Guardian: Evaluating Trust in Online Social Networks with Graph Convolutional Networks

Wanyu Lin, Zhaolin Gao, Baochun Li
University of Toronto, {wylin, bli}@ece.utoronto.ca, zhaolin.gao@mail.utoronto.ca

***Abstract*—In modern online social networks, each user is typically able to provide a value to indicate how trustworthy their direct friends are. Inferring such a value of social trust between any pair of nodes in online social networks is useful in a wide variety of applications, such as online marketing and recommendation systems. However, it is challenging to accurately and efficiently evaluate social trust between a pair of users in online social networks. Existing works either designed handcrafted rules that rely on specialized domain knowledge, or required a significant amount of computation resources, which affected their scalability.**

In recent years, graph convolutional neural networks (GCNs) have been shown to be powerful in learning on graph data. Their advantages provide great potential to trust evaluation as social trust can be represented as graph data. In this paper, we propose *Guardian*, a new end-to-end framework that learns latent factors in social trust with GCNs. *Guardian* is designed to incorporate social network structures and trust relationships to estimate social trust between any two users. Extensive experimental results demonstrated that *Guardian* can speedup trust evaluation by up to $2,827\times$ with comparable accuracy, as compared to the state-of-the-art in the literature.

I. Introduction

Online social networks, such as Facebook, have become a norm in our social and personal lives. Users routinely share their opinions and life experiences in these social networks. The explicit or implicit social relationships established on these online social networks can be leveraged to market products and services, or to make recommendations.

However, the inherent nature of online social networks provides a favorable environment for malicious users to spread incorrect information, either for financial gains [1] or to increase social influence [2]. Therefore, *social trust* has become an important concern in online social networks. In particular, it is helpful to evaluate the pairwise trust relationship between two users who are not directly connected within online social networks. Such estimates of trustworthiness help indicate to what extent a user could expect someone else to perform given actions [3].

With the popularity of online social networks and the importance of social trust, an extensive amount of work on evaluating pairwise social trust has been reported in the literature [3]–[8]. For example, OpinionWalk [7] evaluated trust relationships by performing path searches throughout the network. Discounting and combining operators are designed to model trust propagation and aggregation along network paths. These hand-crafted rules rely heavily upon the knowledge of domain experts, and may be difficult to be generalized to different domains. Liu *et al.* [8] proposed NeuralWalk, a framework based on neural networks, to learn trust propagation and aggregation rules with machine learning techniques. However, it required a significant amount of computation resources for its matrix operations, which is not scalable to real-world online social networks.

Existing trust evaluation approaches were designed based on the propagative and composable nature of social trust in online social networks. In particular, *the propagative nature of social trust* refers to the fact that trust may be passed from one user to another, creating chains of social trust that connects two users who are not explicitly connected [9]. The *composable nature of social trust* refers to the fact that trust needs to be aggregated if several chains of social trust exist [9]. In a nutshell, trust propagation and aggregation rules are the keys to effectively evaluate pairwise social trust in online social networks.

In recent years, we have witnessed encouraging developments in deep graph convolutional neural networks (GCNs) for graph-structured data [10]–[12]. With graph convolutional neural networks, feature information from local graph neighborhoods is iteratively aggregated. By stacking multiple convolutions and transformations, local information can be propagated throughout the entire graph. In online social networks, social trust can be similarly represented as graph data, including both social network structures and associated trust relationships between users. Thus, given their advantages, excellent opportunities may exist in use the GCNs to capture trust propagation and aggregation rules for evaluating social trust relationships between pairs of users.

Yet, evaluating social trust using graph convolutional neural networks is quite challenging. Online social networks not only contain the social graph structure (social connections between users) but also include pairwise social trust relationships. In this context, the first challenge is how social connections and associated trust relationships can be represented jointly so that the propagative nature and composable nature of social trust are able to be captured simultaneously. In addition, social trust is typically asymmetric; one user may trust someone else more than she is trusted back. Therefore, the second challenge is how to characterize such an asymmetric property in social trust.

In this paper, we propose to address these challenges in social trust evaluation based on graph convolutional neural net-

This research was supported in part by the NSERC Discovery Research Program.

works. More specifically, we aim to effectively and efficiently estimate the value of trustworthiness between any two users who are not explicitly connected, given the social network structure and associated trust relationships between users. For this purpose, we propose an end-to-end framework that stacks multiple trust convolutional layers, which is designed to discover hidden and predictive latent factors of trust in online social networks.

The key component of our proposed framework is the trust convolutional layer, which employs the notion of localized graph convolutions [10]. It is designed to capture the propagative nature and composable nature of social trust. The parameters to be learned in each layer are shared across all users, making the parameter complexity of our proposed framework independent of the size of the input network graph. In particular, in order to capture the asymmetric property of social trust, each of our trust convolutional layers consists of two components: popularity trust propagation and engagement trust propagation. The former is used to learn the extent that a user is trusted by the others, while the latter is for capturing the willingness that a user trusts the others. Finally, by stacking a fully-connected layer, *Guardian* is able to explicitly represent both popularity trust and engagement trust of individual users in a collaborative manner. As such, effective pairwise trust relationships can be established.

Highlights of our original contributions are as follows. *First*, we introduce a principled methodology to jointly capture both social connections and associated trust relationships of the users within online social networks. *Second*, we propose a new approach to jointly characterize the popularity trust and engagement trust of users so that the asymmetric property of the social trust can be captured implicitly. *Third*, we demonstrate the effectiveness and efficiency of our proposed framework using two online social networks from different domains — Advogato and Pretty Good Privacy. Our extensive array of experiments on benchmarking datasets demonstrated that *Guardian* can speedup trust evaluation by up to $2,827\times$ with comparable accuracy as compared to NeuralWalk [8], and increase accuracy by up to 18.8% and 19.8% compared with Matri [3] and OpinionWalk [7], respectively.

II. Problem Setup

Throughout this paper, we consider a social trust evaluation problem in an online social network, which is modeled as a directed graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where any vertex $u, v \in \mathcal{V}$ represent users, and $e_{u\to v} \in \mathcal{E}$ denotes the observed trust relationships. $w_{u\to v}$ measures the trustworthiness of the trustor-trustee pair $\langle u, v\rangle$, where the trustworthiness domain is typically application-specific. For example, in Epinion[1], $w \in \{\text{Trust, Distrust}\}$, while in Advogato[2] and in Pretty-Good-Privacy[3] (PGP), $w \in \{\text{Observer, Apprentice, Journeyer, Master}\}$. Let $\mathcal{W} = \{\langle u, v\rangle, w_{u\to v} | e_{u\to v} \in \mathcal{E}\}$ be the set of observed trust relationships in the given online social graph. $\tilde{\mathcal{W}} = \{\langle u, v\rangle, \tilde{w}_{u\to v} | \tilde{e}_{u\to v} \notin \mathcal{E}\}$ denotes the set of unobserved/missing trust relationships that are to be evaluated.

