

Friendship Inference in Mobile Social Networks: Exploiting Multi-Source Information With Two-Stage Deep Learning Framework

Yi Zhao¹, Member, IEEE, ACM, Meina Qiao², Haiyang Wang, Member, IEEE, Rui Zhang³, Member, IEEE, Dan Wang, Senior Member, IEEE, and Ke Xu⁴, Senior Member, IEEE, Member, ACM

Abstract—With the tremendous growth of mobile social networks (MSNs), people are highly relying on it to connect with friends and further expand their social circles. However, the conventional friendship inference techniques have issues handling such a large yet sparse multi-source data. The related friend recommendation systems are therefore suffering from reduced accuracy and limited scalability. To address this issue, we propose a Two-stage Deep learning framework for Friendship Inference, namely TDFI. This approach enables MSNs to exploit multi-source information simultaneously, rather than hierarchically. Therefore, there is no need to manually set which information is more important and the order in which the various information is applied. In details, we apply an Extended Adjacency Matrix (EAM) to represent the multi-source information. We then adopt an improved Deep Auto-Encoder Network (iDAEN) to extract the fused feature vector for each user. Our framework also provides an improved Deep Siamese Network (iDSN) to measure user similarity. To provide a substantial description and evaluation of the proposed methodology, we evaluate the effectiveness and robustness on three large-scale real-world datasets. Trace-driven evaluation results demonstrate that TDFI can effectively handle the sparse multi-source data while providing better accuracy for friendship inference. Through the comparison with numerous state-of-the-art methods, we find

that TDFI can achieve superior performance via real-world multi-source information. Meanwhile, it demonstrates that the proposed pipeline can not only integrate structural information and attribute information, but also be compatible with different attribute information, which further enhances the overall applicability of friend-recommendation systems under information-rich MSNs.

Index Terms—Mobile social networks, friendship inference, multi-source information, deep learning.

I. INTRODUCTION

IN THE past decade, the mobile Internet has profoundly promoted the prosperity of mobile social networks (MSNs) [1], [2], [3]. According to the global digital population statistics as of April 2020 [4], over 4.2 billion people constitute unique mobile Internet users, encompassing 91.9% of the global active Internet users. The number of active mobile social media users has also exceeded 3.76 billion, up to 98.7% of the active social media users. Meanwhile, each mobile social service provider is accelerating the expansion of its user base to occupy a broader market. For example, the number of daily active *Instagram* users is up to a staggering increase from 400 millions in June 2018 to 500 millions in January 2019 [5]. These users not only use social media to find information, watch movies, buy and sell products, but also highly rely on it to connect with others and expand their social circles. Under such an incredible large-scale MSN, however, it is impossible for mobile social service providers to check the information of each user and quickly pinpoint their potential friends. This requires relevant MSNs to have the ability to automatically match potential friends. Consequently, friend-recommendation service has been widely adopted by most mobile social service providers, like *Facebook*, *Instagram*, and *Twitter* [6].

Finding the accurate potential friend relationship among all the social media users is like searching for a needle in a haystack. To reduce such complexity, conventional friendship-inference methods mainly depend on mutual friends, which provides long potential friend lists with very low precision [6]. In particular, friend-recommendation based on social graph representation [7], [8], [9], [10] is largely exploited. However, real-world mobile social networks are much more sparse than expected [11], [12], [13] (*i.e.*, the number of true friends is much smaller than that of non-friends, as illustrated in Table III), which poses a challenge

Manuscript received 24 October 2020; revised 20 June 2022; accepted 21 July 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Shakkottai. Date of publication 25 August 2022; date of current version 18 April 2023. This work was supported in part by the National Science Foundation for Distinguished Young Scholars of China under Grant 61825204, in part by the National Natural Science Foundation of China under Grant 61932016, in part by the Beijing Outstanding Young Scientist Program under Grant BJJWZYJH01201910003011, in part by the China Postdoctoral Science Foundation under Grant 2021M701894, and in part by the China National Postdoctoral Program for Innovative Talents. The work of Dan Wang was supported by GRF under Grant 15210119, Grant 15209220, Grant 15200321, Grant 15201322, Grant ITF-ITSP ITS/070/19FP, and Grant C5018-20G. (Corresponding author: Ke Xu.)

Yi Zhao is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: zhao_yi@tsinghua.edu.cn).

Meina Qiao is with the Department of Computer Vision Technology (VIS), Baidu Inc., Beijing 100193, China (e-mail: qiaomeina@baidu.com).

Haiyang Wang is with the Department of Computer Science, University of Minnesota Duluth, Duluth, MN 55812 USA (e-mail: haiyang@d.umn.edu).

Rui Zhang is with the School of Artificial Intelligence, OPTics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: ruizhang8633@gmail.com).

Dan Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csdwang@comp.polyu.edu.hk).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with the Zhongguancun Laboratory, Beijing 100094, China (e-mail: xuke@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TNET.2022.3198105

1558-2566 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

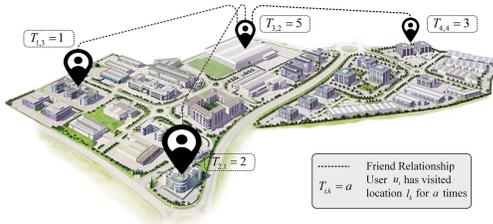


Fig. 1. An overview of the two types of information.

to existing approaches. Moreover, friend information is highly privacy-sensitive and deeply connected with our social identity [14], [15], [16]. More and more people choose to hide their friend information, such that the social network we can build become more sparse. For example, almost 17.2% *Facebook* users in New York hid their friend information in 2010 [17]. Worse more, these approaches cannot fully reflect the real preferences on friend selection [18]. This is due to the fact regarding missing of important real-world information such as users' different lifestyles [19], interests [20], and locations [21].

To better understand users' preferences, recent studies revealed that modern social network applications have largely changed and diversified the activities of MSN users [6], [22], [23]. For example, instead of sending text messages to their friends, the mobility enables MSN user to share a variety of moments, including various locations, landscape photos, and pleasant videos to their social circles. The deep utilization of such multi-source information, which can more truly reflect the behavior of users making friends, has attracted the attention of the academic community [6], [22], [24], [25], [26]. Moreover, through the complementary advantages, the utilization of multi-source information can also alleviate the impact of sparse problem in real-world social networks. When some information is insufficient, other information can still perform the role of friendship inference. This can be further illustrated in Fig. 3. Since the utilization of multi-source information increases complexity, the current mainstream approach is utilizing multi-source information hierarchically. Unfortunately, it requires mobile social service providers to manually give the importance of different information as well as the order in which the various information is applied. However, in real-world mobile social networks, there are various types of information that can be used for friendship inference, and the required manual settings are difficult to give accurately.

To cope with above challenges, we propose and implement a novel Two-stage Deep learning framework for Friendship Inference, namely TDFI. Via treating multi-source information as a whole input, the newly proposed TDFI can smartly process the multi-source information, including structure information (*i.e.*, friendship information among different users) and different attribute information (*e.g.*, location information). In other words, our TDFI enables MSNs to simultaneously exploit multi-source information for friendship inference, rather than hierarchically. Therefore, there is no need to manually set which information is more important and the order in which the various information is applied. In the details

of implementing our TDFI, it consists of three complementary strategies. To ensure scalability, we apply an Extended Adjacency Matrix (EAM) to better represent the multi-source information. After that, an improved Deep Auto-Encoder Network (iDAEN) is proposed to extract one fused feature vector for each user from the multi-source information. Furthermore, TDFI provides an improved Deep Siamese Network (iDSN) to measure the similarity of the fused feature obtained by the iDAEN network, identifying potential friends. In addition, to protect user privacy from being leaked, our framework only requires coarse-grained information to achieve sufficiently satisfactory friendship inference performance.

As illustrated in Fig. 1, in this paper, we utilize friend relation and location as an example of the multi-source information. Note that from the overall perspective of mobile social networks, friend relationships are structural information. On the contrary, from a local perspective of each user, which friends the user has can also be regarded as a kind of attribute information. Location is another common attribute information, because it is a property that naturally exists with MSNs, and location sharing services (*e.g.*, check-in service) are already built into most MSN applications. Meanwhile, location information can further reflect the user's behaviors, which is very representative. Eagle *et al.* [21] confirmed that data such as location information obtained through mobile devices has extraordinary potential in social network analysis. And Scellato *et al.* [27] also indicated that synchronous check-ins information among users can imply potential friendship. Additionally, Backes *et al.* [23] found that check-in information can denote the mobility characteristics, which are significant for inferring friendship. With the cooperation of two DL-based modules, our proposed pipeline can not only integrate structural information and attribute information, but also be compatible with different attribute information. It is worth noting that the location information is applied as a case study, TDFI has good scalability and can incrementally consider different categories of information while obtaining a reasonable complexity. For the purpose of facilitating real-world deployment, the effectiveness and robustness of TDFI are carefully evaluated on three large-scale real-world datasets collected from *Instagram* [23]. Furthermore, the trace-driven comparison has demonstrated that the newly proposed TDFI can effectively cope with the ubiquitous sparse problem in MSNs, and significantly outperforms numerous state-of-the-art methods for friendship inference.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III formally defines the friendship inference problem and introduce the newly proposed TDFI. In Section IV, we introduce the experimental setup and baseline methods. Furthermore, we conduct experiments to evaluate the performance comparison between our TDFI and referred baseline methods in Section V. Finally, in Section VI, we conclude this paper.

II. RELATED WORK

The rapid expansion of mobile social networks involves many aspects of human life, such as leisure and entertainment,

advertising for third-party markets, *etc.* As the scale of MSNs expands, it is extremely important to improve the quality of friend-recommendation services. Traditional friendship-inference methods mainly emphasized the mutual friends or the same groups. The intrinsic nature is that a person is more likely to know a person of his friends rather than a random person [28]. Following this idea, multiple literatures [7], [8], [29], [30] started to use social graph representation to infer friendship among different users. For example, through learning the representation of the social network, node2vec [8] extended the feature of an individual user to a pair of users, aiming to find their potential friends. However, real-world social networks are much more sparse than expected [11], [12], [13]. The graph composed of real-world social networks is sparse, which is not conducive to extracting essential information. Thus, with the limitation of sparseness, friendship inference with only friend information might lead to awkward consequences.

Mobile social networks make it possible to diversify social forms, and thus derive various user-related attribute information, such as the location and photo shared by *Instagram*, the videos shared by *TikTok*, and the texts shared by *Twitter*. To utilize these attribute information, another line of work [24], [25], [31], [32] focuses on analyzing attribute information with the help of structural information. For example, Zhang *et al.* [32] propose a novel graph recurrent neural framework, which can jointly exploit social information (*i.e.*, structural information) and text information (*i.e.*, attribute information). Based on graph convolutional network (GCN), Kipf and Welling [25] demonstrate that the combination of graph-structure information and attribute information can produce better performance than graph-based methods [7], [8]. However, these methods are difficult to be compatible with the mutual interference of multiple attribute information, which can be found in Section IV-B.2.

