

Received December 28, 2020, accepted January 13, 2021, date of publication January 26, 2021, date of current version February 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054818

# Collaborative Deep Forest Learning for Recommender Systems

SOHEILA MOLAEI<sup>1</sup>, AMIRHOSSEIN HAVVAEI<sup>1</sup>, HADI ZARE<sup>1</sup>,  
AND MAHDI JALILI<sup>2</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Network Science and Technologies, University of Tehran, Tehran 1417466191, Iran

<sup>2</sup>School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

Corresponding author: Hadi Zare (h.zare@ut.ac.ir)

**ABSTRACT** Collaborative filtering (CF) is one of the most practical approaches on recommendation systems by predicting users' preferences for items based on the user-item interaction information. Besides the connections between users and items, social networks among users can provide auxiliary information to improve the performance of recommender systems. Here, we propose an end-to-end deep learning framework by learning latent social features to embed in a CF approach. First, representation learning is employed on the rating matrix to extract the latent social features. Then, a novel deep learning approach based on cascade tree forest is used in the recommendation process. Experiments on real-world datasets from different domains demonstrate that the proposed Collaborative Deep Forest Learning (CDFL) outperforms the state-of-the-art CF recommendation methods.

**INDEX TERMS** Recommender systems, social networks, deep learning, collaborative filtering, representational learning.

## I. INTRODUCTION

Recommender systems (RS) have become an integral part of many online services, aiming to enhance users' experience by exposing them to relevant contents [1]. On e-commerce platforms, for example, RS's are used to help consumers search rapidly through potentially huge inventories and allow them to find items they want to purchase easily [2]. RS's are developed to model users' behaviors to predict the likelihood of a user's preference for a new item [3]. Two commonly used RS techniques are content-based filtering (CBF) and collaborative filtering (CF) [4], [5]. CBF is based on recommending items according to their contextual information [6]. The popular CF approach uses a vast amount of past users' experience to predict the items that are likely to be seen (positively rated) by a target user.

Several CF algorithms have been developed in the literature, which can be divided into two main categories, model-based [2], [7], and memory-based [8], [9]. Memory-based methods generate a prediction using the entire user-item data based on statistical techniques to find a set of similar users. Model-based methods seek to provide

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

a prediction model from user-item ratings using machine learning [2], [10].

There are two main challenges of the CF approach: *data sparsity* and *cold start* [11]. There are often many users and items in the dataset; however, there are very few ratings from each user to some of the items. Almost in all real-world examples of RS applications, the user-item interaction matrix is heavily sparse. The cold start problem refers to the lack of enough rating history from (new) users, which makes it difficult, if not impossible, to identify the taste of these users. A number of methodologies have been proposed in the literature to deal with cold start and sparsity problems such as overlapping community detection in a time-aware approach [12] and non-negative matrix factorization [13]. These problems are usually alleviated by matrix factorization (MF) algorithm [14]. MF enables predictions as a linear combination of variables, thereby allowing better scalability [13]. Probabilistic MF (PMF) [15], proved to be better performing than the typical MF method and was later used extensively [16], [17].

Recently, the technology of deep learning has advanced in machine learning, and several deep learning-based RS's have been proposed to produce accurate recommendations. Deep learning enables multi-layered computational models to learn

data representations with multiple abstraction levels [18]. Due to the feature enrichment power of the deep learning approach, there is a tendency of employing them in RS [19], [20]. In this work, we propose a novel end-to-end deep forest learning approach on recommendation models based on the ideas of CF. Initially, the latent features are learned based on the rating matrix through an unsupervised representation approach. Furthermore, the raw input of the deep forest model is extracted using the nod2vec [21] framework. The experimental results demonstrate the strength of the proposed approach compared with the earlier works. We evaluate the performance of the proposed method using four real-world datasets: Epinions, FilmTrust [22], Ciao [23], and Flixster [24].

