NES-TL: Network Embedding Similarity-Based Transfer Learning

Chenbo Fu, Yongli Zheng, Yi Liu, *Member, IEEE*, Qi Xuan, *Member, IEEE*, and Guanrong Chen, *Fellow, IEEE*

***Abstract*—The transfer learning methodology leverages knowledge from the source domain with abundant training data to the insufficient target domain. Recently, new approaches continue to be developed and used to solve different classification tasks, ranging from public news to videos and to many others. Most transfer learning methods are based on the assumption that both source and target data are in the same feature space or with the same data distribution, which however is not always true in real applications where it would lead to a negative transfer. In order to overcome this hurdle, the multiple-source transfer learning framework is useful. Since many real systems can be represented by networks, how to utilize the structural similarity between different networks so as to increase the transfer effectiveness becomes important. In this paper, the NES specification index is used to quantitatively measure the structural similarity between two networks, based on which a new transfer learning method (named NES-TL) is developed. Experiments on tag popularity prediction in StackExchange Q&A communities verify the effectiveness of the proposed approach, showing that it behaves better than existing baseline methods.**

***Index Terms*—Network Embedding, Network Similarity, Transfer Learning, Q&A Community, Tag Network.**

I. Introduction

TRADITIONAL machine learning methods have achieved great success to date, thereby been being widely applied to many real applications [1]–[7]. However, in some circumstances, it is difficult and expensive to collect sufficient training data, leading the models to perform very poorly. Transfer learning methodology [8], [9] aims to solve the problem by leveraging knowledge from one source domain with fruitful training data to one target domain, which has been widely applied to computer vision [5], [10]–[12], sentiment analysis [13], [14], wifi localization [15], and so on. However, applying transfer learning to target task involved with network analysis across networks has not been sufficiently investigated, such as node popularity classification.

Real complex systems contain abundant structural information, thus can be represented by networks, where nodes and links capture their structures [16]–[22]. These structural data have attracted lots of attention, and a new frontier inter-discipline called *network science* had emerged [23]. For example, in open source software projects, developers communicate by sending/receiving emails to/from others and thus construct an email network [24]. The structural information in the networks has been proven very useful and reliable for tasks such as node classification [25], link prediction [4], [26] and so on. The present work focuses on the problem of node popularity prediction in the tag network of Q&A community. To build accurate prediction models, a sufficient number of labeled nodes are necessary. In many situations, although structural information is easy to collect, the node labels are either expensive to obtain or simply not available. For example, there are many small Q&A communities on StackExchange platform, these small Q&A communities are lack of sufficient labeled nodes and make it difficult to train a good classifier. Fortunately, there are also some communities with abundant labeled data from different topic yet structural information are similar. Thus, one can take advantage of the network with rich labeled information to help build a good classifier for the small communities, e.g., by using transfer learning technology.

Transfer learning, which utilizes the labeled instances from one source domain, is called one-source transfer learning. For example, Yang *et al.* [11] proposed the adaptive support vector machine (SVM) to learn a high-quality classifier in the target domain, which is adapted from an existing classifier trained with the instances from one source domain. However, the instances from the source domain are not necessarily helpful for the desired task in the target domain. To resolve this problem, Jiang and Zhai [27] analyzed the domain adaptation problem from the instance weighting viewpoint and proposed a general instance weighting method for natural language processing. Dai *et al.* [28] introduced a kind of boosting-based transfer learning framework, called TrAdaBoost, which uses boosting to filter out the training data with different distributions. Later Duan *et al.* [29] proposed a multiple kernel

Manuscript received July 17, 2018; revised July 30, 2019; accepted September 12, 2019. Date of publication September 19, 2019; date of current version September 2, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 11505153, Grant 61572439, and Grant 61973273, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR19F030001, and in part by the Hong Kong Research Grants Council under the GRF under Grant CityU11200317. Recommended for acceptance by H. Liu. *(Corresponding author: Qi Xuan.)*

C. Fu, Y. Zheng, and Q. Xuan are with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: cbfu@zjut.edu.cn; 2435287691@qq.com; xuanqi@zjut.edu.cn).

Y. Liu is with the Institute of Process Equipment and Control Engineering, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: yliuzju@zjut.edu.cn).

G. Chen is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China (e-mail: eegchen@cityu.edu.hk).

Digital Object Identifier 10.1109/TNSE.2019.2942341

learning framework called the domain transfer multiple kernel learning, where the kernel is comprised of a linear combination of multiple predefined base kernels. Long *et al.* [30] proposed an adaptation regularization based transfer learning method, which learns an adaptive classifier by simultaneously optimizing the structural risk functional. For the online Question and Answer (Q&A) Community, Wang *et al.* [31] proposed a tag transfer learning model for effective cross-domain collaborative filtering, which can improve the performances of recommender systems. According to the type of transferred information, transfer learning approaches can be roughly divided into four categories, i.e., instance-based, feature-based, parameter-based and relation-based transfer learning [8]. The first category aims to perform transfer learning through instances, by usually reweighting the instances from the source domain to correct for marginal distribution differences [28], [32]. The second category aims to transfer knowledge through features, features based on TL methods are to learn a new feature transformation rather than weights or find a common latent feature space which can better represent both source and target domains [30], [33]. The third category aims to perform transfer learning through sharing parameters of source and target domain learner models [34], [35]. And the last category aims to perform transfer learning through the relationship between the source and target domains [36], [37]. The present work belongs to the first transfer learning category, i.e., the instance-based transfer learning, where the instances are reweighted by the network embedding similarity.

The effectiveness of transfer learning depends on the similarity between the source and the target domains, which is relatively difficult to measure. If the source and target are weakly related, brute-force transferring may result in poor performances of the target classifier, which is called a negative transfer. Thus, it is vulnerable for the one-source transfer learning, and the negative transfer is always like the Achilles' heel against variations. One solution to decrease the risk of generating negative transfer is to utilize multiple sources, leading to the idea of multi-source transfer learning. In recent years, great effort has been devoted to this idea. Schweikert *et al.* [38] introduced a multiple convex combination of pre-trained classifiers from both source and target domains for multi-source transfer learning. Tan *et al.* [39] leveraged different views from different source domains to assist the target task. Yao *et al.* [34] effectively extended the work of Dai [28] (TrAdaBoost) and adapted the transfer boosting algorithm to multiple-source domains. Meanwhile, attempts have been made to integrate the domain similarity into the transfer learning framework. For example, Chattopadhyay *et al.* [40] proposed a multi-source transfer learning framework based on a weighting scheme, where the weights of the source domains are calculated with a conditional probability. Duan *et al.* [33] introduced a domain-dependent regularization approach and developed two domain-adaptation methods, called FastDAM and UniverDAM, which were evaluated on the video concept detection task and the document retrieval task, revealing that those methods can achieve better performances in general. Yang *et al.* [41] proposed a method to effectively evaluate the relatedness among the source and target domains through the co-occurrence data.

Most of the existing transfer learning studies focus on non-structural data, ignoring the significant structural features such as node centrality and the relationship between node pairs. Fortunately, in the last decade, more and more researchers have begun to investigate transfer learning method on network data [42], [43]. Recently, Fang *et al.* [44] discovered some latent features by constructing the label propagation matrices in source and target networks and mapping them into a shared latent feature space, through which the proposed method achieves successful knowledge transfer between some real networks. Similar method also adopted by Shen *et al.* [45], they proposed cross-network deep network embedding model to address node classification problem by embedding the nodes from the source and the target networks into a unified low-dimensional latent space. Furthermore, a TrGraph algorithm based on the common signature subgraphs between networks was proposed, which can efficiently transfer knowledge between networks [46]. Ye *et al.* [47] proposed an AdaBoost-like transfer learning algorithm to classify positive and negative links in signed social networks. Moreover, Long *et al.* [48] refined the latent factors by graph co-regularization, and proposed a transfer learning framework, which can preserve the statistical and geometric properties across domains. Niu *et al.* [49] proved that when the networks exhibit positive network correlation, it is often effective to simply apply the learned models directly to the target network without modification. Tang *et al.* [50] developed a framework called TranFG to predict the types of social relationships in a target network by borrowing knowledge from a different source network. Besides, Qi *et al.* [51] utilized transfer learning method to solve the link prediction problem across networks. These recent works proved that transfer learning method is powerful on classical network problems, e.g., node classification and link prediction.

