TDFI: Two-stage Deep Learning Framework for Friendship Inference via Multi-source Information

Yi Zhao[1,2], Meina Qiao[3], Haiyang Wang[4], Rui Zhang[5], Dan Wang[6], Ke Xu[1,2,*], and Qi Tan[1,2]

[1]Department of Computer Science and Technology, Tsinghua University, P.R. China
[2]Beijing National Research Center for Information Science and Technology (BNRist), P.R. China
[3]School of Automation Science and Electrical Engineering, Beihang University, P.R. China
[4]Department of Computer Science, University of Minnesota at Duluth, USA
[5]School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University, P.R. China
[6]Department of Computing, The Hong Kong Polytechnic University, Hong Kong
Email: zhaoyi16@mails.tsinghua.edu.cn, meinaqiao@buaa.edu.cn, haiyang@d.umn.edu
ruizhang8633@gmail.com, csdwang@comp.polyu.edu.hk, xuke@tsinghua.edu.cn, tanq16@mails.tsinghua.edu.cn

***Abstract*—Due to the explosive growth of social network services, friendship inference has been widely adopted by Online Social Service Providers (OSSPs) for friend recommendation. The conventional techniques, however, have limitations in accuracy or scalability to handle such a large yet sparse multi-source data. For example, the OSSPs will be required to manually give the order in which the various information is applied. This unavoidably reduces the applicability of existing friend recommendation systems. To address this issue, we propose a Two-stage Deep learning framework for Friendship Inference (TDFI). This approach can utilize multi-source information simultaneously with low complexity. In particular, we apply an Extended Adjacency Matrix (EAM) to represent the multi-source information. We then adopt an improved Deep AutoEncoder Network (iDAEN) to extract the fused feature vector for each user. The TDFI framework also provides an improved Deep Siamese Network (iDSN) to measure user similarity from iDAEN. Finally, we evaluate the effectiveness and robustness of TDFI on three large-scale real-world datasets. It shows that TDFI can effectively handle the sparse multi-source data while providing better accuracy for friend recommendation.**

I. Introduction

The Internet has profoundly changed the human experience in the past decades. People not only use the Internet to find information, watch movies, buy and sell products but also highly rely on it to connect with others. In particular, the number of monthly active Instagram users is now up to a staggering increase from 800 millions in September 2017 to $1,000$ millions in June 2018 [1]. Given the huge number of these OSN (Online Social Network) users, it is impossible for Online Social Service Providers (OSSPs) to check the information of each user and quickly pinpoint their potential friends. Consequently, friend-recommendation service has been widely adopted by most OSSPs like Facebook, Instagram, and Twitter [2].

Accurately learning the potential friend relationship is an important yet challenging job. Conventional friendship-inference methods mainly depend on mutual friends, which provides long potential friend lists with very low precision [2]. In particular, friend-recommendation based on social graph representation [3]–[5] is largely exploited. However, real-world social networks are much more sparse than expected [6]–[8] (i.e., the number of true friends is much smaller than that of non-friends, as illustrated in Table I), which poses a challenge to existing approaches. Moreover, friend information is highly privacy-sensitive and deeply connected with our social identity [9]. 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 [10]. Worse more, these approaches cannot fully reflect the real preferences on friend selection [11]. This is due to the fact regarding missing of important real-world information such as users' different lifestyles [12], interests [13] and locations [14].

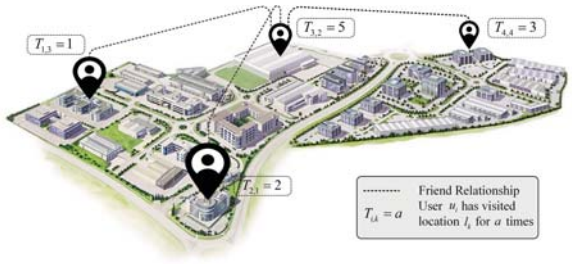


Fig. 1. An overview of the two types of information.

