FA-GNN: Filter and Augment Graph Neural Networks for Account Classification in Ethereum

Jieli Liu, Jiatao Zheng, Jiajing Wu, *Senior Member, IEEE*, and Zibin Zheng, *Senior Member, IEEE*

***Abstract*—As the first blockchain platform supporting smart contracts, Ethereum has become increasingly popular in recent years and generates a massive number of transaction records. Nowadays, millions of accounts in Ethereum have been reported to participate in a variety of businesses, and some of them have been found to be involved in illegal behaviors or even cyber-crimes by exploiting the pseudonymous nature of blockchain. Therefore, there is an urgent need for an effective method to conduct account classification and audit transaction behaviors of each account. In this paper, we model the Ethereum transaction records as a transaction network, and the account classification problem is converted to a node classification problem. Based on the Ethereum transaction network, we propose a novel framework named Filter and Augment Graph Neural Network (FA-GNN), which can retain the information of important neighbors and augment node features with high-order information. Experimental results demonstrate that our proposed FA-GNN outperforms state-of-the-art methods in Ethereum account classification.**

***Index Terms*—Blockchain, complex network, ethereum, network embedding, transaction network.**

I. Introduction

ETHEREUM [1] is currently the largest blockchain platform enabling Turing-complete smart contracts, and its cryptocurrency Ether (abbreviation ETH), now is holding a market cap of more than 200 billion dollars according to CoinMarketCap.com. As the fundamental technology of Ethereum, blockchain [2] provides a distributed environment for cryptocurrency transactions with no need of trusted third parties by employing cryptography, peer-to-peer transmission, and consensus mechanism. Participants in blockchains are pseudonymous and trade without identity verification, which is referred to as the privacy-preserving nature of blockchain. Besides, all cryptocurrency transaction records are irreversible and publicly accessible, and these records are maintained by each peer in the blockchain network.

Manuscript received November 18, 2021; revised March 4, 2022; accepted April 1, 2022. Date of publication April 12, 2022; date of current version June 27, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 62032025 and 61973325, and in part by the Natural Science Foundations of Guangdong Province under Grant 2021A1515011661. Recommended for acceptance by Dr. Caterina Scoglio. *(Corresponding author: Jiajing Wu.)*

Jieli Liu and Zibin Zheng are with the School of Software Engineering, Sun Yat-sen University, Zhuhai 519082, China (e-mail: liujli7@mail2.sysu.edu.cn; zhzibin@mail.sysu.edu.cn).

Jiatao Zheng and Jiajing Wu are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zhengjt7@mail2.sysu.edu.cn; jjingwu@outlook.com).

Digital Object Identifier 10.1109/TNSE.2022.3166655

However, every coin has two sides. The pseudonymous nature of blockchain has also been exploited in rampant cybercrimes, causing huge losses to blockchain users. During 2021, more than 14 billion dollars had been lost due to cryptocurrency security incidents according to the report [3] from a famous blockchain security company named Chainalysis. Specifically in Ethereum, a variety of financial scams have been reported. Based on the statistics of Etherscan's label word cloud,[1] the known categories to which pseudonymous accounts belong include gambling, mining, exchange, phish, hack, etc. Therefore, it is necessary to distinguish different categories of pseudonymous accounts in Ethereum, which can assist to identify abnormal transaction behaviors and provide more clues for transaction tracking, improving the transparency and auditability of the whole system.

Networks are an effective tool to describe the relationships between objects in various real-world systems, and a series of prior studies have modeled cryptocurrency transaction records as complex networks and investigated cryptocurrency transaction data from a network perspective [4]–[8]. Unlike Bitcoin [9], Ethereum adopts the account-centered model [10] as its transaction model, and each transaction can be regarded as a cryptographically signed operation from one account to another account, such as money transfer and contract invocation. Thus the Ethereum cryptocurrency transaction records can be straightforwardly represented in the network form with accounts as nodes and transactions as edges.

Existing studies on cryptocurrency transactions from a network perspective are facing great challenges brought by the sparsity and large scale of the network. As a fundamental tool for large-scale network analysis, network embedding technology aims to map node features into a low-dimensional vector space, so that the downstream tasks such as node classification and link prediction can take benefits from the low-dimensional embedding and be accomplished with ready-made machine learning methods. Recently, graph neural network (GNN) methods including graph convolutional network (GCN) [11], GraphSAGE [12], and Cluster-GCN [13] have shown promising performance for various downstream applications. These methods can capture the dependence in networks via deep learning techniques and message passing mechanisms among nodes of networks.

Given the available labeled information, the task of Ethereum account classification can be formulated as a node

[1] https://etherscan.io/labelcloud

classification task in the Ethereum transaction network. With handcrafted features, Farrugia *et al.* [14] developed a classifier to detect illicit accounts over the Ethereum transaction network. To analyze and classify the Ethereum smart contract accounts, Hu *et al.* [15] trained a Long Short-Term Memory (LSTM) network with features extracted from manual analysis. Since the process of finding effective handcrafted features depends on human experience, researchers turn to applying network embedding techniques to automatically capture the account interaction features in networks. In [16], a network embedding algorithm called trans2vec was designed to extract the features of the addresses for subsequent phishing identification. Chen *et al.* [17] proposed a cascade feature extraction method for phishing account detection in Ethereum, which can not only use the features of an account, but also use the features of its neighbors. The idea of cascade feature extraction is similar to the feature aggregation process of GraphSAGE. In [18], GCN method is utilized to learn account features in the phishing detection task. These studies are the first try of GCN models on blockchain transaction networks. However, they do not modify the original GCN models according to the characteristics of blockchain transaction networks.

When applying GCN models on the Ethereum transaction network, there exist two main challenges. *The first challenge comes from the account features*. Since the pseudonymous nature, the real identity of each account is concealed in Ethereum. Thus the account profiles are mainly determined by the account balances and transaction relationships, making this task different from traditional account classification in social networks and transaction networks. *The second challenge is brought by the transaction relationships between two quite dissimilar accounts, which is common in a payment system*. For example, a hot wallet account of exchange usually trades with different services and users. Thus directly aggregating the neighborhood information may introduce noises to the feature learning process. To sum up, how to select the neighbor subset and aggregate more effective information from the neighborhood of target nodes is the key issue that can greatly affect the performance of a GCN model on the Ethereum transaction network.