Furthermore, notice that StackExchange is one of the most popular Q&A websites, containing a number of Q&A communities in different special domains. In these Q&A communities, tags (e.g., python, ios8, swift2) play an important role in filtering information. Typically, they are selected by the users to broadly cover the domain of the questions, and a good tag will help related questions to be easily searched so as to obtain satisfactory answers. Some tags are frequently used and thus become more and more popular today, while some others are rarely used and finally diminished. An effective tag classifier will help the administrator in better managing the tags or designing tag recommendation algorithms, e.g., Flickr tags [52] and Twitter hashtag [53]. Moreover, predicting the popularity is a well-defined problem in social media [54], [55] and in scientific communities [56]. By using both non-structural and structural features, it was found that the performances of the prediction algorithms can be significantly improved [57].

In the StackExchange Q&A communities, the tags can construct a network through co-occurrence. When new tags join, the network grows. How to predict the popularity of these new tags in the coming future is a challenging task, especially in small communities. In this study, therefore, this question is addressed from the transfer learning perspective, with focus on the similarity between the target and the source networks.

TABLE I
DATA DESCRIPTION IN STACKEXCHANGE Q&A COMMUNITIES

Category	Community	#Labeled Tag	Time Frame
Technology	Stack Overflow	3420	2008/07/31–2016/09/01
	Ask Ubuntu	226	2009/01/08–2016/09/01
	Super User	393	2008/09/15–2016/09/01
	Server Fault	260	2008/08/01–2016/09/01
	Tex-Latex	93	2010/07/26–2016/09/01
Culture/Recreation	English Language & Usage	74	2010/08/05–2016/09/01
	Arqade	280	2010/07/07–2016/09/01
Life/Arts	Science Fiction& Fantasy	140	2011/01/11–2016/09/01
	Home Improvement	53	2010/07/21–2016/09/01
Science	Mathematics	107	2010/07/20–2016/09/01
	Physics	72	2010/11/02–2016/09/01

Specifically, the main contributions of this paper are three-fold, summarized as follows:

- First, a new index called the NES index is formulated based on the Weisfiler-Lehman (WL) network embedding method [58]. Experiments show that NES is highly related to the negative transfer.
- Second, the NES index is used as the weight to integrate into the multi-source transfer learning framework. Experimental results show that the multi-source transfer learning method can thus be significantly improved, showing the effectiveness of NES in transfer learning.
- Third, NES index is compared to six statistical properties and networks that are embedded with representative strategies under the transfer learning framework. Experimental results show that NES outperforms the other baseline methods on most tasks.

The rest of the paper is organized as follows. In Section II, datasets and the present research objective are introduced. In Section III, the description and analysis of the NES index are presented. In Section IV, the NES index is integrated into multi-source transfer learning, with experiment results demonstrated. Finally, in Section V, conclusions and discussions are drawn with some discussions.

II. Dataset and Tag Prediction

The present study uses the publicly available dataset from StackExchange[1]. StackExchange provides data dumps of different Q&A communities. In each community, the data dump provides all posts, including information on questions and answers, tags, posting dates, and user reputation and badges. Here, the focus is on the four largest categories with 11 communities in the dataset, with each category containing more than two communities, as summarized in Table I.

Consider the task of predicting tag popularity based on both structural and non-structural features in a transfer learning framework. The structural features are based on the tag network, i.e., two tags are connected if they belong to the same question and the link-weight is defined as the frequency of co-occurrences of the two associated tags in the same question. Since the tag network at the time when they emerged is quite fragmented, therefore the local structure around these tags at that time will not provide much information for predicting their future evolution. For this reason, only 90% latest emerged tags are considered in this study. To label popular and unpopular tags, all the considered tags are sorted according to their usage frequencies within two years in descending order, and the top 5% tags are chosen as popular tags. Thus, the 85% least frequently used tags are put into the unpopular tag pool. For example, in the Stack Overflow community, the tags with frequency greater than 138 are chosen as popular tags and those with frequency lower than 49 are grouped into the unpopular tag pool. From the unpopular tag pool, choosing a corresponding unpopular tag closest to each popular tag in time, and thus two balanced classes are obtained.

Specifically, six typical and commonly used structural features [59], [60] are extracted from their tag networks, including centrality-based and neighbor-based features. These features are the typical measures in network science regardless of the background of networks, which have been widely used in node classification [61], link prediction [4], [62], [63] and so on.

1) Centrality-Based Features: Centrality is often used to identify important nodes in a network. In the Q&A community, larger value of centrality means the tag is more important in the network, and this tag has high probability to be a popular tag.

- **Normalized Weighted Degree Centrality.** Degree centrality is defined as the number of links connected to the node in the network. That is to say, the higher the degree value of a node is, the more important of the node. When considering the degree centrality in different networks, it is essential to calculate the normalize degree by dividing the total number of nodes in the network. In tag network the degree of a node (tag) represents the number of different tags that co-occurrences with this tags. The normalized weighted degree centrality of tags is defined as

$$D_c(t_i) = \frac{1}{N-1} \sum_{t_j \in N_e(t_i)} w_{t_i,t_j} a_{t_i,t_j} , \tag{1}$$

[1] https://archive.org/download/stackexchange

where $N_e(t_i)$ is the neighbor set of tag t_i and a_{t_i,t_j} is an element of the adjacency matrix $\mathbf{A}$, i.e., $a_{t_i,t_j} = 1$ if tag t_i has a link with tag t_j, and $a_{t_i,t_j} = 0$ otherwise; w_{t_i,t_j} is the weight of the link between tag t_i and tag t_j, and N is the total number of tags in the network.

- **Eigenvector Centrality**. Eigenvector centrality is a measure of the influence of nodes. high eigenvector centrality means the node is connected to many nodes who themselves have high eigenvector centrality. The eigenvector centrality is defined as

$$E_c(t_i) = \frac{1}{\lambda} \sum_{t_j \in N_e(t_i)} a_{t_i,t_j} E_c(t_j) , \tag{2}$$

where λ is the greatest eigenvalue of the adjacency matrix.

- **Closeness Centrality**. Closeness centrality measures the average shortest distance from one node to others in a network. Therefore, the closer it is to other nodes, the more important a node is. Closeness centrality can also be understood as determining the importance of nodes by using the average propagation time of information in the network. It is defined as

$$C_C(t_i) = \frac{N-1}{\sum_{j \neq i} w_{t_i,t_j} d_{t_i,t_j}} , \tag{3}$$

where d_{t_i,t_j} denotes the shortest path length between tag t_i and tag t_j in the network. The shorter the distances between node t_i and the rest nodes are, the more central the node t_i is, and thus the larger this C_C index is.

2) Neighbor-Based Features: Neighbor-based features characterize the structural properties of the node and its neighbor. The importance of a node in a network not only depends on itself but also on its neighbors. In the tag network, if the tag often co-occurs with popular tag, it has a high possibility to be a popular tag.

- **Cluster Coefficient**. Cluster coefficient measure the connectivity of node in the network. In a social network, the cluster coefficient captures the triangular friendship in the network, defined as

$$C(t_i) = \frac{2L_{t_i}}{k_{t_i}(k_{t_i} - 1)} , \tag{4}$$

where k_{t_i} is the degree of tag t_i, and L_{t_i} is the number of links among the k_{t_i} neighbors of tag t_i.

- **Average Clustering Coefficient of Neighbors**. The average clustering coefficient of neighbors of a tag is calculated according to the clustering coefficient of its neighbors. It is defined as

$$\langle C_N(t_i) \rangle = \frac{\sum_{t_j \in N_e(t_i)} C(t_j)}{N_e(t_i)} , \tag{5}$$

which describes the social relationship of neighbors.

- **Average Normalized Degree of Neighbors**. Average normalized degree of neighbors is calculated according to the average normalized degree centrality of the nodes

$$\langle C_{NC}(t_i) \rangle = \frac{\sum_{t_j \in N_e(t_i)} D_c(t_j)}{N_e(t_i)} , \tag{6}$$

which captures the importance of the neighbors of a tag.

Moreover, consider the following four non-structural features:

- **Number of Posts** (N_p). The number of posts [64] is defined as the number of questions and answers tagged with the same tag t_i. Tags sometimes describe the topic of a post. The more posts use tag t_i means the more popular of this topic is.
- **Experience of Questioner** (E_q). The experience of questioner [65] is defined as the number of tags provided by the questioner.
- **Average Number of Votes** (N_v). The number of votes [66] is defined as the difference between the upvotes and downvotes of the questions tagged with t_i, normalized by the number of questions tagged by t_i.
- **Length of Question** (L_q). The length of a question is defined as the average number of words in the question tagged by t_i. It is also used as a feature to predict the popularity of Tweet and online news [67], [68].

