire training process is as follows:

$$\mathcal{L} = \mathcal{L}(\theta) + \lambda_2 \mathcal{L}_{human} \tag{14}$$

where λ_2 is a hyperparameter.

5 Experiments

In this section, we conduct experiments on a real-world telecom dataset (introduced in Sec. 2) and aim to answer the following questions:

- **Q1:** Does the proposed GTF model perform effectively?
- **Q2:** How does each component of the dual attention module contribute to the detection task?
- **Q3:** Is the detecting process of GTF interpretable?

5.1 Experimental Setup

To meet the demand for practical scenarios, we sample 60% of normal users and fraudsters for training respectively, and we test different methods on the remaining users. We also regard 1/3 of users from the training set as a validation set, to avoid overfitting. The ratio of fraudsters to normal users is the same in both the training set, validation set, and test set. For evaluation, we use the following metrics: Precision, Recall, and F1 score, which are commonly used in an imbalanced classification task. Considering the unbalanced nature of our labels, we pay more attention to the F1 score.

Baselines. To comprehensively validate the effectiveness of GTF, we compared it with several different types of baselines.

- Traditional method. The first type of baseline is the traditional classifiers. We select MLP and XGBoost [8], where MLP$_N$ and XGBoost$_N$ take personal information as input to identify fraudsters, and MLP$_E$ and XGBoost$_E$ evenly aggregate call information for each user as input.
- Basic GNNs. The second type of baseline is the basic graph neural network, which aggregates the personal information of neighbors evenly. We select three classic models among them: GCN [4], SGC [9], and GIN [10].
- Attention-based GNNs. The third type of baseline is the attention-based graph neural network, which calculates attention coefficients to aggregate with weights. We select three representative methods, including GAT [5], GATv2 [11], and AGNN [6].
- GNNs with heterogeneous. The fourth type of baseline is the graph neural network with heterophily graphs, and we select GraphSage [12], FAGCN [13], and H2GCN [14]. These methods try to alleviate the smoothing of node features by neighbors with different labels.
- Edge-featured GNNs. The fifth type of baseline is the graph neural network using edge features. We select GAT with edge features[6]. There still remain some other edge-featured methods, i.e. EGNN [15] and CensNet [16], which are conducted on small datasets. Under our large-scale network, the parameters of these methods are too large for our experimental environment, so we do not choose these methods as baselines.
- Telecom fraud detection. We further compare the existing telecom fraud detection method FFD [17] with our model.

6. see https://pytorch-geometric.readthedocs.io/en/latest/index.html for details.

Method	Precision	Recall	F1	Method	Precision	Recall	F1
MLP_N	66.68	71.02	68.78	$XGBoost_N$	**80.54**	59.72	68.59
MLP_E	43.16	38.24	40.55	$XGBoost_E$	73.39	23.57	35.68
GCN	60.68	61.62	61.15	SGC	59.40	58.57	58.98
GIN	57.58	61.64	59.43	GAT	66.30	62.68	64.44
GATv2	67.69	64.33	65.97	AGNN	62.50	62.87	62.69
GraphSage	68.12	67.08	67.60	FAGCN	70.50	65.97	68.16
H2GCN	69.97	67.97	68.96	GAT-E	60.80	63.79	61.12
FFD	67.86	53.07	59.56				
GTF	68.95	**71.66**	**70.28**				

TABLE 2: Performance (%) of identifying fraudsters. The bold indicates the best performance of all methods.

Implementation details. We set hyperparameters in GTF: (1) in the human-in-the-loop framework, we sample 100 fraudsters in the training set for every 20 epochs; (2)for the dual attention module, we adopt 2-layer locality-aware node aggregation and 1-layer mask attention edge aggregation with K being set as 10. we implement all deep learning methods with PyTorch [18] and all GNN methods with PyTorch-Geometric [19]. For XGBoost, we implement it with scikit-learn [20]. For FFD, we use resource code in [17]. Besides, we use the Adam optimizer [21] with a learning rate of 0.001, and the regularizer parameter is 0.0001. The batch size is set as 1024, and the hyperparameter of human loss λ_2 is 0.1. We optimize hyperparameters of all methods on the validation set. All experiments are carried out under Ubuntu 18.04.6 operating system, equipped with an Intel Xeon Gold 6240 CPU and a single Nvidia GTX 2080Ti GPU.