In this paper, we propose a novel GNN framework named **F**ilter and **A**ugment **G**raph **N**eural **N**etwork (FA-GNN) to tackle the account classification task in Ethereum transaction network. First, we construct an Ethereum transaction network with real-world transaction records and conduct network property analysis on it. Based on the obtained network properties, we then design two key components for network embedding learning, namely neighbor filtering and feature augmentation. The neighbor filtering component aims to retain important neighbor information according to three proposed transaction network-based preference strategies. The feature augmentation component enhances node features with higher-order neighborhood information. Finally, we combine the proposed neighbor filtering component, feature augmentation component, and deep learning techniques to build a novel Ethereum transaction network embedding model FA-GNN. Experimental results on real-world transaction records demonstrate the effectiveness of FA-GNN for the account classification task in Ethereum.

The main contributions of this paper are summarized as follows:

1) We comprehensively analyze the network properties of the Ethereum transaction network and propose two measurement metrics of feature similarity. An important insight we obtained is that there exist great differences between connected nodes in the network.
2) Based on the strong heterophily nature of the Ethereum transaction network, we propose the neighbor filtering and feature augmentation component in our framework to tackle the Ethereum account classification problem, which can identify the influential neighbors and generate enhanced features in the Ethereum transaction network.
3) We conduct extensive experiments to evaluate the performance of our proposed framework on real-world Ethereum transaction data. The experimental results show the effectiveness of the proposed framework in Ethereum account classification.

The remaining sections of this paper are organized as follows. In Section II, we discuss the problem formulation and conduct network analysis on the constructed Ethereum transaction network. According to the analysis results, we then propose our model FA-GNN in Section III. In Section IV, we empirically evaluate FA-GNN on the Ethereum account classification task. Finally, we conclude the whole paper in Section V.

II. Problem Formulation and Network Analysis

In this section, we first discuss our problem formulation for Ethereum account classification, and then conduct a network property analysis on the network-form transaction data. From the analysis, we can understand the characteristics of Ethereum transaction network, which assists us in designing an appropriate model.

A. Problem Formulation

The Ethereum transaction data can be straightforwardly constructed into a network $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each node $v \in V$ represents an Ethereum account and carries a m dimensional node feature vector. The node features are represented in a feature matrix $X \in \mathbb{R}^{|V| \times m}$. Each edge $e = (u, v) \in E$ represents that there exist transaction relationships between node u and node v. The Ethereum account classification task is a semi-supervised node classification task where only a subset of nodes has ground truth labels available on Etherscan's label word cloud.

Our goal is to classify the category of each node in the graph by developing a GCN model. The architecture of the original GCN model [11] for each layer can be summarized by $H^{l+1} = \sigma(\hat{A}H^l W^l)$, where $\hat{A}$ is a normalization of the adjacency matrix, H^l contains the hidden representation of

TABLE I
STATISTICS OF THE ETHEREUM DATASET

Nodes	Edges	Labeled nodes	Classes	Features
1,402,220	2,815,028	816	6	12

TABLE II
DISTRIBUTION OF BUSINESS LABELS IN THE COLLECTED ETHEREUM TRANSACTION DATASET

Mining	Phish/hack	ICO	Exchange	Converter	Gambling
60	252	167	171	139	27

each node in layer l, H^0 is the given node feature matrix X, W^l is a weight matrix and σ is the nonlinear activation function. Since this model is designed in a transductive setting and requires that the full graph Laplacian is known, Hamilton *et al.* [12] proposed a *spatial-based* extension of the original GCN framework named GraphSage, which generates node representation by sampling and aggregating the neighborhood features of a node. For each node v, the updating process of the hidden representation in layer l can be formalized as:

$$h_v^{l+1} = combine(h_v^l, aggregate(\{h_u^l, u \in S(v)\})), \quad (1)$$

where h_v^l is the hidden representation of node v in layer l, h_v^0 is initialized as the m dimensional node feature vector of v, $S(v)$ is the node set containing the sampled neighbors of v, $aggregate(\cdot)$ is the aggregation function such as mean pooling and max pooling, and $combine(\cdot)$ is the combine function such as linear-layer combination. Besides, recent years many other spatial-based GCN frameworks have been proposed to handle the node classification tasks in large-scale networks [12], [19], [20], temporal networks [21], [22], heterophilous networks [23], [24], etc. To tackle the Ethereum account classification problem, this paper proposes a spatial-based GCN model according to the properties of the Ethereum transaction network.

B. Dataset and Network Analysis

The openness of blockchain provides facilities in acquiring cryptocurrency transaction data. In this paper, we collect the Ethereum account labels and the transaction records through a famous Etherscan blockchain explorer Etherscan in the same way as the previous study [25]. Then, the transaction records are constructed into a network $G = (V, E)$, where V is the set of nodes representing Ethereum accounts, and E is the set of edges indicating that there exist transaction relationships between each pair of connecting nodes. The statistics of the Ethereum dataset are presented in Table I. As shown in Table I, the constructed Ethereum transaction network contains 1,402,220 nodes and 2,815,028 edges. For each node, 12 features such as transaction amount, account balance, number of transactions, and frequency of transactions are extracted. Among the collected nodes, 816 are labeled, and these labels fall into 6 different categories including mining, phish/hack, Initial Coin Offering (ICO), exchange, converter, and gambling. The distribution of the node labels is shown in Table II.

TABLE III
NETWORK METRICS OF THE ETHEREUM DATASET

Average degree	Degree assortativity	Size of the largest connected component	Clustering coefficient	Homophily ratio
2.01	-0.647	1,191,951	0.099	0.276

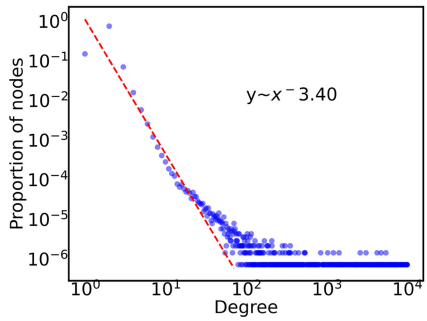


Fig. 1. Degree distribution.

