A Bayesian Graph Embedding Model for Link-Based Classification Problems

Yichao Zhang [ID], *Member, IEEE*, Huangxin Zhuang [ID], Tiantian Liu, Bowei Chen [ID], *Member, IEEE*,
Zhiwei Cao [ID], Yun Fu [ID], Zhijie Fan [ID], and Guanrong Chen [ID], *Life Fellow, IEEE*

Abstract—In recent years, the analysis of human interaction data has led to the rapid development of graph embedding methods. Topological information is typically interpreted into embedded vectors or convolution kernels for link-based classification problems. This paper introduces a Bayesian graph embedding model for such problems, integrating network reconstruction, link prediction, and behavior prediction into a unified framework. Unlike the existing graph embedding methods, this model does not embed the topology of nodes or links into a low-dimensional space but sorts the probabilities of upcoming links and fuses the information of node topology and data domain via sorting. The new model integrates supervised transaction predictors with unsupervised link prediction models, summarizing local and global topological information. The experimental results on a financial trading dataset and a retweet network dataset demonstrate that the proposed feature fusion model outperforms the tested benchmarked machine learning algorithms in precision, recall, and F1-measure. The proposed learning structure has a fundamental methodological contribution and can be extended and applied to various link-based classification problems in different fields.

Index Terms—Ensemble learning, Bayesian Network, interaction prediction, trader network.

I. INTRODUCTION

THE rapid growth of human interaction networks has led to a growing amount of interaction-generated data and the impediment of finding reliable information [1], [2]. Interaction logs, such as the web of sexual contacts, air-transport records, blue tooth traces, email traces, mobile sensing logs, and accessing logs of WiFi hotspots, play an essential role in helping individuals find relevant information, for instance, heterogeneous social behaviors and upcoming interactions. In human interaction records, social relationships among individuals are typically unknown. The latent social relationships can only be inferred through other information channels, such as their similarities on the observable social topologies, profiles, and behavior trajectories [3]–[5]. To predict upcoming interactions with interaction logs, researchers typically translate the task to a classification problem. However, researchers need to address two challenges beforehand. First, networked structures are non-Euclidean, to which many classification methods, such as support vector machines and convolutional neural networks, can not be applied directly. Second, the size of the networks is normally large, which can be computationally expensive and lead to storage costs.

Network representation learning models were proposed to address the above challenges. They map the nodes in topological spaces into low-dimension real-valued vectors and preserve their proximity in the original spaces as much as possible. They can be broadly classified into three categories: matrix factorization-based models, random walk-based models, and neural network models. Matrix factorization-based models include Laplacian Eigenmaps [6], Graph Factorization [7], GraRep [8], and HOPE [9]. Random walk-based models include DeepWalk [10], Node2vec [11], GraphGAN [12], and GraphSAGE [13]. Neural network models include Graph Neural Networks [14], Graph Autoencoders [15], Graph Convolutional Networks (GCN) [16], Graph Differentiable Pooling [17], and Graph Attention Networks (GAT) [18]. Generally, the graph embedding methods can automatically learn topological features without complicated feature engineering procedures, while their robustness to network types and memory requirements for large networks are generally not satisfactory, and their interpretability is highly limited.

In this paper, we propose a Bayesian graph embedding (in short BGE) model for the link-based classification problem, which integrates network reconstruction, link prediction, and behavior prediction into a unified framework. In a social network, individuals can be denoted by nodes. Friends are connected with an undirected link, representing the interaction between them. When the interaction network is given, the interaction prediction can be translated to estimating $\mathbb{P}(T_{ij} = 1|E)$, where E denotes a set of links inferred by the interaction logs, and $T_{ij} = 1$ denotes that individual i and j would interact

Yichao Zhang , Huangxin Zhuang , Tiantian Liu, and Zhiwei Cao are with Tongji University, Shanghai 201804, China (e-mail: yichaozhang@tongji.edu.cn; 1930809@tongji.edu.cn; liutiantian@tongji.edu.cn; zhiweicao@126.com).

Bowei Chen is with the University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: boweichen@glasgow.ac.uk).

Yun Fu is with the University College London, WC1E 6BT London, U.K. (e-mail: yfu@cs.ucl.ac.uk).

Zhijie Fan is with Fudan University, Shanghai 200433, China (e-mail: aaronzfan@126.com).

Guanrong Chen is with the City University of Hong Kong, Kowloon Tong 999077, Hong Kong (e-mail: eegchen@cityu.edu.hk).

Digital Object Identifier 10.1109/TNSE.2021.3131223

at least once. The truth is that the topology of networks is typically unknown.

The interaction log data used in this paper is provided by a British bank. The log covers the transaction records of a department for 12 months. Each transaction record contains several features, mainly including the ID of the buying trader and that of the selling trader. The learning task of the proposed model is to predict upcoming transactions among traders. The transaction prediction in the financial field is closely related to risk management, since the transaction risk is diffusing over the trader network by transactions [19]–[21]. Therefore, the significance of the transaction prediction is self-evident both theoretically and technically. In the available dataset, the trader interaction network can only be inferred from the transaction logs. With the logs including IDs, one can infer a subgraph of the entire trader network, which is useful in the following prediction of the upcoming transactions. The rest of the features can be extracted from their trading patterns, such as the preference on product groups and trading time patterns. Notably, the available interaction logs in a dataset are rather limited. The inferred subgraph of the trader network is likewise small and sparse, where a large number of traders are not connected. These limitations highly increase the difficulty in embedding them into useful vectors and training a convincing binary classifier. To tackle this problem, we introduce a different solution in this paper. The fusion of the node topology and data domain is accomplished through a Bayesian network. To further verify the robustness of the proposed model on different link-based classification problems, we also employ an open dataset about users in a retweet network. Our experimental results on both datasets show that the performance of the new model outperforms the state-of-the-art models, demonstrating the feasibility of the proposed framework.

Our contributions of the study in the paper are three-fold. First, we find a new way to integrate the topological information of multi-components and data domains effectively. Second, our model offers a solution to fuse the supervised machine learning models with the unsupervised models. Third, our extensive experiments on a variety of algorithms show that the proposed new model achieves better prediction performance and interpretability than the state-of-the-art algorithms, which demonstrates that the model provides a promising paradigm for tackling link-based classification problems.

The rest of the paper is organized as follows. Section II reviews the preliminaries and related studies. Section III introduces the proposed model. Section IV presents our experimental results, and Section V concludes the paper.

II. Preliminaries and Related Works

For the interaction prediction task, many previous studies adopt supervised learning methods to embed the data domain and topological features of a node into an input vector, with final outputs provided by a binary classifier, including decision trees, support vector machines, K-nearest neighbors, multilayer perceptrons, radial basis function networks, naive Bayes, and different ensemble learning models like random forest

and boosted decision tree [22], [23]. Among various graph representation learning models [24], graph convolution deep neural network (GCN) has shown particularly encouraging performance in many learning tasks involving graph structure data [25]. For link-based classification tasks, GCN is used to embed nodes into a lower-dimensional space [26]. The final output typically depends on a fully connected neural network or other classical binary classifiers.

We do not embed node pairs into a lower-dimensional space in the proposed model but into a ranking vector. Specifically, a link prediction algorithm based on topology information is adopted to embed node pairs. The probability ranking result is then fused with non-topological information to derive the final prediction. Thus, one can efficiently train a binary classifier based on the non-topological information to infer the probability of transactions between the disconnected node pairs. The ranking of the transaction probability is likewise easy to derive. With the probability ranking, the two modes of data can be fused by a simple linear combination.