With such diverse multi-source information, many studies [6], [18], [26] focus on integrating diverse information into the friend-recommendation systems. For example, Huang *et al.* [6] designed a topic model, which can utilize text information, friend information and image information. Specifically, the friend information and the text information are first used to give a candidate list of possible friends. Then, using the image information, a topic model is adopted to further optimize the candidate list. Among the multi-source information, user location is widely suggested [21], [33], [34]. This is because the location information can reflect some user behaviors in physical space [21]. For example, based on the observation of *Gowalla*, Cho *et al.* [35] found that mobility and social constraints are related. And Pham *et al.* [36] investigated an entropy-based model to entirely utilize location information, which can not only infer friendship but also measure the strength of friendship. Due to the rich information of user mobility, Zhou *et al.* [37] innovatively proposed a heterogeneous data unification method for friendship inference. First, user mobility and friend information are represented by a matrix and a graph, respectively. Subsequently, this method [37] uses a probabilistic factor model and neural network embedding to extract low-dimensional representations

of user mobility and friend information, respectively, which are then concatenated for friendship inference. Besides, location information can also be applied to social network attacks [23], [38]. For example, with the assistance of random walk and feature learning, Backes *et al.* [23] can obtain the features of users' mobility, which can be used to attack the friendship among different users. However, how to utilize these multi-source information (*e.g.*, friend information and location information) simultaneously rather than hierarchically remains a challenging issue.

Different from existing friend-recommendation systems, the proposed TDFI framework can automatically handle multi-source information simultaneously, rather than hierarchically. Note that the referred multi-source information, can be compatible with structural information and also with different attribute information of multiple scales. In other words, TDFI can complement the advantages of different information, avoiding the performance issues caused by insufficient single-source information. In addition, the existing methods that can utilize multi-source information require manually set some factors (*e.g.*, the order in which the various information is applied). Our proposed TDFI is not constrained by this requirement, thus MSNs can automatically exploit various information with the proposed TDFI. This further facilitates the development of TDFI in the real-world MSN applications.

III. TWO-STAGE DEEP LEARNING FRAMEWORK

In this section, we first define the friendship inference problem. We then describe each component of the newly proposed framework. Mobile social network is a social structure connecting individuals. We can model MSN as a graph, denoted by $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U} = \{u_1, u_2, \dots, u_N\}$ denotes the set of users and $\mathcal{E} = \{e_{i,j} | i \in [1, N], j \in [1, N]\}$ indicates the edges, *i.e.*, relationship between two users. If there exists friendship between u_i and u_j , then $e_{i,j} = 1$ or $e_{i,j} = 0$ otherwise. Friendship information is represented by $\mathcal{F} = \{(u_i, u_j) | e_{i,j} = 1, e_{i,j} \in \mathcal{E}, i < j\}$, where $F = |\mathcal{F}|$ is the total number of friend pairs. With the mobility brought by MSNs, it is naturally convenient for users to share their location information. We define $\mathcal{L} = \{l_1, l_2, \dots, l_M\}$ to represent the set of all different locations. For location information, $T_{i,k}$ is used to represent the frequency that u_i has visited l_k . For example, $T_{i,k} = 0$ denotes that u_i has never visited l_k . The value of $T_{i,k}$ can be calculated from the check-in dataset $\mathcal{C} = \{(c_n, u_i, l_k) | n \in [1, C]\}$, where $C = |\mathcal{C}|$ is the total number of check-ins. For instance, (c_3, u_2, l_1) means that the third record in the check-in dataset shows that u_2 made a check-in at l_1 . However, the value of $T_{2,1}$ may be larger than 1, because $T_{2,1} = |\{(c_n, u_i, l_k) | i = 2, k = 1, n \in [1, C]\}|$. Note that our framework only requires some coarse-grained check-in information (*i.e.*, the user id, and the location id) for respecting the user privacy.

For the sake of clarity, Table I lists the major notations to be used extensively in the rest of this paper. It is worth noting that the specific meaning of the symbol also depends on its superscript and subscript. In addition, we list the frequently

TABLE I
 MAJOR NOTATION EXPLANATION

Symbol	Description
u_i or u_j	A specific user with identity i or j
$e_{i,j}$	Indication of whether u_i and u_j are friends
l_k	A specific location with identity k
$T_{i,k}$	Frequency of user u_i going to location l_k
N	The total number of all different users
M	The total number of all different locations
\vec{a}_i	Initial overall multi-source information of u_i
α_i	Coefficient of friend information
β_i	Coefficient of location information
$\vec{\tilde{a}}_i$	Overall multi-source information of u_i after rescaling operation (<i>i.e.</i> , local and global normalizations)
ϕ	Parameters in the encoder network
φ	Parameters in the decoder network
γ	Hyperparameter to leverage the tradeoff between non-zero element penalty and reconstruction error
ψ	All parameters in the iDSN network

 TABLE II
 FREQUENTLY USED TERMS

Term	Description
EAM	Extended adjacency matrix to represent the multi-source information
iDAEN	The first module of TDFI, which can extract fused feature for each user
iDSN	The second module of TDFI, which can measure the similarity between users
GCN [24]	An efficient variant of convolutional neural networks, which operate directly on graphs
GAE [25]	Extended implementation of GCN [24] with Auto-Encoder (AE)
VGAE [25]	Extended implementation of GCN [24] with Variational Auto-Encoder (VAE) [39]

used terms in Table II for a more concise understanding of the subsequent content.

To accurately infer friendship via multi-source information, three complementary strategies are proposed in this paper, *i.e.*, the EAM, iDAEN, and iDSN. More specifically, they play the role of multi-source information representation, fused feature extraction, and friendship inference, respectively, all of which are indispensable for the proposed framework. More discussion and analysis of each component of TDFI can be found in the following content, which can clarify the importance of each component and its performance contribution.

A. EAM: Extended Adjacency Matrix

We first introduce the proposed extended adjacency matrix (EAM) for multi-source information representation of MSNs, which is denoted by A . The intrinsic of EAM is to be compatible with various information, to ensure the scalability of TDFI. As illustrated in Fig. 2, the indexes in the rows denote user id $\{u_i\}_{i=1}^N$, and the indexes in the columns are user id $\{u_j\}_{j=1}^N$ and location id $\{l_k\}_{k=1}^M$. In Fig. 2, the yellow area is the *friendship group*. If user u_i has friendship with user u_j , then the value $a_{i,j} = 1$ or $a_{i,j} = 0$ otherwise. Moreover, our *friendship group* definition follows a symmetric property, *i.e.*, $a_{i,j} = a_{j,i}$. Note that the referred three datasets in Section IV-A only provide friend relationships between users

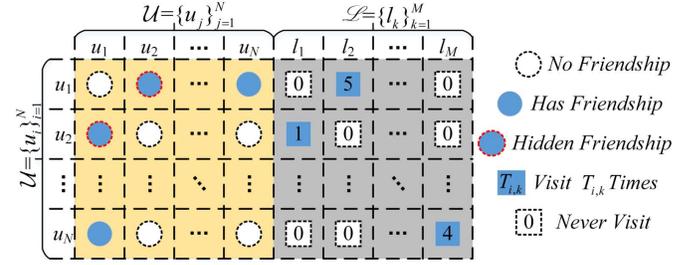


Fig. 2. The initial EAM without local and global normalizations.

with different ids. To be consistent with these datasets and to make a fair comparison with baseline algorithms, we consider each user to have no friend relationship with himself, *i.e.*, $a_{i,i} = 0$. In other words, in our defined EAM, the diagonals of the *friendship group* (*i.e.*, the yellow area) are all zero elements. The gray area is the *check-in group*. The value of $a_{i,N+k}$ denotes the times of user u_i visiting location l_k , *i.e.*, $T_{i,k}$. Note that $\vec{a}_i = \{a_{i,1}, \dots, a_{i,N}, a_{i,N+1}, \dots, a_{i,N+M}\}$ represents the overall multi-source information of u_i .

Regarding our proposed EAM, it can simplify manual settings of multi-source information serving friendship inference simultaneously. For example, when dealing with information from different sources, the heterogeneous data unification method proposed by Zhou *et al.* [37] requires that a specific feature extraction method should be selected separately for each type of source, and then utilizes the concatenated features for friendship inference. In our method, multi-source information can be uniformly represented only by numerical definition and subsequently used for low-dimensional feature extraction simultaneously, avoiding the selection of feature extractors.

For a variety of multi-source information, it is easy to construct the initial EAM like Fig. 2. However, the metrics for different categories of information are different, which can easily lead to imbalances between information. For example, in the friend information (*i.e.*, the yellow area) of the initial EAM, any unit can only be 0 or 1, to indicate whether there is a friend relationship between two users. In the location information (*i.e.*, the gray area), each unit may be any non-negative integer. When the value of the location information is relatively large, the friend information is easily regarded as a value close to 0, thereby causing the friend information ignored by the neural network. Similar information imbalances are more likely to occur when there is more variety of multi-source information to use. To make multiple different source information comparable, therefore, we design the rescaling operation. It consists of local and global normalizations. Based on the initial EAM A , the newly designed rescaling operation enable all information to contribute to the following feature extraction. More specifically, the local normalization is carried out separately in the *friendship group* and *check-in group*. As for *friendship group*, the local normalization is calculated by $a'_{i,j} = \frac{a_{i,j}}{\sum_{j=1}^N a_{i,j}}$. Likewise, the local normalization in *check-in group* is calculated by $a'_{i,N+k} = \frac{a_{i,N+k}}{\sum_{k=1}^M a_{i,N+k}}$.

Due to the gap between the number of friends and check-ins, global normalization method is employed in addition to the local normalization of group information. For user u_i , the information strength of friend information and location

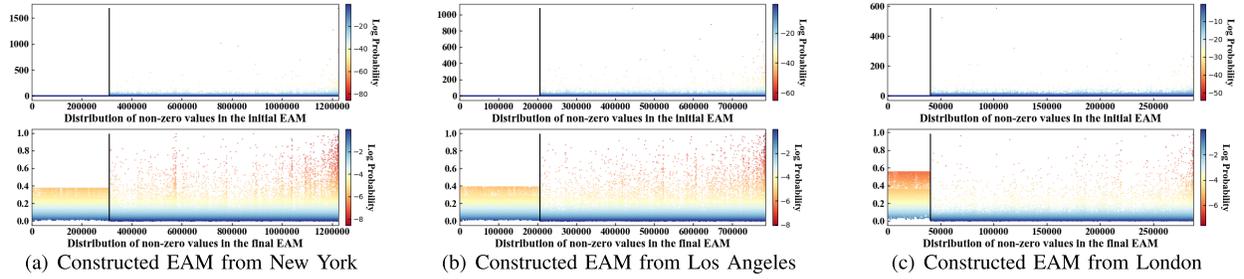


Fig. 3. Distribution of various non-zero elements in the EAM, where the final EAM refers to extended adjacency matrix after undergoing the rescaling operation. The left part of the black vertical line is the distribution of all friendship information, and the right part is the distribution of all users' location information. Note that the value of Y-axis is the logarithm of each non-zero element, avoiding the excessive value of some non-zero elements.

information can be achieved via Eq. (1) and Eq. (2).

$$\alpha_i = \frac{F_i/\bar{F}}{F_i/\bar{F} + C_i/\bar{C}} \quad (1)$$

$$\beta_i = \frac{C_i/\bar{C}}{F_i/\bar{F} + C_i/\bar{C}} \quad (2)$$

where $\bar{F} = F/N$ and $\bar{C} = C/N$ represent the average number of friends and check-ins per user, respectively. Additionally, F_i and C_i are the corresponding number of friends and the number of check-ins of u_i . Moreover, α_i and β_i are the coefficients of *friendship group* and *check-in group*, respectively. \tilde{A} is specifically used to denote the matrix after local and global normalizations. In other words, the values in each group are $\tilde{a}_{i,j} = \alpha_i * a'_{i,j}$ and $\tilde{a}_{i,N+k} = \beta_i * a'_{i,N+k}$.