Since people live in a multi-source environment, the deep utilization of multi-source information, which can more truly reflect the behavior of users making friends, has attracted the attention of the academic community [2], [15], [16]. Moreover, through the complementary advantages, the utilization of multi-source information can also alleviate the impact of sparse problem in real-world social networks. And when some information is insufficient, other information can

*Ke Xu is the corresponding author.

still perform the role of friendship inference. Since the use of multi-source information increases complexity, the current mainstream approach is utilizing multi-source information hierarchically. Unfortunately, it requires OSSPs to manually give the importance of different information as well as the order in which the various information is applied. However, in real-world 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.

In this paper, we propose a novel Two-stage Deep learning framework for Friendship Inference (namely TDFI). This approach can utilize multi-source information simultaneously with low complexity. Via treating multi-source information as a whole input, this approach can smartly process the multi-source information for friendship inference. Therefore, there is no need to manually set which information is more important and the order in which the various information is applied. To ensure scalability, we apply an Extended Adjacency Matrix (EAM) to better represent the multi-source information. After that, an improved Deep AutoEncoder 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 user similarity by measuring the similarity of the fused feature obtained by the iDAEN network.

In this study, we adopt friend relation and location as an example of the multi-source information, as illustrated in Fig. 1. This is because most OSSPs have built-in location sharing services (e.g., check-in service) in their mobile applications. In particular, Facebook, Instagram, and Twitter all allow their users to add location information to their shared photo or message. On the other hand, location information can further reflect the user's behaviors, which is very representative. Eagle *et al.* [14] confirmed that data such as location information obtained through mobile devices has extraordinary potential in social network analysis. And Scellato *et al.* [17] also indicated that synchronous check-ins information among users can imply potential friendship. Additionally, Backes *et al.* [18] found that check-in information can denote the mobility characteristics, which are significant for inferring friendship. 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. The effectiveness and robustness of TDFI are also carefully evaluated on three large-scale real-world datasets collected from Instagram [18]. Furthermore, the trace-based evaluation demonstrates that TDFI significantly outperforms state-of-the-art methods for friendship inference.

Our contributions. The following summarizes the contributions of this paper:

- TDFI can successfully realize the **simultaneous** fusion of multi-source information for friendship inference, rather than using different information hierarchically.
- To respect the user privacy, TDFI only uses coarse-grained information to achieve high-precision friendship inference.
- TDFI can effectively deal with the sparse problem, which is ubiquitous in real social networks.
- Compared with the existing state-of-the-art methods, TDFI is more accurate and more suitable for real friend recommendation systems.

The rest of this paper is organized as follows. In Section II, we present the related work. Section III describes the proposed TDFI. Furthermore, we introduce the experimental setup and analyze the performance of the proposed framework in Section IV. Finally, Section V concludes the paper.

II. Related Work

The fast growth of social networks has led to a large amount of interest in friendship and user relation analysis. Traditional friendship-inference methods mainly emphasized the mutual friends or the same groups. This is because a person is more likely to know a person of his friends rather than a random person [19]. Following this idea, multiple literatures [3], [20]–[22] started to use social graph representation to infer friendship among different users. For example, through learning the representation of the social network, node2vec [3] 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 [6]–[8]. 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.

To consider information from different sources, many studies added diverse information to their friend recommendation systems [2], [11], [16]. For example, a recent study by Huang *et al.* [2] 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 [14], [23], [24]. This is because the location information can reflect some user behaviors in physical space [14]. For example, based on the observation of Gowalla, Cho *et al.* [25] found that mobility and social constraints are related. And Pham *et al.* [26] investigated an entropy-based model to entirely utilize location information, which can not only infer friendship but also measure the strength of friendship. Besides, location information can also be applied to social network attacks [18], [27]. For example, with the assistance of random walk and feature learning, Backes *et al.* [18] can obtain the features of users' mobility, which can be used to attack the friendship among different users.

Intuitively, single-source information is frequently insufficient. However, 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), thus the variety of information in social networks in the real world makes it difficult to apply these methods to real-world social

networks. Different from the conventional friend recommendation systems, the proposed TDFI framework can automatically handle multi-source information simultaneously with reasonable complexity. The trace-based evaluation shows that TDFI can successfully leverage the highly abstract combination of different information to provide better friendship inference.