We then discuss the link prediction algorithms. The topology-based link prediction algorithms can be roughly classified into four categories: likelihood-based methods, probabilistic methods, graph embedding methods, and similarity-based methods [27]. Typical examples of the maximum likelihood-based methods are the stochastic block model (SBM) [28], fast probability block model (FBM) [29], hierarchical structure model (HSM) [30], [31]. The community structure in networks is usually not easy to detect because numerous cycles break their hierarchical structures. The maximum likelihood methods are very time-consuming, and their scalability to the types of networks is limited. The probabilistic methods include probabilistic relational model (PRM) [32], probabilistic entity-relationship model (PERM) [33], and stochastic relational model (SRM) [34]. As they depend on the global topological information, their time complexities are normally high, and their accuracy is generally not very satisfactory. The similarity-based models can be sorted into three categories: local, quasi-local, and global algorithms. The local algorithms only consider the local information of node pairs. Typical local algorithms include the common neighbors (CN) algorithm, preferential attachment (PA) algorithm, resource allocation (RA) algorithm, Adamic-Adar (AA) algorithm, and Cannistraci-Hebb (CH) algorithm [35], [36]. In this category, the performance and robustness to network types of the RA algorithm are relatively better. The scale of the information used in quasi-local algorithms is between local and global algorithms. Typical quasi-local algorithms include the local random walk algorithm (LRW), local path algorithm (LP), Propflow algorithm [37], and Quantum-inspired Ant Colony Optimization (QACO) algorithm [38]. The information used in the quasi-local algorithms is normally less than the global algorithms, while their performances are often promising. In this category, the performance and robustness to network types of the QACO algorithm are more competitive. The global algorithms, as the name implies, leverage the global topology information of the network. Typical global

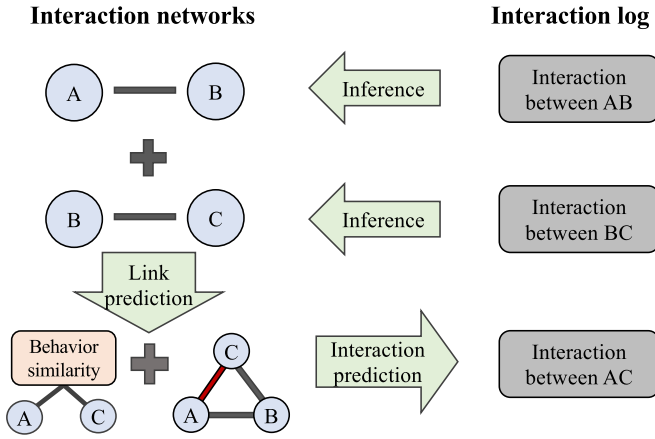**Interaction networks**   **Interaction log**

Fig. 1. Schematic view of interaction prediction.

algorithms include the Katz algorithm, matrix forest index (MFI), average commute time algorithm (ACT), low-rank algorithm (LR), structural perturbation method (SPM), and the like [39]. Among them, the performance and robustness of the SPM are relatively better.

III. BAYESIAN GRAPH EMBEDDING MODEL

A. Theoretical Framework

For each individual, his attributes are composed of his topological properties in the interaction network and his interaction patterns, such as behavior preference and time trajectory. The attributes are used for each pair to calculate their topological similarities, behavior preference similarities, time pattern similarities, and the likes. Based on these similarities or a direct concatenation of the embedded vectors, one can train a binary classifier, but its performance usually is not satisfactory. One reason is that labels are imbalanced [40]. As many real networks are sparse, the feature extraction or graph embedding procedures inevitably sacrifice some topological information. To tackle this problem, we propose the Bayesian graph embedding method. Our model is illustrated in Fig. 1. From the interaction logs, one can infer links between individuals A and B (B and C likewise). Based on the inferred links, one can derive a linking probability of A and C. Finally, the linking probability and the behavior similarities together determine whether A and C will interact with each other. Our goal is to compute the probability

$$\mathbb{P}(T_{ij} = 1 | E, \theta(\mu_i, \mu_j)), \qquad (1)$$

where μ_i and μ_j are the attributes learned from i and j interaction logs; $\theta(\mu_i, \mu_j)$ is a vector representing the similarities between individual i and j, where link $e(i, j) \in \overline{E}$.

As also shown in Fig. 2, one can infer whether there are links between the individuals in the probe set from the topology in the training set. Based on the links and the behavioral similarities between them, one can eventually predict whether they will interact. Take traders in investment banks as an example, μ_i represents trader i's preference on product
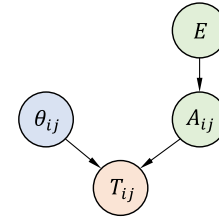

Fig. 2. The Bayesian network of interaction prediction.

groups, trading time patterns, and so on. For the traders i and j, $\theta(\mu_i, \mu_j)$ encapsulates their similarity on the product group preference, similarities of time patterns, and so on. Accordingly, $T_{ij} = 1$ means that at least one transaction is accomplished between traders i and j. To predict whether a transaction will be accomplished between two traders who have no transaction in the available transaction logs, one can directly abstract a series of features from the network mentioned above and establish a binary classification model by fusing the features and trading patterns of the traders. Nevertheless, the solution will be challenged by the links not recorded in the transaction logs.

Based on the interaction logs, one can easily infer the set of links E, since an interaction will surely leave a link, while the inferred links are merely a part of the set of links in the interaction network. The reason is that a dataset of logs can hardly cover all the interactions in history, since it is a trade secret itself. On the other hand, some individuals may be acquaintances, but they have not interacted yet. Clearly, the links among them can not be easily detected. To resolve the problem, we assume that a link in the interaction network can highly promote the probability of interaction between two individuals. To facilitate the downstream calculation, we uniformly set the probability $\mathbb{P}(T_{ij} = 1 | A_{ij} = 1) = \kappa$ in this paper, where κ is a constant. Under this assumption, the task is naturally interpreted as first predicting the missing links with E and then predicting the upcoming interactions on the predicted links.

Based on the Bayesian network shown in Fig. 2, then

$$\mathbb{P}(T_{ij} = 1 | E, \theta(\mu_i, \mu_j))$$
$$= \mathbb{P}(T_{ij} = 1 | A_{ij} = 1, \theta(\mu_i, \mu_j)) \mathbb{P}(A_{ij} = 1 | E)$$
$$= \frac{\mathbb{P}(T_{ij} = 1, A_{ij} = 1, \theta(\mu_i, \mu_j)) \mathbb{P}(A_{ij} = 1 | E)}{\mathbb{P}(A_{ij} = 1, \theta(\mu_i, \mu_j))}, \qquad (2)$$

where $A_{i,j}$ denotes an entry in the adjacency matrix of the network, and link $v(i, j) \in \overline{E}$. As $\mathbb{P}(A_{ij} = 1 | T_{ij} = 1, \theta(\mu_i, \mu_j)) = 1$, the right-hand side of Eq. (2) is reduced to

$$\frac{\mathbb{P}(T_{ij} = 1 | \theta(\mu_i, \mu_j)) \mathbb{P}(A_{ij} = 1 | E)}{\mathbb{P}(A_{ij} = 1 | \theta(\mu_i, \mu_j))}. \qquad (3)$$