To summarize, our work makes the following contributions:

- We propose a novel deep representative method called CDLF, which operates directly on user-item graphs in recommender systems.
- We present how this type of graph-based deep neural network could be used in a graph for fast and robust prediction on recommendation systems.
- We explore the possibility of building recommender systems based on combining differentiable and non-differentiable modules.
- We demonstrate the effectiveness of the proposed method in real-world data and validate the importance of extracting features in accurate recommendations.

The rest of this paper is arranged as follows. Section II gives a useful summary of related works. Section III describes the proposed deep learning approach in recommendation models. Section IV points out the experimental results. Finally, Section V concludes the paper with some suggestions on this interesting field.

## II. LITERATURE REVIEW

Data sparsity is a critical problem in recommender systems for collaborative filtering (CF) techniques, particularly for new users and items. There are a variety of hybrid methods to mitigate data sparsity that integrate auxiliary data sources into recommendation strategies, such as material, meaning, or social relationships [25]–[28]. In this work, we propose an end-to-end method that jointly learns representations with the help of auxiliary information. The representations, which can be thought of as mapping inputs to embedded features and estimating user preferences. Our proposed method is a model-based method. In the following, we first give a brief overview of model-based factorization CF, and then discuss some existing deep learning-based recommendation methods.

### A. MATRIX FACTORIZATION CF

Sparsity of the rating matrix and the cold start problem make it difficult in some cases to find similar users for target users [29]. Several techniques have been developed to alleviate this problem, such as using trust [30], [31] and

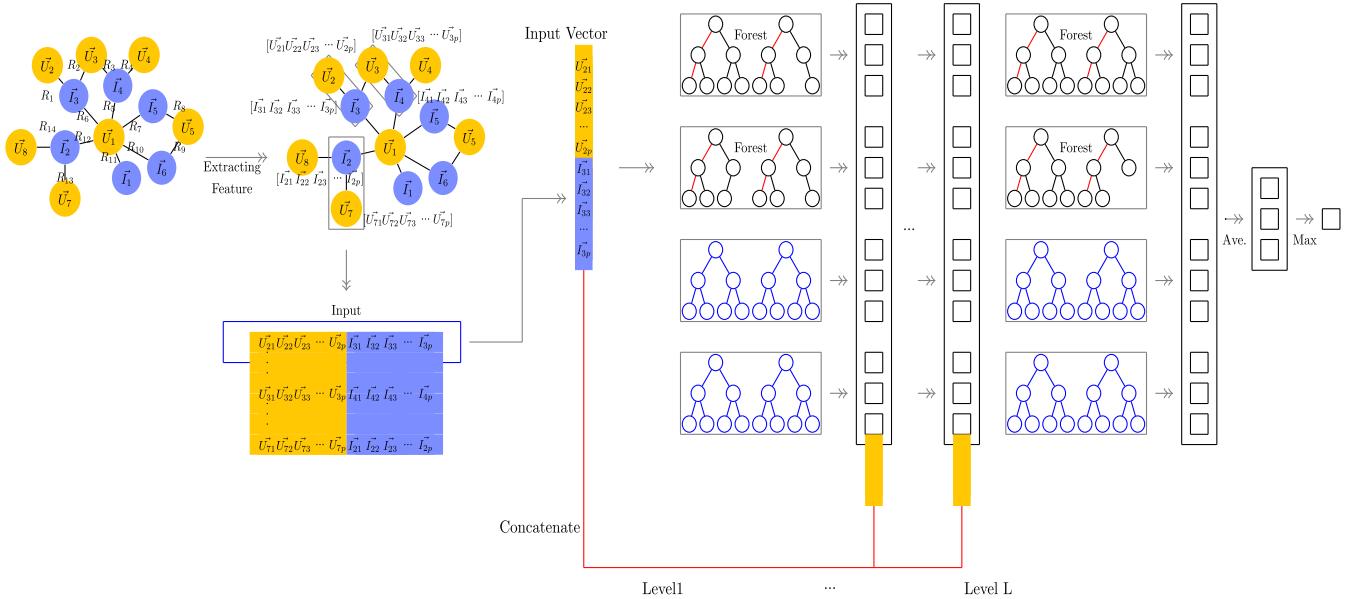
emotional information [32]. Koren [31] suggested a recommendation model (SVD++) to simultaneously consider the explicit and implicit trust between users and the ratings. In this filed, Guo *et al.* [2] Integrated several sources of information into the recommendation model to minimize data sparsity and cold start problems called TrustSVD, thereby expanding on the new recommendation algorithm, SVD++. Duricic *et al.* [33] discussed the use of usual equivalence for constructing the similarity matrix on the trust network. Davoudi and Chatterjee [34] dealt with the issue of modeling social trust where trust values dominated the characteristics of users in a social network. An imputation based strategy on rating information is employed to tackle the sparsity challenge [35]. Qian *et al.* [32] proposed an emotion-aware recommendation system based on a hybrid fusion of information to evaluate the features of users, that are considered as the rating data as explicit information, user social network data as implicit information and sentiment from user reviews as emotional information.