III. Two-stage Deep Learning Framework

The social network can be represented by a graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ 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 is 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. We define $\mathscr{L} = \{l_1, l_2, \ldots, l_M\}$ to represent the set of all different locations. For location information, $T_{i,k}$ is used to represent the times 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, $T_{2,1}$ 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 we only need some coarse-grained check-in information (i.e., the user id, and the location id) for respecting the user privacy.

To accurately infer friendship via multi-source information, three complementary strategies are proposed in this paper, i.e., the EAM, iDAEN, and iDSN. 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.

A. EAM: An Extended Adjacency Matrix

We at first introduce the EAM as proposed in this paper. To ensure the scalability of multi-source information, we build up a novel extended adjacency matrix represented by A. As shown 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. 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 $\vec{a}_i = \{a_{i,1}, \ldots, a_{i,N}, a_{i,N+1}, \ldots, a_{i,N+M}\}$ represents the overall multi-source information of u_i.

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

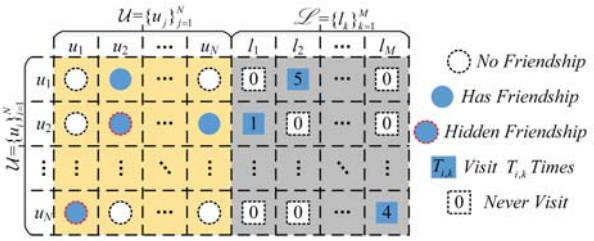


Fig. 2. The initial EAM without local and global normalizations.

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. Therefore, based on the initial EAM A, local and global normalizations are further required. 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 information can be achieved via Eq. (1).

$$\alpha_i = \frac{F_i/\overline{F}}{F_i/\overline{F} + C_i/\overline{C}}, \quad \beta_i = \frac{C_i/\overline{C}}{F_i/\overline{F} + C_i/\overline{C}} \tag{1}$$

where $\overline{F} = F/N$ and $\overline{C} = C/N$ 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. $\widetilde{A}$ is specifically used to denote the matrix after local and global normalizations. In other words, the values in each group are $\widetilde{a}_{i,j} = \alpha_i * a'_{i,j}$ and $\widetilde{a}_{i,N+k} = \beta_i * a'_{i,N+k}$.

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. 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 TDFI is evaluated for inferring the specific friendship (i.e., manually hidden friendship), from check-in information and the remaining friendship information.

B. Feature Extraction

Although the proposed EAM can represent multi-source information, it still cannot directly withdraw the essence of multi-source information. Moreover, considering the scalability, as the type of information used increases, the dimensions

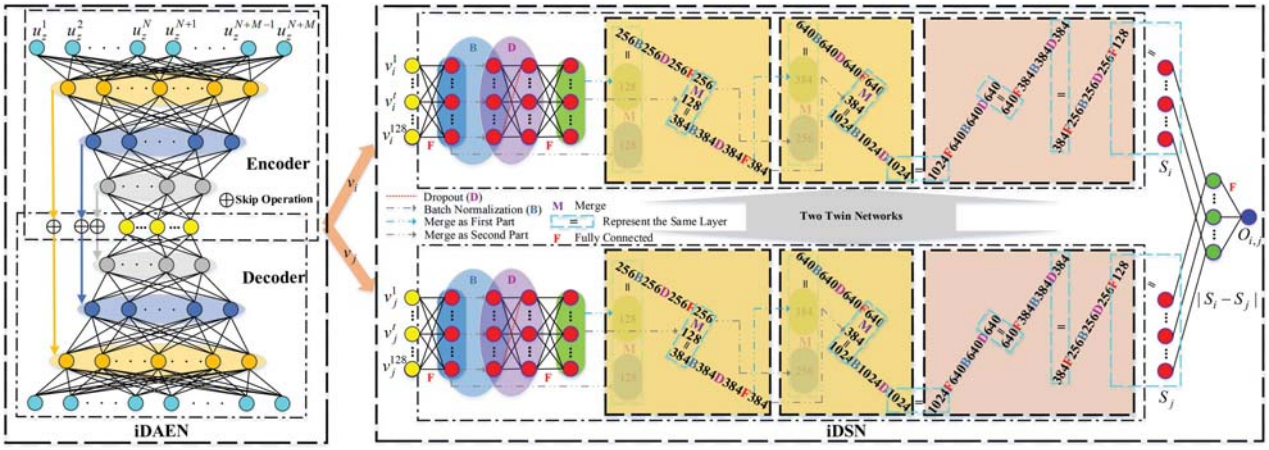


Fig. 3. The framework of TDFI, which consists of an improved deep autoencoder network (iDAEN) and an improved deep siamese network (iDSN).