To this end, the traditional metric, *accuracy*, is adopted to measure the goodness of an algorithm [69].

III. Network Embedding Similarity

In this section, a new metric, named *NES*, is introduced to measure the similarity between networks, based on network embedding. Then, the relationship between the NES index and the transfer learning effectiveness will be discussed.

A. Definition of NES

An undirected network is described by a graph $G(V, E)$, where V and E are the sets of nodes and links, respectively. In this work, network embedding vector is used to calculate NES. Fig. 1 presents two simple networks of different structures, and illustrates how to calculate NES, where the Weisfeiler-Lehman (WL) scheme is chosen as the network embedding method.

Specifically, suppose that there are two networks, G_1 and G_2. For each network, the nodes are labeled based on their degrees. Let D_G^h and f_G^h be the label sequence of network G in ascending order and the frequency sequence of labels in D_G^h, respectively, where h denotes the iteration number. Initially, the label of a node is its degree. For example, network G_1 has three values of its degree: 1 (blue), 2 (yellow), and 3 (red). The degree sequence of network G_1 thus is $D_{G_1}^0 = \{1, 2, 2, 2, 2, 3\}$. Similarly, for G_2, one has $D_{G_2}^0 = \{1, 2, 2, 3, 3, 3\}$. Since both networks have the same set of degree values, the label frequencies for G_1 and G_2 are $f_{G_1}^0 = (1, 4, 1)$ and $f_{G_2}^0 = (1, 2, 3)$, respectively. In step 1, continue to label the nodes based on the degree sequence of

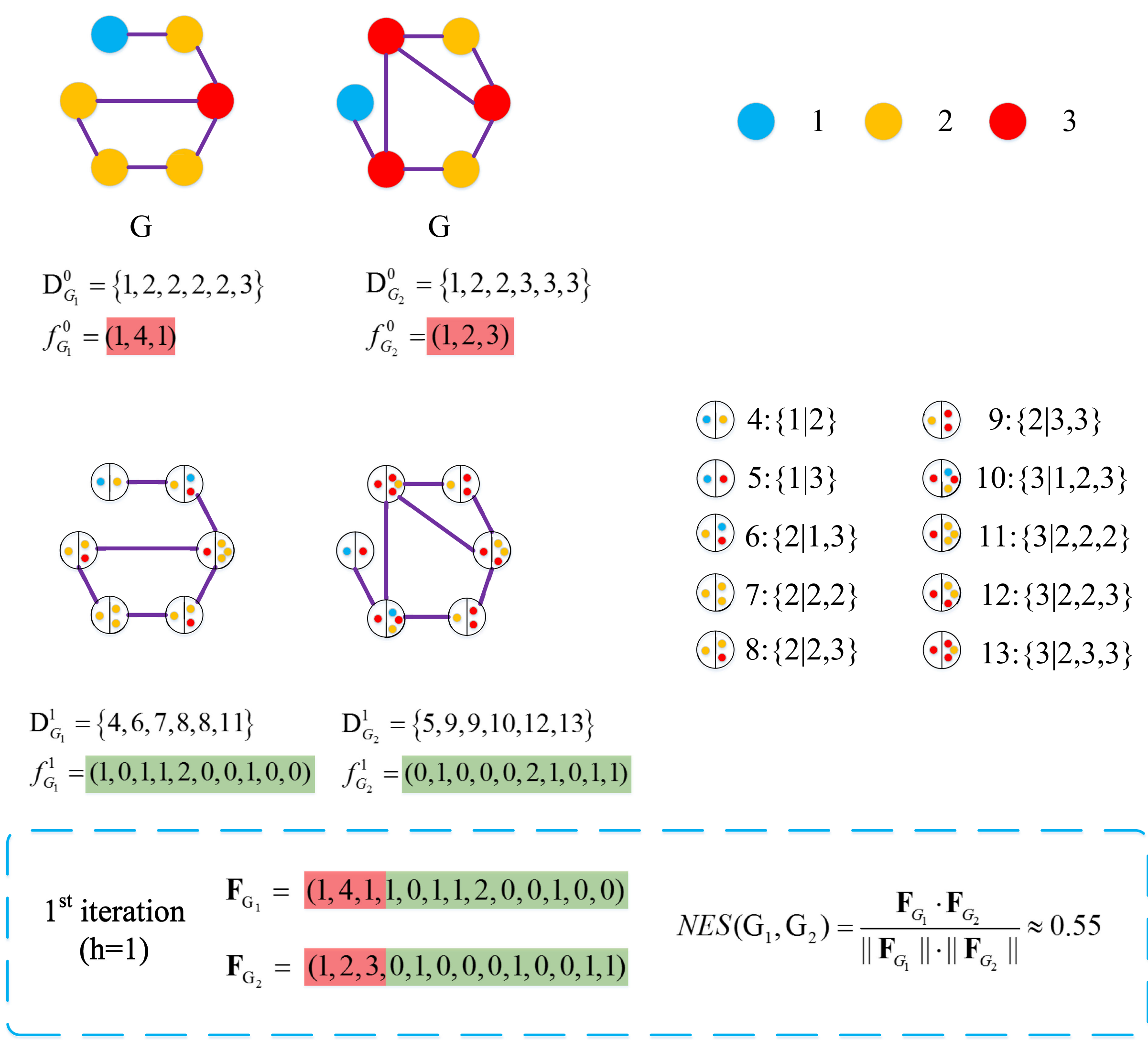


Fig. 1. An example of calculating the NES between two simple networks, G_1 and G_2.

their neighbors. In particular, first, sort the nodes based on their degree, from small to large, where for nodes of the same degree, sort them based on the smallest degree of their neighbors, then second smallest degree, and so on. For example, the blue node of G_1 (with label 1) has a yellow neighbor (with label 2) initially. Thus, $\{1|2\}$ is used to represent it and relabel it with a new number 4. Meanwhile, the blue node of G_2 (with label 1) has a red neighbor (with label 3) initially, and in this case, $\{1|3\}$ is used to represent it and relabel it by 5. The sequence of these relabeled nodes is assigned as $D^1_{G_i}$ and the frequency sequence of labels in $D^1_{G_i}$ is assigned as $f^1_{G_i}$, for $i = 1, 2$, as shown in Fig. 1. The length of $f^h_{G_1}$ is exactly the same as that of $f^h_{G_2}$ at the same iteration, where an element equal to 0 means that there is no such label in the corresponding network. After p iterations, the whole graph embedding vector $\mathbf{F}_{G_1}$ can be obtained by appending $f^p_{G_1}$ to all the former frequency sequences $f^0_{G_1}, f^1_{G_1}, f^2_{G_1}, \ldots, f^p_{G_1}$. Finally, the NES between G_1 and G_2 is calculated by the cosine similarity $NES(G_1, G_2) = \mathbf{F}_{G_1} \cdot \mathbf{F}_{G_2}/(||\mathbf{F}_{G_1}|| \cdot ||\mathbf{F}_{G_2}||)$. In this paper, the iteration number is fixed as $h = 2$.

In order to verify the effectiveness of the NES index, generate four networks, i.e., two BA scale-free networks [70] and two ER random networks [71]. The four networks have the same size of 100 nodes. The parameters for scale-free networks are $m = m_0 = 3$; while for random networks, the linking probability is 0.06. For the purpose of comparison, the average degrees of these four networks are kept to be nearly the same. As shown in Fig. 2, the NES indexes between the same types of networks are much higher than those between different types of networks, indicating that this index indeed captures the structural similarity between networks. Furthermore, because the scale-free network and random network have significantly different degree distribution, it is meaningful to investigate if the difference of NES is only dependent on the degree distribution. To verify this, degree-preserving randomization [72] is used on scale-free network $BA1$. Though this method one can get a new scale-free network, call $BA1'$, and its degree distribution is as the same as $BA1$. Calculating the average NES of $BA1$ and $BA1'$, the result is 0.91, smaller than 1, which implies that the NES index is not

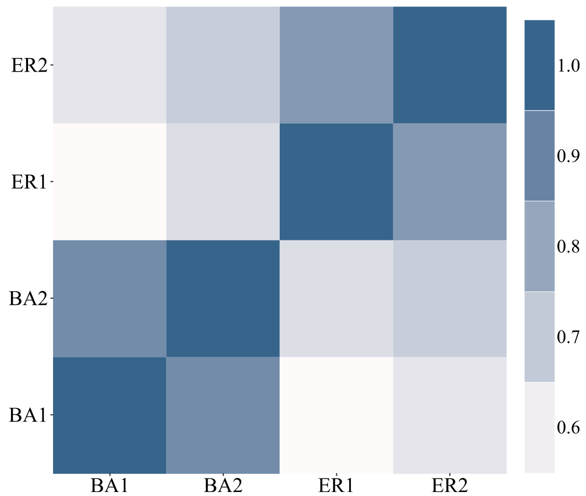


Fig. 2. The heatmap of four networks, two BA and two ER networks. The cell color is used to represent the NES between two networks, with deeper color corresponding to higher NES.