Computing $\mathbb{P}(A_{ij} = 1 | E)$ is a typical topology-based link prediction problem while computing $\mathbb{P}(T_{ij} = 1 | \theta(\mu_i, \mu_j))$ can be translated to an attribute-based binary classification problem. Since $\mathbb{P}(A_{ij})$ is irrelevant to $\theta(\mu_i, \mu_j)$ for T_{ij} is not given, $\mathbb{P}(A_{ij} = 1 | \theta(\mu_i, \mu_j))$ is equal to $\mathbb{P}(A_{ij} = 1)$, which is only
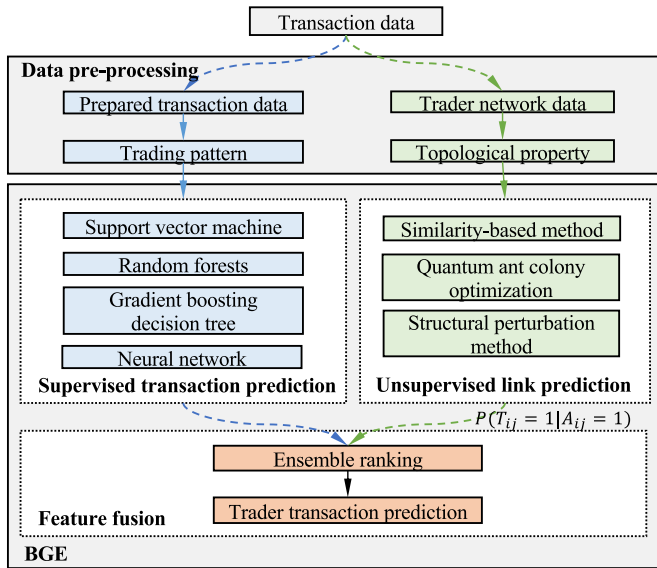
Fig. 3. Schematic view of the BGE model.

dependent on the density [41] of the entire interaction network including the missing links. Therefore, to predict an interaction between individual i and j, one is required to calculate $\mathbb{P}(T_{ij} = 1|\theta(\mu_i, \mu_j))$ and $\mathbb{P}(A_{ij} = 1|E)$ for all link $e(i, j) \in \overline{E}$ with their behavior similarities and topological properties, respectively. Computing $\mathbb{P}(T_{ij} = 1|\theta(\mu_i, \mu_j))$ is a binary classification problem, while estimating $\mathbb{P}(A_{ij} = 1|E)$ is a typical link prediction problem. Furthermore, the interaction prediction problem is divided into a supervised machine learning task and an unsupervised machine learning task [27]. The difficulty lies in that there is no suitable way to integrate supervised and unsupervised learning models into the same framework.

To unify the supervised and unsupervised machine learning models, we implement the proposed model with three stages: topology-based link prediction, attribute-based interaction prediction, and feature fusion. A formal definition of the link prediction problem is presented as follows. For each link, the interactions between the individuals on both ends are not recorded in the interaction logs, i.e., thus for links in $\overline{E}$, define the linking probability ranking L as $L = [B_{ij}, B_{kl}, \ldots]$, with $e(i, j), e(k, l) \in \overline{E}$ and $B_{ij} \geq B_{kl}$, where the node pairs at the top are more likely to be the missing or upcoming links. For L, a series of representative topology-based algorithms are introduced to score the links in $\overline{E}$, including the local, quasi-local, and global algorithms.

For the attribute-based interaction prediction, we mine the individual behavior patterns such as product preference similarity and time pattern similarity in trader networks based on the limited transaction information. A series of well-performed supervised machine learning methods are then applied to score each link in $\overline{E}$. Finally, a ranking P of interaction probability on $\overline{E}$ derived from the attribute-based model and L derived from the topology-based model are required to be fused to output the final ranking of interaction probabilities.

A schematic view of the framework of the proposed model is presented in Fig. 3, where the supervised and unsupervised

learning algorithms are used as base learners to predict the transactions and links in parallel. They are then fused into a stronger classifier for transaction prediction. Specifically, the supervised learning algorithms use the non-topological properties of the transaction data as the input, while the topological features of the trader network are taken into account by unsupervised learning algorithms. In the following, we will show that our model can appropriately integrate the topological information of the sparse trading network with the behavior similarities of pairwise traders in a way called ranking fusion.

B. Interaction Prediction

If the data domain of nodes is non-empty in a dataset, previous studies commonly adopt the supervised machine learning algorithms for the convenience of integrating different types of information. Instead, our model takes the supervised algorithms as a component to learn the linking probability from the data domain of nodes. In the following, we briefly introduce attribute-based interaction prediction, topology-based link prediction, and interaction prediction based on network embedding. They are either the components of our model or the methods to be compared.

1) Attribute-Based Interaction Prediction: Applying the supervised learning algorithms to the problem of interaction prediction [42] is typical for machine learning practitioners as the occurrence of interaction can be predicted by a binary classifier. Whereas the implementation of these algorithms is dragged by the imbalanced data classes resulting from the low density of real networks and difficulty in abstracting the topological features into independent features [15]. A robust classifier should be built either on adequately interpreting the topological similarity measurements to features or on learning the representation of the features through optimizing the prediction accuracy [11]. The tested classification models in previous studies include support vector machines, K-nearest neighbors, logistic regression, random forest, multilayer perceptron, radial basis function network, naive Bayes, and gradient boost decision tree. To present a feasible ranking vector, we select a series of classical binary classifiers to evaluate the interaction probabilities between node pairs comprehensively.

In short, the support vector machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. Specifically, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane that categorizes new examples. In [22], a comparison between a few link prediction models is reported, and SVM with RBF kernel was very successful in terms of accuracy. Therefore, we choose SVM as our first binary classifier. Random forests or random decision forests are a supervised ensemble learning method for classification, regression, and other tasks that operate through constructing a multitude of decision trees in training and outputting the labels for classification or values for regression from the trees. It is a flexible, easy-to-use machine learning algorithm that usually produces good results, which is our second choice. Gradient boosting decision tree (in short, GBDT) is an iterative decision tree algorithm composed of

multiple decision trees. During each iteration, the algorithm uses the current ensemble to predict the label of each sample and then compare the label with the ground truth. The dataset is remarked with the corresponding "residual" to emphasize the training sample with poor prediction performance. Generally, the GBDT algorithm performs well in various data mining and machine learning competitions, which is thus our third choice. The multilayer perceptron (in short, MLP) is regarded as the simplest form of a feedforward neural network. Despite its simple structure, the perceptron can learn and solve quite complex learning problems. We use this basic deep neural network framework to test the possibility of using other deep learning models. Table III shows the parameter settings for all the candidate models.

2) Topology-Based Link Prediction: To acquire the ranking of linking probabilities for node pairs in networks, one can turn to the similarity-based algorithms in which each pair of nodes, $x - y$, is assigned a similarity score s_{xy}. All the missing or upcoming links are ranked with their scores, directly proportional to their linking possibilities. Typically, these algorithms are interpretable, with computational complexities lower than the machine learning approaches. To present a feasible ranking vector, we next introduce three representative algorithms on different scales.

RA Algorithm. Let x and y denote two randomly selected nodes in a network. Let $\Gamma(x)$ and $\Gamma(y)$ denote the sets of x and y's neighbors, and k_x and k_y denote the degrees of x and y, respectively. The similarity between nodes x and y is defined as

$$S_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}. \tag{4}$$

The performance of the RA algorithm has been demonstrated to be one of the best local algorithms [43]. Therefore, it is selected as a component of the BGE model.