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 autoencoder network has outstanding performance for feature extraction and information fusion [28], 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 time efficiency, in addition to reducing the fused information dimension, we apply special mechanisms, skip operation and feature-reuse, in the design of iDAEN and iDSN, which will be introduced in the descriptions of Fig. 3 and Fig. 4.

As shown in Fig. 3, 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. However, considering the time overhead, the design of 7 hidden layers is suitable. Equipped with the encoder network $\mathbb{E}$, which projects the input $\mathcal{X} = \{\vec{x}_i\}_{i=1}^{N} = \{\vec{a}_i\}_{i=1}^{N}$ 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 autoencoder 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. (2).

$$\begin{aligned} &\underset{\Theta=\{\phi,\varphi\}}{\arg\min} \quad ||\widehat{\mathcal{X}} - \mathcal{X}||^2 \\ &s.t. \quad \mathcal{Z} = \mathbb{E}(\mathcal{X}, \phi), \widehat{\mathcal{X}} = \mathbb{D}(\mathcal{Z}, \varphi) \end{aligned} \tag{2}$$

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

Although the deep autoencoder 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. (2) is rewritten as Eq. (3).

$$\begin{aligned} \mathcal{L}_A(\Theta) &= ||(\widehat{\mathcal{X}} - \mathcal{X}) \circ \mathcal{X}'||^2 \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N+M} ((\widehat{x}_{i,j} - x_{i,j}) \cdot (\gamma \cdot x_{i,j} + 1))^2 \end{aligned} \tag{3}$$

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 $\widehat{x}_{i,j} \neq 0$) 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. The skip connection operation can be represented as Mg in Eq. (4).

$$Mg(m) = E_i^m \oplus D_j^m \tag{4}$$

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. 3, 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.

C. Friendship Learning

Compared to the EAM representing multi-source information, the encoder network $\mathbb{E}$ of a well-trained iDAEN network can 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. Therefore, an improved deep siamese network is utilized as the second stage of TDFI to infer friendship, as illustrated in the iDSN part of Fig. 3.

The siamese network was originally proposed as an energy-based model [29], which was used to judge the similarity between pairwise samples. Similarly, 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. 3. 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. (5).

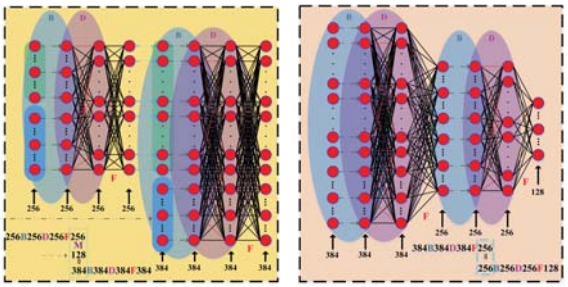
$$O_{i,j} = f(W|S_i - S_j| + b) \tag{5}$$

where $f(\eta) = \frac{1}{1+e^{-\eta}}$. 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. (6).

$$\mathcal{L}_S(\psi; \mathcal{P}) = -\frac{1}{P}\sum_{r=1}^{P}(y_r log(\hat{y}_r) + (1 - y_r)log(1 - \hat{y}_r)) \tag{6}$$

where ψ represents all the parameters of the iDSN network. $\mathcal{P} = \{\vec{p}\}_{r=1}^{P}$ 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. $\widehat{y}_r$ 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.



(a) Common Layers (b) Neurons Reduction

Fig. 4. The detailed structure of iDSN network.