Matrix factorization (MF) is one of the most commonly utilized CF methods. Since the typical MF suffers from the problem of matrix sparsity, some earlier works explored the content information of items to boost the performance. Salakhutdinov *et al.* [36] used restricted Boltzmann machines to perform CF and Georgiev and Nakov [19] expanded it by integrating associations between user-user and item-item. Sainath *et al.* [37] reduced the number of model parameters and speed up training by using low-rank matrix factorization. Yang *et al.* [38] suggested a recommendation model focused on matrix factorization that would map users into a low-dimensional latent feature space considering the relationship of trust. Silva *et al.* [39] have proposed PoissonMF-CS which uses factorization of the Poisson matrix to model the preferences of users and the contents of items. Ren *et al.* [40] suggested SCVR that item scores would be forecast based on user opinions and social relationships.

### B. DEEP LEARNING

Recently, deep learning models have been used to improve the performance of recommender algorithms due to their non-linear modeling capabilities. These include NeuMF (Neural Matrix factorization) [41] and CDAE (Collaborative Denoising Auto-Encoder) [42]. Sainath *et al.* [37] reduced the number of model parameters and speed up training by using low-rank matrix factorization in the last layer of a deep network. Yuan Jiang *et al.* proposed DTR [10] by utilizing stacked denoising autoencoder (SDAE) to transform the trust information to latent feature space. In [11], a neural network architecture by a denoising autoencoder is proposed based on the integration of the rating and explicit trust. Deep learning works can be broadly grouped into the following:

- **Embedding and Feature extracting methods:** These types of methods learn to map nodes on a low-dimensional feature space that maximizes the likelihood of node neighborhood preservation in the network [21].



**FIGURE 1.** The flow-graph of the proposed approach.

Such embedding methods could be used to embed users and items [43], [44], or directly provide recommendations. Following this, [45] suggested a novel Deep Attention-based Sequential, consisting of an embedding block that embeds users and items in low-dimensional spaces into an attention block. Reference [46] suggested a deep bias probabilistic matrix factorization model by using the fully convolutional neural network to remove latent user-item features and by applying a bias to probabilistic matrix factorization to monitor user rating activity and item popularity.

- **Autoencoders methods:** Deep learning can easily control non-linear or non-trivial user-item relationships and empower more complicated abstractions to be compiled as data representations in the higher layers. Most network models, such as Feedforward or Autoencoder networks are exploited for acquiring this goal [47]. Sedhain *et al.* [48] introduced a new Autoencoder framework called AutoRec, which had representational and mathematical strengths over existing approaches, including collaborative filtering. Wu *et al.* [42] present a novel method, called Collaborative Denoising Auto-Encoder, for the top-N recommendation that utilizes the idea of Denoising Auto-Encoders.

These approaches, considering their efficacy, might not be enough to provide adequate embeddings for CF. Their major drawback is lack of user-item interaction to display users (or items) similarities. More precisely, most approaches construct an embedding function only with attributes without addressing interactions between users and items [49]. Here, we use embedding and feature extraction techniques based on deep learning approaches to propose a new RS algorithm.