To pave the way for designing an appropriate network embedding method, we investigate the Ethereum transaction network with several metrics in graph analysis as shown in Table III and Fig. 1. As we can see, the average degree of all nodes in the network is 2.01, and the degree distribution of the collected Ethereum transaction network follows a power-law distribution [26], indicating that most of the nodes are small-degree. The degree assortativity coefficient of the network is -0.647, indicating that large-degree accounts tend to trade with small-degree accounts. Similar to other complex networks such as the Twitter network [27], the largest connected component contains most of the nodes in the network. The global clustering coefficient of this network is 0.099, implying that there exist a few triangles in the transaction network, and any two neighbor accounts are less likely to have trading relationships with each other.

Specially, we measure the neighborhood similarity of nodes in the network from the perspectives of node labels and node features, which is important in designing a spatial-based GCN model. To measure the tendency of nodes to connect with nodes having the same class label, we adopt the homophily ratio $h = \frac{|\{(u,v) \in E_y : y_u = y_v\}|}{|E_y|}$ in [23], where E_y represents the set of edges between the labeled nodes, y_u and y_v represent the label of nodes u and v respectively. A network with strong homophily has a homophily ratio close to 1, while a network with strong heterophily has a homophily ratio close to 0. As shown in Table III, the Ethereum transaction network is heterophilous with a homophily ratio $h = 0.276$, which indicates that any labeled nodes connected in pair have a low probability of belonging to the same category.

Based on the widely-used distance function Euclidean distance, we propose two metrics to measure the similarity of features between any pair of connected nodes. To investigate the average feature distance between each pair of connected

TABLE IV
STATISTICS OF FEATURE DISTANCE IN DIFFERENT NETWORKS

	Ethereum	**Cora***	**Citeseer***	**Pubmed***	**FB100-Cornell5***	**FB100-Penn94***
Global average distance[1]	43.65±4.45	5.46±0.44	7.1±0.47	0.48±0.06	2.72±0.35	2.73±0.33
Local average distance[2]	36.62±1.82	5.42±0.35	6.9±0.31	0.48±0.03	2.67±0.35	2.66±0.33

[1]The global average distance (Formula 2) measures the average difference between each pair of connected nodes.
[2]The local average distance (Formula 3) measures the average difference between each node and its neighbors.
*Cora [29], Citeseer [30] and Pubmed [31] are citation networks, and FB100-Cornell5, FB100-Penn94 [32] are social networks.

nodes in the network, we define a global average distance metric as follow:

$$d_1 = \frac{1}{|E|} \sum_{(u,v)\in E} \|F_u - F_v\|_2, \tag{2}$$

where E is the set of edges in the network, $|E|$ is the size of E, $\|\cdot\|_2$ denotes the 2-norm, F_u and F_v denote the features of nodes u and v. Then we focus on the feature difference between nodes and their neighbors, and define a local average distance metric as follow:

$$d_2 = \frac{1}{|V|} \sum_{v\in V} \frac{\sum_{u\in N(v)} \|F_v - F_u\|_2}{|N(v)|}, \tag{3}$$

where V is the set of nodes in the network, $|V|$ is the size of V, $N(v)$ is the neighbors of node v. Table IV presents the statistical results of these two metrics in different networks, including the Ethereum transaction network, citation networks, and social networks. As we can see, the global average distance and the local average distance of the Ethereum transaction network are both much higher than those of the other five common networks, implying the dissimilarity between the connected nodes in the Ethereum transaction network from the dimension of node features.

In short, the Ethereum transaction network is a scale-free network [26] with some topological properties such as low average degree, degree disassortative mixing, and low clustering coefficient. Most notably, the Ethereum transaction network has a low homophily ratio and a large feature distance, indicating that neighbor nodes in the network may have distinct labels and dissimilar features. As many popular GNNs are designed based on the strong homophily property of network data [28], i.e., connected nodes in a network tend to have the same class label and similar features, these methods are hard to be generalized to the Ethereum transaction network which is with strong heterophily. If we directly aggregate the information from the local neighborhood, the category of center nodes may be confused by the information from the dissimilar neighbors. Thus the analysis results motivate us to explore the higher-order neighborhood for more effective feature extraction in the network.

III. METHODS

In this section, we propose a novel GNN model that can explore more effective neighborhood information during the aggregation process for a network with strong heterophily. The proposed GNN model is called Filter and Augment Graph Neural Network (FA-GNN), which introduces two key components for Ethereum transaction network embedding learning, namely, neighbor filtering and feature augmentation. By utilizing these two components, FA-GNN can retain information with preference and explore information from the higher-order neighborhood. We first describe the general framework of FA-GNN, and then detail the design of FA-GNN.

A. Framework

The framework of FA-GNN is shown in Fig. 2. As shown in the figure, it contains three main components including neighbor filtering, feature augmentation, and network embedding.
1) Neighbor Filtering: In a transaction network, the relationship between a pair of connected nodes is related to transaction features such as transaction amount and network features such as degree. To remove the redundant information brought by neighbors, we design this component and filter out a certain ratio of neighbors through various preference strategies.
2) Feature Augmentation: Aiming to explore the higher-order information in the network embedding learning process, we propose a feature augmentation component to generate enhanced features for each node by aggregating the higher-order information.
3) Node Representation Learning: With the augmented node features as the input, the node representation learning component learns a low-dimensional representation vector for each node in the network via a two-layered neural network. Subsequently, the obtained representation vectors are used as an input of downstream Ethereum transaction network analysis tasks.

B. Neighbor Filtering

As discussed in Section II-B, compared with traditional network data like Cora, there exists a significant difference between connected nodes in the Ethereum transaction network in terms of node features. This is because that nodes in the Ethereum transaction network tend to interact with nodes in other classes. Instead of using the information from the full neighbor set, we restrict the neighbor search of each node within a neighbor set obtaining after neighbor filtering, aiming to identify more effective neighborhood information by exploring the node importance of the neighbors.