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 needed for training the 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. 4(a). As shown in Fig. 3 and Fig. 4, 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. 4(a), where *F* is a *fully connected* layer, *M* is a *merge* layer to combine 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. 4. Through the complementary advantages of *F-M-B-D*, the iDSN network designed for TDFI can show excellent performance in friendship inference. After the *common layers*, we decrease the number of neurons by *neurons redection*, to achieve a more efficient comparison of the similarity between different vectors. More specifically, *neurons redection* 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 iDAEN is 128, as shown in Fig. 4(b).

IV. Experiments

The proposed TDFI aims at inferring friendship from multi-source information. In this section, we will introduce the

TABLE I
STATISTICAL PROPERTIES OF THE THREE DATASETS.

	$\lvert\mathcal{U}\rvert(N)$	$\lvert\mathscr{L}\rvert(M)$	$\lvert\mathcal{F}\rvert(F)$	$\lvert\mathcal{C}\rvert(C)$	$\overline{C}=C/N$	$\overline{F}=F/N$	$F/(N(N-1)/2)$
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**

experimental setup, the detailed deployment of the network, and the experimental results of TDFI compared with the state-of-the-art methods. To fully evaluate the performance of the baseline model and our proposed TDFI, the experiment mainly involves two aspects, i.e., effectiveness and robustness.

A. Datasets

To evaluate the framework proposed in this paper, experiments are performed on three real-world datasets collected from Instagram in 2016, which is originally used for friendship attack [18]. Specifically speaking, 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 cites, 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 followees of the users via Instagram's API. The statistical properties of the three datasets are demonstrated in Table I. Here, $\overline{C}$ and $\overline{F}$ 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.* [18] 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, the 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.* [30] 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 use 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

We employ walk2friends [18] 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 [17], [26], [31]. In terms of effectiveness and robustness, walk2friends has consistently outperformed these 14 models by 13% to 20% on the same datasets. As a result, walk2friends serves as the state-of-the-art method in friendship inference on these three large-scale multi-source information datasets. Therefore, the following comparisons including effectiveness and robustness are made between walk2friends and the proposed TDFI.

As for the walk2friends method, it uses 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 is constructed according to the pairwise similarity, such as consine similarity, euclidean distance, chebyshev 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 used in [18]. And the results of the cosine pairwise similarity method which maximizes the AUC in [18], are chosen to compare with the proposed TDFI.

C. Parameter Setting

For a fair comparison, we set the parameters for walk2friends in line with the default parameters suggested in [18] to gain the optimal performance. In particular, the walk length $l_w = 100$, walk times $t_w = 20$, and the dimensionality of feature vector $d_w = 128$.

As for the iDAEN network, the hyperparameter γ in Eq. (3) 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* to avoid gradient vanishment. Thus, 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 IV-E will show that TDFI has achieved good performance on all three datasets, which further demonstrates the universality of the proposed TDFI.

D. Evaluation Metrics

For the comparison between the baseline algorithm and the proposed TDFI, one of the evaluation metrics we adopt here is AUC (i.e., the Area Under the Curve of ROC (Receiver Operating Characteristic) [32]). And the higher the AUC of the algorithm is, the better the performance is. Moreover, AUC is the same metrics as previous inference algorithms (i.e., walk2friends), which is conducive to the fairness of TDFI and baseline algorithms comparison.

In addition to AUC criterion, Equal Error Rate (EER) and Rate of Detection (RD) are also used. In friendship inference, the pair of users with friendship is regarded as positive, otherwise, the case is negative. So the *FPR* (False Positive Rate), *TPR* (True Positive Rate) and *FNR* (False Negative Rate) in the ROC curve are defined by Eq. (7):

$$\begin{aligned} FPR &= \frac{\sharp\text{of false-positive pairs}}{\sharp\text{of negative pairs}} \\ TPR &= \frac{\sharp\text{of true-positive pairs}}{\sharp\text{of positive pairs}} \\ FNR &= \frac{\sharp\text{of false-negative pairs}}{\sharp\text{of positive pairs}} \end{aligned} \tag{7}$$

EER, defined as the *FPR* value of the point on the ROC curve when *FPR* equals to *FNR*. It is a 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 and has a better user experience.