### III. PROPOSED METHOD

Our contributions are two-fold. First, we introduce a simple feature extraction rule which operates directly on user-item interaction graphs. Second, we explain how this type of graph-based deep neural network can be used in a graph for efficient and scalable supervised prediction. Experiments on four benchmark datasets demonstrate that our model compares favorably both in accuracy and efficiency against state-of-the-art RS methods. The overview of the proposed approach is shown in Figure 1:

#### A. PROBLEM DEFINITION

Let's consider  $m$  users and  $n$  items. In our setting  $R = [R_{ij}]_{m \times n}$ . Each element of  $R$  refers to the observed rating of item  $j$  given by user  $i$ . The entities of  $R_{ij}$  can be varied between 1 and  $R_{max}$ , but unknown elements indicate that the user has not rated the item. Our goal is to predict unknown entries values of user ratings in the rating matrix.

#### 1) CDFL METHOD

RS is usually based on the data that represents relationships between a set of users and items. Feeding the raw data into the algorithm may not result in a proper outcome. Some methods can produce information with dependencies between interconnected objects that can be used to pre-process the data before giving it to RS algorithms. Additional input information can be added to the recommendation process, where this useful information could be a better representation of data to help the model perform better. Node2Vec is a semi-supervised scalable algorithm for feature learning in networks. This method returns feature representations that increase the probability that node neighborhoods will

be retained in d-dimensional feature space (steps 1 and 2 in III-A1 algorithm).

Let  $G = (V, E)$  be a given user-item graph which is built from a user-item rating matrix. Let  $f$  be the node mapping function to feature d-size representations. We define  $N_s(u) \subset V$  as a  $u$  node network neighborhood given for each  $u \in V$  source node through a neighborhood sample of  $S$ . We try to optimize the objective (1) function, which maximizes the log likelihood of  $N_s(u) \subset V$  for a  $u$  node conditioned on its representation of the feature, given by  $f$ :

$$\max_f \sum_{u \in V} \log Pr(N_s(u)|f(u)). \quad (1)$$

In this step, we create a heterogeneous network which has two types of nodes. The main challenge of deep learning methods when applied to RS is how to obtain input from a heterogeneous network while keeping computational complexity low. In the first step, we try to convert the heterogeneous network to a homogeneous network. To do so, we first extract features from heterogeneous network. After extracting features for every node, our method embeds feature vectors of any user  $u$  and item  $i$  as a final feature in which the dimension of the new embedding feature is  $2d$ , as shown in Figure 1. After embedding, we produce a homogeneous network as an input which could be trained with a deep learning algorithm.

As the embedded feature vectors are obtained, Cascade Forest Structure is used for the perdition of ratings where each level of cascade receives feature information processed by its preceding level and outputs its processing result to the next level and each level is an ensemble of decision tree forests. This method employs a cascade structure to perform layer-by-layer processing, as shown in Figure 1, where each level of a cascade receives information about features processed at the previous level. Suppose that we generate some completely-random forests. Each random forest contains  $n$  completely-random trees generated by randomly choosing a feature for the split at each node of the tree, and growing tree until each leaf node contains only the same class of instances. Each forest will produce an estimate of class distribution, by counting the percentage of different classes of training examples at the leaf node (steps 3, 4, and 5 in algorithm 1).

To avoid overfitting, a  $k$ -fold cross-validation procedure is used for producing a class vector with each forest. The class vector formed by each forest is created by  $k$ -fold cross-validation to reduce the possibility of overfitting. That instance would be used in detail as training data for  $k - 1$  times, resulting in  $k - 1$  class vectors which are then averaged to generate the final class vector as augmented features for the next cascade level [50]. A summary of the main steps of CDFL is bring in III-A1.

#### IV. EXPERIMENTS

We demonstrate the efficacy of our model on some real-world datasets from different domains. The proposed method is compared with state-of-the-art recommendation algorithms.