5.2 Performance Comparison

We first compare the experimental results of GTF with that of other baselines to answer **Q1**. As shown in Tab. 2, our model, GTF, achieves better performance than all baseline methods and improves the F1 score to 1.32. Simple classification methods for personal information, including MLP and XGBoost, perform normally, which demonstrates that the personal information we extracted through observation is very effective in distinguishing fraudsters and normal users. Besides, although the performance of user average call information is worse than that of personal information, using call information to distinguish fraudsters can reach an F1 of about 40 in an unbalanced dataset, where fraudsters only take up a small proportion of total users. It indicates that call information is a significant characteristic to detect fraudsters. The basic GNNs, including GCN, SGC, and GIN, perform poorly. As stated above, the reason is that neighbors of fraudsters are mostly normal users and mean aggregation will over-smooth fraudsters' personal information. Attention-based GNNs, including GAT, GATv2, and AGNN, get better performance (an average +4.51 improvement of F1) by attention mechanism, which could selectively aggregate effective neighbor information. However, the performance is still 5.91 lower than the results achieved by our model on average in terms of F1. The reason is that the global attention, as we stated above, can not adaptively aggregate fraudster neighbors for each

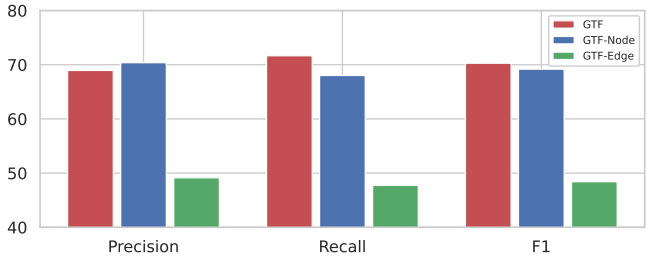


Fig. 4: Results of ablation study. (-) means part module with hilt. For example, GTF-Node means Locality-aware node aggregation module training with hilt framework.

fraudster because there is a mutual disguise between fraudsters. GraphSage, FAGCN, and H2GCN consider processing neighbor messages instead of conducting direct aggregation, which alleviates over-smoothing of node features by neighbors. These methods achieve greater performance than basic GNNs. Among them, H2GCN, a method that concatenates multi-layer aggregating information instead of conducting direct aggregation, obtains the highest F1 score among all baselines. However, H2GCN does not consider the different importance of neighbors in the same hop. As Fig. 4 shows, our submodule GTF-Node, a locality-aware node aggregation method that aggregates neighbors with the same identity, also outperforms H2GCN. Edge-featured GNNs leverage edge features during aggregation. GAT with edge feature uses edge features when calculating attention coefficients. The performance of it is worse than that of the normal GAT (-3.32 of F1). This demonstrates that randomly putting personal information and call information together to calculate aggregation weights can even lead to learning the wrong weights. As a traditional method for telecommunication fraud detection, FFD is inferior to GNNs as it manually extracts features of the graph structure.

5.3 Model Effectiveness

Ablation study. We evaluate the performance of each module in the dual attention module to answer **Q2**. As Fig. 4 shows, GTF-Node (locality-aware node aggregation) has some performance degradation compared to GTF, which proves that edge features have a significant improvement on the model performance. Besides, GTF-node gets the best performance compared with all other GNN baselines. GTF-Edge (mask attention edge aggregation) has the best performance compared with all other baselines that only use edge features in Tab. 2. Therefore our node and edge aggregation methods are able to aggregate more effective information.