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

E. Experimental Results

To fully verify the effectiveness of our method, 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 the baseline algorithm.

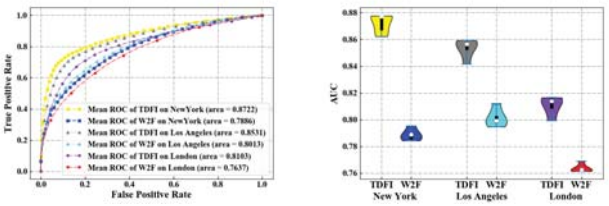
1) Comparison of Effectiveness: Due to the sparsity of the social networks indicated in the last column of Table I, the number of user pairs with friendship is far less than the number of user pairs without friendship such that the high imbalance of the labels is brought about. Hence, we utilize the same down-sampling strategy as in [18]. 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 train set $\mathcal{P} = \{\mathcal{Q}_h\}_{h=1,h\neq i}^{5}$, and the remaining part $\mathcal{Q}/\mathcal{P}$ as the testing set. The implementation details of our framework are shown in Algorithm 1. Note that the friendships for testing is hidden in the training of both the iDAEN network and the iDSN network, as illustrated in Fig. 2.

Algorithm 1: Inferring friendship via TDFI

Input: $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{C}$
Output: 0 or 1

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

The AUC results of 5-fold cross-validation on three datasets are demonstrated in Table II. The samples of cross-validation for TDFI and walk2friends are equally the same. The results of walk2friends are selected with optimal performance among the 7 pairwise similarity methods in [18] (i.e., cosine similarity). The ROC curves of both TDFI and walk2fiends on New York, Los Angeles and London are shown in Fig. 5. Note that the mean value of AUC in Table II is different from that in Fig. 5(a), because the value of each AUC in Table II is calculated separately and then averaged, while the area in Fig. 5(a) is calculated by taking all the five folds of cross-validation data and labels together.



(a) Comparison of Mean ROC (b) AUC Distribution

Fig. 5. Effectiveness comparison between TDFI and W2F.

In Fig. 5(a), the mean ROC curves of TDFI is consistently above the curves of walk2friends, which demonstrates that the proposed TDFI outperforms walk2friends, no matter what pairwise similarity methods walk2friends uses.

TABLE II
EFFECTIVENESS COMPARISON ABOUT AUC DETAILS, WHERE WALK2FRIENDS ADOPTS THE COSINE SIMILARITY.

	Methods	1-Fold	2-Fold	3-Fold	4-Fold	5-Fold	Mean	std. dev.
New York	TDFI	0.8776	0.8621	0.8770	0.8772	0.8666	0.8721	0.0065
	walk2friends	0.7890	0.7953	0.7843	0.7852	0.7891	0.7886	0.0039
Los Angeles	TDFI	0.8521	0.8416	0.8582	0.8592	0.8565	0.8535	0.0065
	walk2friends	0.8030	0.7947	0.8118	0.7975	0.7992	0.8012	0.0059
London	TDFI	0.7993	0.8160	0.8083	0.8138	0.8165	0.8108	0.0064
	walk2friends	0.7626	0.7615	0.7691	0.7620	0.7631	0.7636	0.0028

In addition to the mean values of AUC (i.e., the area below mean ROC curves) in Fig. 5(a), we also draw the distribution of AUC in each fold cross-validation. Specifically, Table II has been designed to see the specific AUC values, while Fig. 5(b) is for showing the advantages and statistical characteristics of the method. Fig. 5(b) uses the violin plots to show the comparison of AUC distribution on each city. The white circle is the median value, and the thick black line within the violin indicates interquartile range. The shape of the violin shows the distribution of the AUC values. 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. For our algorithm, we 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 baseline models, the proposed TDFI pushes a great improvement over these baseline approaches.