From the global normalization method, it can be found that when the provided friend information is richer, the *friendship group* has a higher proportion, otherwise the *check-in group* has a higher proportion. Accordingly, we can utilize friendship information and check-in information with different proportions. In other words, through the coordination of the corresponding coefficients of each type of information sources with different physical meanings in the global normalization, the information-rich sources can be more prominent in the subsequent feature extraction, and the potential interference caused by the information-poor sources can be weakened. Due to the purpose of inferring friendship in the subsequent process, certain friendship may be hidden randomly, as illustrated by the blue circle with the red dash line in Fig. 2. Therefore, our proposed TDFI is evaluated for inferring the specific friendship (*i.e.*, manually hidden friendship), from check-in information and the remaining friendship information.

To support the above analysis, we further analyze the distribution of various non-zero elements in the EAM (*i.e.*, the initial EAM and the final EAM) and the corresponding statistical information, illustrated in Fig. 3. As notified in Section V-B, 5-fold cross-validation is used in our experiments to fully verify the effectiveness of the newly proposed method. For three different datasets, we have to construct 15 EAMs. For simplicity, we use the overall friend information to construct 3 EAM (*i.e.*, one EAM per dataset, Fig. 3(a), Fig. 3(b), and Fig. 3(c)). Note that in the experimental evaluation, each constructed EAM will not have any friend information for testing. For each dataset, with regard to the range of the X-axis, we can see that the right part of the black vertical line is much larger than the left part. This means that there

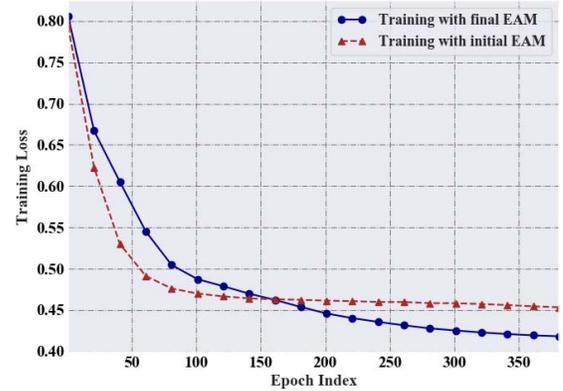


Fig. 4. The difference in loss values caused by rescaling operation during the training phase.

are far more non-zero elements in the location information (*i.e.*, the right part) than that in the friend information (*i.e.*, the left part), that is, the imbalance of different categories of information exists. This emphasizes the necessity of global normalization. Specifically, the number of non-zero elements in the location information of New York, Los Angeles, and New York, is 2.9 times, 2.8 times, and 6.1 times that of the friend information, respectively. In addition, for friend information, the proportion of non-zero elements in the three datasets is about 0.00016, 0.00021, and 0.00023, respectively. For location information, the proportion of non-zero elements in the three datasets is about 0.00137, 0.00116, and 0.00216, respectively. This once again demonstrates the imbalance of different categories of information, but also reflect the sparse nature of the EAM. In Section III-B, we will further solve the sparse problem. For the initial EAM illustrated in Fig. 3 (*i.e.*, top three subfigures), all non-zero elements in friend information are the same value (*i.e.*, $0 = \log(1)$, where 1 means there is a friend relationship.), while the range of values of location information is extremely wide. After undergoing the rescaling operation (*i.e.*, local and global normalizations), the distribution of non-zero elements in the final EAM is more balanced, which is conducive to the feature extraction through the deep neural network-based iDAEN in Section III-B.

To elucidate the necessity of rescaling operation (*i.e.*, local and global normalizations), in addition to coping with the aforementioned data imbalance and sparseness issues, we also provide simulation experiments to highlight the advantages

that rescaling operation brings to model training. First of all, our proposed iDAEN in Section III-B is based upon AE, and its training goal is to minimize the reconstruction error. However, the value ranges of initial EAMs constructed by the three datasets (*i.e.*, London, Los Angeles and New York) are $[0, 585]$, $[0, 1076]$ and $[0, 1687]$, respectively. After using the rescaling operation to convert initial EAM to final EAM, the corresponding value range is only $[0, 1]$. If initial EAM and final EAM are used as inputs respectively, there will be a significant difference in the reconstruction error for iDAEN. Therefore, we utilize a third-party model, *i.e.*, GAE in Section IV-B.2, to observe the differences caused by the rescaling operation in the training process. This is because, regardless of whether there is a difference in the value range of the user's attribute information (*i.e.*, a row of initial EAM or final EAM), the loss value of GAE is in $[0, 1]$, which is convenient for observation. Fig. 4 illustrates the training details of GAE on the London dataset. It can be found that after converting initial EAM into final EAM through the rescaling operation (*i.e.*, local and global normalizations), the converged model will have a lower loss value, which means better training results.

B. Feature Extraction

With the proposed EAM, we already have the ability to represent multi-source information simultaneously. However, this representation still cannot directly extract the essence of multi-source information. Moreover, considering the scalability and complexity, accompanied by more different sources of information, the dimensions of the EAM become larger. This will make it very hard for the iDSN network to directly learn the similarity between a pair of users based on the labels. Considering that deep auto-encoder network has outstanding performance for feature extraction and information fusion [40], [41], we propose the iDAEN network, utilized as the first stage of TDFI to extract one fused feature vector for each user via multi-source information represented by EAM. The iDAEN network can obtain more general nonlinear combinations of variables compared to both linear and nonlinear approaches. In addition to extracting robust features, it can also reduce the dimensionality of information such that computational resources can be saved for the second stage of TDFI (*i.e.*, the iDSN network). Regarding improving convergence time efficiency, in addition to reducing the fused information dimension, we apply special mechanisms, *i.e.*, skip operation and feature-reuse, in the design of iDAEN and iDSN, which will be introduced in the following descriptions of Fig. 5 and Fig. 6.

As illustrated in Fig. 5, the iDAEN network is composed of two symmetric structures (*i.e.*, Encoder and Decoder). Here it is specifically designed to include 7 hidden layers. Like most deep neural networks, adding more layers may result in better accuracy, but also increase time overhead. Combining various factors (*e.g.*, inference accuracy and time overhead) and generalizability for three datasets, the design of 7 hidden layers is suitable. Equipped with the encoder network \mathbb{E} , which projects the input $\mathcal{X} = \{\tilde{x}_i\}_{i=1}^N = \{\vec{a}_i\}_{i=1}^N$ onto a latent space \mathcal{Z} with low dimensionality, the decoder network \mathbb{D} is to

reconstruct the input from the latent space. The standard loss function of deep auto-encoder network is for measuring the reconstruction error between input and output, and the goal of training is to calculate suitable parameters Θ to minimize the loss function, as represented by Eq. (3).

$$\begin{aligned} \arg \min_{\Theta=\{\phi, \varphi\}} \quad & \|\hat{\mathcal{X}} - \mathcal{X}\|^2 \\ \text{s.t.} \quad & \mathcal{Z} = \mathbb{E}(\mathcal{X}, \phi), \hat{\mathcal{X}} = \mathbb{D}(\mathcal{Z}, \varphi) \end{aligned} \quad (3)$$

where $\hat{\mathcal{X}} = \{\vec{x}_i\}_{i=1}^N$ denotes the output, ϕ and φ are the parameters in the encoder and decoder network, respectively.

Although the deep auto-encoder network is effective for feature fusion, it still needs to be improved to yield to our scheme. As mentioned earlier, the sparse problem of friends is ubiquitous in real social networks. That is, the user population is huge, while the number of friends is relatively small. Thus, the proposed EAM is sparse (*i.e.*, the number of non-zero elements is far less than that of zero elements), such that the network might be unable to extract sufficient features. To address this problem, penalty on non-zero elements is added to force the iDAEN network to learn the non-zero features. In particular, the loss function in Eq. (3) is rewritten as Eq. (4).

$$\begin{aligned} \mathcal{L}_A(\Theta) &= \|(\hat{\mathcal{X}} - \mathcal{X}) \circ \mathcal{X}'\|^2 \\ &= \sum_{i=1}^N \sum_{j=1}^{N+M} ((\hat{x}_{i,j} - x_{i,j}) \cdot (\gamma \cdot x_{i,j} + 1))^2 \end{aligned} \quad (4)$$

where the symbol \circ means the pointwise product, $\mathcal{X}' = \{x'_{i,j} = (\gamma \cdot x_{i,j} + 1) | i \in [1, N], j \in [1, N + M]\}$, and γ is a hyperparameter to leverage the tradeoff between penalty on the non-zero elements and the reconstruction error. Regarding the operation of adding 1, this is primarily to ensure that the prediction of zero elements with error (*i.e.*, $x_{i,j} = 0$, but $\hat{x}_{i,j} \neq 0$) can has a non-zero loss value.

Since the gradients may vanish or explode, skip connection is also added between the layers in encoder network and decoder network in addition to changing the activation functions. More specifically, the skip connection operation can be represented as Mg in Eq. (5).

$$Mg(m) = E_i^m \oplus D_j^m \quad (5)$$

where E_i^m and D_j^m are the layers with m neurons in the i -th layer in encoder and j -th layer in decoder, respectively. More specifically, this strategy could largely shorten the path to calculate the gradient, so as to avoid the inconvenience of gradient vanishment or explosion. In addition, it can also speed up convergence and quickly completes training on the iDAEN network. To retain the symmetry of the iDAEN network, the value of the neurons in the layers of encoder are added to the corresponding layer with the same neurons in the decoder network, and the operation is shown in the iDAEN part of Fig. 5, where the circles with the same color are the symmetric layers in the encoder and decoder. The skip connection is demonstrated as the lines with the arrow which has the same color as the corresponding layers.