TABLE I
Notations

Notation	Descriptions
$w_{u\to v}$	the trustworthiness of v from the perspective of u
$\mathcal{W}$	the set of observed pairwise trust relationships
$\lvert w\rvert$	the number of trustworthiness types
$N_O(u)$	the set of observed trustees whom u endorses her trust on (out-neighbors of u)
$N_I(u)$	the set of observed trustors who endorse trust on u (in-neighbors of u)
$x[u]$	initial embedding of user u
D_e	the dimension of initial embedding vector
pTr, eTr	the popularity trust and the engagement trust
$h_I[u]$	the latent factor of the popularity trust from in-neighbors $N_I(u)$ of user u
$h_O[u]$	the latent factor of the engagement trust from out-neighbors $N_O(u)$ of user u
$h[u]$	the trust latent factor of user u
$\tilde{h}_{u\to v}$	the pairwise trustworthiness latent factor
$\tilde{w}_{u\to v}$	predicted trust relationship
$\otimes$	the concatenation operator of two vectors
$\oplus$	the mean aggregator
σ	non-linear activation functions, e.g., $\tanh(\cdot)$, $\text{softmax}(\cdot)$
W, b	the model parameters (weight matrices and bias) in *Guardian*

Notably, as in most existing online social networks, trustworthiness is represented by categorical values. In this context, the social trust evaluation problem is equivalent to a social trust prediction problem. We can define $|w|$ to be the total number of types of trustworthiness, which is application-specific. For example, in PGP or Advogato, $|w| = 4$.

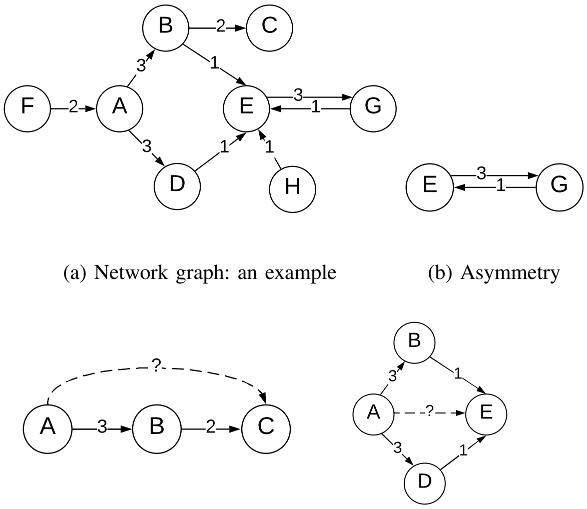
Before we formulate the problem of pairwise social trust evaluation, we introduce some important notations and necessary properties of social trust to facilitate a better understanding of the problem and our solution. For any user $u \in \mathcal{V}$, let $N_O(u)$ be the set of observed trustees whom u endorses her trust on (out-neighbors of u), $N_I(u)$ be the set of observed trustors who endorse trust on u (in-neighbors of u). In this sense, we can define $|N_I(u)|$ and $|N_O(u)|$ to represent in-degree and out-degree of u, respectively. The mathematical notations used in this paper are summarized in Table I.

In the literature [13], widely used trust properties include the propagative nature, composable nature, and asymmetric property. For the sake of clarity, we use an example to illustrate the properties of social trust, which will also be used throughout this paper. Fig. 1a shows a social network graph, where nodes represent users, directional edges denote trust relationships of the trustor-trustee pairs, and the numbers are the associated trustworthiness (0 for the lowest trustworthiness and 3 for the highest trustworthiness).

Fig. 1c illustrates the propagative nature of social trust. Since user A trusts user B with a trust value of 3 and user B trusts user C with 2, user A trusts user C with 2. In this

[1] https://snap.stanford.edu/data/soc-Epinions1.html
[2] http://www.trustlet.org/datasets/advogato/
[3] http://networkrepository.com/arenas_pgp.php

example, A → B → C forms a trust chain for A → C. To establish a trust relationship for A → E as shown in Fig. 1d, there exist two trust chains that need to be aggregated. Because both the trust value for B → E and D → E are 1, it is unlikely for A → E to achieve a high trust value. The asymmetric property of the social trust can also be illustrated in Fig. 1b: the trustworthiness of user G from the perspective of user E (E → G) is different from that of E from G (G → E), even though they endorse trust explicitly to each other.



(a) Network graph: an example (b) Asymmetry

(c) Propagative nature (d) Composable nature

Fig. 1. The property illustrations of social trust: an example.

With the aforementioned notations and definitions, we can now formally define the problem of *social trust evaluation* (or *prediction*). Given an online social network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, the social trust evaluation problem aims to evaluate (or *predict*) the trustworthiness of the trustor-trustee pair $\tilde{w}_{u \rightarrow v}$, where $u, v \in \mathcal{V}$, $u \neq v$ and $\tilde{e}_{u \rightarrow v} \notin \mathcal{E}$.

III. *Guardian*: PROPOSED FRAMEWORK

We now present *Guardian*, our proposed framework for social trust evaluation, the architecture of which is illustrated in Fig. 2. There are three components in the framework: (1) an embedding layer that offers an initialization of user embeddings; 2) multiple trust convolutional layers that refine the popularity trust embedding and engagement trust embedding by injecting high-order social trust relationships; and 3) a prediction layer that consists of a fully-connected layer followed by a softmax function. It first transforms the latent representations of users into the latent factor of trust, and then outputs the probability of the prediction. In what follows, we first conceptually discuss the efficiency and effectiveness of our proposed framework, and then discuss more influential factors for social trust evaluation and limitations of our framework.

A. *Embedding Layer*

With the recent emergence of representation learning, the network embedding technique has been extensively studied to discover and encode network structural properties into a low-dimensional latent space. More formally, network embedding learns a representation vector $x[u] \in \mathcal{R}^{D_e \times 1}$ for each user u in the network graph $\mathcal{G}$. In *Guardian*, we use a pre-trained embedding layer to map each user into a D_e-dimensional representation. It is worth noting that these representations serve as an initial state for user embeddings, to be optimized in an end-to-end fashion. In *Guardian*, we refine the embeddings by propagating them along the online social network graph. With a specially designed transformation layer, these refined user embeddings can be transformed into pairwise trustor-trustee embeddings for trustworthiness prediction.