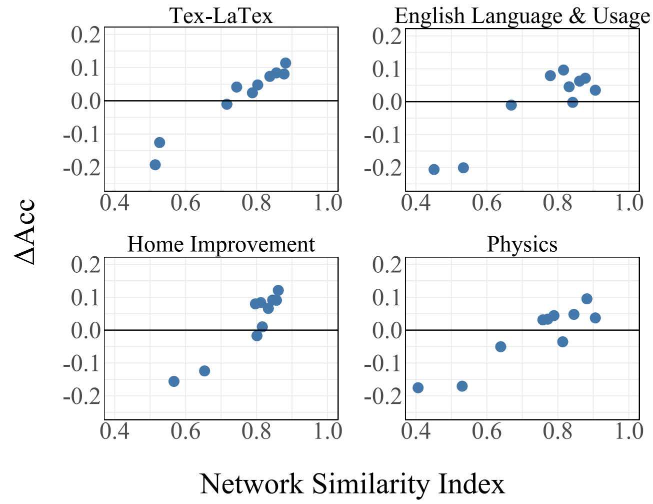


Fig. 3. The relationship between NES (using WL method) and the improved accuracy ΔAcc. Once the target community is selected, while the remaining communities are chosen as source domains.

only related to the degree distribution. This experiment has also been performed on the random network, and a similar result was obtained, i.e., NES is 0.85. All the results are averaged over 64 network randomizations.

B. NES and Transfer Learning Effectiveness

A domain is defined by $D = \{(\mathbf{x}_i, y_i)|_{i=1}^n\}$ with size n, where $\mathbf{x}_i$ is a feature vector of d dimension and y_i is the label of $\mathbf{x}_i$. In this work, consider the binary classification task, i.e., predicting the popular or unpopular tags as discussed in Section II, with respect to a given community in a target dataset $D^T = D_T^l \cup D_T^u$, where $y_i \in \{-1, +1\}$ is a binary label, i.e., it is a popular tag if $y_i = +1$ but an unpopular one otherwise. Whereas $D_T^l = \{(\mathbf{x}_i^l, y_i^l)|_{i=1}^{n_l}\}$ and $D_T^u = \{\mathbf{x}_i^u|_{i=1}^{n_u}\}$ are the labeled and unlabeled data in the target domain, respectively, with $n = n_l + n_u$ being the size of the target domain.

Now, the relationship between the NES index and the transfer learning effectiveness is investigated by experiments, where Support Vector Machine (SVM) [73]–[75] is used to train the model. Gaussian kernel $(k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2))$ is adopted and 10-fold cross-validation is applied to optimize the model parameters. The regularization parameter C is set as 1. Each experiment is performed 30 times and then the average is computed and recorded as the results. Then, the one-source transfer learning method, which is trained with mixed domain labeled data, and the non-transfer learning method (SVM-T), which uses the target labeled data only [76], are compared. Whereas, the transfer learning effectiveness is defined as the improvement of the accuracy ΔAcc of the transfer learning method compared with the non-transfer learning method on the same test dataset.

Fig. 3 plots the relationship between the transfer learning effectiveness and the NES index. Four communities from four different categories are selected as the target communities, including Tex-Latex, English Language & Usage, Home Improvement and Physics. Then, the remaining communities

TABLE II
CORRELATION TEST BETWEEN ΔACC AND NES INDEX

Community	Pearson Correlation Coefficient
Tex-LaTex	0.979 **
English Language & Usage	0.905 **
Home Improvement	0.937 **
Physics	0.920 **

$*p < 0.5, **p < 0.01, ***p < 0.001$

are chosen as the source domains. In each target domain, ten instances are sampled randomly, including five positive and five negative ones. The results show that there is a positive correlation between the NES index and the transfer learning effectiveness, i.e., the higher degree of NES between the source and target community networks corresponds to the more effective transfer learning. To show more details, the Pearson correlation between ΔAcc and the NES index are calculated, as shown in Table II. The results demonstrate that there is a significant positive correlation between the transfer learning effectiveness and the NES index. In other words, based on NES, one can roughly estimate the effectiveness of transfer learning before using it, which can effectively help avoid negative transfer. In fact, the lower the NES, the higher the probability of the negative transfer. For example, in the performed experiments, when the NES index is lower than 0.7, the negative transfer occurs in all the four communities.

IV. NES-BASED TRANSFER LEARNING

In this part, the NES index is integrated into multiple domain transfer learning. The description and analysis of this NES-based transfer learning (NES-TL) algorithm is presented.

Let $D_{S_k} = \{(\mathbf{x}_i^{S_k}, y_i^{S_k})|_{i=1}^{n_k}\}$ be the dataset from the kth source domain, $k = 1, \ldots, q$, where q is the total number of sources, n_k is the size of the kth source domain. As shown in Fig. 4, the target classifier $f(\mathbf{x})$ is obtained from q trained classifiers mentioned in Section III, with each being trained by the

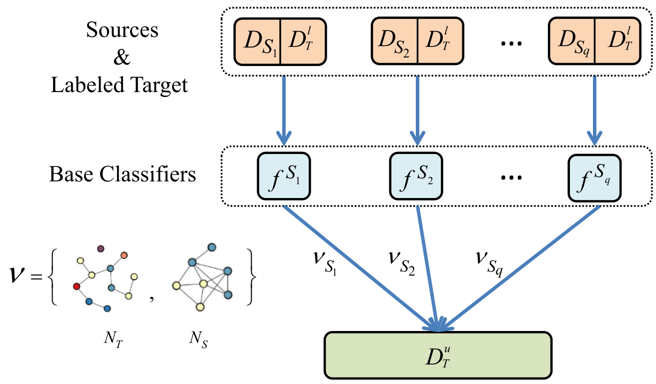


Fig. 4. The model for NES-TL. The base classifiers (f^{S_k}) are trained by mixed training instances, including sources (D_{S_k}) and the labeled target (D_T^l). The final classifier is obtained from $f(\mathbf{x}) = \sum_k^q \nu_{S_k} f^{S_k}$, where ν_{S_k} is the normalized NES.

Algorithm 1: NES-TL

Input: The source domain data sets $S = \{D_{S_1}, D_{S_2}, \ldots, D_{S_q}\}$ and the target domain data set $D_T = \{D_T^l, D_T^u\}$

q: The number of source domains

$\Omega = \{G^{S_1}, G^{S_2}, \ldots, G^{S_q}, G^T\}$: The network of source and target

Output: Target classifier function $f(\mathbf{x}) : X \to Y$

Begin:

for $i = 1, 2, \ldots, q$ **do**

calculate the structural similarity between source domain and target domain: $w_i = NES(G^{S_i}, G^T)$

learn weak learner $f^{S_i} : X \to Y$ over the combined set $D_{S_i} \cup D_T^l$

end for

Merge the similarities of all sources: $\mathbf{w} = \{w_1, w_2, \ldots, w_q\}$

Set $\nu = \frac{1}{|\log(\mathbf{w})|}$

Normalize ν to 1 such that $\sum_{i=1}^q \upsilon_{S_i} = 1$

The final model $f(\mathbf{x}) = \sum_i^q \upsilon_{S_i} f^{S_i}(\mathbf{x})$

Return: $f(\mathbf{x})$

one-source transfer learning method. Specifically, the target classifier is formulated as

$$f(\mathbf{x}) = \sum_{k=1}^{q} \nu_{S_k} f^{S_k}(\mathbf{x}), \tag{7}$$

where $\nu_{S_k} \in [0, 1]$ is the normalized weight of classifier f^{S_k} and is obtained from the NES index $1/|\log[NES(G^T, G^{S_k})]|$. A more detailed description of the NES-TL is given in Algorithm 1.