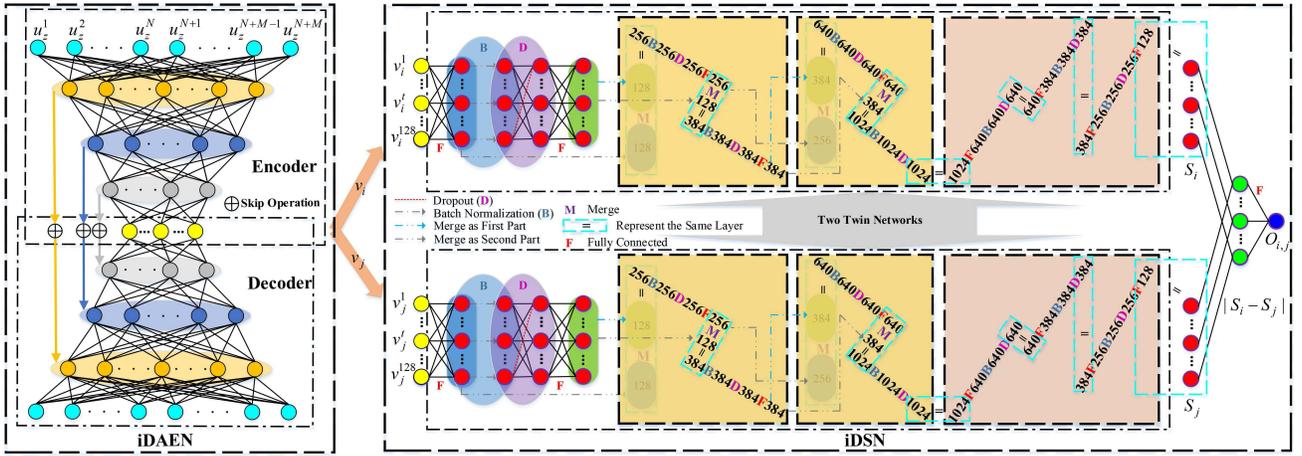


Fig. 5. The framework of TDFI, which consists of an improved deep auto-encoder network (iDAEN) and an improved deep siamese network (iDSN).

C. Friendship Learning

Compared to the EAM representing multi-source information, the encoder network \mathbb{E} of a well-trained iDAEN network can further compress the essence of multi-source information into one fused feature vector for each user. However, inferring friendship directly from certain common distance (e.g., euclidean distance) of these vectors is irrational. This is because the fused feature comes from multi-source information, which is highly abstract and difficult to be compared with each other directly. To cope with this issue, an improved deep siamese network is utilized as the second stage of TDFI to infer friendship, as illustrated in the iDSN part of Fig. 5.

The siamese network was originally proposed as an energy-based model [42], which was used to judge the similarity between pairwise samples. For example, Ahrabian and BabaAli [41] creatively established a siamese network to determine the probability of pairwise samples belonging to the same category, thereby realizing the verification of handwritten signatures. The characteristic of our iDSN serves as a twin-network that projects the pairwise input $\{v_i, v_j\}$ to the pairwise vector $\{S_i, S_j\}$, as shown in the iDSN part of Fig. 5. To be more specific, S_i and S_j represent the features of the input extracted by the twin networks. Particularly, the associated twin networks share the same weights, such that similar samples can be mapped onto the ambient feature space. Moreover, the $L1$ distance between S_i and S_j is evaluated, which is followed by a fully-connected layer with *Sigmoid* activation (i.e., $f(\cdot)$), formulated as Eq. (6).

$$O_{i,j} = f(W|S_i - S_j| + b) \quad (6)$$

where $f(\eta) = \frac{1}{1+e^{-\eta}}$. In addition, W and b represent the 128-dimension weight vector and the bias of the last fully connected layer, respectively.

Thus, the output of the iDSN network $O_{i,j}$ is the predicted label of the pairwise input $\{v_i, v_j\}$, which is in the range of *Sigmoid* function, i.e., $(0, 1)$. If the pairwise input has a friendship with each other, then $O_{i,j}$ should be close to 1 or close to 0 otherwise. That is to say, given a threshold, inferring friendship is equivalent to a binary classification problem.

For training the iDSN network, our goal is to calculate suitable parameters ψ to minimize the loss function, which is formulated as Eq. (7).

$$\mathcal{L}_S(\psi; \mathcal{P}) = -\frac{1}{P} \sum_{r=1}^P (y_r \log(\hat{y}_r) + (1 - y_r) \log(1 - \hat{y}_r)) \quad (7)$$

where ψ represents all the parameters of the iDSN network. $\mathcal{P} = \{\vec{p}_r\}_{r=1}^P$ represents the training set for the iDSN network, which contains pairwise users with or without friendship. $P = |\mathcal{P}|$ and $\vec{p}_r = (p_r^1, p_r^2)$ represents the pairwise users $\{u_i, u_j\}$, which can be converted into $\{v_i, v_j\}$ through the encoder network of iDAEN. \hat{y}_r represents the output of the iDSN network, which has the same meaning as $O_{i,j}$. And y_r is the true label of input (p_r^1, p_r^2) , i.e., if (p_r^1, p_r^2) is of friendship, then $y_r = 1$ or $y_r = 0$ otherwise.

To predict with high precision and low time overhead, we employ the feature-reuse method, which is similar to that in the first stage of TDFI. However, different from the iDAEN of TDFI, the skip connection here is carried out between layers L_i and L_{i+1} , i.e., the feature-reuse is achieved by the concatenation of the neurons in layers L_i and L_{i+1} . Specifically, the skip connection may increase the number of trainable parameters in the iDSN network, which may increase the time required to train the deep neural network. However, the skip connection can shorten the path of gradient flowed backward, which is useful for training. As a result, the iDSN network will benefit from the skip connection operation, and the training time will be reduced. And more details about our iDSN with skip connection is illustrated in Fig. 6(a). As shown in Fig. 5 and Fig. 6, the twin-network in iDSN consists of *common layers* and *neurons reduction*. In particular, the *common layers* can be divided into *F-M-B-D* as shown in Fig. 6(a), where *F* is a *fully connected* layer, *M* is a *merge* layer combining the output of *F* layer in L_i and L_{i+1} by concatenation, *B* is a *Batch Normalization* layer to avoid intern variance shift, and *D* is a *dropout* layer to avoid overfitting, which is indicated by the red dashed line in Fig. 6. Through the complementary advantages of *F-M-B-D*, the iDSN network designed for TDFI can show excellent performance in

TABLE III
STATISTICAL PROPERTIES OF THE THREE DATASETS

	$ \mathcal{U} (N)$	$ \mathcal{L} (M)$	$ \mathcal{F} (F)$	$ \mathcal{C} (C)$	$\bar{C} = C/N$	$\bar{F} = F/N$	$F/(N(N-1)/2)$
New York	44,371	25,868	193,995	1,843,187	41.5404	4.3721	0.000197075
Los Angeles	30,679	22,260	129,004	1,301,991	42.4392	4.2050	0.000274135
London	13,187	10,693	25,413	500,776	37.9750	1.9271	0.000292299

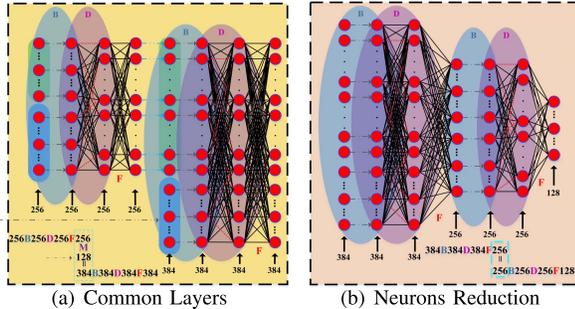


Fig. 6. The detailed structure of iDSN network.

friendship inference. After the *common layers*, we decrease the number of neurons by *neurons reduction*, to achieve a more efficient comparison of the similarity between different vectors. More specifically, *neurons reduction* is different from *common layers*. For example, it also contains *dropout layer*, but the dropout ratio are different from that of *common layers*. In addition, compared to *common layers*, it has no *merge layer* (*i.e.*, *F-B-D*). By reducing the number of neurons layer by layer, the last layer contains 128 neurons. That is to say, the dimensionality of feature (*i.e.*, S_i and S_j) obtained from iDSN is 128, as shown in Fig. 6(b).

IV. EXPERIMENTAL CONFIGURATION

This section introduces the relevant three real-world datasets for performance evaluation. Meanwhile, to promote fair comparisons, we perform analysis of baseline methods, which is also conducive to revealing the insight of our proposed TDFI. Finally, we provide the parameter configuration details of all methods.

A. Real-World Datasets

To match the characteristics of real-world mobile social networks, we conduct experiments on three real-world datasets collected from *Instagram* in 2016. These datasets are originally utilized for friendship attack [23], which is one of the baseline algorithms for comparison. More specifically, the data of the three datasets are from New York, Los Angeles, and London, respectively. That is, our TDFI will be evaluated on the data from three different cities, to fully verify the performance of the proposed method. It mainly contains friend information and location information (*i.e.*, check-in information). The ground truth, *i.e.*, the friendship is collected by the followers of the users via *Instagram*'s API. The statistical properties of the three datasets are introduced in Table III. Here, \bar{C} and \bar{F} represent the average number of check-ins and the average number

of friends for each user, respectively. The $N(N-1)/2$ in the last column is calculated as the number of user pairs. Therefore, the value in the last column indicates the sparsity of the social networks, which empirically validates that the EAM in Section III-A is sparse.

Although Backes *et al.* [23] mentioned that the three datasets had a lot more detailed check-in information, such as latitude, longitude, and the category of location, only part of the information in these datasets is publicly shared, *i.e.*, friendship information \mathcal{F} and check-in information \mathcal{C} . Despite the lack of detailed information, our proposed TDFI still has excellent performance with coarse-grained information. This is primarily for achieving privacy protection, which is a meaningful and popular research topic. For example, Shen *et al.* [43] proposed an encryption-based privacy protection method to avoid leakage of privacy when the graph (*e.g.*, social graph) is outsourced to the cloud computing paradigm. Unlike privacy protection from the outside, we utilize the coarse-grained information directly to achieve fundamental privacy protection. Moreover, from the coarse-grained information used by TDFI, it is impossible to distinguish a specific person, which is more conducive to protecting user privacy.

B. Baseline Algorithms

As far as the various baseline algorithms are concerned, the methods utilized for comparison in this paper are mainly divided into two categories. The first category is methods that only rely on location information. These methods take the user's mobility as the key feature, and then judge the probability of different users becoming friends by comparing the similarity of the mobility features. The second category is to use the interaction of structural information between users and other attribute information to infer friend relationships. Further detailed introduction is as follows.

1) Mobility-Based Link Prediction Methods:

- Exploiting Place Features in Link Prediction [27].
- Entropy-Based Model (EMB) [36].
- Personal, Global and Temporal (PGT) [44].
- **walk2friends** [23].

First of all, we employ walk2friends [23] as baseline algorithm (may be abbreviated as W2F). This is because walk2friends has high effectiveness and robustness. Specifically, in addition to walk2friends, there are another 14 baseline algorithms from [27], [36], [44]. In terms of effectiveness and robustness, walk2friends has consistently outperformed these 14 models by 13% to 20%. Note that the comparison between walk2friends and the other 14 baseline algorithms are also carried out on the three dataset adopted in Section V. As a result, walk2friends serves as the state-of-the-art method in friendship

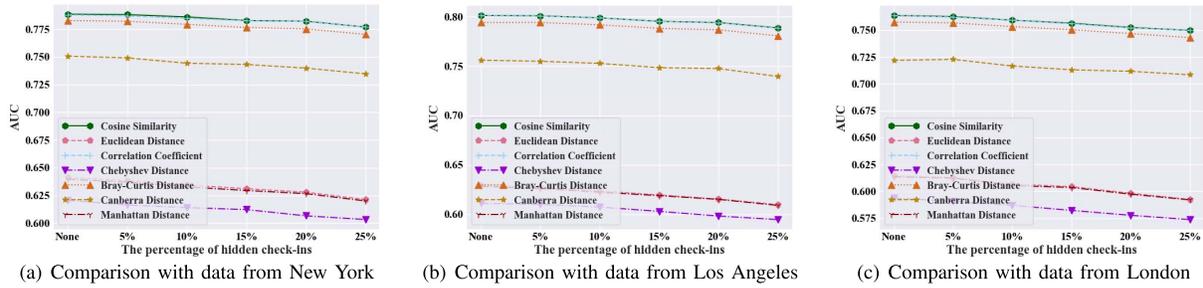


Fig. 7. Comparison of performance brought by 7 different pairwise similarity methods of walk2friends. The different proportions of location information hiding are used to assess robustness. Note that a higher AUC value means better performance.

inference on these three large-scale multi-source information datasets. The following comparisons, including effectiveness and robustness, are performed between walk2friends and the proposed TDFI.