Locality-aware node aggregation. To demonstrate that the improvement comes from our locality-aware node aggregation method, we follow the experiment in [22] to analyze the learning ability of label-agreement in four methods (GTF-Node, GATv2, GAT, and AGNN) for fraudsters in the test set, where the ideal attention should give all weights to fraudster neighbors. In [22], they take label agreement vectors as ground truth and use Kullback-Leibler divergence to measure the similarity between aggregation weights and label agreement vectors. As Figure 5 shows,

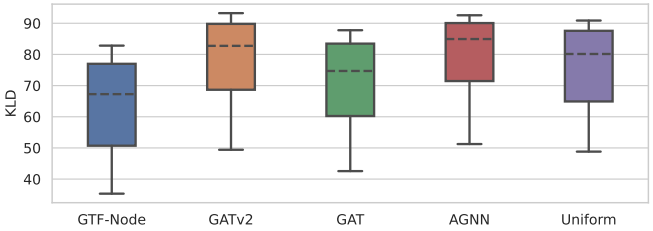


Fig. 5: Distribution of KL divergence.

the KLD distribution of GTF-Node is lower than that of all other methods, and UNIFORM stands for mean aggregation like GCN. It illustrates that GTF-Node has a better learning ability for label agreement. Besides, KLD distributions of global attention methods are similar to that of UNIFORM. It shows that when fraudsters disguise features mutually, the methods of calculating aggregation weights globally do not effectively assign high weights to connected fraudsters.

Mask-attention edge aggregation. GTF-Edge selects edges to aggregate representative call information to distinguish fraudsters. To validate the effectiveness of the selecting mechanism on GTF-Edge, we test performance with different K values. As Fig. 6 shows, low values and high values of K result in poor performance, but GTF-Edge with K close to 10 (average degree of fraudsters) has good performance. This is because the information is not fully captured when K is small. When K is large, invalid information observed in Sec. 3.2 is aggregated and representative information is over smoothed, which proves that our GTF-Edge can effectively capture representative information.

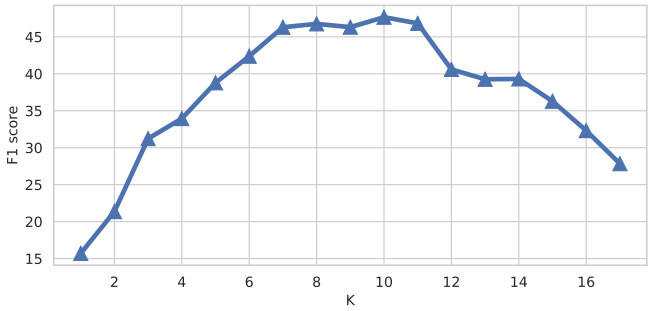


Fig. 6: Performance of GTF-Edge with different K values.

Hyperparameter analysis of hilt. In order to demonstrate the importance of the subgraph-level hilt framework, we conduct experiments with different values of the parameter of hilt loss λ_2. As fig. 7 shown, the hilt framework indeed improves the performance of the model (+0.34 of F1). As λ_2 gradually increases, the F1 of GTF keeps increasing and attains the maximum value when λ_2 takes 0.1. When the value of λ is greater than 0.2, the performance of GTF decreases instead, which may be caused by the fact that the value of hilt loss becomes larger and affects the optimizer to optimize the cross-entropy loss $L(\theta)$.

5.4 Simulation Experiment

To further demonstrate the effectiveness of the human-in-the-loop (hitl) framework, we conduct simulation experiments on the synthetic dataset.

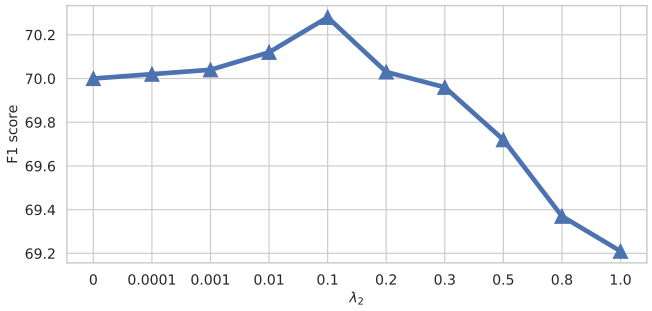


Fig. 7: Performance of GTF with different values of λ_2.

The synthetic dataset is a graph generated according to certain rules. We randomly generate a graph that consists of 2277 nodes and 31371 edges, of which 1756 nodes are positive and 521 nodes are negative. For node features, we generate two different kinds of feature distribution based on node labels. For edge features, we also generate two different distributions, and for each positive node, we randomly assign positive features to more than half of its neighbor edges and negative features to the rest of the neighbor edges, which is the same for each negative sample. In this way, we have prior knowledge that parts of the node's ego-graph should be aggregated when GNN aggregates this node. So our domain experts in the hitl framework can conduct the following operations with this prior knowledge and the accuracy is 100%.