In the experiments, the task of tag popularity prediction is performed in the StackExchange communities. Six communities are chosen as the target communities, i.e., Super User (SU), Server Fault (SF), Tex-LaTex (Tex), English Language & Usage (Eng), Science Fiction & Fantasy (Sci) and Physics (Phys). Here, three communities belong to technology category, which is the largest one in StackExchange. In addition, five communities are chosen as the source domains, including Stack Overflow, Askubuntu, Arqade, Home Improvement and Mathematics. The sizes of the datasets are presented in Table III, and the basic topological properties for target and source domian networks are presented in Table IV. Here, SVM with Gaussian kernel is adopted as the basic classifier, and the regularization parameter C is set to 1.

A. Baseline Methods

To evaluate the performance of the approach developed in this work, the proposed NES-TL method is compared to the following baseline transfer learning methods:

- **SVM-T**. Training the model only using labeled data in the target domain [76].
- **FR**. For q source domains, FR training q base classifiers use the corresponding labeled data in the source domain and the labeled data in the target domain, then fuse the decisions of q base classifiers equally [33], [77].
- **MSTrAdaBoost**. Multi-Source TrAdaBoost (MSTrAdaBoost) extends the framework of the TrAdaBoost method to multiple sources. It adopts a weight-update strategy in source training instances, which is the same as TrAdaBoost, and in target instances, which is the same as AdaBoost [78], [79].
- **FastDAM**. FastDAM incorporates the domain-dependent regularizer into the least-squares SVM [33].
- **UniverDAM**. UniverDAM incorporates both domain-dependent regularizer and Universum-based [80] regularizer into the least-squares SVM [33].

In the performed experiments, linear kernel is used for MSTrAdaBoost, and Gaussian kernel is used for the others. In SVM-T and FR, the regularization parameter is set to $C = 1$, and the other parameters are set to default values. In FastDAM and UniverDAM, the regularization parameter C is set to 1, and the kernel parameter γ is set to $\gamma = 1.2^\delta \gamma'$, where $\gamma' = 1/d$ with $d = 9$ being the feature dimension, and $\delta \in \{-0.5, 0, 0.5, \ldots 4\}$. The tradeoff parameter λ_L and λ_D are both set to 1. Therefore, totally 50 base classifiers from 5 sources and 10 kernel parameters are obtained and used for the DAM methods. In MSTrAdaBoost, the regularization parameter C is set to 20, and the maximum number of iterations is fixed to 20.

B. Experimental Results

In the experiments, 20 instances are randomly selected as the labeled data in the target domain D_T^l, including 10 positive and 10 negative instances, and the rest in the target domain are used as the test set. The experiments are repeated for 30 times, and their mean accuracy are then calculated and recorded.

Table V and Table VI present the experimental results of the five baseline methods and the proposed NES-TL method, where one can see that the proposed method performs best in all the six transfer learning tasks. More precisely, compared with the SVM-T, the proposed method has improvements over 10.68% (SU), 10.69% (SF), 7.08% (Tex), 7.61% (Eng), 18.65% (Sci) and 15.89% (Phys) in terms of accuracies. And under the F1-measures, the improvements are 14.17% (SU), 14.14% (SF), 10.40% (Tex), 7.90% (Eng), 31.25% (Sci) and 6.87% (Phys), respectively. It should be noted that, although the linear weighted SVM is selected as the weak classifier for

TABLE III
DATA DESCRIPTION OF SIX TASKS IN STACKEXCHANGE Q&A COMMUNITIES

Category	Target Community (TC)	Source Community (SC)	TC Size	SC Size
Technology	Super User	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	393	4051
	Server Fault	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	260	4051
	Tex-LaTex	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	93	4051
Culture/Recreation	English Language & Usage	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	74	4051
Life/Arts	Science Fiction& Fantasy	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	128	4051
Science	Physics	Stack Overflow, Ask Ubuntu, Arqade, Home Improvement, Mathematics	72	4051

TABLE IV
BASIC TOPOLOGICAL FEATURES OF TARGET AND SOURCE NETWORKS

Community		#Node	#Link
Target	Super User	1881	24910
	Server Fault	1390	20452
	Tex-LaTex	410	3380
	English Language & Usage	327	3030
	Science Fiction & Fantasy	282	1120
	Physics	346	2914
Source	Stack Overflow	8351	130314
	Ask Ubuntu	613	4722
	Arqade	296	1114
	Home Improvement	261	1538
	Mathematics	339	2810

MSTrAdaboost, the Gaussian kernel has also been tested. The results show that the NES-TL method outperforms all MSTrAdaboost method, no matter which kernel is used. Since it seems that linear kernel performs better than Gaussian kernel on most tasks, here, only the results of MSTrAdaboost with linear kernel is presented.

The proposed method has also been tested for different numbers of labeled target data, and results are compared to the baseline methods. Fig. 5 and Fig. 6 show the variation of accuracies and F1-measures with different labeled target instances respectively. As shown in the figures, the proposed method performs better than the five baseline methods, especially when there are less than 10 instances in the labeled target data. Note that all the methods tend to converge when the number of labeled target data increases.

C. Comparing With Other Strategies

There are many methods that can be used to represent the similarity between two networks, including traditional statistical properties and network embedding methods. In this part, experiments are performed on six target tasks and the prediction accuracy on popular tags are compared under proposed NES-TL framework with different similarity indexes computing strategies, including statistical properties based indexes, i.e., *Avg-degree*, *Avg-cluster*, *D-distance*, and embedding methods based indexes, i.e., *node2vec*, *WL*, *graph2vec*. The baseline is *FR*, which ignores the differences between domains and uses the same weight for different source domains. These methods calculating network similarity indexes for comparison are briefly introduced as follows:

- ***Avg-degree***. Average degree centrality is an important statistical index of a network [23]. Here, one first calculates the absolute difference of the average degree centrality between the source and the target domains, then uses the reciprocal of this value.
- ***Avg-cluster***. Average clustering coefficient is another import structural property of a network, which captures the global degree of node aggregation in the network [23]. Similarly to the Avg-degree, one also calculates the absolute difference of the average clustering coefficient between the source and the target domains, and then uses the reciprocal of this value.
- ***D-distance***. D-distance is an index for measuring the difference of distance probability distributions between different networks [81]. The higher the D-distance, the more dissimilar of the two networks; thus, one can take the value of ($1-$D-distance) as the network similarity between the two domains.
- ***Node2vec***. Node2vec is a popular method for mapping nodes in a network to the feature vectors in a low-dimensional space, preserving the neighborhood information of these nodes [25]. To calculate the similarity index of two networks, one can average the embedding vectors of all the nodes in source and target network respectively, and then calculate the cosine similarity between average embedding vectors of the source and the target network.
- ***Graph2vec***. Graph2vec views a network as a document. It extracts the rooted subgraphs by the WL method as a vocabulary, and then learn the network representation by going through the doc2vec skip-gram training process [82]. In the experiments, the cosine similarity between the source and the target domains are calculated.

For better comparison, 10 labeled instances are randomly selected as the target labeled instances, including 5 positive and 5 negative ones, and the above indexes are used to replace WL based NES index in the proposed transfer learning framework. Each experiment is performed 30 times and then the average is computed and recorded as the results. The results are shown in Table VII, where one can see that, in most cases the proposed NES-TL method outperforms the transferring methods based on other indexes, indicating the effectiveness of the NES index in measuring the similarity between networks.