As for the walk2friends method, it utilizes the random walk, which is often used in network embedding to obtain the random traces on the user-location bipartite graph. The random traces containing both users and locations can represent the mobility neighbors. Then the traces are fed to the skip-gram model with one hidden layer to be mapped to continuous vectors. Finally, the prediction of social link among users can be constructed according to the pairwise similarity, such as cosine similarity, euclidean distance, and so on.

Although walk2friends method is effective, it relies on the pairwise similarity methods. To ensure a fair comparison, we evaluate all pairwise similarity methods for walk2friends, which are utilized by Backes *et al.* [23]. More specifically, these pairwise similarity methods involve 7 common distance or similarity measures, including cosine similarity, euclidean distance, correlation coefficient, chebyshev distance, bray-curtis distance, canberra distance, and manhattan distance. As illustrated in Fig. 7, cosine similarity, correlation coefficient, and bray-curtis distance can achieve relatively better performance, which is consistent with the findings of Backes *et al.* [23]. Note that although the performance of cosine similarity and correlation coefficient are very similar, cosine similarity achieve 0.0699% improvement over correlation coefficient in New York (*i.e.*, Fig. 7(a)), 0.0202% in Los Angeles (*i.e.*, Fig. 7(b)), and 0.0313% in London (*i.e.*, Fig. 7(c)). This further demonstrates that walk2friends with cosine pairwise similarity can achieve the best performance, which is consistent with the optimal experimental configuration by Backes *et al.* [23]. Moreover, the performance of all pairwise similarity methods will decrease as the proportion of hidden location information increases. This indicates that walk2friends is excessively dependent on the integrity of location information, and thus the vulnerability increases. In Section V-B.2, we will further analyze this phenomenon and compare it with other relevant methods. Considering the fair comparison, the optimal cosine pairwise similarity is adopted by walk2friends in all experiments to compare with the newly proposed TDFI and other baseline methods.

2) *Graph-Based Methods With Single or Multiple Sources of Information:*

- Line [7].
- node2vec [8].

- Graph Convolutional Network (GCN) [24].
- Graph Auto-Encoder (GAE) [25].
- Variational Graph Auto-Encoder (VGAE) [25].

As we all know, the widely applied Convolutional Neural network (CNN) uses discrete convolution. In essence, the discrete convolution refers to the weighted summation. Taking image data as an example, CNN utilizes a filter with shared parameters to construct spatial features by calculating the weighted sum of the central pixel and adjacent pixels. However, CNN requires that the data to be processed conform euclidean structure. For non-euclidean structure data, it is difficult to apply CNN. In reality, there are a lot of non-euclidean structure data, such as social network topology. To efficiently extract the features of non-euclidean structure data, Kipf and Welling [24] propose the Graph Convolution Network (GCN). The original GCN algorithm [24] can operate directly on graphs (*i.e.*, non-euclidean structure data, such as the bipartite graph of friendship). In addition, GCN can simultaneously utilize the attribute information (*e.g.*, location information of each user) of each node in the graph. Overall, it is designed for semi-supervised learning in a transductive setting, and requires that the full graph Laplacian is know during training.

Based on the GCN, Kipf and Welling [25] further propose Graph Auto-Encoder (GAE) and Variational Graph Auto-Encoder (VGAE). In other words, both of them are the implementation of GCN on Auto-Encoder (AE) architecture. More specially, VGAE is implemented based on Variational Auto-Encoder (VAE) [39]. And both GAE and VGAE consist of a GCN encoder and a simple inner product decoder, introduced by Kipf and Welling [25]. Similar to GCN, GAE and VGAE can simultaneously exploit graph-structured data (*e.g.*, friendship information) and attribute data (*e.g.*, location information) of each graph node. It has been demonstrated that both GAE and VGAE can achieve competitive results on the link prediction task in citation networks [25], which is consistent with the friendship inference based on location and other information involved in this paper. Moreover, GAE and VGAE can naturally incorporate node features (*i.e.*, attribute information), which significantly outperform other graph-based link prediction-oriented methods (*e.g.*, Line [7] and node2vec [8]) on various benchmark datasets, demonstrated by Kipf and Welling [25]. In addition to walk2friends, therefore, we also employ GAE and VGAE as the baseline methods, to compare with the newly proposed TDFI.

As notified above, GCN-based methods (*i.e.*, GAE and VGAE) can utilize graph-structured information and attribute

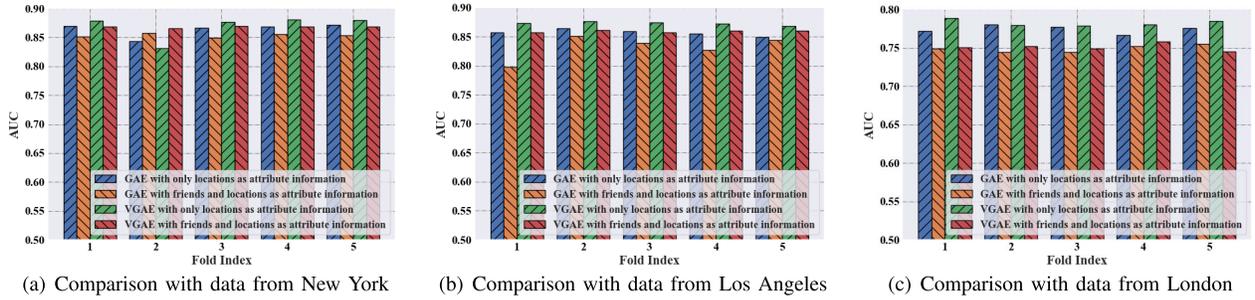


Fig. 8. How different types of attribute information (*i.e.*, *one type of attribute information* or *multiple types of attribute information*) affect GCN-based methods (*i.e.*, GAE and VGAE). Note that a higher AUC value means better performance.

information of each graph node. Specifically, the user’s location belongs to attribute information. When we look at all users globally, the buddy relationship can be regarded as graph-structured information, and thus datasets concerned in this paper can achieve the input requirements of GAE and VGAE. In fact, when we treat each user locally, his or her friend is actually a type of attribute information. The newly proposed iDAEN utilizes these two types of attribute information (*i.e.*, friend and location). To facilitate fair comparison, we first analyze which of *one type of attribute information* (*i.e.*, location) or *multiple types of attribute information* (*i.e.*, friend and location) is more effective for improving GCN-based methods. Note that even for *one type of attribute information*, GAE and VGAE also utilize multi-source information because it regard friend information as graph-structured information to optimize the corresponding network.

Fig. 8 shows the comparison results with different type of attribute information. It can be found that, except for the results of the 2nd fold experiment with New York dataset, in all other experiments, the GCN-based methods (*i.e.*, GAE and VGAE) using only location information as *one type of attribute information* are obviously superior to the methods using location and friend as *multiple types of attribute information*. Moreover, based on the average results of five cross-validation experiments, the performance of GAE with *one type of attribute information* has been improved by 1.24%, 3.03%, and 3.39% in New York, Los Angeles, and London, respectively. And the performance improvements of VGAE with *one type of attribute information* are 0.15%, 1.63%, and 4.19%, respectively. Therefore, in the following experimental comparison, we will adopt the GCN-based methods (*i.e.*, GAE and VGAE) with *one type of attribute information* to compare with our proposed method.

These phenomena indicate that GCN-based methods (*i.e.*, GAE and VGAE) are difficult to deal with the mutual interference of different types of attribute information. In other words, as the types of attribute information increases, the performance of GCN-based methods will gradually decline. Based on this perspective, we can leverage the degradation of GCN-based methods performance to evaluate the richness of multi-source information after the iDAEN processing. When we use the attribute information processed by iDAEN as the attribute information of GCN-based methods, the more performance degradation, it means that the iDAEN module is compatible with more different types of attribute information. More analysis will be

discussed in conjunction with the experimental results in Section V-B.

C. Parameter Setting

In terms of parameter setting, we fully follow the optimal parameter configuration of the comparison methods to promote fair comparison. First of all, we set the parameters for walk2friends in line with the default parameters introduced by Backes *et al.* [23], to gain the optimal performance. In particular, the walk length $l_w = 100$, walk times $t_w = 20$, and the dimensionality of feature vector $d_w = 128$. Similarly, for GAE and VGAE, we also follow the network structure and corresponding parameter settings introduced by Kipf and Welling [25], which is conducive to achieving the best performance of GCN-based methods (*i.e.*, GAE and VGAE).

As for the iDAEN network, the hyperparameter γ in Eq. (4) is set as $\gamma = 9$. In addition, since the range of the input in iDAEN is in $(0, 1)$, excluding the *Sigmoid* function in the last layer to adapt to the distribution of the input, other layers use *Relu* function rather than *Sigmoid* function to avoid gradient vanishment. The activation function here and skip connection in Section III-B are both designed to avoid the occurrence of gradient vanishment. The number of neurons of each layer in the encoder network for New York, Los Angeles and London are 70239 – 500 – 400 – 256 – 128, 52939 – 500 – 400 – 256 – 128, 23880 – 500 – 400 – 256 – 128, respectively. In addition, the structure of the decoder network is symmetrical with that of the encoder network.

Since the dimensionality of feature obtained from iDAEN is 128, the number of neurons per layer for three cities is the same in the iDSN network, *i.e.*, 128 – 256 – 384 – 640 – 1024 – 640 – 384 – 256 – 128. It can be found that for the datasets of three different cities, the network structures of iDAEN and iDSN are identical except that the number of iDAEN input neurons is different. The experimental results in Section V-B will show that TDFI has achieved excellent performance on all three datasets, which further demonstrates the universality of the proposed TDFI.

V. PERFORMANCE EVALUATION

Based on the experimental configuration in Section IV, we further evaluate the performance of TDFI in friendship inference via multi-source information. In this section, we first introduce the evaluation metrics. And then, via the comparison with the state-of-the-art methods, the effectiveness and

robustness are analyzed respectively to evaluate the performance of our TDFI.