To effectively verify whether hitl can help our dual attention module learn aggregation weights, we delete the feature linear transformation parameters and only retain the parameters that are used for calculating weights in the model, and we assign 2-dimensional one-hot encoding based on positive and negative features. Also, to prevent the model from being fitted before the aggregation weights are learned completely, we delete some parts: the concat operation in the GTF-Node and node features used in the calculation of aggregation weights in GTF-Edge.

To better reflect the effect of hitl, we add different error rates to the marked ego-network. As Fig. 8 shows, the addition of hitl to all three models improves the ability to learn the ideal aggregation weights. Meanwhile, when the error rate is small, adding hitl still achieves an improvement in performance. As the error rate increases, the introduction of error information leads to a dramatic decrease in model performance.

(a) GTF (b) GTF-Node (c) GTF-Edge

Fig. 8: Performance of the proposed model on the synthetic dataset by varying different error rates of the simulated domain expert.

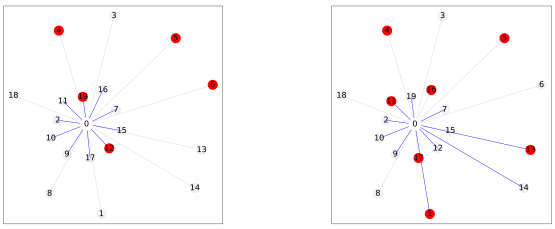
5.5 Interpretability

User study. In order to validate the improvement of interpretability of the prediction process after introducing sub-graph level human-in-the-loop (hitl), we conduct a user study. To this end, we invite some domain experts to score the aggregation patterns derived from the models with and without hitl, respectively. For every aggregation pattern, experts give a score ranging from 0 to 4 based on empirical knowledge. A higher score means that the aggregation pattern meets the expectation of the expert. We randomly sample 1000 users from the test set for expert evaluation. The specific statistics are shown in Tab. 3. It can be seen that our module with the hitl framework gets a higher score than that without the hitl, especially the proportion of scores greater than 2 exceeds 75%. This illustrates that the aggregation patterns obtained by the model trained with the hitl framework are more interpretable.

Model	0	1	2	3	4
W hitl	0.10	2.70	20.50	39.20	37.50
W/O hitl	1.50	9.70	35.30	33.40	20.10

TABLE 3: Distribution of scores in user study marked by human annotators.

Case study. To further illustrate the interpretable prediction process of our dual attention module, we present a specific case of a fraudster identified correctly by our model. In Fig. 9, We draw the ego-network of this fraudster. Based on the aggregation weights output by the two models, we mark neighbor nodes (red) and edges (blue). As Fig. 9 shows,



(a) GTF (b) GTF-hitl

Fig. 9: Ego-graph marked by aggregation weights.

the marked nodes and edges are quite different between Fig. 9(a) and Fig. 9(b). For neighbor nodes, the marked nodes in Fig. 9(a) contain almost fraudster neighbors and only have one normal user, while the number of the marked nodes in Fig. 9(b) is more than that of the marked nodes in Fig. 9(a) and their labels are a mess. It obviously illustrates that the node aggregation pattern that is output by the model with hitl is more consistent with human understanding. For neighbor edges, the marked edges also have a difference between Fig. 9(a) and Fig. 9(b). We find that the marked edges in Fig. 9(a) are calls that match the abnormal characteristics which are that fraudsters make calls less frequently and with a shorter duration compared with normal users. For example, the model with hitl selects the edge connected to node 16, while the model without hitl does not select this. By observing the call logs, we find that there is only one short call between node 0 and node

16 in a month. Therefore, our introduction of hitl makes the prediction process of our model interpretable.