D. Time Complexity

The time complexity of the above six transfer learning methods is shown in Table VIII, where $n_L = n_l + n_s$ is the number of labeled instances in the target and the source domains,

TABLE V
AVERAGE ACCURACIES OF DIFFERENT TASKS

Target Community	SU	SF	Tex	Eng	Sci	Phys
SVM-T	0.6946±0.0456	0.7135±0.0407	0.7287±0.0528	0.8284±0.0453	0.5716±0.0588	0.6987±0.0605
FR	0.7419±0.0309	0.7719±0.0382	0.7205±0.0422	0.8870±0.0358	0.6748±0.0326	0.7978±0.0428
FastDAM	0.6969±0.0625	0.7107±0.0616	0.6982±0.0574	0.7722±0.0474	0.6019±0.0489	0.7635±0.0494
UniverDAM	0.7081±0.0885	0.7494±0.0841	0.6936±0.0704	0.7537±0.0430	0.6215±0.0652	0.7789±0.0420
MSTrAdaboost	0.7317±0.0631	0.7756±0.0823	0.7630±0.0550	0.8751±0.0516	0.6675±0.0477	0.7933±0.0420
NES-TL	**0.7688±0.0256**	**0.7896±0.0257**	**0.7803±0.0502**	**0.8914±0.0251**	**0.6782±0.0246**	**0.8097±0.0473**

TABLE VI
AVERAGE F1-MEASURES OF DIFFERENT TASKS

Target Community	SU	SF	Tex	Eng	Sci	Phys
SVM-T	0.6770±0.0517	0.6872±0.0542	0.7170±0.0473	0.8275±0.0451	0.5430±0.1083	0.7543±0.0556
FR	0.7628±0.0155	0.7714±0.0257	0.7575±0.0279	0.8852±0.0307	0.6892±0.0419	0.7689±0.0252
FastDAM	0.6487±0.1024	0.6502±0.1046	0.6990±0.0455	0.7601±0.0677	0.5385±0.1808	0.7881±0.0614
UniverDAM	0.6952±0.1629	0.7472±0.1140	0.7039±0.0553	0.7126±0.0612	0.5705±0.1953	0.7904±0.0489
MSTrAdaboost	0.7270±0.0635	0.7682±0.0976	0.7548±0.0393	0.8780±0.0673	0.7053±0.0636	0.7974±0.0355
NES-TL	**0.7729±0.0171**	**0.7844±0.0197**	**0.7915±0.0378**	**0.8936±0.0256**	**0.7127±0.0279**	**0.8061±0.0378**

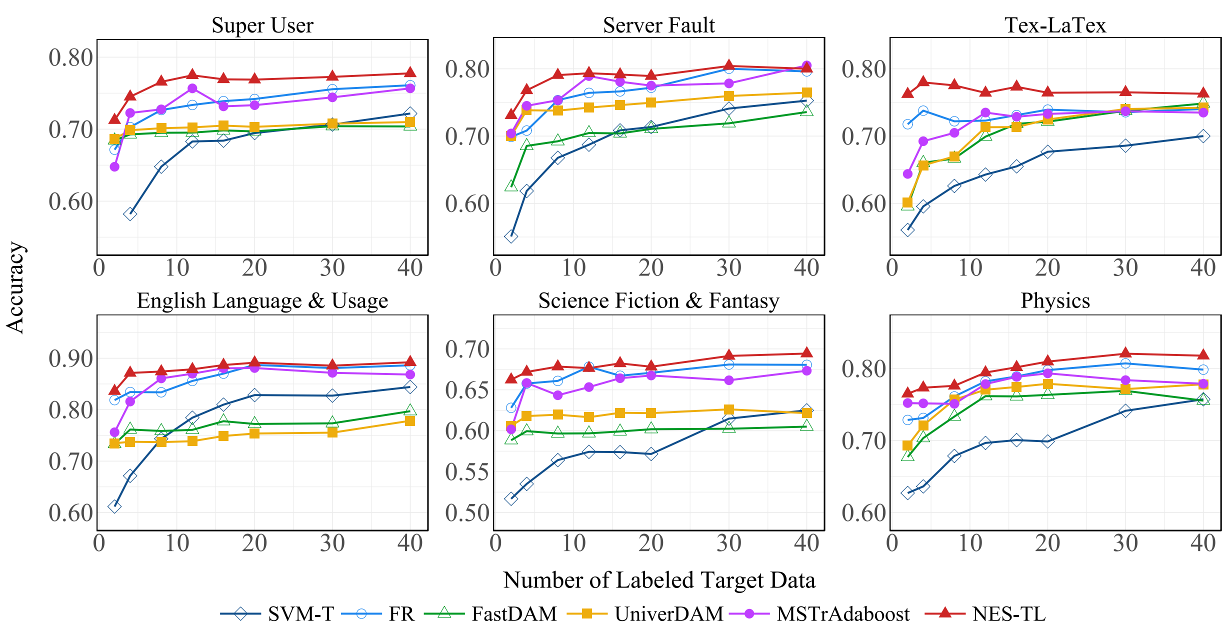


Fig. 5. The accuracies achieved by the six methods as functions of the number of labeled instances in the target domain.

$n_s = \sum_{k=1}^{q} n_k$, and n_δ is the number of possible values of δ, and I is the maximum number of iterations for MSTrAdaBoost. In these methods, FastDAM and UniverDAM are based on FR, i.e., they need pre-trained classifiers. Since they use different numbers of δ values, the total time complexity should be n_δ times the complexity of FR. Although the proposed method also needs pre-trained classifiers, it does not need to train the weights of these classifiers or the kernel parameter expansion, therefore has lower time complexity overall.

V. CONCLUSION AND DISCUSSION

In this paper, a new index, NES, is proposed for measuring the similarity between source and target networks, which is then integrated into a multi-source transfer learning framework to improve the transfer effectiveness. It is found that, for single-source transfer learning, a smaller NES between source and target corresponds to a higher probability of negative transfer. Furthermore, this index is integrated as the weight into the multi-source transfer learning and then applied to the task of tag popularity prediction in StackExchange Q&A communities on the real StackExchange dataset. The experimental results show that the proposed approach can get relatively better performances on this task than the baseline methods. The transfer learning performance has also been compared to the others using different network representation strategies, and it is found that the proposed NES outperforms the others, demonstrating its effectiveness in measuring the similarity between networks. Moreover, it also should be noted that, in this work, our main task is tag popularity prediction, which the

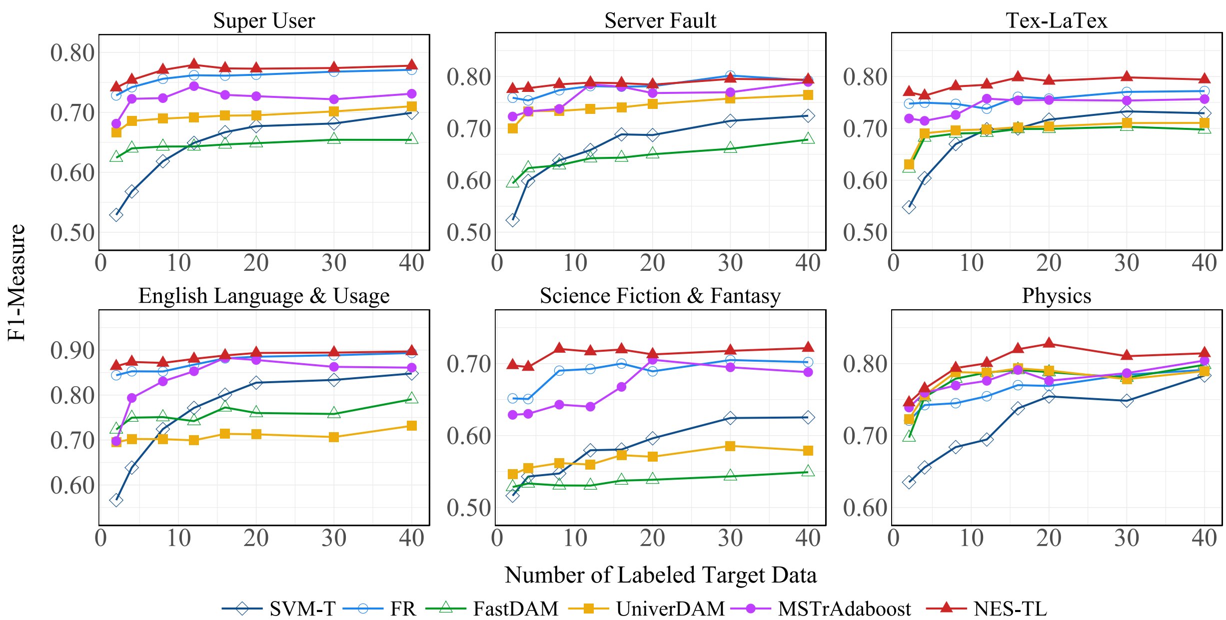


Fig. 6. The F1-measures achieved by the six methods as functions of the number of labeled instances in the target domain.