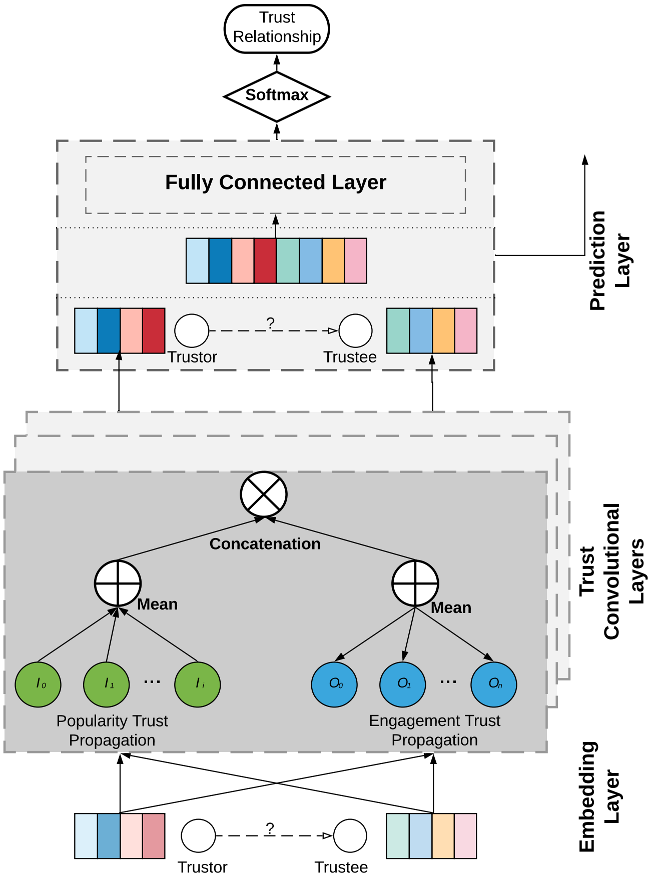


Fig. 2. Illustration of *Guardian* framework.

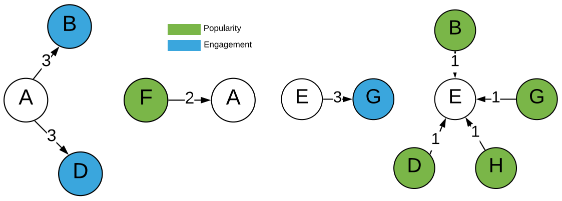
B. *Trust Convolutional Layers*

As an online social network graph contains not only social connections between users but also trust interactions between the trustors and the trustees, we provide a principled approach to jointly capture the social connections and associated trust relationships for learning the embeddings $h[u]$ of the users. In particular, due to the asymmetric property of social trust, a user can assume different roles, either as a trustor or a trustee. To be able to capture the asymmetric property of social trust, we first separate pairwise trust interactions into two groups: *popularity interactions* and *engagement interactions*.

Popularity-based interactions refer to the trustworthiness of a user as observed by the others. In this sense, the more a user is trusted by the others, the more popularity-based trust this user gains. Similarly, engagement-based interactions refer to the trustworthiness of the others from a user's perspective. The popularity trust indicates the extent that a user is trusted by the others, while the engagement trust reveals the willingness that a user trusts the others.

In what follows, we consider two types of trust aggregation to characterize the popularity trust and engagement trust, represented as $h_I[u]$ and $h_O[u]$, respectively. For each of them, we use mean-aggregator to aggregate its associated trust interactions with its neighbors. It is worth mentioning that, mean-aggregator is the main operation of aggregating information from local graph neighborhoods [10], [11].

Let's see an example in our example social network graph, originally shown in Fig. 1a. With our trust model, the popularity interactions of user A and E are depicted in green in Fig. 3, while the engagement interactions are shown in blue. More specifically, for user E, there are four incoming neighbors, all of which have a trust value of 1. The popularity trust of E is, therefore, 1 by averaging over its incoming trust relationships, and the engagement trust of E is 3. Similarly, the popularity trust of A is 2, as there is only one incoming neighbor with trust value 2, while its engagement trust is 3, averaging over its two outgoing trust relationships.



(a) (b)

Fig. 3. The popularity trust and the engagement trust: an example.

Popularity Trust Propagation (pTr). Intuitively, the incoming social connections and associated trust relationships provide direct evidence on the popularity trust of a user in online social networks. We build upon this basis to propagate the popularity trust between connected users.

In particular, to model categorical trustworthiness, we first use one-hot encoding to represent each type of trustworthiness. Taking the Advogato dataset as an example, for trustworthiness $w_{u\rightarrow v} \in \{\text{Observer, Apprentice, Journeyer, Master}\}$, we model them as the following one-hot representations: $[0,0,0,1]^T$, $[0,0,1,0]^T$, $[0,1,0,0]^T$, and $[1,0,0,0]^T$. Then *Guardian* employs a linear transformation to convert the one-hot encodings into dense vector embeddings through Eq. (1) and Eq. (4). For a trust relationship with trustworthiness $w_{u\leftarrow v}$ (u is being a trustee in this trust relationship), we model the popularity trust of u as observed by v with a combination of v's embedding $x[v]$ and the embedding of associated trustworthiness $e_{w_{u\leftarrow v}}$.

$$e_{w_{u\leftarrow v}} = W_{u\leftarrow v} \cdot w_{u\leftarrow v} \tag{1}$$

$$\text{pTr}_{u\leftarrow v} = x[v] \otimes e_{w_{u\leftarrow v}} \tag{2}$$

where $W_{u\leftarrow v} \in \mathcal{R}^{D_e \times D_{|w|}}$ is a trainable transformation matrix, $\otimes$ denotes the concatenation operation between two vectors, and $|w|$ denotes the number of trustworthiness types.