A. Evaluation Metrics

To measure the performance of friendship inference, we utilize three common evaluation metrics for the comparison between the newly proposed TDFI and baseline algorithms. One of the evaluation metrics we adopt here is AUC (*i.e.*, the Area Under the Curve of ROC (Receiver Operating Characteristic) [45]). In our context, the AUC means the probability that an inference method accurately identify friends and non-friends. And the higher the AUC of the algorithm is, the better the performance is. In other words, an inference method with an AUC of 1 is a perfect friendship inference method on the given dataset, whereas an inference method with an AUC of 0.5 is randomly inferring friendship. Moreover, AUC is the same evaluation metrics as previous friendship inference algorithms (*i.e.*, walk2friends [23], GAE [25], and VGAE [25]), which is conducive to the fairness of comparison between TDFI and baseline algorithms.

In addition to AUC criterion, Equal Error Rate (EER) and Rate of Detection (RD) are also used for evaluation. For friendship inference, the pair of users with friendship is regarded as positive, otherwise, the case is negative. Thus the *FPR* (False Positive Rate), *TPR* (True Positive Rate) and *FNR* (False Negative Rate) are defined as Eq. (8), Eq. (9), and Eq. (10), respectively.

$$FPR = \frac{\# \text{of false-positive pairs}}{\# \text{of negative pairs}} \quad (8)$$

$$TPR = \frac{\# \text{of true-positive pairs}}{\# \text{of positive pairs}} \quad (9)$$

$$FNR = \frac{\# \text{of false-negative pairs}}{\# \text{of positive pairs}} \quad (10)$$

EER, defined as the *FPR* value of the point on the ROC curve when *FPR* equals to *FNR*. It is a measure which captures an algorithm's tradeoff between accuracy and recall, and one method with lower EER is evaluated to have better performance. Similarly, RD is defined as the *TPR* value of the point where *FPR* equals to *FNR*. This criterion is expected to be higher to have better performance. Considering the application of friendship inference, a method with lower EER and higher RD will infer friendship more precisely, which can promote a better user experience.

We evaluate the above criteria on the methods in two aspects: each fold cross-validation and the mean value of all five folds of cross-validation data. In the following contents, we will introduce relevant results in detail.

B. Experimental Results

To fully verify the effectiveness of the newly proposed TDFI, 5-fold cross-validation is used in our experiments. Equipped with the evaluation metrics mentioned above, we evaluate the proposed TDFI from the perspectives of the effectiveness and the robustness via comparing to various baseline algorithms, introduced in Section IV-B.

Algorithm 1 Inferring Friendship via TDFI

Input: \mathcal{Q} , \mathcal{P} and \mathcal{C}

Output: 0 or 1

- 1 Initialize the EAM A according to \mathcal{P} and \mathcal{C}
 - 2 Calculate \tilde{A} with local and global normalizations
 - 3 **repeat**
 - 4 Train the iDAEN network through $\{\tilde{a}_i\}_{i=1}^N$
 - 5 Minimize $\mathcal{L}_A(\Theta)$ in Eq. (4)
 - 6 **until convergence**
 - 7 Compute the fused feature vector v_i for each user u_i
 - 8 **repeat**
 - 9 Train the iDSN network through \mathcal{P}
 - 10 Minimize $\mathcal{L}_S(\psi; \mathcal{P})$ in Eq. (7)
 - 11 **until convergence**
 - 12 Test \mathcal{Q}/\mathcal{P} by the well-trained iDSN network
-

As notified in Section IV-B.2 and Fig. 8, we can leverage the degradation of GCN-based methods performance to evaluate the richness of multi-source information after iDAEN processing. In addition to TDFI and the referred baseline algorithms, therefore, we also evaluate the performance of the combination of GCN and iDAEN, denoted by GCN-iDAEN. More specifically, we utilize GAE to implement the GCN framework, and use the fusion features processed by iDAEN as the attribute information of GAE. Compared with the GAE, if the performance of GCN-iDAEN is significantly degraded, it means that iDAEN can save more valuable data of different types of attribute information. In other words, iDAEN is compatible with more different types of attribute information.

1) *Comparison of Effectiveness:* Due to the sparsity of the social networks indicated in the last column of Table III, the number of user pairs with friendship is far less than the number of user pairs without friendship such that the high imbalance of the labels is brought about. Hence, we utilize the same down-sampling strategy, introduced by Backes *et al.* [23]. To ensure a fair comparison, we randomly sample the same number of pairs without friendship and integrate them with the pairs with friendship as a set, namely \mathcal{Q} . Then \mathcal{Q} is divided into 5 parts, *i.e.*, $\mathcal{Q} = \{\mathcal{Q}_h\}_{h=1}^5$, each of which contains the same number of pairs with or without friendship. Consequently, we select 4 parts of \mathcal{Q} as training set and the remaining one part as testing set. Specifically, regarding the i -th cross-validation, we select 4 parts of \mathcal{Q} as training set $\mathcal{P} = \{\mathcal{Q}_h\}_{h=1, h \neq i}^5$, and the remaining part \mathcal{Q}/\mathcal{P} as the testing set. That is, we will evaluate each dataset 5 times. The implementation details of the newly proposed framework are shown in Algorithm 1. Note that the friendships for testing are hidden in the training of both the iDAEN network and the iDSN network, as illustrated in Fig. 2.

All AUC results of 5-fold cross-validation on three datasets are demonstrated in Table IV, Table V, and Table VI. The samples of cross-validation for TDFI, baseline algorithms (*i.e.*, walk2friends, GAE, and VGAE), and GCN-iDAEN are equally the same. As notified in Section IV-B.1, all results of walk2friends are selected with optimal performance among the 7 pairwise similarity methods, *i.e.*, cosine similarity. GAE

TABLE IV

EFFECTIVENESS COMPARISON ABOUT AUC DETAILS UPON NEW YORK DATASET, WHERE WALK2FRIENDS ADOPTS THE COSINE SIMILARITY

Fold Index	TDFI	walk2friends	GAE	VGAE	GCN-iDAEN
1st	0.87757174	0.78903732	0.87010984	0.87840411	0.82619540
2nd	0.86211580	0.79528344	0.84328034	0.83160593	0.81209048
3rd	0.87703591	0.78430959	0.86638506	0.87667507	0.81183298
4th	0.87720409	0.78522337	0.86891467	0.88050297	0.80022821
5th	0.86664682	0.78911064	0.87131375	0.88012855	0.80199179
Mean	0.87211487	0.78859287	0.86400073	0.86946333	0.81046777
std. dev.	0.00647727	0.00387116	0.01048778	0.01897762	0.00925793

TABLE V

EFFECTIVENESS COMPARISON ABOUT AUC DETAILS UPON LOS ANGELES DATASET, WHERE WALK2FRIENDS ADOPTS THE COSINE SIMILARITY

Fold Index	TDFI	walk2friends	GAE	VGAE	GCN-iDAEN
1st	0.85210393	0.80303692	0.85779028	0.87381104	0.80002399
2nd	0.84158310	0.79472212	0.86419931	0.87656799	0.77788659
3rd	0.85824482	0.81175503	0.85961440	0.87495446	0.79469024
4th	0.85924307	0.79753727	0.85506418	0.87277607	0.81034464
5th	0.85646976	0.79917494	0.84969943	0.86901061	0.79589589
Mean	0.85352894	0.80124526	0.85727352	0.87342404	0.79576827
std. dev.	0.00645404	0.00590343	0.00481583	0.00254065	0.01050252

TABLE VI

EFFECTIVENESS COMPARISON ABOUT AUC DETAILS UPON LONDON DATASET, WHERE WALK2FRIENDS ADOPTS THE COSINE SIMILARITY

Fold Index	TDFI	walk2friends	GAE	VGAE	GCN-iDAEN
1st	0.79928732	0.76259451	0.77209545	0.78876053	0.72520433
2nd	0.81603062	0.76147544	0.78027583	0.77998905	0.73820485
3rd	0.80829673	0.76908639	0.77772562	0.77919809	0.74235968
4th	0.81382569	0.76201198	0.76677061	0.78006913	0.74273295
5th	0.81648814	0.76305543	0.77595216	0.78464942	0.73187834
Mean	0.81078570	0.76364475	0.77456393	0.78253324	0.73607603
std. dev.	0.00644478	0.00277249	0.00471971	0.00365856	0.00669545

and VGAE also achieve the optimal implementation, which has been discussed in Section IV-B.2. The ROC curves of all relevant methods on New York, Los Angeles, and London are illustrated in Fig. 9. Note that the mean value of AUC in Table IV, Table V, and Table VI may be different from that in Fig. 9 (*i.e.*, Fig. 9(a), Fig. 9(b), and Fig. 9(c)), because the value of each AUC in the referred tables is calculated separately and then averaged, while the area in Fig. 9 is calculated by taking all the five folds of cross-validation data and labels together. Moreover, in addition to being intuitive, the mean ROC curves can mitigate the impact of some accidental events, such that the derived conclusion is more reliable and in line with reality.

As illustrated in Fig. 9, no matter which dataset is used for evaluation, all mean ROC curves of TDFI, GAE, and VGAE, are consistently above the curves of walk2friends. This demonstrates that the proposed TDFI outperforms walk2friends, no matter what pairwise similarity methods walk2friends uses. More specifically, TDFI gain 10.59% improvement over walk2friends in New York, 6.53% in Los Angeles, and 6.18% in London. Since walk2friends has gain 13% to 20% improvement over other 14 mobility-based baseline methods [23], the proposed TDFI pushes a great improvement over these baseline approaches. Since TDFI, GAE, and VGAE all simultaneously exploit multi-source information, it also can be demonstrated from Fig. 9 that the comprehensive utilization

of multi-source information can indeed bring significant performance improvements.

Among these methods (*i.e.*, TDFI, GAE and VGAE) that all utilize multi-source information, in terms of mean ROC (equivalent to the average AUC), except for the Los Angeles dataset, our proposed TDFI is superior to GAE and VGAE. Specifically, we gain 0.94% improvement over GAE in New York, and 4.68% in London. TDFI also achieves 0.30% improvement over VGAE in New York, and 3.61% in London. As far as Los Angeles is concerned, the difference in performance is no more than 0.44% with GAE and 2.28% with VGAE. Overall, in terms of global average performance, our TDFI achieves a 7.77% performance improvement over walk2friends, 1.63% over GAE, and 0.44% over VGAE, respectively. This means that the newly proposed TDFI outperforms other methods that utilize multi-source information in terms of effectiveness.

Moreover, Fig. 9(a), Fig. 9(b) and Fig. 9(c) all show that the combination of the GCN framework and iDAEN (*i.e.*, GCN-iDAEN) will cause a significant degradation in effectiveness. This further demonstrates that the GCN-based methods cannot cope with the interference between multi-source attribute information, which has been initially discussed in Fig. 8. This phenomenon also demonstrates that the information processed by iDAEN still maintains the effective data of multi-source attribute information. That is, iDAEN has

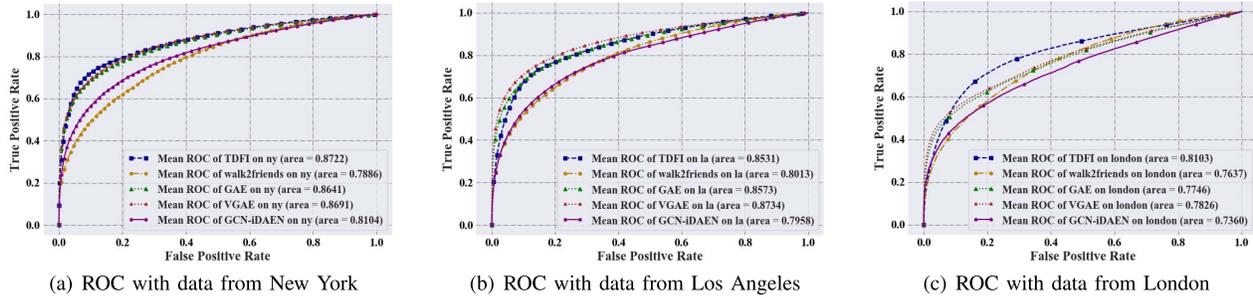


Fig. 9. Comparison of overall performance in terms of Mean ROC between different methods.