QACO Algorithm. As a representative quasi-local algorithm, the QACO algorithm integrates ant colony optimization and quantum computing [38]. Consider an undirected network $G = (V, E)$, and let a number of artificial ants randomly diffuse in G, where each node and node pair are respectively allocated a certain amount of pheromone. The probability that a link is visited by an ant is proportional to its pheromone. Assume an ant visits n nodes, which leaves a walking path. Let the probability that an ant travels from node v_i to node v_j be p_{ij}. The value of p_{ij} relies on the pheromone of the path, the visibility of the node pair (v_i, v_j), and the quantum pheromone of v_j. After an ant v_j reaches its destination, the pheromone on the node pairs and nodes in the path will be updated according to a certain rule. In turn, the updated pheromone on the links and nodes will affect the paths of the ants in the next iteration. Generally, the pheromone and visibility of the node pairs will heuristically lead the ants to approach the globally optimal paths, since following the quantum pheromone is an effective way to avoid local optima. Finally, the pheromone τ_{ij} and visibility η_{ij} on node pair (v_i, v_j) can effectively reflect the similarity between v_i and v_j.

SPM Algorithm. The SPM algorithm is a structural perturbation method in which a new matrix is generated by perturbing the eigenvalues of the original adjacent matrix while keeping the eigenvectors [44]. Randomly select a fraction p^H of the links in E to constitute a perturbation set ΔE, and define E^R as the set $E - \Delta E$. Let A^R and ΔA be the corresponding adjacency matrices for the networks composed of E^R and ΔE, respectively. Note that $A = A^R + \Delta A$, where the real symmetric matrix A^R can be diagonalized as

$$A^R = \sum_{k=1}^{N} \lambda_k x_k x_k^T, \tag{5}$$

where λ_k and x_k are the k_{th} eigenvalue and its corresponding orthogonal and normalized eigenvector of A^R, respectively. Using the perturbed eigenvalues and unchanged eigenvectors, the perturbed matrix of A^R can be rewritten as

$$\tilde{A} = \sum_{k=1}^{N} (\lambda_k + \Delta \lambda_k) x_k x_k^T, \tag{6}$$

where $\Delta \lambda_k$ denotes the difference on the k_{th} eigenvalue induced by the perturbation. $\tilde{A}$ can be used as the linear approximation of the given adjacency matrix A.

If the perturbation does not significantly change the structural features, the differences between the eigenvector of the observed matrix x_k and $x_k + \Delta x_k$ for all k will be negligible. In this case, $A^R + \Delta A$ can be approximated as $\tilde{A}$. Comparing $\tilde{A}$ and A^R, one can obtain the scores of the node pairs in question, which are then used to predict the missing links in ΔE. If there are missing links in a network, the SPM algorithm can effectively detect them.

3) Interaction Prediction Based on Network Embedding: Network embedding is a method of network representation learning. The purpose is to learn the lower-dimensional potential representation of nodes in the network while maintaining the original network structure as much as possible [45]. The features learned through graph embedding can be conveniently used in various machine learning tasks. For example, in link prediction, graph embedding is also a standard method to obtain the feature representation of nodes. In order to compare it with the method proposed in this paper, we use two classic graph embedding models for link prediction tasks.

Node2vec is a classical model in network representation learning and is mainly designed for feature learning of automated prediction tasks. This method is an extension of another graph embedding model, Deepwalk, which comprehensively considers the depth-first sampling neighborhood and breadth-first sampling neighborhood [46].

Due to the wide application of GCN [25] in the field of graph embedding, recent link prediction methods are closely related to it. Among them, the most representatives are Graph Auto Encoder (GAE) and variational Graph Auto Encoder (VGAE) [47]. GAE is an application of the sparse autoencoder model [48] to the field of graph embedding. The idea is to use GCN to fuse node features with topological information and decode the embedding by reconstructing the graph. VGAE [47] introduces Gaussian noise to GAE, which is an

---

Algorithm 1. Preprocessing of Z for probability-based fusion

1: **Input:** X ▷ Original outputs of candidate models
2: $Z \leftarrow X$
3: **for** $j \in \mathcal{U}$ **do** ▷ $\mathcal{U}$ is the index set of unsupervised learning models
4:      **for** $i = 1, \ldots, N$ **do**
5:
$$Z_{(i,j)} \leftarrow \frac{X_{(i,j)} - \min\{X_{(\cdot,j)}\}}{\max\{X_{(\cdot,j)}\} - \min\{X_{(\cdot,j)}\}}$$

6:      **end for**
7: **end for**
8: **Output:** Z

Algorithm 2. Preprocessing of Z for ranking-based fusion

1: **Input:** X ▷ Original outputs of candidate models
2: **for** $j = 1 \cdots, D$ **do**
3:      $\widetilde{X}_{(\cdot,j)} \leftarrow$ Index of sorting $X_{(\cdot,j)}$ in decreasing order
4: **end for**
5: **for** $i = 1, \ldots, Tp$ **do**
6:      **for** $j = 1, \ldots, D$ **do**
7:          $Z_{(i,j)} \leftarrow \frac{1}{\widetilde{X}_{(i,j)}}$
8:      **end for**
9: **end for**
10: **Output:** Z

Algorithm 3. Optimal weights

1: **Input:** Z, y
2:
$$w^* \leftarrow \arg\min_w -\frac{1}{N}\left\{\sum_{i=1}^{N}[y_i \ln f(w^\top Z_{(i,\cdot)}) \right.$$
$$\left. + (1 - y_i)\ln(1 - f(w^\top Z_{(i,\cdot)}))]\right\}, \quad (7)$$

$$\text{subject to } \sum_{j=1}^{D} w_j = 1, \quad (8)$$

$$w_j \geq 0, j = 1, 2, \ldots, D, \quad (9)$$

3: **Output:** w^*

---

unsupervised models be N_s and N_u, respectively. The fused interaction probability for node pair i can be defined as follows:

$$p_i = w_{(1:N_s)}^\top Z_{(i,1:N_s)} + w_{(N_s+1:N_s+N_u)}^\top \cdot Z_{(i,N_s+1:N_s+N_u)}. \quad (10)$$

Here, we divide w into two segments, since the physical meanings of the outputs of the candidate models are different for supervised and unsupervised learning models. The outputs of the former are the predicted probabilities of the transaction links of node pairs, while the outputs of the latter are the similarity scores between two nodes, which are typically out of the range between 0 and 1. Therefore, the scores are required to be normalized before fusion. We adopt the 'Min-Max Scaling' [49] to normalize them in the probability-based fusion. The preprocessing of Z is presented in Algorithm 1. Although the predicted probabilities and similarity scores possess different scales and physical meanings, their functions are similar. A more significant probability or similarity score implies that the transaction between the two traders is more likely to occur. Therefore, one can fuse them to predict upcoming transactions.

Considering the differences in scales and physical meanings, we propose a ranking-based fusion. As the physical meanings of the rankings are the same, the scaling is not necessary anymore. In the fusion, $Z = \frac{1}{\widetilde{X}_{(i,j)}}$, where $\widetilde{X}_{(i,j)}$ denotes i's index in the ranking of $X_{(\cdot,j)}$ in descending order. The preprocessing of Z is presented in Algorithm 2. The predicted label of node pair i depends on whether the index of p_i in the ranking of p is less than or equal to a predetermined threshold K, where K is a hyper-parameter of our model. If the index is less than or equal to K, it will be labeled 1, and 0 otherwise. The mapping is denoted by function $f(\cdot)$.