In this work, we assume that the impact of a node v's neighbor on the node v is correlated with the importance of the neighbor within the node v's neighborhood, and the importance of the neighbor can be defined from both feature and topology aspects. In terms of network topology, we utilize the most direct node importance metric in network science,

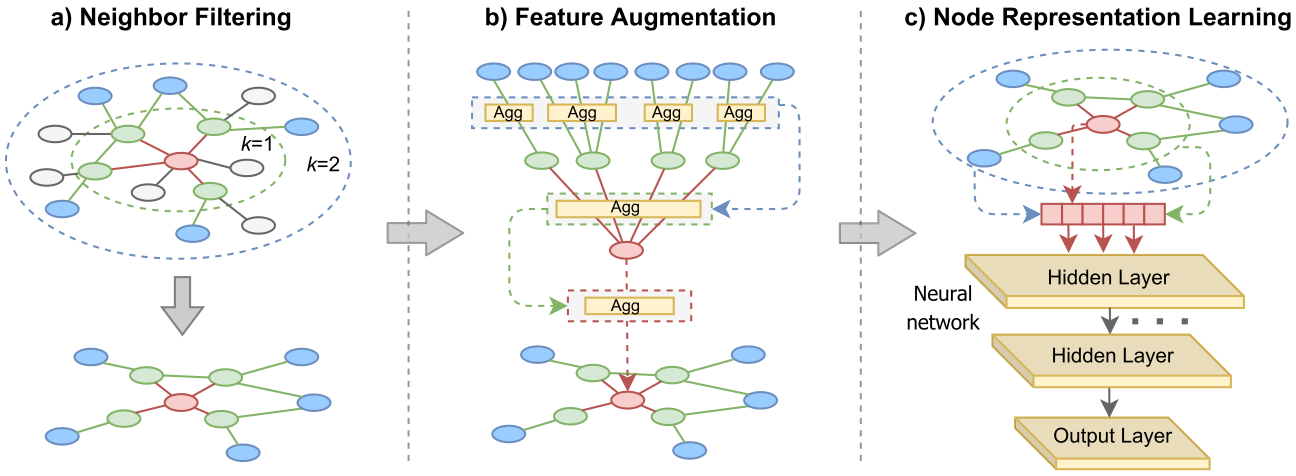


Fig. 2. The framework of FA-GNN.

Algorithm 1: Feature augmentation.

Input: The entire network $G = (V, E)$; input features $\{x_v, \forall v \in V\}$; the set of filtered neighbors $\{S(\lambda, v, p), v \in V\}$; depth k;
Output: Enhanced features Z;

$h_v^0 \leftarrow x_v, \forall v \in V$;
for $i = 1, \ldots, k$ do
  for $v \in V$ do
    $h_v^i \leftarrow \frac{1}{|S(\lambda,v,p)|+1}(h_v^{i-1} + \sum_{u \in S(\lambda,v,p)} h_u^{i-1})$;
  end
end
$z_v \leftarrow combine(x_v, h_v^k), \forall v \in V$;
$Z \leftarrow \{z_v, \forall v \in V\}$;
return Z;

namely degree, to measure the importance of neighbors in the ego-network. In terms of node features, we make use of the total transaction amount and transaction times in the importance measurement of neighbors. Here three preference strategies are proposed in measuring the neighbor importance, and the importance value of a neighbor u with strategy p, $g(u, p)$, is given by:

$$g(u,p) = \begin{cases} \text{Transaction amount of } u, & p = 1 \\ \text{Transaction times of } u, & p = 2 \\ \text{Degree of } u, & p = 3 \end{cases} \quad (4)$$

We then rank the neighbors of each node $v \in V$ according to their importance. Given a neighbor retention rate $\lambda \in [0, 1]$, we retain top $\lambda|N(v)|$ important neighbors in a set $S(\lambda, v, p)$, and filter out the remaining neighbor information. With the three preference strategies, we can obtain those relatively active accounts from the network interaction and transaction behavior perspectives for feature aggregation, which can increase the receptive field of center nodes. The set $S(\lambda, v, p)$ which contains the $\lambda|N(v)|$ neighbors of v that can maximize the sum of the importance value can be expressed as

$$S(\lambda, v, p) = \underset{S' \subset N(v), |S'| = \lambda|N(v)|}{\arg\max} \sum_{u \in S'} g(u, p). \quad (5)$$

C. Feature Augmentation

In order to learn from the higher-order neighborhood, we introduce the feature augmentation component, which expands the node features via aggregating higher-order neighborhood information. Algorithm 1 describes the feature augmentation process with the entire network $G = (V, E)$, node features set $\{x_v, \forall v \in V\}$, filtered neighbor set $\{S(\lambda, v, p), v \in V\}$, and depth k as input. For each round i in the outer loop, every node $v \in V$ aggregates information from its filtered neighbors belonging to $S(\lambda, v, p)$, into a vector h_v^i. Intuitively, each round i aggregates different information from the network based on the feature vectors $\{h_v^{i-1}, v \in V\}$, and k controls the depth in exploring the higher-order neighborhood. With a smaller setting of k, h_v^k can capture the local neighbor information of a node $v \in V$, while a larger setting of k can capture more global information from the network. Finally, we obtain the enhanced features of each node by combining the original features and the aggregated neighborhood features with a concatenation operation. In the downstream tasks with the enhanced features as input, we can explore the effect of the original node features, local neighborhood information, and higher-order neighborhood information separately.

D. Node Representation Learning

After the feature augmentation process, we obtain an enhanced feature set Z. Then we use it as the input of a neural network model to learn the representations of nodes in the network. This learning process for node representations facilitates the mining of hidden information in the Ethereum transaction network and helps us obtain an effective low-dimensional representation vector for each node. In our model, the representation updating rule of the lth layer is:

$$H^l = \sigma(H^{l-1} W^{l-1}), \quad (6)$$

where W^{l-1} is a trainable weight matrix in layer $l-1$, H^l and H^{l-1} are the hidden representation matrix in layer l and layer $l-1$, H^0 is initialized as the feature matrix constructed from Z, and σ is a nonlinear activation function such as the

TABLE V
THREE WAYS FOR TRAIN-VALIDATION-TEST SPLIT

	Train (Ratio)	Validation (Ratio)	Test (Ratio)
Split 1	70%	10%	20%
Split 2	50%	20%	30%
Split 3	30%	30%	40%

$ReLU(\cdot) = max(0, \cdot)$. The output layer of our model is:

$$O = H^l W^l, \tag{7}$$

where O is the output of FA-GNN. In our model we adopt $l = 1$. For node classification tasks, we evaluate our model with the cross-entropy loss function over all labeled examples.

IV. RESULTS AND DISCUSSIONS

In this section, we investigate the effectiveness of the proposed FA-GNN on the account classification task in Ethereum transaction network. We first present the experimental settings including dataset, experimental metrics, and baselines. Then, we detail the experimental results to answer the following questions: *RQ1)* Dose FA-GNN outperform the state-of-art benchmark methods in the downstream task? *RQ2)* How do different neighbor retention rate λ and different aggregation depth k influence the model performance? *RQ3)* What is the impact brought by different neighbor filtering preferences on the model performance? *RQ4)* How does the combine strategy benefit the prediction? Finally, we conduct an ablation study on our model.