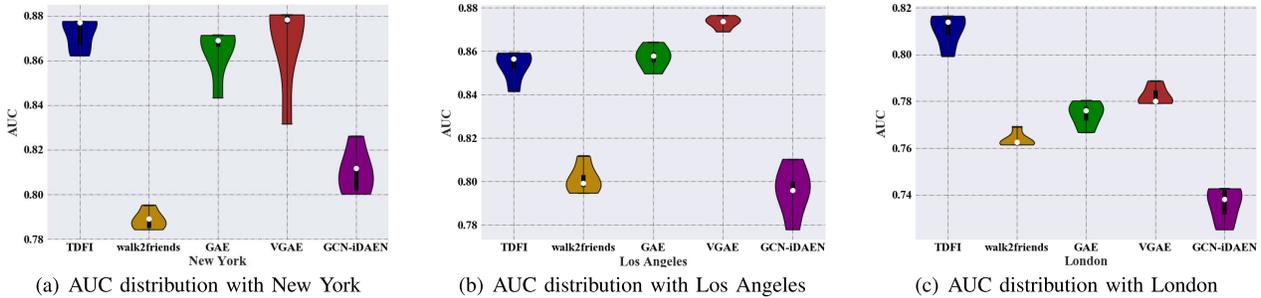


Fig. 10. Comparison of effectiveness and stability between different methods via statistical AUC distribution.

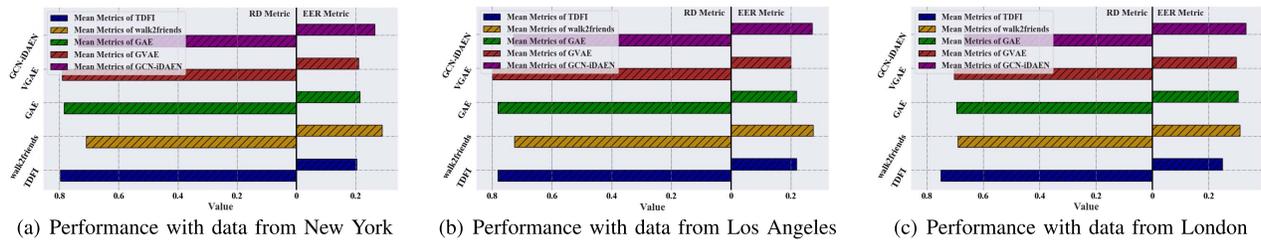


Fig. 11. EER and RD comparison.

excellent compatibility. Therefore, our TDFI can comprehensively utilize multi-source attribute information, achieving superior performance.

In addition to the mean values of AUC (*i.e.*, the area below mean ROC curves in Fig. 9) for evaluating effectiveness, we also draw the distribution of AUC in each fold cross-validation. Specifically, Table IV, Table V, and Table VI have been designed to check the specific AUC values in each fold cross-validation, while Fig. 10 is applied to further perform the comparison in statistical distribution. In Fig. 10, each violin diagram shows the distribution of the experimental results of a specific method on a specific dataset. The white circle is the median value, whereas the thick black line within the violin indicates interquartile range. The shape of the violin shows the distribution of the AUC values. First of all, the results illustrated in Fig. 10 are consistent with the conclusions involved in Fig. 9. Moreover, it is obvious that the results of TDFI mainly converge on the higher AUC value, indicating that most of the testing results are stable at a high level, whereas the results of walk2friends mainly converge on the lower AUC value, indicating that the overall effect is relatively poor. Compared with GAE and VGAE, TDFI also achieved better performance

on three different datasets. In addition, all GCN-iDAEN results are clearly distributed at a lower level. This again demonstrates that, although GCN-based methods can utilize structural information and attribute information simultaneously, it cannot handle the interference between attribute information from different sources. On the contrary, while iDAEN maintains the diversity of attribute information, the iDSN module of TDFI can efficiently exploit rich multi-source information to obtain more outstanding performance.

We also record the results of EER and RD criteria, which are illustrated in Fig. 11. We can find that regardless of whether the dataset is from New York, Los Angeles or London, TDFI always has a lower EER and a higher RD compared with walk2friends. Specifically, for the three datasets from New York, Los Angeles, and London, EERs are reduced by 29.61%, 22.06%, and 18.67%, respectively. And RDs increase by 12.50%, 7.91%, and 8.27%, respectively. Regarding comparison with GAE and VGAE, our TDFI also achieves similar performance to AUC in the two metrics of EER and RD. Overall, in terms of global average EER, our TDFI achieves 23.45%, 7.87%, and 3.23% reduction compared to walk2friends, GAE, and VGAE, respectively. Meanwhile,

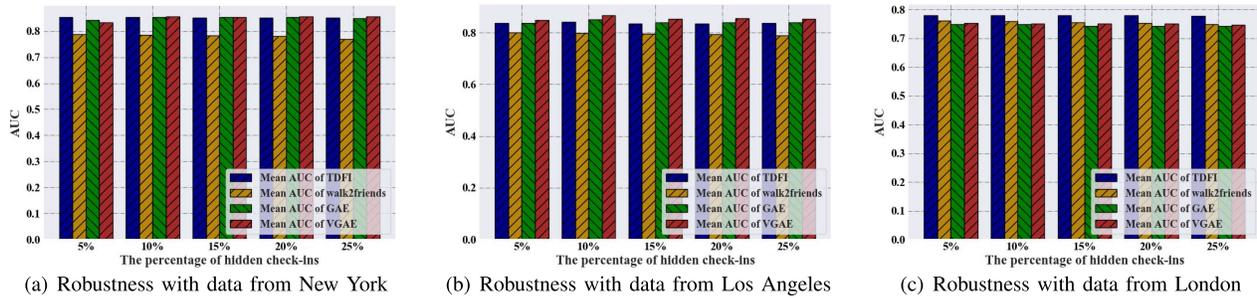


Fig. 12. Comparison of mean AUC for evaluating the robustness.

in terms of global average RD with different datasets, TDFI achieves 9.56% improvement over walk2friends, 3.16% over GAE, and 1.76% over VGAE. As notified in Section V-A, the relatively small EER and relatively large RD mean that TDFI can achieve better performance in terms of effectiveness. That is, our proposed TDFI is more suitable for the deployment of friend-recommendation systems in real-world applications.

2) *Comparison of Robustness*: As noted in the last row of Table IV, Table V and Table VI, it can be found that the standard deviation is very small, indicating that our method is relatively stable. In addition, TDFI and the referred baseline algorithms (*i.e.*, walk2friends, GAE and VGAE) all utilize the check-in information for feature extraction. To investigate the robustness of the friendship inference models with respect to the check-in information, we randomly discard the 5%, 10%, 15%, 20%, and 25% of the number of check-in information and evaluate the results. Except for the check-in information, all the other information (*e.g.*, friend and non-friend information) is fixed as previously mentioned with the same default parameters. Accordingly, the experimental results on robustness comparison of the three different datasets, *i.e.*, New York, Los Angeles and London, are summarized as the histograms in Fig. 12(a), Fig. 12(b) and Fig. 12(c), respectively.

First of all, we can find from Fig. 12 that for each method (*i.e.*, TDFI, walk2friends, GAE, and VGAE), as check-ins decrease from 5% to 25%, the height of these bars has changed slightly, which shows that all methods are relatively stable. This is consistent with the phenomenon reflected in the last row of Table IV, Table V, and Table VI. From the perspective of different methods, regardless of the percentage of hidden check-ins, the methods (*i.e.*, TDFI, GAE, and VGAE) that can exploit multi-source information, always achieve superior performance than walk2friends that only utilizes location information. This further demonstrates that the simultaneous utilization of multi-source information not only improves performance, but also offsets the negative effects of partial information loss. In other words, the comprehensive utilization of multi-source information is of great significance for friendship inference.

In terms of qualitative analysis, the newly proposed TDFI is always more prominent than walk2friends. Moreover, as the percentage of hidden check-ins increases, the performance of TDFI may occasionally increase, rather than consistently decline. This is because information from different sources will inevitably interfere with each other, and sometimes hiding

part of the redundant information may further reduce interference. This phenomenon is also reflected in the results of GAE and VGAE. In addition, the performance improvement effect brought by multi-source information complementation is significantly higher than the interference effect. On the contrary, the AUC results of walk2friends reduce as check-ins decrease, indicating that walk2friends has a higher dependence on the check-in information. This phenomenon has also been illustrated in Fig. 7. Moreover, almost all the experiments involved in Fig. 7 clearly show that no matter which distance or similarity measures are adopted, the performance of walk2friends will decrease as the percentage of hidden check-in increases. Note that in real-world social networking scenarios, due to the high correlation between location and privacy [16], many people choose to hide their location information to protect privacy [17]. In terms of quantitative analysis, in all experiments on the comparative evaluation of robustness, the newly proposed TDFI achieves 5.804%, 1.260%, and 0.483% performance improvement, over walk2friends, GAE, and VGAE, respectively. Combining qualitative and quantitative analysis, it can be demonstrated that our TDFI still achieves the best performance in terms of robustness.

With numerous comparative experiments, we also demonstrated the DL-based TDFI is more suitable for deployment in the real-world MSN applications. As notified in Section IV-B, among the 7 common distance or similarity methods, cosine similarity enables walk2friends to achieve the optimal average performance. Note that in Fig. 7, the drawing data used is the average of 5-fold cross-validation. As far as the *comparison of effectiveness* in Section V-B.1, walk2friends with cosine similarity has always achieved the best performance. However, in terms of walk2friends, within all 90 (*i.e.*, $6 \times 3 \times 5$) folds of experiments on the three datasets, we found some special cases, where the cosine similarity degenerates from the optimal option to the sub-optimal option.