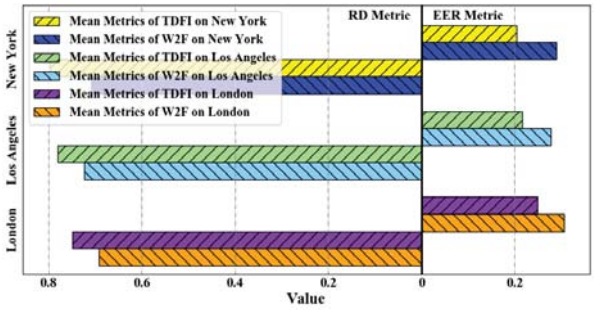
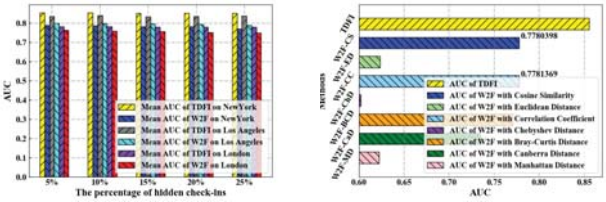


Fig. 6. EER and RD comparison between TDFI and W2F.

We also record the results of EER and RD criteria, which are shown in Fig. 6. 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. As introduced in Section IV-D, this means that TDFI is better in terms of effectiveness and more suitable for real-world applications.

2) Comparison of Robustness: From the last column of Table II, we can see that the standard deviation is very small, indicating that our method is relatively stable. In addition, both TDFI and walk2friends use the check-in information for feature extraction of each user. 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 is fixed as previously mentioned with the same default parameters. Accordingly, the results of robustness comparison are summarized in Fig. 7.



(a) Comparison of Mean AUC (b) One Special Case

Fig. 7. Robustness comparison between TDFI and W2F.

It can be clearly seen from Fig. 7(a) that as check-ins decrease from 5% to 25%, the height of these bars has changed slightly, which shows that both our TDFI and the baseline method are relatively stable. More specifically, the AUC results of walk2friends reduce as check-ins decrease, indicating that walk2friends has a higher dependence on the check-in information. Fig. 7(a) demonstrates that despite the lack of some check-in information and no matter how many percentages of check-in are hidden, our method still outperforms walk2friends with cosine similarity. Moreover, except for some special cases similar to Fig. 7(b), the cosine similarity (i.e., used in Fig. 7(a)) is the best one among the 7 pairwise similarity methods for walk2friends in each cross-validation.

Specifically, as shown in Fig. 7(b), in the 3-th 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. And 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 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 is more general. In addition, even if walk2friends with correlation coefficient has achieved the best performance in the special case, our TDFI still outperforms the best result of walk2friends (i.e., walk2friends with correlation coefficient), which further demonstrates the effectiveness and robustness of the proposed TDFI.

V. Conclusion

In this paper, we designed and implemented a novel framework, TDFI, for friendship inference via multi-source information. This system can smartly handle different types of user-related data simultaneously with low complexity. In detail, we 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. Our qualitative and quantitative evaluations indicated that TDFI outperforms the existing recommendation systems with improved accuracy and robustness.

As for future work, we are particularly interested in the understanding of trade-offs between system accuracy and overhead in real-world systems. Moreover, we are also expanding our framework to better handle incomplete or even biased datasets. We believe that the proposed TFDI framework will bring new features to further enhance the overall applicability of friend recommendation systems.

Acknowledgment

This work in this paper was in part supported by the National Key R&D Program of China under Grant No. 2018YFB0803405, China National Funds for Distinguished Young Scientists under Grant No. 61825204 and Beijing Outstanding Young Scientist Project.

References

[1] Statista, "Number of monthly active Instagram users from January 2013 to June 2018," ([Online], Available: https://www.statista.com/statistics/253577/number-of-monthly-active-instagram-users/), 2018.

[2] 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 TMM*, vol. 19, no. 6, pp. 1314–1326, 2017.

[3] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of KDD*, 2016, pp. 855–864.

[4] C. Tu, Z. Zhang, Z. Liu, and M. Sun, "TransNet: Translation-Based Network Representation Learning for Social Relation Extraction," in *Proceedings of IJCAI*, 2017, pp. 2864–2870.

[5] Y. Gu, Y. Sun, Y. Li, and Y. Yang, "RaRE: Social Rank Regulated Large-scale Network Embedding," in *Proceedings of WWW*, 2018, pp. 359–368.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online Learning of Social Representations," in *Proceedings of KDD*, 2014, pp. 701–710.