A. Experimental Setup

Ethereum Dataset and metrics: Ethereum is currently the largest blockchain platform with smart contracts. Due to the pseudonymous nature of Ethereum, the identities of users and services are hidden behind the pseudonymous accounts. Conducting account classification in Ethereum can help us enrich the label library and identify abnormal accounts, and therefore can improve the transparency and auditability of Ethereum. We evaluate the performance of our proposed FA-GNN using the Ethereum dataset whose details are given in Table I. This dataset contains 6 common kinds of account labels from Etherscan including mining, phish/hack, ICO, exchange, converter, and gambling. To make a comprehensive evaluation, we split the set of nodes with ground labels in three ways as shown in Table V. And the train set in split 1, split 2, and split 3 contains 70%, 50%, and 30% of the labeled nodes selected randomly, respectively. Then, the whole network is fed into the learning model which generates node representations for all nodes in the network. Only the classification performance of 816 labeled nodes are considered in the model training, validation and test process. We use Macro-Precision, Macro-Recall, Micro-F1, and Macro-F1 as the evaluation metrics in the experiments.

Baselines: To evaluate the performance of the proposed FA-GNN, we consider the following state-of-the-art methods.

1) GCN [11]: GCN is the first graph convolutional network achieved by a localized first-order approximation of spectral graph convolutions.
2) SGC [33]: SGC is a simplification of GCN which reduces the unnecessary complexity and redundant computation by removing nonlinearities and collapsing weight matrices between consecutive layers.
3) GraphSAGE [12]: GraphSAGE is an inductive GNN method based on neighborhood information sampling and aggregation. It introduces a set of aggregator functions such as $mean$ and LSTM.
4) Cluster-GCN [13]: Cluster-GCN is an effective GCN algorithm for stochastic gradient descent-based training. It improves the memory and computational efficiency by sampling a block of nodes associated with an identified dense subgraph and restricting the neighborhood search within this subgraph.
5) H_2GCN [23]: H_2GCN is a graph neural network which is effective in networks with low-homophily settings. It consists of the following key designs: ego- and neighbor-embedding separation, higher-order neighborhoods, and combination of intermediate representations.

Noting that GCN and SGC are traditional GCN models without neighbor sampling mechanisms, while GraphSAGE, Cluster-GCN are sampling-based GCN methods. Besides, H_2GCN is designed for networks with low homophily.

Parameter setup: We use Adam [34] as the optimization method in FA-GNN. The learning rate of FA-GNN and the baseline methods except SGC is set to 0.01. And for SGC, the learning rate is set to 0.2. In GraphSAGE, we use the Mean aggregator and the sample size is set to 100 in each layer. In Cluster-GCN, the number of clusters is set to 6. We adopt the concatenation operation to combine the original features and the aggregated neighborhood features in FA-GNN. The hidden dimension for all the compared methods is set to 128. And we report the average score of 10 independent experiments with different random seeds.

B. Ethereum Account Classification Results (RQ1)

We first compare the performance of FA-GNN and the baseline methods in the Ethereum account classification task. Since the neighbor retention rate λ, depth k, and neighbor filtering preference p are adjustable parameters, using Macro-F1 as the main metric, we report the average results under the best parameters learned with a grid search over $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$, $k \in \{1, 2, \ldots, 10\}$, and $p \in \{1, 2, 3\}$. From the results displayed in Tables VI and VII, it is evident that the proposed FA-GNN outperforms the baseline methods in different dataset splits. Specifically, in split 1, we achieve *8.93% gain* over the highest Macro-F1 score obtained by the baseline methods with transaction amount as the preference ($p = 1$), neighbor retention rate $\lambda = 0.2$, and aggregation depth $k = 4$. In split 2, we achieve *7.82% gain* over the best Macro-F1 score of baselines with transaction times as the preference ($p = 2$), neighbor retention rate $\lambda = 0.1$, and

TABLE VI
ETHEREUM ACCOUNT CLASSIFICATION RESULTS (PRECISION AND RECALL)

Algorithm	Split 1		Split 2		Split 3	
	Precision	Recall	Precision	Recall	Precision	Recall
GCN	0.479±0.035	0.529±0.020	0.624±0.025	0.660±0.030	0.527±0.042	0.579±0.042
SGC	0.769±0.016	0.749±0.011	0.748±0.011	0.707±0.010	0.777±0.033	0.690±0.011
GraphSAGE	0.795±0.056	0.808±0.022	0.769±0.020	0.746±0.019	**0.784±0.092**	0.707±0.015
Cluster-GCN	0.817±0.025	0.793±0.037	0.754±0.039	0.756±0.038	0.724±0.045	0.722±0.036
H_2GCN	0.682±0.011	0.691±0.010	0.689±0.058	0.650±0.016	0.642±0.020	0.597±0.012
FA-GNN	**0.876±0.015**	**0.871±0.028**	**0.810±0.017**	**0.822±0.019**	0.782±0.059	**0.722±0.021**

TABLE VII
ETHEREUM ACCOUNT CLASSIFICATION RESULTS (MICRO-F1 AND MACRO-F1)

Algorithm	Split 1		Split 2		Split 3	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
GCN	0.529±0.020	0.479±0.035	0.660±0.030	0.624±0.025	0.579±0.042	0.527±0.042
SGC	0.799±0.013	0.746±0.012	0.788±0.007	0.716±0.010	0.798±0.009	0.704±0.013
GraphSAGE	0.826±0.032	0.789±0.044	0.808±0.015	0.754±0.019	0.822±0.012	0.706±0.023
Cluster-GCN	0.829±0.022	0.795±0.031	0.804±0.025	0.749±0.034	0.778±0.021	0.719±0.039
H_2GCN	0.791±0.011	0.684±0.010	0.752±0.007	0.654±0.021	0.708±0.012	0.604±0.014
FA-GNN	**0.888±0.008**	**0.866±0.021**	**0.863±0.005**	**0.813±0.016**	**0.832±0.010**	**0.730±0.027**

aggregation depth $k = 6$. In split 3, by aggregating the information of the top 10% one-hop neighbors with the highest degree ($p = 3,\ \lambda = 0.1,\ k = 1$), we obtain the best performance which provides *1.53% gain* over the best Macro-F1 score obtained by baseline methods.