As illustrated in Fig. 13, in the 3rd cross-validation on New York with hiding 20% check-in information, the result of walk2friends with correlation coefficient is better than the result of walk2friends with cosine similarity, and we regard it as one special case. Note that the referred special case is about the result of a specific experiment, whereas the values in Fig. 7 are the average results of 5-fold cross-validation experiments. In addition, in our 90 folds of experiments on the three datasets, similar special cases occurred 16 times. That is to say, the performance of walk2friends method relies on the

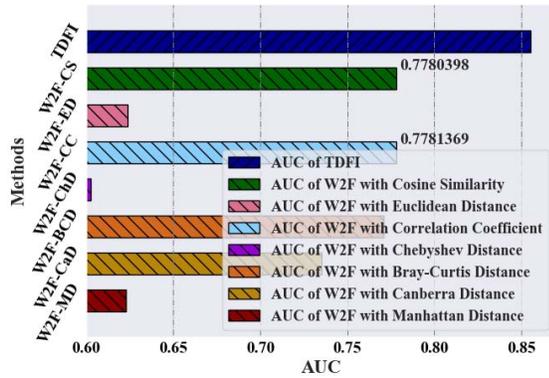


Fig. 13. One special case with robustness comparison between TDFI and walk2friend (*i.e.*, W2F).

pairwise similarity methods. And there is no single pairwise similarity method that works best in all scenarios. In contrast, the friendship inference of our proposed TDFI depends on the iDSN network, whose structure remains unchanged in all scenarios (*i.e.*, different cities and different percentage of hidden check-ins). In other words, our TDFI does not require manual operations to select similarity methods, which can facilitate the deployment of friend-recommendation systems in real-world MSN applications. In addition, even if walk2friends with correlation coefficient has achieved the best performance in these special cases, our proposed TDFI still outperforms the best result of walk2friends (*i.e.*, walk2friends with correlation coefficient), which further demonstrates the effectiveness and robustness of the newly proposed TDFI.

VI. CONCLUSION

Motivated by the diversification of social forms and various data with mobile social networks, we proposed, implemented, and evaluated TDFI, a novel two-stage deep learning framework for friendship inference. Our TDFI enables MSNs to smartly exploit multi-source user-related data simultaneously, rather than hierarchically. In terms of details, we first adopted an extended adjacency matrix with both local and global normalizations for absorbing different information. This matrix then serves as an input to the iDAEN network to extract fused feature with low dimensionality. After that, the iDSN network is utilized to determine whether the pair of users has friendship by measuring the similarity of the fused feature. We conducted extensive experiments on three real-world datasets to evaluate the performance of TDFI and baseline methods. The trace-driven evaluation results demonstrated that TDFI can complement the advantages of different information, avoiding the performance issues caused by insufficient single-source information. In addition to being compatible with structural information and attribute information simultaneously, our TDFI can also exploit different attribute information. Overall, our qualitative and quantitative evaluations indicated that the newly proposed TDFI outperforms the existing recommendation systems with improved accuracy and robustness. Regarding future work, an interesting open issue is whether our framework can be extended to the cooperation of different mobile social networks.

REFERENCES

- [1] Y. Zhao *et al.*, “TDFI: Two-stage deep learning framework for friendship inference via multi-source information,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1981–1989.
- [2] X. Chen, B. Proulx, X. Gong, and J. Zhang, “Exploiting social ties for cooperative D2D communications: A mobile social networking case,” *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1471–1484, Sep. 2015.
- [3] Z. Su, Q. Xu, J. Luo, H. Pu, Y. Peng, and R. Lu, “A secure content caching scheme for disaster backup in fog computing enabled mobile social networks,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4579–4589, Oct. 2018.
- [4] (2020). Statista. *Global Digital Population as of April 2020*. [Online]. Available: <https://www.statista.com/statistics/617136/digital-population-worldwide/>
- [5] (2020). Statista. *Number of Daily Active Instagram Stories Users From October 2016 to January 2019*. [Online]. Available: <https://www.statista.com/statistics/730315/instagram-stories-dau/>
- [6] S. Huang, J. Zhang, D. Schonfeld, L. Wang, and X.-S. Hua, “Two-stage friend recommendation based on network alignment and series expansion of probabilistic topic model,” *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1314–1326, Jun. 2017.
- [7] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proc. ACM WWW*, vol. 2015, pp. 1067–1077.
- [8] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.
- [9] C. Tu, Z. Zhang, Z. Liu, and M. Sun, “TransNet: Translation-based network representation learning for social relation extraction,” in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2864–2870.
- [10] Y. Gu, Y. Sun, Y. Li, and Y. Yang, “RaRE: Social rank regulated large-scale network embedding,” in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 359–368.
- [11] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.
- [12] D. Wang, P. Cui, and W. Zhu, “Structural deep network embedding,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1225–1234.
- [13] M. Danisch, O. Balalau, and M. Sozio, “Listing K-cliques in sparse real-world graphs,” in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 589–598.
- [14] P. Mittal, C. Papamanthou, and D. Song, “Preserving link privacy in social network based systems,” in *Proc. NDSS*, 2013, pp. 1–19.
- [15] P. Zhao *et al.*, “Synthesizing privacy preserving traces: Enhancing plausibility with social networks,” *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2391–2404, Dec. 2019.
- [16] S. Boukoros, M. Humbert, S. Katzenbeisser, and C. Troncoso, “On (the lack of) location privacy in crowdsourcing applications,” in *Proc. USENIX Security*, 2019, pp. 1859–1876.
- [17] R. Dey, Z. Jelveh, and K. Ross, “Facebook users have become much more private: A large-scale study,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2012, pp. 346–352.
- [18] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, “Friendbook: A semantic-based friend recommendation system for social networks,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 538–551, Mar. 2015.
- [19] M. Tomlinson, “Lifestyle and social class,” *Eur. Sociol. Rev.*, vol. 19, no. 1, pp. 97–111, 2003.
- [20] W. Alex, “Why is it hard to make friends over 30?” *New York Times*, Tech. Rep., 2012. [Online]. Available: <https://www.nytimes.com/2012/07/15/fashion/the-challenge-of-making-friends-as-an-adult.html>
- [21] N. Eagle, A. S. Pentland, and D. Lazer, “Inferring friendship network structure by using mobile phone data,” *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 36, pp. 15274–15278, 2009.
- [22] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, “Social friend recommendation based on multiple network correlation,” *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 287–299, Feb. 2016.
- [23] M. Backes, M. Humbert, J. Pang, and Y. Zhang, “walk2friends: Inferring social links from mobility profiles,” in *Proc. ACM CCS*, 2017, pp. 1943–1957.
- [24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. ICLR*, 2016, pp. 1–14.
- [25] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” in *Proc. NIPS Workshop Bayesian Deep Learn.*, 2016, pp. 1–3.
- [26] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “User profile preserving social network embedding,” in *Proc. IJCAI*, 2017, pp. 3378–3384.

- [27] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 1046–1054.
- [28] D. J. Hruschka and J. Henrich, "Friendship, cliquishness, and the emergence of cooperation," *J. Theor. Biol.*, vol. 239, no. 1, pp. 1–15, Mar. 2006.
- [29] J. Liu, Q. Lian, L. Fu, and X. Wang, "Who to connect to? Joint recommendations in cross-layer social networks," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1295–1303.
- [30] M. Yoon, W. Jin, and U. Kang, "Fast and accurate random walk with restart on dynamic graphs with guarantees," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 409–418.
- [31] F. Xu, J. Lian, Z. Han, Y. Li, Y. Xu, and X. Xie, "Relation-aware graph convolutional networks for agent-initiated social E-commerce recommendation," in *Proc. ACM CIKM*, 2019, pp. 529–538.
- [32] C. Zhang, X. Wu, W. Yan, L. Wang, and L. Zhang, "Attribute-aware graph recurrent networks for scholarly friend recommendation based on internet of scholars in scholarly big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2707–2715, Apr. 2020.
- [33] J. Chang and E. Sun, "Location3: How users share and respond to location-based data on social," in *Proc. Int. AAAI Conf. Web Social Media*, 2011, pp. 74–80.
- [34] A. Likhyani, S. Bedathur, and P. Deepak, "LoCaTe: Influence quantification for location promotion in location-based social networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2259–2265.
- [35] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. ACM KDD*, 2011, pp. 1082–1090.
- [36] H. Pham, C. Shahabi, and Y. Liu, "EBM: An entropy-based model to infer social strength from spatiotemporal data," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 265–276.
- [37] F. Zhou, B. Wu, Y. Yang, G. Trajcevski, K. Zhang, and T. Zhong, "vec2Link: Unifying heterogeneous data for social link prediction," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1843–1846.
- [38] A.-M. Olteanu, K. Huguenin, R. Shokri, M. Humbert, and J.-P. Hubaux, "Quantifying interdependent privacy risks with location data," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 829–842, Mar. 2017.
- [39] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014, pp. 1–14.
- [40] D. Charte, F. Charte, S. García, M. J. del Jesus, and F. Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines," *Inf. Fusion*, vol. 44, pp. 78–96, Nov. 2017.
- [41] K. Ahrabian and B. BabaAli, "Usage of autoencoders and Siamese networks for online handwritten signature verification," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 9321–9334, Dec. 2019.
- [42] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 1735–1742.
- [43] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 940–953, Apr. 2018.
- [44] H. Wang, Z. Li, and W.-C. Lee, "PGT: Measuring mobility relationship using personal, global and temporal factors," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 570–579.
- [45] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.



and game theory. He is a member of ACM. He is a recipient of the Shuimu Tsinghua Scholar Program.

Yi Zhao (Member, IEEE) received the B.Eng. degree from the School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an, China, in 2016, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2021. He is currently an Assistant Researcher and a Post-Doctoral Fellow at the Department of Computer Science and Technology, Tsinghua University. His research interests include network economics, network security, machine learning, social network,



Meina Qiao received the M.D. degree from the School of Automation Science and Control Engineering, Beihang University, in 2019. She is currently a Senior Researcher with the Department of Computer Vision Technology (VIS), Baidu Inc. Her research interests include computer vision, machine learning, and pattern recognition.



Haiyang Wang (Member, IEEE) received the Ph.D. degree in computing science from Simon Fraser University, Burnaby, BC, Canada, in 2013. He is currently an Associate Professor with the Department of Computer Science, University of Minnesota Duluth, Duluth, MN, USA. His research interests include cloud computing, peer-to-peer networking, social networking, big data, and multimedia communications.



Rui Zhang (Member, IEEE) received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2018. He is currently an Associate Professor with the School of Artificial Intelligence, OPTics and ElectroNics (iOPEN), Northwestern Polytechnical University.



Dan Wang (Senior Member, IEEE) received the B.Sc. degree in computer science from Peking University, Beijing, China, the M.Sc. degree in computer science from Case Western Reserve University, Cleveland, OH, USA, and the Ph.D. degree in computer science from Simon Fraser University, Vancouver, BC, Canada. He is currently a Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include Internet architecture and QoS, smart buildings, and green computing. He is a Steering Committee Member of ACM e-Energy. He won the Best Paper Award from ACM e-Energy 2018 and ACM Buildsys 2018. He won the Global Innovation Award from TechConnect, in 2017. He served as the TPC Co-Chair for IEEE/ACM IWQoS 2020. He has served as the TPC Co-Chair for the ACM e-Energy 2020. He is the Steering Committee Chair of IEEE/ACM IWQoS.



Ke Xu (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He currently serving as a Full Professor with the Department of Computer Science and Technology, Tsinghua University. He has published more than 200 technical articles and holds 11 U.S. patents in the research areas of next-generation Internet, blockchain systems, the Internet of Things, and network security. He is a member of ACM. He was the Steering Committee Chair of IEEE/ACM IWQoS. He has guest-edited several special issues in IEEE and Springer journals. He is an Editor of IEEE INTERNET OF THINGS JOURNAL.