6 Related work

Graph Neural Networks. The general paradigm of GNNs is node feature transformation and aggregation of features of neighbor nodes alternately. As pioneering works, [4], [23] generate neighbor embedding by mean aggregation. [5], [6], [11], [22] introduce an attention mechanism into GNN that aggregates the most relevant neighbors. In addition to these attention methods that consider aggregate weights, some other methods improve GNNs by the way of introducing edge features. [15], [16] utilize edge features in neighbor aggregation to obtain more effective node embeddings. However, these methods are inefficient. [15] regards edge features as multi-dimensional weights. The dimension of final node embedding becomes huge when the dimension of the edge features is large. [16] uses a line graph to convert edges to nodes and alternately aggregates node features and edge features. With the high time complexity, these methods cannot handle large-scale graphs. We propose a dual attention mechanism to aggregate nodes and edges simultaneously and efficiently even on large-scale graphs.

Fraud detection on graphs. Fraud detection on graphs has attracted considerable research efforts recently [24], [25], [26], [27], [28], [29], [30], [31]. For example, [32] proposes a GNN-based imbalanced learning approach to solve heavily skewed label distribution problem in fraud detection. [33] uses a semi-supervised GNN model with a hierarchical attention mechanism for explainable fraud prediction in financial fraud detection. However, there are few works that focus on telecom fraud scenarios. Instead, [17] proposes a traditional factor graph model to distinguish frauds in the telecommunication network. Therefore, using GNN to solve telecom fraud detection is a promising exploration.

Human-in-the-loop. With the increasing recognition of human-centered AI as a new paradigm for AI, human-in-the-loop (hitl) has emerged to enable collaborative human-machine-driven decision making [34], [35], [36]. With the involvement of humans, deep learning models are more likely to meet human expectations in specific tasks. Thus, the model can have excellent performance while allowing the inference process to be interpretable. Previous work has studied hitl learning in related recommendation [36], [37], image [38], and medicine [39]. In fact, there are few works that combine hitl and GNNS. As we know, the prediction process of GNN is difficult to explain. But hitl can provide interpretability. It is a necessary research topic to make the prediction process interpretable by adding hilt to GNN.

7 Conclusion

In this paper, we study the problem of telecom fraud detection and analyze the difference between fraudsters and normal users from the node level (personal information) and the edge level (call information). We propose a novel graph neural network model, GTF, to identify telecom frauds. Specifically, the model consists of a dual attention module to fuse both node-level information and edge-level information, and a subgraph-level human-in-the-loop based learning framework to improve the model's interpretability. When evaluated on real-world datasets, the proposed method not only achieves significantly better results than other baselines but also generates an interpretable process. In the future, we intend to apply our model GTF to cross-network. Cross-network is a telecommunication network that contains different telecom operators' numbers and calls between them, which is more in line with the telecommunication network in the real world. In this way, our model GTF can be generalized to cross-network fraud and fully applied to the real world.

References

[1] E. E. C. Centre, "Cyber-telecom crime report 2019," 2019.

[2] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, and S.-Y. Chen, "Generative adversarial network based telecom fraud detection at the receiving bank," *Neural Networks*, vol. 102, pp. 78–86, 2018.

[3] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[6] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," *arXiv preprint arXiv:1803.03735*, 2018.

[7] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[8] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[9] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.

[10] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.

[11] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[12] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.

[13] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 3950–3957.

[14] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[15] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9211–9219.

[16] X. Jiang, P. Ji, and S. Li, "Censnet: Convolution with edge-node switching in graph neural networks." in *IJCAI*, 2019, pp. 2656–2662.