We now take the element-wise mean of the vectors in $\{\text{pTr}_{u\leftarrow v}, \forall v \in N_I(u)\}$. This mean-based aggregator is a linear approximation of a localized spectral convolution [11], as the following function:

$$h_I[u] = \frac{1}{N_I(u)} \cdot \sum_{v\in N_I(u)} \text{pTr}_{u\leftarrow v} \tag{3}$$

Engagement Trust Propagation (eTr). Accordingly, we characterize the engagement trust of a user through its outgoing social connections and associated trust relationships. We build upon this basis to perform the propagation and aggregation of engagement trust between the connected users. Thus, the engagement trust of user u can be captured by the following functions:

$$e_{w_{u\rightarrow v}} = W_{u\rightarrow v} \cdot w_{u\rightarrow v} \tag{4}$$

$$\text{eTr}_{u\rightarrow v} = x[v] \otimes e_{w_{u\rightarrow v}} \tag{5}$$

$$h_O[u] = \frac{1}{N_O(u)} \cdot \sum_{v\in N_O(u)} \text{eTr}_{u\rightarrow v} \tag{6}$$

where $\text{eTr}_{u\rightarrow v}$ denotes the involvement trust of the user u to user v in online social networks.

Learning Trust Latent Factors of Users. In order to learn better latent factors of users for downstream trustworthiness prediction, the popularity trust and engagement trust are needed to be considered jointly. Here, we propose to combine these two types of trust through a standard fully connected (FC) layer, where $h_I[u]$ and $h_O[u]$ are concatenated before feeding into the FC. Formally, the latent factor of user u, $h[u]$, can be characterized as follows:

$$h[u] = \sigma(W \cdot (h_I[u] \otimes h_O[u]) + b) \tag{7}$$

where W is a trainable transformation matrix, b is a learnable bias, and σ denotes the non-linear activation function.

Higher-order Trust Propagation. By stacking l trust convolutional layers, a user is capable of receiving the social trust (the popularity trust and engagement trust) propagated from its l-hop neighbors. In the l-th step, the representation of user u is recursively formulated as Eq. (8) - Eq. (12):

$$\text{pTr}^l_{u\leftarrow v} = h^{l-1}[v] \otimes \{W^l_{u\leftarrow v} \cdot w_{u\leftarrow v}\} \tag{8}$$

$$\text{eTr}^l_{u\rightarrow v} = h^{l-1}[v] \otimes \{W^l_{u\rightarrow v} \cdot w_{u\rightarrow v}\} \tag{9}$$

$$h^l_I[u] = \frac{1}{N_I(u)} \cdot \sum_{v\in N_I(u)} \text{pTr}^l_{u\leftarrow v} \tag{10}$$

$$h_O^l[u] = \frac{1}{N_O(u)} \cdot \sum_{v \in N_O(u)} \mathrm{eTr}_{u \to v}^l \tag{11}$$

$$h^l[u] = \sigma(W^l \cdot (h_I^l[u] \otimes h_O^l[u]) + b^l) \tag{12}$$

where $h^0[u] = x[u]$ is the pre-trained embedding of user u obtained in the embedding layer, $w_{u \to v}$ and $w_{u \leftarrow v}$ are the observed trust relationships, and $W_{u \leftarrow v}^l$, $W_{u \to v}^l$, W^l, and b^l are the model trainable parameters, to be optimized in an end-to-end fashion with *Guardian*. Note that, by stacking multiple trust convolutional layers, we not only enrich the initial user embedding with its propagated popularity trust and engagement trust in online social networks, but also allow controlling the range of trust propagation by adjusting l.

C. Prediction Layer

In order to learn the latent factor of trust relationship, we first concatenate the latent embeddings of the trustor and the trustee, and then fit them to a standard fully-connected (FC) layer followed by a softmax layer. Formally, the latent representation of the trustor-trustee pair is formulated as Eq. (13), where W_{fc} is a trainable weight matrix defined in the FC layer, and σ is the softmax function, defined as $\mathrm{softmax}(x_i) = \frac{\exp(x_i)}{Z}$ with $Z = \sum_i \exp(x_i)$.

$$\tilde{h}_{u \to v} = \sigma(W_{fc} \cdot (h[u] \otimes h[v])) \tag{13}$$

The advantage of using concatenation lies in its simplicity and expressiveness, which have been shown in a recent work of graph convolutional neural networks [10]. In addition, the fully connected layer leads to a more effective representation of a trust relationship for prediction, as this step explicitly injects the popularity trust and the engagement trust of individual users in a collaborative fashion. The outcome of this step is the probabilistic prediction values of the trustworthiness. As a consequence, the trustworthiness of user v from the perspective of user u is computed as $\tilde{w}_{u \to v} = \underset{j}{\mathrm{argmax}}(\tilde{h}_{u \to v})$. Note that $\tilde{w}_{u \to v} \neq \tilde{w}_{v \to u}$, due to the asymmetric property of social trust in online social networks. The detailed forward propagation algorithm of *Guardian* is shown as **procedure Guardian**.

D. Model Training

To learn the model parameters in *Guardian*, we define the objective function as the cross-entropy loss between the predicted values and the ground-truth trustworthiness from the observed set $\mathcal{W}$. Formally, it is formulated as:

$$\mathcal{L} = -\frac{1}{|\mathcal{W}|} \sum_{(\langle u,v \rangle, w_{u \to v}) \in \mathcal{W}} \log \tilde{h}_{u \to v, w_{u \to v}} + \lambda \cdot ||\Theta||_2^2 \tag{14}$$

where $\mathcal{W} = \{\langle u, v \rangle, w_{u \to v}\}$ is the set of observed trustor-trustee pairs and associated trust relationships, $\Theta = \{\{W_{u \leftarrow v}^l, W_{u \to v}^l, W^l, b^l\}_{l=1}^L, W_{fc}\}$ denotes all trainable model parameters, and λ controls the L_2 regularization strength to prevent over-fitting. In particular, we adopt Adam [14] as the optimizer in our implementation, as it has been shown to be effective in updating the model parameters [10].

```
procedure Guardian: TRUST RELATIONSHIP PREDICTION (I.E, FORWARD PROPAGATION)
    Generate initial states of user embeddings for $\mathcal{G}$
    $h^0[u] \leftarrow x[u]$, for all $u \in \mathcal{V}$
                                        ▷ Trust latent factors of observed users
    for all $u \in \mathcal{V}$ do
        for $l = 1 \cdots L$ do
                                        ▷ Popularity Trust
            $h_I^l[u] = \frac{1}{N_I(u)} \cdot \sum_{i \in N_I(u)} \mathrm{pTr}_{u \leftarrow i}^l$
                                        ▷ Engagement Trust
            $h_O^l[u] = \frac{1}{N_O(u)} \cdot \sum_{i \in N_O(u)} \mathrm{eTr}_{u \to i}^l$
            $h^l[u] = \sigma(W^l \cdot (h_I^l[u] \otimes h_O^l[u]) + b^l)$
                                        ▷ Trust relationship prediction vector
    for all $\langle u, v \rangle \in \mathcal{W}$ do
        $h[u] \leftarrow h^L[u]$
        $h[v] \leftarrow h^L[v]$
        $\tilde{h}_{u \to v} = \sigma(W_{fc} \cdot (h[u] \otimes h[v]))$
```