The optimal weights of candidate models in Eq. (10) can be derived by Algorithm 3, in which a loss function is formulated based on cross-entropy [49]. To optimize the hyper-parameter

application of the variational auto-encoder(VAE) to the field of graph embedding. The decoders of both GAE and VGAE are built on link prediction tasks. Therefore, one can naturally use them to accomplish such tasks. We will compare our model with these two graph-embedding models in Section IV.

C. Feature Fusion

For the interpretability of the model, we adopt a linear model to fuse the ranking vectors. Suppose that there are $Tp = |\overline{E}|$ samples of trader pairs. The existence of their trading transactions (i.e., ground truth or labels) is denoted by a vector $y = [y_1, \ldots, y_{Tp}]^\top$. Suppose also that there are D candidate models in total (including both supervised and unsupervised learning models), and the corresponding model weight in the linear fusion is denoted by $w = [w_1, \ldots, w_D]^\top$. For N samples, the outputs of the candidate models are denoted by a matrix $Z = [Z_{(1,\cdot)}, \ldots, Z_{(Tp,\cdot)}]^\top = [Z_{(\cdot,1)}, \ldots, Z_{(\cdot,D)}]$, where $Z_{(i,\cdot)}$ is a vector of the predictions given by the candidate models for node pair i, $i = 1, \ldots, Tp$. In the probability-based fusion, the predictions are the interaction probabilities of node pairs for supervised models and similarity scores for unsupervised models, respectively. In the ranking-based fusion, the predictions are the reciprocals of the ranking indices of the interaction probabilities and similarity scores mentioned above. Their formal definitions will be provided later in this section. $Z_{(\cdot,j)}$ is a vector of the predictions for all the node pairs given by the candidate model j, $j = 1, \ldots, D$. Use a unified weight vector w to fuse the rankings of interaction probabilities and linking probabilities. Let the numbers of supervised and

TABLE I
DISTRIBUTIONS OF FINANCIAL PRODUCTS

Family	Group	No. of trans (%)	Amount (%)	Buy (%)	Sell (%)
IRD	BOND	6.8300	23.1408	6.8300	23.1408
IRD	CF	0.2900	0.0414	0.2900	0.0414
IRD	CS	0.7100	0.5560	0.7100	0.5560
IRD	FRA	1.5000	2.4039	1.5000	2.4039
IRD	INFLS	0.1400	0.0173	0.1400	0.0173
IRD	IRS	10.6800	12.0112	10.6800	12.0112
IRD	LFUT	18.1000	0.1870	18.1000	0.1870
IRD	LN_BR	26.8300	55.1268	26.8300	55.1268
IRD	OPT	0.1000	0.5306	0.1000	0.5306
IRD	OSWP	0.1500	0.0515	0.1500	0.0515
IRD	REPO	1.3000	3.0031	1.3000	3.0031
IRD	SFUT	2.4500	2.1191	2.4500	2.1191
CURR	FXD	25.5600	0.7190	25.5600	0.7190
CURR	OPT	5.2800	0.0643	5.2800	0.0643
CURR	MP	0.0008	0.0000	0.0008	0.0000
CRD	CDS	0.0700	0.0278	0.0700	0.0278
CRD	FDB	0.0017	0.0000	0.0017	0.0000
COM	SWAP	0.0008	0.0000	0.0008	0.0000

w in Algorithm 3, we adopt the grid search in $(0, 1)$ with a step length 0.1 in our experiments.

IV. EXPERIMENTS

This section introduces our datasets, provides data preparation and model training settings, and presents the experimental results and comparative analysis.

A. Data

We use a real-world trading dataset from a British investment bank. It is composed of 120,648 transaction records of its U.K. trading department from 2 January 2014 to 31 December 2014. Each transaction record contains 36 entries (known as *features* in machine learning [49]) such as unique transaction ID, trading product grouping levels, instrument description, selling trader ID, buying trader ID, the quantity of trade, transaction time, the date that the trade goes live or goes to mature in the system, currency rate, market operations feedback, and the likes.

To further verify the robustness of the proposed model on different link-based classification problems, we also employ a publicly available dataset about users in a retweet network[1]. This dataset contains a network of over 100,000 users. For each user, several content-related, network-related, and activity-related features were provided, such as average sentiment and subjectivity of his neighbors' tweets, his number of tweets, followers, followees, and favorites.

B. Data Preparation and Model Training

Our main dataset is the trading data. To properly use the transaction records, entries in the original dataset are carefully selected, and in the meantime, domain knowledge of financial

[1] https://www.kaggle.com/manoelribeiro/hateful-users-on-twitter?select=users_hate_glove.content

trading and investment is likewise considered to generate some new features. In the original dataset, the 'family' is a high-level grouping of products. Transactions can be categorized into four family types. The 'group' is a medium-level grouping of products, including 17 categories. Clearly, the groups provide a more accurate picture of traders' trading patterns than the families. The distributions of the products on groups are shown in Table I. For each transaction, there are also a number of temporal attributes in the record, such as the time that a transaction is requested from a trader, the time that the transaction is processed by the system, and the time that the transaction is confirmed by two sides. As the differences among these time records are small, we uniformly choose the last one. The value of each transaction is in pounds sterling based on the exchange rate of the date of the transaction.

Considering traders may have certain customary time patterns in trading, we reconstruct the time-associated features into three levels of time granularity: month, weekday, and time slice. The trading time pattern of traders varies widely, which seems not relevant to the transaction prediction. Whereas, one should recall the necessary condition of a transaction that there must be at least a small overlap between the traders' time patterns. The overlap will more or less contribute to the similarity of their trading time patterns. In addition, the details of each transaction are integrated, but different products are indiscriminately aggregated to calculate the attribute 'trading volume' as the product details are not the focus of this paper.

As transaction prediction is based on a trading network, one needs to further understand the structure of the trading network and extract the topological features of node pairs in it from the transaction records. We first integrate the features associated with two traders, including their identities, the groups of financial products that they have traded, the time distributions of transactions in terms of months, weekdays, and time slices. For example, the 'group' has a total of 17 categories, so the group feature of each node can be encoded by a 17-dimensional vector representing the group distribution of transactions. Based on the transformed features, we are able to build the feature vector of links with the similarities of the two nodes. In the measure of similarity, we use cosine similarity. After inferring a trader network, extracting the features of the nodes, and then using the similarities between the pairwise nodes to constitute the feature vectors of the links, we finally acquire the features shown in Table II.