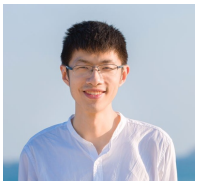
[17] Y. Yang, Y. Xu, Y. Sun, Y. Dong, F. Wu, and Y. Zhuang, "Mining fraudsters and fraudulent strategies in large-scale mobile social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 169–179, 2019.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[19] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.
[22] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with self-supervision," in *International Conference on Learning Representations*, 2020.
[23] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," in *Proceedings of ICLR'16*, 2016.
[24] V. S. Tseng, J.-C. Ying, C.-W. Huang, Y. Kao, and K.-T. 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*, 2015, pp. 2157–2166.
[25] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "Fraudar: Bounding graph fraud in the face of camouflage," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 895–904.
[26] J. M. Kleinberg, "Hubs, authorities, and communities," *ACM computing surveys (CSUR)*, vol. 31, no. 4es, pp. 5–es, 1999.
[27] M. Onderwater, "Detecting unusual user profiles with outlier detection techniques," 2010.
[28] 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 *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM'20)*, 2020.
[29] K. Ding, Q. Zhou, H. Tong, and H. Liu, "Few-shot network anomaly detection via cross-network meta-learning," in *Proceedings of the Web Conference 2021*, 2021, pp. 2448–2456.
[30] Y. Li, X. Huang, J. Li, M. Du, and N. Zou, "Specae: Spectral autoencoder for anomaly detection in attributed networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2233–2236.
[31] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, "Spam review detection with graph convolutional networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2703–2711.
[32] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, "Pick and choose: A gnn-based imbalanced learning approach for fraud detection," in *Proceedings of the Web Conference 2021*, 2021, pp. 3168–3177.
[33] 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.
[34] G. Li, "Human-in-the-loop data integration," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 2006–2017, 2017.
[35] Y. Zhang, F. Zhang, P. Yao, and J. Tang, "Name disambiguation in aminer: Clustering, maintenance, and human in the loop." in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1002–1011.
[36] Z. Fu, Y. Xian, Y. Zhu, S. Xu, Z. Li, G. De Melo, and Y. Zhang, "Hoops: Human-in-the-loop graph reasoning for conversational recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2415–2421.
[37] Z. Fu, Y. Xian, S. Geng, G. De Melo, and Y. Zhang, "Popcorn: Human-in-the-loop popularity debiasing in conversational recommender systems," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 494–503.
[38] S. Reddy, A. Dragan, and S. Levine, "Pragmatic image compression for human-in-the-loop decision-making," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[39] E. Fosch-Villaronga, P. Khanna, H. Drukarch, and B. H. Custers, "A human in the loop in surgery automation," *Nature Machine Intelligence*, vol. 3, no. 5, pp. 368–369, 2021.

Teng Ke is currently a master student of the College of Computer Science and Technology, Zhejiang University, China. He received his bachelor's degree from the College of Electronics and Information Technology, Sun Yat-sen University. His main research interests include data mining and graph anomaly detection.



Yang Yang received his Ph.D. degree from Tsinghua University in 2016. He is an associate professor in the College of Computer Science and Technology, Zhejiang University. His main research interests include data mining and social network analysis. He has been visiting scholar at Cornell University and Leuven University. He has published over 40 research papers in major international journals and conferences including: KDD, WWW, AAAI, TKDE and TOIS.

Shiliang Pu received his Ph.D. degree from Zhejiang University. He is the chief expert of Hikvision and the director of Hikvision Research Institute. He has received the Qiu Shi outstanding youth achievement transformation award from the China Association for Science and Technology (CAST). His main research interests include artificial intelligence and large visual data.

Xuan Yang is currently a master student of the College of Computer Science and Technology, Zhejiang University, China. She received her bachelor's degree from the College of Computer Science and Technology, Zhejiang University. Her main research interests include data mining, graph modeling and computational social science. She has been published in TKDE.

Quanjin Tao is currently a master student of Polytechnic Institute of Zhejiang University, China. He received his bachelor's degree from Pharmaceutical preparation, Tianjin Medical University. His main research interests include data mining and graph modeling.

Yifei Sun is currently a CS Ph.D candidate at Zhejiang University, China. He received his CS bachelor degree and Japanese bachelor degree from Dalian University of Technology, in 2019. His current research interests include data mining and graph modeling.

Weihao Jiang received his Ph.D. degree from Chinese academy of social science. He is a senior algorithm director of the big data intelligence department, Hikvision Research Institute. His main research interests include data mining and knowledge graph.

Hui Wang received his master's degree from Fuzhou University. He is a senior algorithm manager in the big data intelligence department, Hikvision Research Institute. His main research interests include data mining, intelligence optimum algorithms and large scale graph modeling.

Yingye Yu received her master's degree from City University of Hong Kong. She is a senior engineer in the big data intelligence department, Hikvision Research Institute. Her main research interests include data mining and large scale graph modeling.