Since the baseline methods can be divided into three groups, namely GCN methods without neighbor sampling mechanisms, sampling-based GCN methods, and GCN methods for networks with low homophily, we also have the following observations from the results.

1) GCN and SGC are traditional graph neural network methods without sampling, and they do not take the characteristics of transaction networks into account. Among these two methods, SGC aggregates the neighborhood features first and then learns node representations with a neural network. Yet GCN performs the neighborhood information aggregation within the learning of a neural network, which can lead to the propagation and amplification of errors with the deepening of the neural network. Thus the performance of GCN is worse than SGC in our experiments.
2) GraphSAGE is a sampling-based method which samples a fixed small number of neighbors to aggregate. However, there exist sampling variances and repeated sampling problems in GraphSAGE. ClusterGCN samples subgraphs with graph clustering algorithms and restricts the neighborhood search according to the subgraphs, which also advances the learning of higher-order information. However, the performance of ClusterGCN largely depends on the graph clustering algorithm.
3) H_2GCN is designed for networks with low homophily by utilizing the higher-order information and intermediate representation combination. However, it has not adopted any neighborhood sampling strategy, leading to an exponential growth of computation and memory cost as the depth increases. During the experiments, H_2GCN cannot achieve the expected good results since the out of memory problem when $k \geq 2$. Compared with these baselines, the proposed FA-GNN performs best by considering the characteristics of the Ethereum transaction network, retaining important neighborhood information, and augmenting node features with higher-order information.

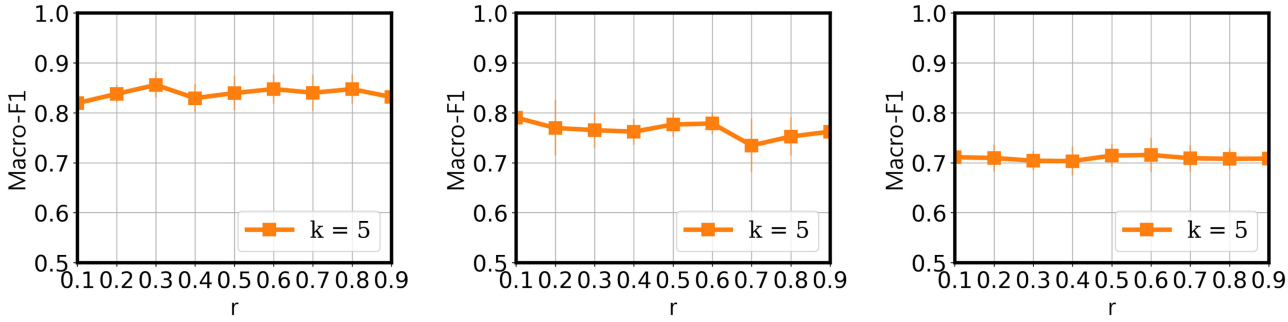
C. Sensitivity Analysis of λ and k (RQ2)

Next, we investigate the performance of FA-GNN under different settings of neighbor retention rate λ and aggregation depth k. Based on the transaction amount preference, we plot the Macro-F1 score of Ethereum account classification with respect to different λ and k in split 1, split 2, and split 3.

Firstly, we vary λ from 0.1 to 0.9 and plot the performance changes of FA-GNN with $k = 5$. As demonstrated in Fig. 3, FA-GNN can achieve a better performance when λ is set as a relatively small value such as $\lambda = 0.2$ and $\lambda = 0.5$, which indicates that the retained neighborhood information has a higher information-to-noise ratio in these neighbor retention rates. Secondly, by setting λ to 0.5 and vary k from 1 to 10, we can investigate the impact from different k on FA-GNN. As shown in Fig. 4, the model performance can be promoted after aggregating higher-order neighborhood information with $k \leq 7$, which demonstrates the effectiveness of higher-order neighborhood information.

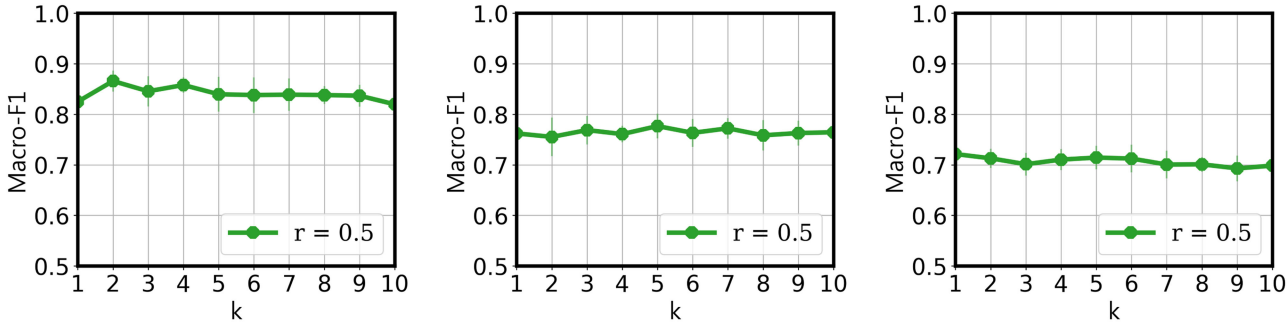
D. Evaluation of Filtering Preferences (RQ3)

Based on the same parameter setting $\lambda = 0.5$ and $k = 5$, we then investigate the results that can be obtained under different neighbor filtering preference strategies. Fig. 5 displays the Marco-F1 performance results of the model with transaction amount, transaction times, node degree as the measure of



(a) Split 1 (b) Split 2 (c) Split 3

Fig. 3. Sensitivity analysis with respect to different r.



(a) Split 1 (b) Split 2 (c) Split 3

Fig. 4. Sensitivity analysis with respect to different k.