E. Analysis and Discussions

Different from the state-of-the-art trust evaluation solutions in the literature [7], [8], our framework does not have any assumptions on the existence of paths between the trustor and the trustee while we compute the pairwise trustworthiness values. This reflects the real-world situation where some of the users are new in the society and may not have any social connections with the other users. However, these newly added users are still able to trust the existing users who have a significant popularity trust (e.g., the authenticated/official users) to some extent. Surprisingly, our proposed framework can still achieve the best prediction accuracy even if we do not make any assumptions, which, as shown in Sec. IV, can be empirically verified later.

The key computational operations of our framework are the notion of localized graph convolutions [10]. To be able to implicitly capture the asymmetric property of social trust, each trust convolutional layer learns how to aggregate the popularity trust and engagement trust of users from a small graph neighborhood in the social graph. By applying multiple trust convolutional layers that aggregate the trust information from the local neighborhood of users, our approach can obtain the popularity trust and engagement trust of users from their local network topology.

It is worth mentioning that parameters of our proposed trust convolutional layers are shared across all users, making the parameter complexity of *Guardian* independent of the input graph size. Sec. IV empirically verified the efficiency and scalability of our framework. In addition, as *Guardian* is an inductive learning model, it is able to estimate the pairwise trustworthiness for users that were not seen during the training phase. In other words, it does not require any retraining

TABLE II
STATISTICAL DESCRIPTION OF ADVOGATO AND PGP DATASETS.

DATASET	# OF NODES	# OF EDGES	AVG. DEGREE	DIAMETER
ADVOGATO	$6,541$	$51,127$	19.2	4.82
PGP	$38,546$	$317,979$	16.5	7.7

process as the pre-trained parameters can be used for inference for the unseen users.

Incorporating context-aware features. Except for the social network graph and associated trust interactions, context is also an important influential factor for social trust evaluation [15]. In different contexts, trust relationships are typically different. For example, user A trusts user B for movie recommendations, while A may not trust B for restaurant recommendations. Movies and restaurants here represent different contexts. Therefore, it is crucial to distinguish between the different contexts of trust. Our framework can be readily extended to incorporate such context-aware features to further improve prediction accuracy, e.g, concatenating context features and graph structure embedding as the initial representation of a user.

Limitations. One important property of the social trust is that it is dynamic. More precisely, social trust can increase or decrease with new interactions and observations. It may also decay with time. A more recent interaction or observation may be more important than those that have happened earlier. It is intriguing to find out how our proposed framework responds to dynamics in social trust relationships, which will be left as our future work.

IV. EXPERIMENTAL RESULTS

A. Description of Datasets Used

In our experiments, we choose two widely used, real-world and benchmarking datasets for performance comparisons of different trust evaluation models [8]. The first dataset is Advogato, which is an online social network for open source developers. To allow users to certify each other, this network provides four different levels of trustworthiness. More specifically, the types of trustworthiness are {Observer, Apprentice, Journeyer, Master}.

The second dataset is Pretty-Good-Privacy (PGP), an encryption program that provides cryptographic privacy and authentication for data communication by adopting the concept of "web of trust." Similarly, the web of trust in PGP dataset contains four different levels of trustworthiness. The statistics of these two datasets are presented in Table II.

B. Experimental Settings

Baselines for comparisons. To demonstrate the effectiveness, we compared *Guardian*, our proposed framework against three groups of methods including traditional walk-based approach, matrix factorization-based approach, and deep neural network-based approach. For each group, we selected a representative baseline and below we will detail them. All experiments run 20 times to ensure statistical significance.

OpinionWalk [7]: This approach modeled the pairwise trustworthiness using statistical distributions in three-valued subjective logic. In order to establish a trust relationship between two indirectly connected users, it walked throughout the network in a breadth-first search manner. In particular, trust propagation and aggregation along the social paths were modeled with its predefined discounting and combining operators.

Matri [3]: This methodology was proposed to combine trust tendency and trust propagation under a collective matrix factorization framework. Under this framework, the trustor and trustee are mapped into a joint latent space. The trustworthiness of each trustor-trustee pair is modeled as the similarity (measured by the inner product of two vectors) between the latent vector of the trustor and the latent factor of the trustee in the learned latent space.

NeuralWalk [8]: This model was the state-of-the-art trust evaluation solution in the literature, in terms of its prediction accuracy. Its core is to learn single-hop trust propagation and aggregation rules with a neural network architecture, WalkNet. By iteratively executing the learning process of WalkNet multiple times, NeuralWalk is able to evaluate multi-hop social trust within online social networks.

Evaluation metrics. In order to evaluate the effectiveness of our proposed framework, two popular metrics were adopted to evaluate the prediction accuracy, including F1-score and Mean Absolute Error (MAE). All results are reported based on the results of 20 runs. Note that, larger values of F1-score, smaller values of MAE indicate better prediction accuracy. A small improvement in these evaluation metrics implies a significant influence on the quality of prediction. For efficiency and scalability, we used the average wall-clock time over 20 runs.

All the experiments were performed on a machine with Intel Core i7-9700K 8-core 3.6GHz CPU, 32GB RAM, 500GB SSD, and GeForce GTX 1660 Ti GPU.

Data preprocessing. We followed the data preprocessing as reported in NeuralWalk [8]. Specifically, as OpinionWalk is deductive, there is no need to separate the datasets for training and inference. Instead, we randomly selected $1,000$ trustor-trustee pairs for each dataset to statistically compare OpinionWalk with our framework. As for Matri, NeuralWalk, and *Guardian*, we randomly split each dataset into two portions: 80% of the trustor-trustee pairs to constitute the training set, and the remaining 20% as the test set. More precisely, the 20% of trustor-trustee pairs were removed from the network graph to compose the training set.