The constructed trader network is composed of 1,149 nodes and 1,810 links in total, which is a relatively small social network comparing with other online social networks, such as Facebook, Twitter, etc. The density of the network is 0.003 with a small average degree of 3.1. Its clustering coefficient and average path length are 0.111 and 3.48, indicating that it is a typical small-world network. For trader pairs that have transaction records, we set their labels to 1, and 0 for the others. Note that the number of trader pairs without transaction records is much larger

TABLE II
FEATURES DESCRIPTION OF THE TRADING NETWORK

Features	Link-based features	Node-based features	Raw Data
Group	Group similarity of two nodes	17-dimensional vector, each dimension represents the number of transactions in the respective group	NB.INT:7595052 GROUP:LFUT B.TRADER:T251 S.TRADER:T533
Month	Month similarity of two nodes	12-dimensional vector, each dimension represents the number of transactions in the month	Trn.Time.2:340856481 in timestamp, that is, 1980-10-20 10:21:21
Weekday	Weekday similarity of two nodes	5-dimensional vector, each dimension represents the number of transactions in the Working day	Same as above
Time slice	Time slice similarity of two nodes	24-dimensional vector, each dimension represents the number of transactions in the hour	Same as above

TABLE III
HYPER-PARAMETER SETTINGS OF SUPERVISED LEARNING MODELS

Model	Hyperparameter
Support vector machine	Kernel: Radial basis function Regularization parameter: 10 Gamma: 0.001
Random forest	Tree numbers: 135 Min. samples be a leaf:19 Max. depth of the tree: 9
Gradient boosting decision tree	Tree numbers: 120 Learning rate: 0.1 Max. depth of the tree: 3 Min. samples required to split: 284 Min. samples be a leaf: 78 Subsample: 0.7
Multilayer perceptron	Activation: ReLU Weight optimization: Adam [50] L2 penalty parameter: 0.0001

TABLE IV
OVERALL RESULTS OF MODEL PERFORMANCE OF THE TRADER DATA

Model	Ranking-based fusion			Probability-based fusion		
	Prec (%)	Recall (%)	F1 (%)	Prec (%)	Recall (%)	F1 (%)
BGE	75.52	75.57	75.51	71.82	71.81	71.79
Supervised (fus)	70.28	70.25	70.23	35.75	35.75	35.73
Unsupervised (fus)	56.41	56.39	56.37	48.40	48.41	48.38
SVM + RA	63.87	63.87	63.84	37.29	37.30	37.28
SVM + SPM	69.23	69.23	69.20	46.91	46.92	46.89
SVM + QACO	70.94	70.95	70.91	37.02	37.02	37.00
RF + RA	69.12	69.10	69.08	31.88	31.91	31.88
RF + SPM	73.37	73.39	73.35	46.19	46.13	46.14
RF + QACO	74.92	74.39	74.89	31.27	31.30	31.27
GBDT + RA	69.56	69.56	69.53	31.71	31.73	31.71
GBDT + SPM	73.48	73.49	73.46	44.48	44.32	44.38
GBDT + QACO	74.81	74.84	74.79	30.99	31.01	30.99
MLP + RA	68.62	68.61	68.59	33.20	33.22	33.20
MLP + SPM	72.38	72.33	72.32	46.35	46.40	46.36
MLP + QACO	74.09	74.09	74.06	32.04	32.06	32.04

The best method in each column is highlighted in gray colour.

than the number of links in the network. In other words, the network is extremely sparse. Therefore, we randomly pick a number of negative samples, which is equal to the number of positive samples, to compose the final dataset for the purpose of solving the sample imbalance problem. Finally, we collect a total of 3,620 samples, with a positive to negative sample ratio of $1 : 1$.

We do not engineer features for the twitter data, and the constructed retweet network is composed of 1,004 nodes and 1,870 links in total. Its clustering coefficient and average path length are 0.212 and 2.46, indicating that it is a typical small-world network. For user pairs that have retweet records, we set their labels to 1 and 0 for the others. The network is likewise sparse. Therefore, we randomly pick a number of negative samples, which is equal to the number of positive samples, to compose the final dataset for the purpose of solving the sample imbalance problem. Finally, we collect a total of 3,740 samples, with a positive to negative sample ratio of 1:1.

In order to ensure the credibility of the results, we adopt a 10-fold cross-validation method. In the 10-fold cross-validation, the balanced sample set is randomly partitioned into 10 equal-sized subsets. Of the 10 subsets, a single subset is retained as the validation data for testing the model, and the remaining 9 subsets are used as a training set. Our training set is composed of 3,258 samples (3,366 samples for the retweet network), and the testing dataset is composed of 362 samples (374 samples for the retweet network). For the binary classification task, our setting guarantees that each fold contains roughly the same

proportions of the two types of class labels. The cross-validation process is repeated for 10 times, with each of the 10 subsets used exactly once as the validation data. The prediction results are then averaged to produce the final result. As introduced in Section III, we test four different supervised learning algorithms to present a feasible ranking vector, which are SVM, RF, GBDT, and MLP. The hyper-parameters of each algorithm are respectively optimized with the grid search. The detailed setting of hyper-parameters is shown in Table III.

C. Results and Analysis

Precision, recall, and F1-score are the most popular metrics for evaluating a machine learning algorithm's performance for classification problems. When we apply these metrics to the algorithms with the top-K selection in the link prediction [51], Precision is redefined as the ratio of the accurate predictions among the top-K predicted links [52]. For the balanced test sets, we set $K = \frac{|l|}{2}$, and precision is computed as

$$\text{Precision@K} = \frac{m}{K}. \tag{11}$$

For a given K, larger precision means higher prediction accuracy. Recall describes the ratio of the predicted links to the removed links. Let m be the number of correctly predicted links in the top-K of L for the test set, and we compute recall
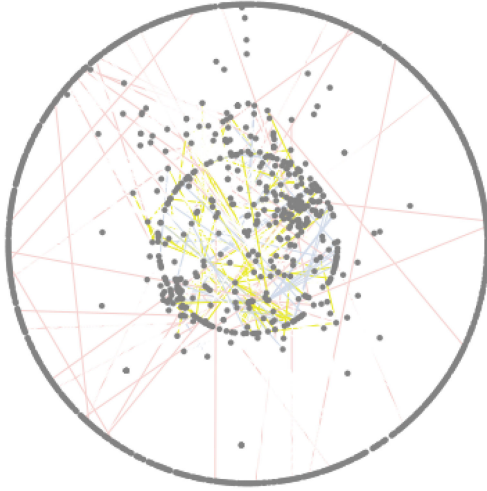
Fig. 4. Network visualizations of transaction prediction. The prediction result on a probe set randomly selected from the transaction logs is visualized here. The transaction log used in the experiment is a fragment of the entire transaction historical log from 2 January 2014 to 31 December 2014. In this inferred network, the traders (gray circles) with just one partner in the entire trading log are placed on the perimeter of a circle. The remaining traders are placed inside this circle, allocated among their neighbors. In the test set, a trader will move toward the center of the circle a certain distance every time he is at the end of a link to highlight the trader. The links in the training set are hidden. Solid blue lines denote the existing but not predicted transactions, solid yellow lines denote the correctly predicted transactions, and solid red lines denote the falsely predicted transactions.

as follows

$$\text{Recall@K} = \frac{m}{n}, \qquad (12)$$

where n is the total number of existing links in the test set. Another evaluation metric F1-score takes the harmonic mean of precision and recall.

Table IV shows the average results of the evaluation metrics with the 10-fold cross-validation for the trader network data. The best-performing feature combinations are highlighted in gray. Compared with the feature combinations provided by pure supervised or unsupervised models, one can see that a reasonable fusion of them achieves the best performance. Concerning the fusion mode, the ranking-based fusion generally performs better. For the ranking-based fusion, the performance of the feature fusion provided by Random Forest and 'QACO' is ranked second, followed by that of 'GBDT' and 'QACO'. The ranking-based fusion of the pure unsupervised link prediction methods performs poorly in the dataset, but its performance is better than most probability-based fusions. To further illustrate the transaction prediction based on the trader network, the prediction result of the BGE model in an experiment is visualized in Fig. 4, where the solid blue lines denote the existing but not predicted transactions, the solid yellow lines denote the correctly predicted transactions, and the solid red lines denote the falsely predicted transactions.