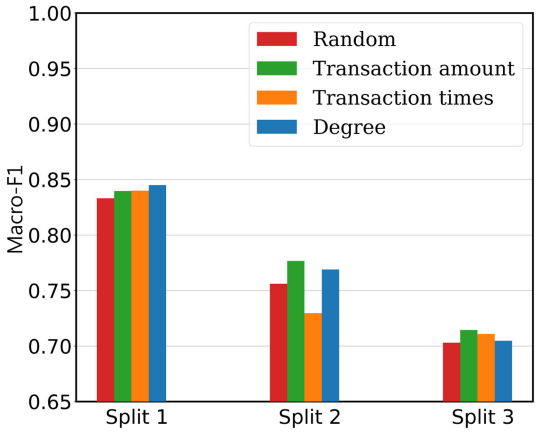


Fig. 5. Performance comparison of different neighbor filtering strategies.

node importance, and also displays the performance result of the model with random filtering strategy. From the experimental results, we can observe that by using the proposed neighbor filtering strategies, we can effectively choose the important neighbors in information aggregation. Moreover, using the transaction amount as the filtering preference can achieve the overall most outstanding result.

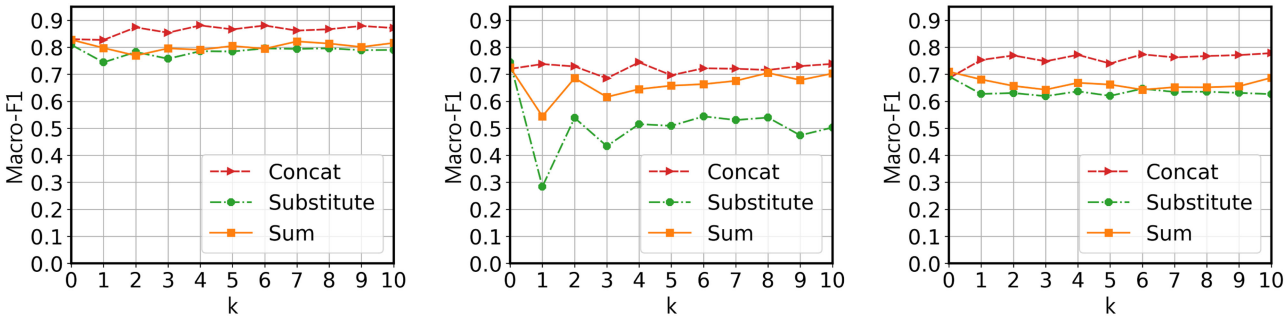
E. Evaluation of Combine Strategies (RQ4)

We further evaluate the performance of the combine strategy in Algorithm 1 line 7. To reduce possible interference, we aggregate the neighbor information directly without adopting any neighbor filtering operation, i.e., $\lambda = 1, S(\lambda, v, p) = N(v)$. We compare the concatenation operation with the summation operation and the substitution operation where z_v is directly substituted by h_v^k for $v \in V$. As shown in Fig. 6, we can obtain the best performance with a concatenation operation compared with the summation and substitution operation under a varying k. And the overall performance increases as k rises, demonstrating the effectiveness of higher-order neighborhood information. We also observe that the substitution operation achieves the worst performance among these compared strategies, indicating that the retention of the original features can improve the performance and robustness to noise of representation learning in the Ethereum transaction network.

F. Ablation Study

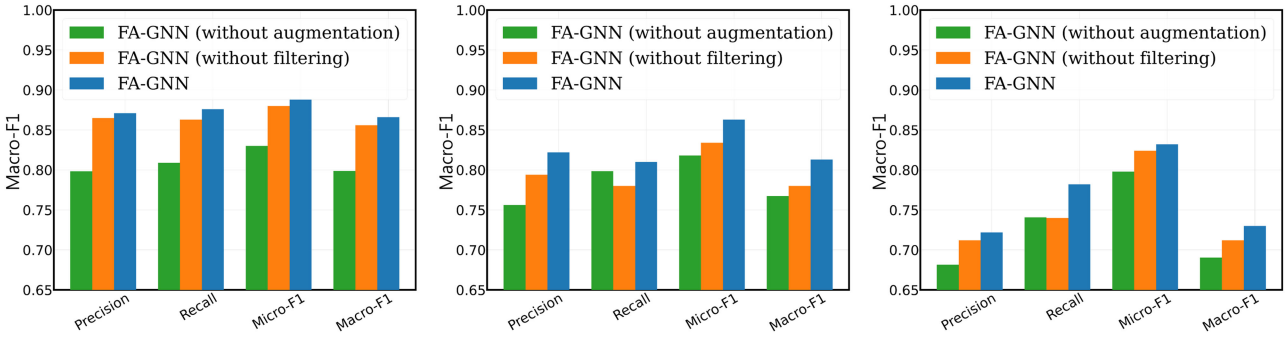
We conduct an ablation study on our model and report the average results of FA-GNN, FA-GNN without the neighbor filtering component, and FA-GNN without the feature augmentation component under the best parameters learned with a grid search over $\lambda \in \{0.1, 0.2, \ldots, 0.9\}$, $k \in \{1, 2, \ldots, 10\}$, and $p \in \{1, 2, 3\}$. Specifically, FA-GNN without the feature augmentation component takes the original node features as the input of node representation learning and does not aggregate any neighborhood information. Therefore, it also disables the neighbor filtering component.

As shown in Fig. 7, FA-GNN without filtering outperforms FA-GNN without augmentation, which highlights the effectiveness obtained by the node features augmented with higher-



(a) Split 1 (b) Split 2 (c) Split 3

Fig. 6. Performance comparison of different combine strategies with respect to different aggregation depth k.



(a) Split 1 (b) Split 2 (c) Split 3

Fig. 7. Ablation analysis for FA-GNN, which shows the average results under the best parameters obtained by FA-GNN, FA-GNN without the neighbor filtering component, and FA-GNN without the feature augmentation component in Ethereum account classification.

order neighborhood information. Besides, FA-GNN achieves the best performance among the compared methods, indicating that the proposed neighbor filtering component and feature augmentation component are effective.

V. Conclusion

In this work, we proposed a GNN-based model named FA-GNN to tackle the account classification task in Ethereum. By conducting a network property analysis, we found that there exist great differences between two connected nodes in the Ethereum transaction network in terms of both the node features and node labels. This motivated us to adopt a different model design from traditional GCNs based on the homophily assumption. Specifically, we proposed two key components in our model, namely neighbor filtering and feature augmentation. According to node features and the newly designed neighbor importance measurements in the transaction network, the neighbor filtering component was proposed to identify the influential neighbors and remove redundant neighborhood information. Then enhanced features can be generated for nodes by higher-order information aggregation in the feature augmentation component. Finally, a deep neural network architecture was adopted to learn node representations for downstream tasks. Extensive experiments demonstrated the effectiveness of the proposed FA-GNN in Ethereum account classification and proved that our model can filter out noised neighbors and explore higher-order information in the Ethereum transaction network. In the future, we will consider more settings under the continuous-time dynamic graph scenario, and build an online transaction analysis system for Ethereum.