OpinionWalk [7] and Matri [3] mapped four trustworthiness levels into scalar values, aka {Observer: 0.1, Apprentice: 0.4, Journeyer: 0.7, Master: 0.9}, and they used MAE as their performance metric. Similarly, to be comparable, we did the same mapping for NeuralWalk and *Guardian* (both are categorical classifiers) to obtain the model MAE. Regarding F1-score, the outputs of OpinionWalk and Matri were rounded

TABLE III
PREDICTION ACCURACY ON ADVOGATO

APPROACHES	F1-SCORE	MAE
Guardian	**74.3%**	0.082
NEURALWALK	74.0%	0.081
OPINIONWALK	64.3%	0.228
MATRI	65.6%	0.127

TABLE IV
PREDICTION ACCURACY ON PGP

APPROACHES	F1-SCORE	MAE
Guardian	**87.1%**	**0.083**
NEURALWALK	–	–
OPINIONWALK	67.3%	0.249
MATRI	68.3%	0.122

to the nearest categorical values, aka {Observer: 0, Apprentice: 1, Journeyer: 2, Master: 3}.

As illustrated in Sec. III, to model categorical trustworthiness, we used one-hot encoding to represent each type of trustworthiness. As the benchmark datasets we used all contain four different types of trustworthiness, we transformed {Observer, Apprentice, Journeyer, Master} as following one-hot representations: $\{[0,0,0,1]^T$, $[0,0,1,0]^T$, $[0,1,0,0]^T$, and $[1,0,0,0]^T\}$. Note that, our framework can be readily generalized to any application domains containing an arbitrary number of trustworthiness levels.

Parameter settings. We implemented our proposed framework in Pytorch[4]. node2vec [16] was used to generate the initial embeddings for each user[5]. The embedding dimension was fixed to 128 for all datasets. In terms of hyperparameters, we applied a grid search for hyperparameters: the learning rate was tuned amongst $\{0.001, 0.005, 0.01, 0.05\}$, the coefficient of L_2 normalization was searched in $\{10^{-5}, 10^{-4}\}$, and the dropout ratio was in $\{0.0, 0.1, \ldots, 0.8\}$. We used the Xavier initializer [17] to initialize the model parameters. In addition, early stopping strategy was performed, *i.e.*, premature stopping if training loss does not increase for 10 successive epochs. Without specification, we report the results of three trust propagation layers $[32, 64, 32]$, learning rate of 0.01, dropout ratio of 0.0 and normalization coefficient of 10^{-5}. The detailed parameter settings for OpinionWalk, NeuralWalk, and Matri refer to [3], [7], [8], respectively.

C. Performance Comparisons

Effectiveness. The Advogato dataset is used to evaluate the performance of different approaches. The results are reported in Table III. *Guardian* offers the best F1-score with 0.3% improvement on NeuralWalk — the state-of-the-art solution — and even higher improvement on Matri, about 8.7%. As F1-score is scaled between 0 and 1, the increases in performance are significant. In terms of MAE, NeuralWalk and *Guardian* achieved approximately the same prediction accuracy, which implies the powerful learning capability of machine learning techniques.

To test that *Guardian* does not rely on datasets, we also evaluated our framework on PGP. We were not able to report the performance of NeuralWalk on PGP, as it ran out of the memory after one out of three iterations on our machine. As shown in Table IV, *Guardian* consistently offers the best F1-score by increasing the accuracy 18.8% for Matri and 19.8% for OpinionWalk. The results reported successfully verify that our proposed trust convolutional layers are able to characterize the trust latent factors of users to establish effective social trust.

Matri was not able to offer comparable performance on two datasets, which indicates that either the collected matrix for factorization or the inner product in the learned latent space of users may not be sufficient to capture the complex relations among the trustors and trustees. We also observed that OpinionWalk achieved the worst performance on both datasets, which shows that the path-search manner or the predefined trust propagation and aggregation rules may not be effective to provide accurate estimations.

Efficiency. For efficiency comparisons, we evaluated different approaches on the same machine as listed above. Because OpinionWalk is a deductive method - evaluating one trustor-trustee pair at a time, it does not generate any model parameters/user latent space for new trustor-trustee pair evaluation. In other words, the time for trust evaluation increases linearly with the number of trustor-trustee pairs to be evaluated. As such, we report the average runtime for evaluating $1,000$ trustor-trustee pairs in Fig. 4 but exclude this method from the following discussions.

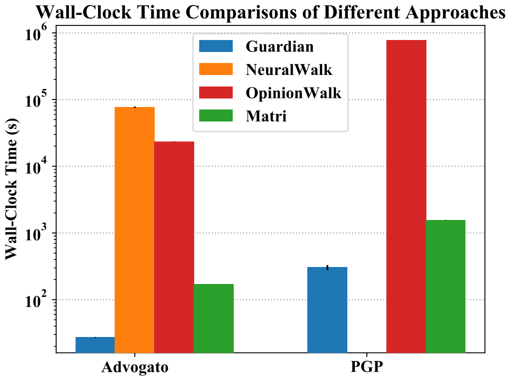


Fig. 4. Wall-clock time on Advogato and PGP[6].

[4] https://pytorch.org

[5] As the benchmarking datasets do not contain context information, we do not consider context-aware feature in our experiments.

TABLE V
TRAINING AND INFERENCE TIME (ON THE FULL TEST SET)

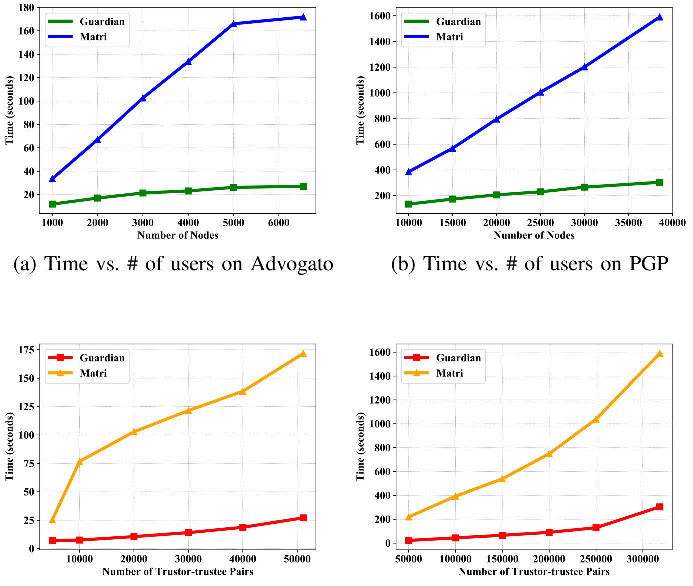
PROCESS	APPROACH	ADVOGATO	PGP
TRAINING TIME (S)	*Guardian*	**28.580**	**304.370**
	MATRI	176.395	$1,593.285$
INFERENCE TIME (S)	*Guardian*	0.102	0.583
	MATRI	0.008	0.044

We compared the total runtime of three approaches (Matri, NeuralWalk and *Guardian*) and the results are also shown in Fig. 4. It is worth noting that *Guardian* consistently outperforms all other baselines on all datasets. In particular, *Guardian* shortens the processing time significantly on NeuralWalk by $2,827\times$. Note that, when we run NeuralWalk on PGP, running one of three iterations has already cost us around 52 hours before it ran out of memory. Comparing to Matri, *Guardian* is $6.17\times$ and $5.23\times$ faster on Advogato and PGP respectively. It demonstrates that our proposed trust convolutional layer greatly speeds up trust evaluation process in online social networks and shows its promising that can be applied to large-scale network applications.