We run feature ablation tests in the two fusion modes to test the importance of features in our model. Table V shows that 'topology' has the most significant impact on model

TABLE V
ABLATION OF THE FEATURES

Fusion	Used features	Precision	Recall	F1
Probability	G, M, W, T1	0.677	0.677	0.677
Probability	G, M, W, T2	0.714	0.714	0.714
Probability	G, M, T1, T2	0.720	0.720	0.720
Probability	G, W, T1, T2	0.719	0.719	0.719
Probability	M, W, T1, T2	0.721	0.721	0.721
Probability	G, M, W, T1, T2	0.725	0.725	0.725
Rank	G, M, W, T1	0.684	0.684	0.684
Rank	G, M, W, T2	0.744	0.744	0.744
Rank	G, M, T1, T2	0.752	0.752	0.752
Rank	G, W, T1, T2	0.746	0.746	0.746
Rank	M, W, T1, T2	0.746	0.746	0.746
Rank	G, M, W, T1, T2	0.762	0.762	0.762

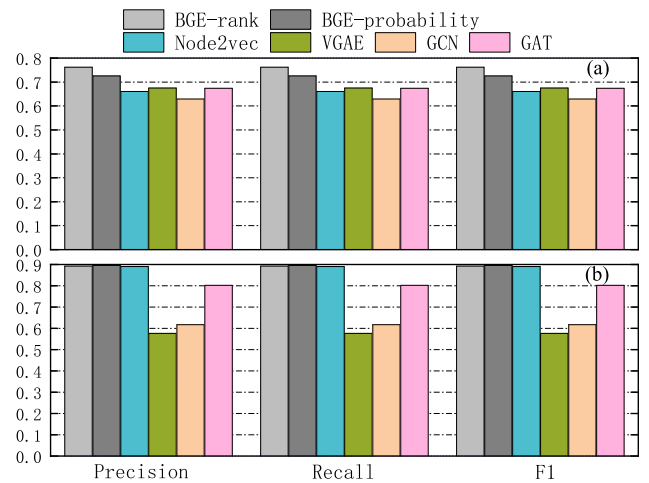G: Group; M: Month; W: Weekday; T1: Timestamp; T2: Topology.

Fig. 5. Performance evaluation of six graph embedding algorithms for (a) the transaction prediction on the trader networks and (b) the retweet prediction on the retweet networks.

performance, which is embedded by the unsupervised algorithms. The second is 'Timestamp,' and the least is 'Group'. Generally speaking, the impacts from 'Group,' 'Month,' 'Weekday,' and 'Timestamp' are relatively close. We do not test the importance of features in the retweet networks, since the number of features in the dataset is over 1,000, and such a test is computationally expensive and trivial.

Fig. 5(a) shows that the BGE-rank model outperforms the rest algorithms in all the metrics. The BGE-probability model is next to the BGE-rank model. The ranking-based fusion successfully tackles this problem, providing a novel pathway to fuse the attributes and topological features of links. At the same time, as three widely used methods in graph embedding, we observe that GCN, GAT, VGAE do not perform well in this task. Here, GCN and GAT are trained by the same decoder in GAE [47]. For the Node2-vec, we linearly combine the inner product of the embedded vectors with the supervised models. For the topological properties, one can see that the fusion performance by GCN is worse than that by Node2vec. The reason is that GCN

cannot effectively integrate the features of high-order neighbors, which play an essential role in the mission. Similar behaviors can be observed in the retweet networks. Fig. 5(b) shows that the F1-measures of the BGE-probability model and BGE-rank model are likewise higher than the rest algorithms in all the metrics, confirming the robustness of our model to network types. Interestingly, the performance of the BGE-probability model is slightly higher than the BGE-rank model, indicating that the reciprocal of the ranking indices may not be the best way to fuse the link-based features in the task. Next to the BGE-rank model, 'Node2vec' also performs well, indicating that the node attributes are crucial in the task.

V. Conclusion

This paper proposes a novel feature ranking framework to predict human interactions with historical interaction logs. Different from the existing methods, we adopt neither a pure supervised method with complicated feature extraction and learning procedures nor a pure unsupervised method, which can hardly integrate the individual properties with their connections. Instead, we apply a Bayesian graph embedding model for fusing the individual properties with the topological information of the network composed of their interconnections. The model successfully integrates the supervised interaction prediction and unsupervised link prediction with interaction probabilities. Extensive experimental results on two datasets of different types show that our model outperforms the tested benchmarked algorithms in precision, recall, and F1-score. We believe that our model provides a promising paradigm for further studies on network embedding and human interaction prediction. Admittedly, as we aim to promote the performance of interaction prediction algorithms, other metrics such as computational complexity, scalability to network size, and robustness to network type are not specifically optimized. To meet specific needs, one can rearrange the components of the model; the fusion among the individual properties and topological properties can be further improved. For instance, a non-linear combination may perform better in some tasks; semi-supervised extensions can likewise be considered if the dataset contains a large number of unlabeled data. Such possible extensions will be explored in our further work.

Appendix - Key Notations

$\mathbb{P}$	=	Probability of interaction
E	=	Set of links inferred by the interaction logs
A	=	Adjacency matrix of the network
L	=	Linking probability ranking
$T_{i,j}$	=	Transaction between two traders
u_i	=	Attributes learned from interaction logs
θ_{u_i,u_j}	=	Similarities between individual i and j
Z	=	Outputs of the candidate models
D	=	Number of candidate models
$\boldsymbol{w}$	=	Model weight
N_s	=	Number of supervised models
N_u	=	Number of unsupervised models

References

[1] K. Zhao, J. Stehlé, G. Bianconi, and A. Barrat, "Social network dynamics of face-to-face interactions," *Phys. Rev. E*, vol. 83, 2011, Art. no. 056109.

[2] A.-L. Barabási, "The origin of bursts and heavy tails in human dynamics," *Nature*, vol. 435, pp. 207–211, 2005.

[3] M. E. J. Newman and A. Clauset, "Structure and inference in annotated networks," *Nature Commun.*, vol. 7, no. 1, pp. 11863–11863, 2016.

[4] M. Yang, J. Liu, L. Chen, Z. Zhao, X. Chen, and Y. Shen, "An advanced deep generative framework for temporal link prediction in dynamic networks," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4946–4957, Dec. 2020.

[5] L. Ma, J. Li, Q. Lin, M. Gong, C. A. C. Coello, and Z. Ming, "Reliable link inference for network data with community structures," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3347–3361, Sep. 2019.

[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 585–591.

[7] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A.-J. Smola, "Distributed large-scale natural graph factorization," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 37–48.

[8] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 891–900.

[9] M.-D. Ou, P. Cui, J. Pei, Z.-W. Zhang, and W.-W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016. pp. 1105–1114.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[11] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[12] H. Wang *et al.*, "Graph representation learning with generative adversarial nets," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2508–2515.

[13] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[14] S. Harada *et al.*, "Dual convolutional neural network for graph of graphs link prediction," 2018, *arXiv:1810.02080*.

[15] W.-L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Eng. Bull.*, vol. 40, no. 3, pp. 52–74, 2017.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[17] R. Ying, J. You, C. Morris, X. Ren, W.-L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning withdifferentiable pooling," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 4805–4815.

[18] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–12.

[19] R. Cont, A. Moussa, and E. Santos, *Handbook on Systemic Risk, Ch. Network Structure and Systemic Risk in Banking Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2013, pp. 327–368.