TABLE VII
COMPARISON OF PREDICTION ACCURACIES USING DIFFERENT NETWORK REPRESENTATION STRATEGIES ON DIFFERENT TASKS

Target Community	FR	Avg-degree	Avg-cluster	D-distance	Node2vec	Graph2vec	NES
SU	0.7029	0.7018	0.6726	0.7000	0.6971	0.7386	**0.7439**
SF	0.7532	0.7324	0.7488	0.7508	0.7304	0.7712	**0.7804**
Tex	0.7000	0.6964	0.6229	0.7506	0.6928	0.7434	**0.7542**
Eng	0.8734	0.8788	0.8203	0.8625	0.7938	0.8750	**0.8808**
Sci	0.6453	0.6564	0.6548	**0.6650**	0.6393	0.6513	0.6603
Phys	0.7661	0.7355	0.6952	0.7500	0.7548	0.7742	**0.7887**

TABLE VIII
TIME COMPLEXITY OF SIX TRANSFER LEARNING METHODS

Methods	Time Complexity	
	Pre-trained Classifiers	Model Training
SVM-T	–	$O\left(dn_l{}^2\right)$
FR	–	$O\left(dn_L{}^2\right)$
FastDAM	$O\left(n_\delta dn_L{}^2\right)$	$O\left(dn_l{}^2\right)$
UniverDAM	$O\left(n_\delta dn_L{}^2\right)$	$O\left(dn_s{}^2\right)$
MSTrAdaBoost	–	$O\left(I\left(dn_s{}^2+n_s\right)\right)$
NES-TL	$O\left(dn_L{}^2\right)$	$O\left(1\right)$

label space may correlate to degree distribution, but we also tested our proposed method on other tasks and datasets, e.g., classification. It is found that our method may have less benefit for the classification task where label spaces are uncorrelated to their degree distribution. This implies that our method may sensitive to network infrastructure definition, and work better when network infrastructure is better defined or obvious.

The present study highlights that the proposed NES can capture the structural similarity between different networks, therefore can be well utilized to improve the performances of multi-source transfer learning. In the further, this NES index will also be tested and applied to other applications such as network classification, and more network data will be used, such as multi-layer network [83], [84] or heterogeneous network [85].

REFERENCES

[1] T. G. Dietterich, "Machine-learning research," *AI Mag.*, vol. 18, no. 4, pp. 97–136, 1997.
[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
[3] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7–8, pp. 2031–2038, 2013.
[4] C. Fu *et al.*, "Link weight prediction using supervised learning methods and its application to yelp layered network," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1507–1518, Aug. 2018.
[5] Q. Xuan, B. Fang, Y. Liu, J. Wang, Y. Zheng, and G. Chen, "Automatic pearl classification machine based on a multistream convolutional neural network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6538–6547, Aug. 2018.
[6] Y. Liu, C. Yang, Z. Gao, and Y. Yao, "Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes," *Chemometrics Intell. Lab. Syst.*, vol. 174, pp. 15–21, 2018.
[7] Y. Liu, Y. Fan, and J. Chen, "Flame images for oxygen content prediction of combustion systems using DBN," *Energy Fuels*, vol. 31, no. 8, pp. 8776–8783, 2017.
[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 10, no. 22, pp. 1345–1359, Oct. 2010.
[9] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 1, no. 3, pp. 1–40, 2016.
[10] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 1785–1792.
[11] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 188–197.
[12] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1304–1309.

[13] C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment,” in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1541–1546.
[14] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification,” in *Proc. 45th Annu. Meet. Assoc. Comput. Linguistics*, 2007, pp. 440–447.
[15] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, “Transfer learning for wifi-based indoor localization,” in *Proc. Workshop Transfer Learn. Complex Task 23rd AAAI Conf. Artif. Intell.*, 2008, pp. 43–48.
[16] M. E. J. Newman, “The structure and function of complex networks,” *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.
[17] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, “Complex networks: Structure and dynamics,” *Phys. Rep.*, vol. 424, no. 4–5, pp. 175–308, 2006.
[18] S. H. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.
[19] C. Fu, J. Wang, Y. Xiang, Z. Wu, L. Yu, and Q. Xuan, “Pinning control of clustered complex networks with different size,” *Physica A, Statistical Mech. Appl.*, vol. 479, pp. 184–192, 2017.
[20] Q. Xuan, Z.-Y. Zhang, C. Fu, H.-X. Hu, and V. Filkov, “Social synchrony on complex networks,” *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1420–1431, May 2018.
[21] Q. Xuan *et al.*, “Modern food foraging patterns: Geography and cuisine choices of restaurant patrons on yelp,” *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 508–517, Jun. 2018.
[22] C. Fu, H. Zhang, M. Zhan, and X. Wang, “Synchronous patterns in complex systems,” *Physical Rev. E*, vol. 85, no. 6, 2012, Art. no. 066208.
[23] A.-L. Barabási, *Network Science*. Cambridge, U.K.: Cambridge Univ. Press, 2016.
[24] Q. Xuan, A. Okano, P. Devanbu, and V. Filkov, “Focus-shifting patterns of oss developers and their congruence with call graphs,” in *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng.*, 2014, pp. 401–412.
[25] 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.
[26] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.
[27] J. Jiang and C. Zhai, “Instance weighting for domain adaptation in NLP,” in *Proc. 45th Annu. Meet. Assoc. Comput. Linguistics*, 2007, pp. 264–271.
[28] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 193–200.
[29] L. Duan, I. W. Tsang, and D. Xu, “Domain transfer multiple kernel learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.
[30] M. Long, J. Wang, G. Ding, S. J. Pan, and S. Y. Philip, “Adaptation regularization: A general framework for transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.
[31] W. Wang, Z. Chen, J. Liu, Q. Qi, and Z. Zhao, “User-based collaborative filtering on cross domain by tag transfer learning,” in *Proc. 1st Int. Workshop Cross Domain Knowl. Discovery Web Social Netw. Mining*, 2012, pp. 10–17.
[32] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, “Multisource domain adaptation and its application to early detection of fatigue,” *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 4, pp. 18:1–18:26, 2012.
[33] L. Duan, I. W. Tsang, and D. Xu, “Domain transfer multiple kernel learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.
[34] Y. Yao and G. Doretto, “Boosting for transfer learning with multiple sources,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1855–1862.
[35] T. Tommasi, F. Orabona, and B. Caputo, “Safety in numbers: Learning categories from few examples with multi model knowledge transfer,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 3081–3088.
[36] L. Mihalkova and R. J. Mooney, “Transfer learning by mapping with minimal target data,” in *Proc. AAAI Workshop Transfer Learn. Complex Tasks*, 2008, pp. 31–36.
[37] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, “Cross-domain co-extraction of sentiment and topic lexicons,” in *Proc. 50th Annu. Meet. Assoc. Comput. Linguistics, Long Papers-Vol. 1*, 2012, pp. 410–419.
[38] G. Schweikert, G. Rätsch, C. Widmer, and B. Schölkopf, “An empirical analysis of domain adaptation algorithms for genomic sequence analysis,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1433–1440.
[39] B. Tan, E. Zhong, E. W. Xiang, and Q. Yang, “Multi-transfer: Transfer learning with multiple views and multiple sources,” *Statistical Anal. Data Mining, ASA Data Sci. J.*, vol. 7, no. 4, pp. 282–293, 2014.
[40] R. Chattopadhyay, J. Ye, S. Panchanathan, W. Fan, and I. Davidson, “Multi-source domain adaptation and its application to early detection of fatigue,” in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 717–725.
[41] L. Yang, L. Jing, J. Yu, and M. K. Ng, “Learning transferred weights from co-occurrence data for heterogeneous transfer learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2187–2200, Nov. 2016.
[42] J. He, Y. Liu, and R. Lawrence, “Graph-based transfer learning,” in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 937–946.
[43] J. Tang, T. Lou, and J. Kleinberg, “Inferring social ties across heterogenous networks,” in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 743–752.
[44] M. Fang, J. Yin, and X. Zhu, “Transfer learning across networks for collective classification,” in *Proc. 13th Int. Conf. Data Mining*, 2013, pp. 161–170.
[45] X. Shen and F.-L. Chung, “Network embedding for cross-network node classification,” 2019, *arXiv:1901.07264*.
[46] M. Fang, J. Yin, X. Zhu, and C. Zhang, “Trgraph: Cross-network transfer learning via common signature subgraphs,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2536–2549, Sep. 2015.
[47] J. Ye, H. Cheng, Z. Zhu, and M. Chen, “Predicting positive and negative links in signed social networks by transfer learning,” in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1477–1488.
[48] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, “Transfer learning with graph co-regularization,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1805–1818, Jul. 2014.
[49] R. Niu, S. Moreno, and J. Neville, “Analyzing the transferability of collective inference models across networks,” in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2015, pp. 908–916.
[50] J. Tang, T. Lou, J. Kleinberg, and S. Wu, “Transfer learning to infer social ties across heterogeneous networks,” *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 7:1–7:43, 2016.
[51] G.-J. Qi, C. C. Aggarwal, and T. S. Huang, “Breaking the barrier to transferring link information across networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1741–1753, Jul. 2015.
[52] B. Sigurbjörnsson and R. Van Zwol, “Flickr tag recommendation based on collective knowledge,” in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 327–336.
[53] E. Zangerle, W. Gassler, and G. Specht, “Recommending #-tags in twitter,” in *Proc. Proc. Workshop Semantic Adaptive Social Web*, 2011, vol. 730, pp. 67–78.
[54] T. Trzciński and P. Rokita, “Predicting popularity of online videos using support vector regression,” *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2561–2570, Nov. 2017.
[55] A. Das, M. Roy, S. Dutta, S. Ghosh, and A. K. Das, “Predicting trends in the twitter social network: A machine learning approach,” in *Proc. Int. Conf. Swarm, Evol., Memetic Comput.*, Springer, 2014, pp. 570–581.
[56] T. Chakraborty, S. Kumar, P. Goyal, N. Ganguly, and A. Mukherjee, “Towards a stratified learning approach to predict future citation counts,” in *Proc. 14th ACM/IEEE-CS Joint Conf. Digit. Libraries*, 2014, pp. 351–360.
[57] C. Fu, Y. Zheng, S. Li, and Q. Xuan, “Predicting the popularity of tags in stackexchange qa communities,” in *Proc. Int. Workshop Complex Syst. Netw.*, 2017, pp. 90–95.
[58] N. Shervashidze, P. Schweitzer, E. J. V. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, “Weisfeiler-Lehman graph kernels,” *J. Mach. Learn. Res.*, vol. 12, no. 9, pp. 2539–2561, 2011.
[59] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, “Identifying influential nodes in complex networks,” *Physica A, Statistical Mech. Appl.*, vol. 391, no. 4, pp. 1777–1787, 2012.
[60] L. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, “Characterization of complex networks: A survey of measurements,” *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, 2007.
[61] S. Bhagat, G. Cormode, and S. Muthukrishnan, “Node classification in social networks,” in *Social Network Data Analytics*. Berlin, Germany: Springer, 2011, pp. 115–148.
[62] L. Lü and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A, Statistical Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.
[63] M. Waniek, T. P. Michalak, M. J. Wooldridge, and T. Rahwan, “Hiding individuals and communities in a social network,” *Nature Human Behav.*, vol. 2, no. 2, pp. 139–147, 2018.