References

[1] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," 2014. [Online]. Available: http://gavwood.com/Paper.pdf

[2] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Serv.*, vol. 14, no. 4, pp. 352–375, 2018.

[3] Chainalysis, "Crypto crime trends for 2022: Illicit transaction activity reaches all-time high in value, all-time low in share of all cryptocurrency activity," 2022. [Online]. Available: https://blog.chainalysis.com/reports/2022-crypto-crime-report-introduction/

[4] J. Wu, J. Liu, Y. Zhao, and Z. Zheng, "Analysis of cryptocurrency transactions from a network perspective: An overview," *J. Netw Comput. Appl.*, vol. 190, p. 103139, 2021.

[5] T. Chen *et al.*, "Understanding ethereum via graph analysis," in *Proc. IEEE Conf. Comput. Commun. IEEE*, 2018, pp. 1484–1492.

[6] Y. Wu *et al.*, "A bitcoin transaction network analytic method for future blockchain forensic investigation," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1230–1241, Apr.–Jun. 2021.

[7] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang, "Detecting mixing services via mining bitcoin transaction network with hybrid motifs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 4, pp. 2237–2249, Apr. 2021.

[8] I. Alqassem, I. Rahwan, and D. Svetinovic, "The anti-social system properties: Bitcoin network data analysis," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 1, pp. 21–31, Jan. 2020.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks and defenses," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 67:1–67:43, 2020.
[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–14.
[12] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[13] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 257–266.
[14] S. Farrugia, J. Ellul, and G. Azzopardi, "Detection of illicit accounts over the ethereum blockchain," *Expert Syst. Appl.*, vol. 150, 2020, Art. no. 113318.
[15] T. Hu *et al.*, "Transaction-based classification and detection approach for ethereum smart contract," *Inf. Process. Manag.*, vol. 58, no. 2, 2021, Art. no. 102462.
[16] J. Wu *et al.*, "Who are the phishers? Phishing scam detection on ethereum via network embedding," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 2, pp. 1156–1166, Feb. 2022.
[17] W. Chen, X. Guo, Z. Chen, Z. Zheng, and Y. Lu, "Phishing scam detection on ethereum: Towards financial security for blockchain ecosystem," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 4506–4512.
[18] L. Chen, J. Peng, Y. Liu, J. Li, F. Xie, and Z. Zheng, "Phishing scams detection in ethereum transaction network," *Assoc. Comput. Machinery Trans. Internet Technol.*, vol. 21, no. 1, pp. 10:1–10:16, 2020.
[19] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. 5th Int. Conf. Learn. Represent.* OpenReview.net, 2018, pp. 1–15.
[20] A. Bojchevski *et al.*, "Scaling graph neural networks with approximate pagerank," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* ACM, 2020, pp. 2464–2473.
[21] D. Xu, C. Ruan, E. Körpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–19.
[22] L. Qu, H. Zhu, Q. Duan, and Y. Shi, "Continuous-time link prediction via temporal dependent graph neural network," in *Proc. Web Conf. Assoc. Comput. Machinery*, 2020, pp. 3026–3032.
[23] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7793–7804.
[24] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[25] T. Huang, D. Lin, and J. Wu, "Ethereum account classification based on graph convolutional network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 5, pp. 2528–2532, May 2022.
[26] M. E. J. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford Univ. Press, 2010.
[27] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *Proc. Int. Conf. Weblogs Soc. Media.*, 2010, pp. 10–17.
[28] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.
[29] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, no. 3, pp. 93–106, 2008.
[30] Q. Lu and L. Getoor, "Link-based classification," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 496–503.
[31] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification," in *Proc. Int. Conf. Mach. Learn. ICML Work. MLG*, 2012, pp. 1–8.
[32] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of facebook networks," *Phys. A, Statist. Mech. Appl.*, vol. 391, no. 16, pp. 4165–4180, 2012.
[33] F. Wu, A. H. S. Jr., T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn. Projection-method Learn. Rule*, 2019, pp. 6861–6871.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–22.

Jieli Liu received the B.Eng. in software engineering in 2019 from Sun Yat-sen University, Guangzhou, China, where she is currently working toward the Ph.D. degree with the School of Software Engineering. Her research interests include blockchain, network science, data mining, and machine learning with graphs.

Jiatao Zheng received the B.Eng. degree in network engineering from South China Normal University, Guangzhou, China, in 2019. He is currently working toward the M.Eng. degree with the School of Computer Science and Engineering, Sun Yat-sen University. His research interests include blockchain, network science, data mining, and machine learning with graphs.

Jiajing Wu (Senior Member, IEEE) received the B.Eng. degree in communication engineering from Beijing Jiaotong University, Beijing, China, in 2010, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2014. In 2015, she joined Sun Yat-sen University, Guangzhou, China, where she is currently an Associate Professor. Her research interests include blockchain, graph mining, and network science. She is an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS. During the Ph. D. study in Hong Kong, she was the recipient of the Hong Kong Ph.D. Fellowship Scheme, from 2010 to 2014.

Zibin Zheng (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2011. He is currently a Professor of data and computer science with Sun Yat-sen University, Guangzhou, China. He is the Chair of the Software Engineering Department, Pearl River Young Scholars, and the Founding Chair of the Services Society Young Scientists Forum. In the past five years, he has authored or coauthored more than 120 international journal and conference papers, including three ESI highly-cited papers, 40 ACM/IEEE TRANSACTIONS papers. According to Google Scholar, his papers have more than 6300 citations, with an H-index of 41. His research interests include blockchain, services computing, software engineering, and financial Big Data. He was the recipient of several awards, including the outstanding Thesis Award of CUHK, in 2012, the ACMSIGSOFT Distinguished Paper Award at ICSE2010, Best Student Paper Award at ICWS2010, and IBM Ph.D. Fellowship Award. He was CollaborateCom'16 General Co-Chair, ICIOT'18 PC Co-Chair, and IoV'14 PC Co-Chair.