[20] H. Amini, R. Cont, and A. Minca, "Resilience to contagion in financial networks," *Math. Finance*, vol. 26, no. 2, pp. 329–365, 2013.

[21] A. Babus and P. Kondor, "Trading and information diffusion in over-the-counter markets," *Econometrica*, vol. 86, no. 5, pp. 1727–1769, 2018.

[22] M. Al-Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *Proc. SDM06: Workshop Link Anal. Counter-Terrorism Secur.*, 2006, vol. 30, pp. 798–805.

[23] W. Cukierski, B. Hamner, and B. Yang, "Graph-based features for supervised link prediction," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1237–1244.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.

[26] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial, "Multi-level graph convolutional networks for cross-platform anchor link prediction," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2020, pp. 1503–1511.

[27] E.-C. Mutlu and T. A. Oghaz, "Review on graph feature learning and feature extraction techniques for link prediction," *Mach. Learn. Knowl. Extraction*, vol. 2, no. 4, pp. 672–704, 2020.

[28] R. Guimerà and M. Sales-Pardo, "Missing and spurious interactions and the reconstruction of complex networks," *Proc. Nat. Acad. Sci.*, vol. 106, no. 52, pp. 22073–22078, 2009.

[29] Z. Liu, J.-L. He, K. Kapoor, and J. Srivastava, "Correlations between community structure and link formation in complex networks," *PLoS One*, vol. 8, no. 9, 2013, Art. no e72908.

[30] A. Clauset, C. Moore, and M.-E.-J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[31] A. Muscoloni and C.-V. Cannistraci, "Local-ring network automata and the impact of hyperbolic geometry in complex network link-prediction," *CoRR*, vol. abs/1707.09496, 2017.

[32] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *Proc. Int. Joint Conf. Artif. Intell.*, 1999, vol. 99, pp. 1300–1309.

[33] D. Heckerman, C. Meek, and D. Koller, *Introduction to Statistical Relational Learning*. Cambridge, MA, USA: MIT Press, 2007.

[34] K. Yu, W. Chu, S. Yu, V. Tresp, and X. Zhao, "Stochastic relational models for discriminative link prediction," in *Proc. Adv. Neural Inf. Process. Syst.* (B. Schölkopf, J. C. Platt, and T. Hoffman, eds.), 2006, vol. 19, pp. 1553–1560.

[35] T. Zhou, L.-Y. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B*, vol. 71, no. 4, pp. 623–630, 2009.

[36] B. Zhu and Y. Xia, "An information-theoretic model for link prediction in complex networks," *Sci. Repots*, vol. 5, no. 13707, pp. 1–11, 2015.

[37] W.-P. Liu and L.-Y. Lü, "Link prediction based on local random walk," *Europhysics Lett.*, vol. 89, no. 5, 2010, Art. no. 58007.

[38] Z. Cao, Y. Zhang, J. Guan, and S. Zhou, "Link prediction based on quantuminspired ant colony optimization," *Sci. Rep.*, vol. 8, 2018, Art. no. 13389.

[39] P. Zhang, X. Wang, F. Wang, A. Zeng, and J. Xiao, "Measuring the robustness of link prediction algorithms under noisy environment," *Sci. Rep.*, vol. 6, no. 18881, pp. 1–7, 2016.

[40] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.

[41] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2005, pp. 177–187.

[42] F. Yang, K. Fan, D. Song, and H. Lin, "Graph-based prediction of protein-protein interactions with attributed signed graph embedding," *BMC Bioinf.*, vol. 21, pp. 1–16, 2020.

[43] B.-L. Chen and L. Chen, "A link prediction algorithm based on ant colony optimization," *Appl. Intell.*, vol. 41, no. 3, pp. 694–708, 2014.

[44] L.-Y. Lü, L. Pan, and T. Zhou, "Toward link predictability of complex networks," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 112, no. 8, pp. 2325–2330, 2015.

[45] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2019.

[46] A. Grover and J. Leskovec, "Node2vec: Scalable Feature Learning for Networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[47] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *Proc. Neural Inf. Process. Syst. Workshop Bayesian Deep Learn.*, 2016, pp. 1–3.

[48] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1293–1299.

[49] C. Bishop, *Pattern Recognition and Machine Learning* New York, NY, USA: Springer, 2006.

[50] D.-P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[51] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[52] Y. Yang, R.-N. Lichtenwalter, and N.-V. Chawla, "Evaluating link prediction methods," *Knowl. Inf. Syst.*, vol. 45, no. 3, pp. 751–782, 2015.

Yichao Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from Tongji University, Shanghai, China. He is currently an Associate Professor with the Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interests include link prediction, modelling of weighted networks, random diffusion on weighted networks, and evolutionary games on networks.

Huangxin Zhuang received the bachelor's degree in computer science and technology from Tongji University, Shanghai, China, where he is currently working toward the master's degree with the Department of Computer Science and Technology. His research interests include information diffusion and social networks.

Tiantian Liu received the B.S. degree from the China University of Petroleum (East China), Dongying, China, in 2018. She is currently working toward the M.S. degree in computer science and technology with Tongji University, Shanghai, China. Her research interests include machine learning, data mining, and social network analysis.

Bowei Chen (Member, IEEE) received the Ph.D. degree in computer science from University College London, London, U.K. He is currently an Associate Professor with the Adam Smith Business School, University of Glasgow, Glasgow, U.K. His research interests include the applications of probabilistic modeling and deep learning in business, with special focuses on marketing and finance. He is on the Editorial Boards of the *Electronic Commerce Research and Applications*, *Frontiers in Big Data*, and *Frontiers in Artificial Intelligence*.

Zhiwei Cao received the master's degree in systems engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2012. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tongji University, Shanghai, China. He is also an Assistant Researcher with the Information Security Technology Division, The Third Research Institute of Ministry of Public Security, Shanghai, China. His current research interests include link prediction and computational intelligence.

Yun Fu received the Ph.D. degree in computer science from the University College London, London, U.K. He is currently a Teaching Fellow with the Department of Computer Science, University College London, U.K. His research interests include machine learning, computer vision, and bioinformatics.
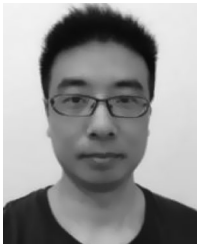
Zhijie Fan received the master's degree from Zhejiang University, Hangzhou, China, and the Ph.D. degree from Tongji University, Shanghai, China. He is currently a Postdoctoral with the School of Computer Science, Fudan University, Shanghai, China. He is a Researcher with the Information Security Technology Division, The Third Research Institute of Ministry of Public Security, Shanghai, China. His research interests include machine learning and network security.

Guanrong Chen (Life Fellow, IEEE) received the M.Sc. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 1981, and the Ph.D. degree in applied mathematics from Texas A&M University, College Station, TX, USA, in 1987. Since 2000, he has been a Chair Professor and the Founding Director of the Centre for Chaos and Complex Networks, City University of Hong Kong, Hong Kong. Prior to that, he was a tenured Full Professor with the University of Houston, Houston, TX, USA. He was awarded the 2011 Euler Gold Medal, Russia, and conferred Honorary Doctorate by the Saint Petersburg State University, Russia in 2011 and by the University of Le Havre, Normandie, France in 2014. He is a Member of the Academia of Europe and a Fellow of The World Academy of Sciences and is a Highly Cited Researcher in Engineering also in Mathematics according to Thomson Reuters.