[64] C. Shah and J. Pomerantz, “Evaluating and predicting answer quality in community qa,” in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 411–418.

[65] D. Correa and A. Sureka, “Fit or unfit: Analysis and prediction of ‘closed questions’ on stack overflow,” in *Proc. 1st ACM Conf. Online Social Netw.*, 2013, pp. 201–212.

[66] K. Lerman and T. Hogg, “Using a model of social dynamics to predict popularity of news,” in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 621–630.

[67] K. Fernandes, P. Vinagre, and P. Cortez, “A proactive intelligent decision support system for predicting the popularity of online news,” in *Proc. Portuguese Conf. Artif. Intell.*, Springer, 2015, pp. 535–546.

[68] H. Alharthi, D. Outioua, and O. Baysal, “Predicting questions’ scores on stack overflow,” in *Proc. 3rd Int. Workshop CrowdSourcing Softw. Eng.*, 2016, pp. 1–7.

[69] D. M. Powers, “Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[70] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[71] P. Erds and A. Rényi, “On the evolution of random graphs,” *Publications Math. Inst. Hung. Acad. Sci.*, vol. 5, pp. 17–61, 1960.

[72] S. Maslov and K. Sneppen, “Specificity and stability in topology of protein networks,” *Science*, vol. 296, no. 5569, pp. 910–913, 2002.

[73] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[74] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proc. 16th Int. Conf. Int. Conf. Mach. Learn.*, vol. 99, 1999, pp. 200–209.

[75] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, vol. 186. Norwell, MA, USA: Kluwer, 2002.

[76] H. Daume III and D. Marcu, “Domain adaptation for statistical classifiers,” *J. Artif. Intell. Res.*, vol. 26, pp. 101–126, 2006.

[77] H. Daume III, “Frustratingly easy domain adaptation,” in *Proc. 45th Annu. Meet. Assoc. Comput. Linguistics*, 2007, pp. 256–263.

[78] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” in *Proc. Eur. Conf. Comput. Learn. Theory*, Springer, 1995, pp. 23–37.

[79] R. Caruana and A. Niculescu, “An empirical comparison of supervised learning algorithms,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 161–168.

[80] V. Vapnik, “Transductive inference and semi-supervised learning,” in *Semi-Supervised Learning*, Cambridge, MA: MIT Press, pp. 454–472, 2006.

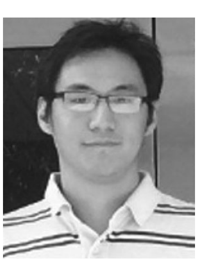
[81] T. A. Schieber, L. Carpi, A. Díaz-Guilera, P. M. Pardalos, C. Masoller, and M. G. Ravetti, “Quantification of network structural dissimilarities,” *Nature Commun.*, vol. 8, 2017, Art. no. 13928.

[82] A. Narayanan, M. Chandramohan, V. Rajasekar, L. Chen, Y. Liu, and S. Jaiswal, “Graph2vec: Learning distributed representations of graphs,” in *Proc. 13th Int. Workshop Mining Learn. Graphs*, 2017.

[83] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” *J. Complex Netw.*, vol. 2, no. 3, pp. 203–271, 2014.

[84] D. Y. Kenett, M. Perc, and S. Boccaletti, “Networks of networks–an introduction,” *Chaos, Solitons Fractals*, vol. 80, pp. 1–6, 2015.

[85] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, “Flavor network and the principles of food pairing,” *Scientific Rep.*, vol. 1, pp. 196:1–196:7, 2011.



Chenbo Fu received the B.S. degree in physics from the Zhejiang University of Technology, Hangzhou, China, in 2007, and the M.S. and Ph.D. degrees in physics from Zhejiang University, Hangzhou, China, in 2009 and 2013, respectively. He was a Postdoctoral Researcher with the College of Information Engineering, Zhejiang University of Technology, and was a Research Assistant with the Department of Computer Science, University of California at Davis, USA, in 2014. He is currently a Lecturer with the College of Information Engineering, Zhejiang University of Technology. His research interests include network-based algorithms design, social networks, data mining, chaos synchronization, network dynamics, and machine learning.

Yongli Zheng received the B.S. degree in electrical and information engineering from Zhejiang University of Technology, Hangzhou, China, in 2016. He is currently working toward the M.S. degree in control theory and engineering at the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests include social network analysis and machine learning.

Yi Liu (M’11) received the Ph.D. degree in control theory and engineering from Zhejiang University, Hangzhou, China in 2009. He was a Postdoctoral Researcher with the Department of Chemical Engineering, Chung-Yuan Christian University, Taiwan, from February 2012 to June 2013. He is currently an Associate Professor with the Zhejiang University of Technology, Hangzhou, China. He has authored and coauthored more than 30 international journal papers. His research interests include data intelligence with applications to modeling, control and optimization of industrial processes.

Qi Xuan (M’18) received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively. He was a Postdoctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University, from 2008 to 2010, and a Research Assistant with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, in 2010 and 2017. From 2012 to 2014, he was a Postdoctoral Fellow with the Department of Computer Science, University of California at Davis, CA, USA. He is currently a Professor with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou. His current research interests include network science, graph data mining, cyberspace security, machine learning, and computer vision.

Guanrong Chen (M’89–SM’92–F’97) received the M.Sc. degree in computer science from Sun Yatsen University, Guangzhou, China, in 1981, and the Ph.D. degree in applied mathematics from Texas A&M University, College Station, TX, USA, in 1987. Since 2000, He has been a Chair Professor and the Founding Director with the Centre for Chaos and Complex Networks, the City University of Hong Kong. Prior to that, he was a tenured full Professor with the University of Houston, Texas, USA. He was awarded the 2011 Euler Gold Medal, Russia, and conferred a Honorary Doctorate by the Saint Petersburg State University, Russia, in 2011 and by the University of Le Havre, Normandy, France, in 2014. He is a member of the Academy of Europe and a fellow of The World Academy of Sciences, and is a Highly Cited Researcher in Engineering as well as in Mathematics according to Thomson Reuters. He is a life fellow of the IEEE.