To enhance the understanding of the time cost for training[7]and inference respectively, we measured the time used for these two processes separately on both datasets. Table V summarizes the training and inference time for different approaches. As reported, *Guardian* is $6.17\times$ faster than Matri on the training phase, which shows the total time cost of Matri mainly comes from its matrix factorization phase. We also noticed that our framework bears longer inference time as compared with Matri. However, Matri can not evaluate the trust relationship for users that were not seen during the learning phase, while our proposed *Guardian* is an inductive model that can be generalized to unseen users. Therefore, a retraining of the dataset is needed for newly added users for Matri. *Guardian*, on the other hand, does not require retraining because the pre-trained parameters can be saved for later inference.

Scalability. The scalability of *Guardian* is evaluated by measuring the wall-clock times with a different number of users and a different number of trustor-trustee pairs, respectively. Both of the selected users and pairs are subgraphs from the main graph of the dataset, and each node in the subgraphs has at least one edge (no singleton node). We observe that the results, shown in Fig. 5, are consistent with the complexity discussions in Sec. III-E.

More specifically, as Fig. 5a and Fig. 5b show, the wall-clock time of Matri increases sharply with the number of users while *Guardian* consistently performs well as the number of users increases. This is because the parameters of our proposed trust convolutional layers are shared across all users, making the parameter complexity of our approach independent of the number of users. For the increasing number of trustee-trustor pairs, the time of Matri increases dramatically and shows similar trends on both datasets, shown in Fig. 5c and Fig. 5d. It is noteworthy that *Guardian* consistently performs well on all benchmarking datasets, indicating that *Guardian* is more scalable and can readily be generalized to large-scale network applications.



(a) Time vs. # of users on Advogato

(b) Time vs. # of users on PGP

(c) Time vs. # of pairs on Advogato

(d) Time vs. # of pairs on PGP

Fig. 5. Scalability: *Guardian* vs. Matri.

TABLE VI
ROBUSTNESS WITH DIFFERENT SIZES OF TRAINING SET ON ADVOGATO

APPROACH	TRAINING SET(%)	F1-SCORE	MAE
Guardian	80%	$\mathbf{74.3\%} \pm \mathbf{0.4\%}$	$\mathbf{0.082} \pm \mathbf{0.002}$
	60%	$\mathbf{72.9\%} \pm \mathbf{0.2\%}$	$\mathbf{0.087} \pm \mathbf{0.001}$
	40%	$\mathbf{70.7\%} \pm \mathbf{0.1\%}$	$\mathbf{0.094} \pm \mathbf{0.001}$
MATRI	80%	$65.6\% \pm 0.4\%$	0.127 ± 0.001
	60%	$63.9\% \pm 0.3\%$	0.132 ± 0.001
	40%	$61.7\% \pm 0.3\%$	0.139 ± 0.001

Robustness. We evaluated the approaches with different training and test set ratio to measure their robustness. The portions of the training set were set as 80%, 60%, 40% of the entire dataset. Table VI and Table VII show the evaluation results of both datasets. As reported, *Guardian* has a minor performance decrease of 3.6% for Advogato and 1.9% for

[6] Due to RAM issue, we are not able to reproduce NeuralWalk on PGP with our machine.

[7] For simplicity, in this paper, we described the factorization phase of Matri as training phase.

TABLE VII
Robustness with different sizes of Training Set on PGP

Approach	Training Set(%)	F1-Score	MAE
Guardian	80%	$\mathbf{87.1\% \pm 0.1\%}$	$\mathbf{0.083 \pm 0.001}$
	60%	$\mathbf{86.5\% \pm 0.1\%}$	$\mathbf{0.088 \pm 0.001}$
	40%	$\mathbf{85.3\% \pm 0.2\%}$	$\mathbf{0.096 \pm 0.001}$
Matri	80%	$68.3\% \pm 0.7\%$	0.122 ± 0.0003
	60%	$64.7\% \pm 0.1\%$	0.131 ± 0.0004
	40%	$60.5\% \pm 0.1\%$	0.144 ± 0.0001

PGP when the size of the training set is reduced to 40% of the entire graph, while Matri has a decrease of 3.9% and 7.8%, respectively. This indicates that our proposed framework has better robustness, with respect to the size of the training set. Notably, *Guardian* also consistently offers the best prediction accuracy, even when the model was trained with 40% training data as compared to Matri with 80% training data. This further suggests that proposed convolutional layers are able to effectively learn social trust relationships.

V. Related Work

In this section, we present and discuss some related works on pairwise social trust evaluation and recent advancements in applying convolutional neural networks to graph-structured data.

A. Pairwise Social Trust Evaluation

Walk-based approaches: In the past decade, most of the existing trust evaluation models were based on the trust propagation along the paths from the trustor to the trustee. For example, ModelTrust [18] and TidalTrust [19] evaluated the pairwise trustworthiness by searching the paths throughout the network. The propagated trust from multiple paths, between the trustor and the trustee, then are aggregated to be the estimated value of trust. Aiming for higher accurate trust evaluation, AssessTrust [6] and OpinionWalk [7] modeled the value of trust using statistical distributions in three-valued subjective logic. In particular, in order to establish a trust relationship between two indirectly connected users, OpinionWalk [7] walked throughout the network in a breadth-first search manner and modeled the trust propagation and aggregation via its predefined discounting and combining operators.