[7] D. Wang, P. Cui, and W. Zhu, "Structural Deep Network Embedding," in *Proceedings of KDD*, 2016, pp. 1225–1234.

[8] M. Danisch, O. Balalau, and M. Sozio, "Listing k-cliques in Sparse Real-World Graphs," in *Proceedings of WWW*, 2018, pp. 589–598.

[9] P. Mittal, C. Papamanthou, and D. Song, "Preserving Link Privacy in Social Network Based Systems," in *Proceedings of NDSS*, 2013, pp. 1–19.

[10] R. Dey, Z. Jelveh, and K. Ross, "Facebook Users Have Become Much More Private: A Large-Scale Study," in *Proceedings of PerCom Workshops*, 2012, pp. 346–352.

[11] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, "Friendbook: A Semantic-Based Friend Recommendation System for Social Networks," *IEEE TMC*, vol. 14, no. 3, pp. 538–551, 2015.

[12] M. Tomlinson, "Lifestyle and Social Class," *European Sociological Review*, vol. 19, no. 1, pp. 97–111, 2003.

[13] W. Alex, "Why Is It Hard to Make Friends Over 30?" The New York Times, 2012.

[14] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *PNAS*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[15] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, "Social Friend Recommendation Based on Multiple Network Correlation," *IEEE TMM*, vol. 18, no. 2, pp. 287–299, 2016.

[16] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "User Profile Preserving Social Network Embedding," in *Proceedings of IJCAI*, 2017, pp. 3378–3384.

[17] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting Place Features in Link Prediction on Location-based Social Networks," in *Proceedings of KDD*, 2011, pp. 1046–1054.

[18] M. Backes, M. Humbert, J. Pang, and Y. Zhang, "walk2friends: Inferring Social Links from Mobility Profiles," in *Proceedings of CCS*, 2017, pp. 1943–1957.

[19] D. J. Hruschka and J. Henrich, "Friendship, cliquishness, and the emergence of cooperation," *Journal of Theoretical Biology*, vol. 239, no. 1, pp. 1–15, 2006.

[20] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale Information Network Embedding," in *Proceedings of WWW*, 2015, pp. 1067–1077.

[21] J. Liu, Q. Lian, L. Fu, and X. Wang, "Who to Connect to? Joint Recommendations in Cross-layer Social Networks," in *Proceedings of INFOCOM*, 2018, pp. 1295–1303.

[22] M. Yoon, W. Jin, and U. Kang, "Fast and Accurate Random Walk with Restart on Dynamic Graphs with Guarantees," in *Proceedings of WWW*, 2018, pp. 409–418.

[23] J. Chang and E. Sun, "Location 3: How Users Share and Respond to Location-Based Data on Social Networking Sites," in *Proceedings of AAAI*, 2011, pp. 74–80.

[24] A. Likhyani, S. Bedathur, and P. Deepak, "LoCaTe: Influence Quantification for Location Promotion in Location-based Social Networks," in *Proceedings of IJCAI*, 2017, pp. 2259–2265.

[25] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and Mobility: User Movement in Location-Based Social Networks," in *Proceedings of KDD*, 2011, pp. 1082–1090.

[26] H. Pham, C. Shahabi, and Y. Liu, "EBM: An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data," in *Proceedings of SIGMOD*, 2013, pp. 265–276.

[27] A.-M. Olteanu, K. Huguenin, R. Shokri, M. Humbert, and J.-P. Hubaux, "Quantifying Interdependent Privacy Risks with Location Data," *IEEE TMC*, vol. 16, no. 3, pp. 829–842, 2017.

[28] 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," *Information Fusion*, vol. 44, pp. 78–96, 2017.

[29] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," in *Proceedings of CVPR*, 2006, pp. 1735–1742.

[30] 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 TIFS*, vol. 13, no. 4, pp. 940–953, 2018.

[31] H. Wang, Z. Li, and W.-C. Lee, "PGT: Measuring Mobility Relationship using Personal, Global and Temporal Factors," in *Proceedings of ICDM*, 2014, pp. 570–579.

[32] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.