Matrix factorization-based approaches: [20] and Matri [3] are matrix factorization-based approaches, which are proposed to analyze the observed trustworthiness to identify the unobserved/missing trust relationships. In this category, the trustor-trustee pairs were analogous to user-item pairs in a recommender system. In general, the matrix factorization methods are used to map the trustors and the trustees to a joint latent factor space, so that the trustworthiness of the trustor-trustee pairs can be modeled as their inner products in that space. In particular, Matri [3] was designed to combine trust tendency and trust propagation under a collective matrix factorization framework, while [20] further considered the similarity of users' trust rating habits. Since these approaches are inherently transductive, expensive re-training process may be required to estimate the trust values for users that were not seen during the training phase.

Neural network-based approach: In contrast to the aforementioned approaches, NeuralWalk [8] was designed to capture the trust propagation and aggregation rules using machine learning techniques. The main component of this model is WalkNet, a neural network architecture, that was designed to model single-hop trust propagation and aggregation. By iteratively employing WalkNet, NeuralWalk is capable of establishing a trust relationship between the trustor and the trustee, as long as there exists at least one social path from the trustor to the trustee. Even though NeuralWalk can achieve state-of-the-art prediction accuracy in the literature, it is highly inefficient due to the massive matrix operations for training and test set selection.

B. Graph Convolutional Neural Networks

More recently, graph convolutional neural networks (GCNs) have been proven to be capable of learning on graph structure data [10], [11], [21], leading to new state-of-the-art results on benchmarks such as node classification and link prediction. These GCN-based approaches consistently outperformed techniques based upon matrix factorization or random walks (e.g, node2vec [16], Line [22], and DeepWalk [23]).Their success has led to a surge of interest in applying GCN-based frameworks to applications ranging from recommendation systems [12], drug design [24], to social influence prediction [25].

Despite the compelling success achieved by previous work, little attention has been paid to social trust evaluation with graph convolutional neural networks. Here we fill this gap and show the effectiveness and efficiency of graph convolutional neural networks-based representation learning for social trust evaluation.

VI. Conclusion and Future Work

In this paper, we devised a new framework *Guardian*, to model social trust for trust evaluation. In this framework, we explicitly incorporated the popularity trust and engagement trust into the latent representations of users to learn effective trust relationships. The key of *Guardian* is the newly proposed trust convolutional layer, which is able to jointly capture social graph structure and associated trust interactions. Extensive experiments on two real-world datasets have demonstrated the rationality and effectiveness of our proposed *Guardian*. In the meanwhile, it enjoys high efficiency due to the notion of localized graph convolutions. In the future, we are interested in improving *Guardian* by incorporating the attention mechanism during trust propagation. Moreover, we will investigate the capability of *Guardian* to address trust dynamics. It will also be interesting to incorporate the context-aware information to further enhance prediction performance.

REFERENCES

[1] P. Gao, H. Miao, J. S. Baras, and J. Golbeck, "Star: Semiring Trust Inference for Trust-Aware Social Recommenders," in *Proc. International Conference on Recommender Systems*. ACM, 2016.

[2] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler, "A 61-Million-Person Experiment in Social Influence and Political Mobilization," *Nature*, vol. 489, no. 7415, p. 295, 2012.

[3] Y. Yao, H. Tong, X. Yan, F. Xu, and J. Lu, "Matri: a Multi-Aspect and Transitive Trust Inference Model," in *Proc. WWW*. ACM, 2013.

[4] X. Li, Q. Yang, X. Lin, S. Wu, and M. Wittie, "ITrust: Interpersonal Trust Measurements from Social Interactions," *IEEE Network*, vol. 30, no. 4, pp. 54–58, 2016.

[5] D.-N. Yang, H.-J. Hung, W.-C. Lee, and W. Chen, "Maximizing Acceptance Probability for Active Friending in Online Social Networks," in *Proc. SIGKDD*. ACM, 2013.

[6] G. Liu, Q. Yang, H. Wang, X. Lin, and M. P. Wittie, "Assessment of Multi-hop Interpersonal Trust in Social Networks by Three-Valued Subjective Logic," in *Proc. INFOCOM*. IEEE, 2014.

[7] G. Liu, Q. Chen, Q. Yang, B. Zhu, H. Wang, and W. Wang, "OpinionWalk: An Efficient Solution to Massive Trust Assessment in Online Social Networks," in *Proc. INFOCOM*. IEEE, 2017.

[8] G. Liu, C. Li, and Q. Yang, "NeuralWalk: Trust Assessment in Online Social Networks with Neural Networks," in *Proc. INFOCOM*. IEEE, 2019.

[9] R. Urena, G. Kou, Y. Dong, F. Chiclana, and E. Herrera-Viedma, "A Review on Trust Propagation and Opinion Dynamics in Social Networks and Group Decision Making Frameworks," *Information Sciences*, vol. 478, pp. 461–475, 2019.

[10] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS 2017)*, 2017.

[11] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in *Proc. International Conference on Machine Learning (ICML 2017)*, 2017.

[12] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph Convolutional Neural Networks for Web-Scale Recommender Systems," in *Proc. SIGKDD*. ACM, 2018.

[13] W. Sherchan, S. Nepal, and C. Paris, "A Survey of Trust in Social Networks," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 47, 2013.

[14] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] S. Nepal, W. Sherchan, and C. Paris, "STrust: A Trust Model for Social Networks," in *Proc. International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, 2011.

[16] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proc. SIGKDD*. ACM, 2016.

[17] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[18] P. Massa and P. Avesani, "Controversial Users Demand Local Trust Metrics: An Experimental Study on epinions. com Community," in *Proc. AAAI*, 2005.

[19] J. Golbeck, J. Hendler *et al.*, "Filmtrust: Movie Recommendations using Trust in Web-Based Social Networks," in *Proc. International Conference on Consumer Communications and Networking*. IEEE, 2006.

[20] X. Zheng, Y. Wang, M. A. Orgun, Y. Zhong, and G. Liu, "Trust Prediction with Propagation and Similarity Regularization," in *Proc. AAAI*, 2014.

[21] M. Zhang and Y. Chen, "Link Prediction Based on Graph Neural Networks," in *Proc. International Conference on Neural Information Processing Systems (NeurIPS 2018)*, 2018.

[22] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-Scale Information Network Embedding," in *Proc. WWW*. ACM, 2015.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online Learning of Social Representations," in *Proc. SIGKDD*. ACM, 2014.

[24] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling Polypharmacy Side Effects with Graph Convolutional Networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[25] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: Social Influence Prediction with Deep Learning," in *Proc. SIGKDD*. ACM, 2018.