#### Algorithm 1 Summary of the Main Steps of CDFL

- **Input:** User Item Matrix
- **Output:** Rate estimation

- 1) Creating a user item graph
- 2) Assigning the feature to each node
  - a) Running the feature extraction algorithm
  - b) Combining each user's features with its connected item's features (Figure 1)
- 3) Giving features to Cascade Forest
- 4) Type of forests:
  - a) Completely random forest
  - b) random forest
- 5) Tree growth till pure leaf

**TABLE 1. Demographic information of the datasets used in this work.**

2*Dataset	Basic Meta			
	Users	Items	Ratings(Scale)	Density
Epinions	40,163	139,738	[1,5]	0.0118%
Ciao	21,019	71,633	[1,5]	0.0379%
FilmTrust	1,508	2,071	[0.5,4.0]	1.14%
Flixster	147,612	48,794	[0.5,5.0]	0.1138%

#### A. DATASETS

We use four benchmark real-world datasets from different real-world domains, i.e, *Epinions*, *Ciao*, *FilmTrust*, and *Flixster*.

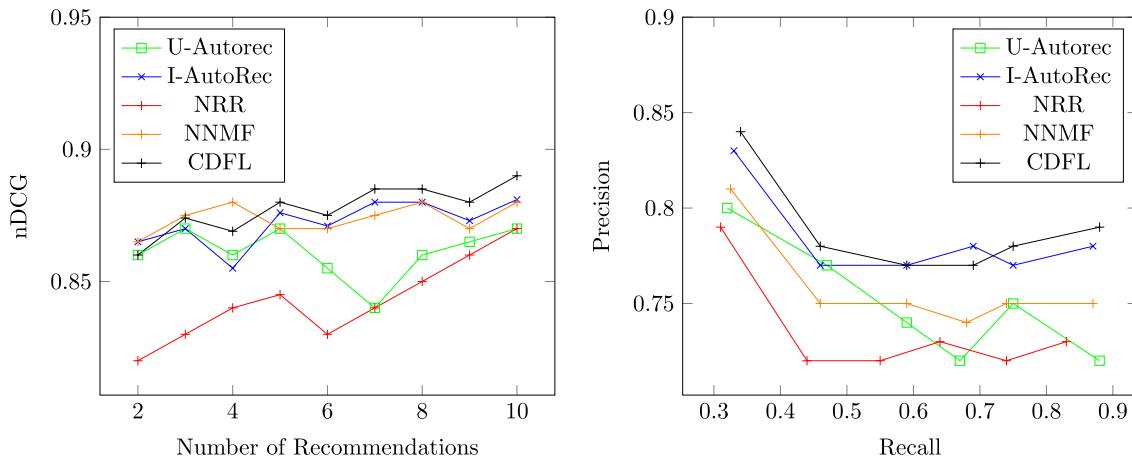
- *Epinions*: This dataset is published by [51] from the *Epinions.com* Web site in November/December 2003. It contains 40,163 users who rated 139,738 movies at least once which has formed a totally 664,823 ratings.
- *FilmTrust*: In June 2011 the FirmTrust dataset was gathered from the whole FilmTrust website [22]. The rating values are discrete numbers in the range [0.5, 4.0]. It contains 35,497 ratings provided by 1508 users for 2071 items.
- *Flixster*: Flixster dataset released by [24] utilizing a social networking tool that enables users to rate movies. In this dataset, users are allowed rate movies in the range [0.5, 5] with step size 0.5.
- *Ciao DVD*: In December 2013, Ciao dataset gathered from the whole DVDs groups on the *dvd.ciao.co.uk* website [23]. There are 1,625,480 ratings given by 21,019 users to 71,633 reviews written by users.

#### B. EVALUATION MEASURES

To assess predictive efficiency, two well-known metrics are used, mean absolute error (MAE) and root mean square error (RMSE), defined by:

$$MAE = \frac{\sum_{i,j} |R_{i,j} - \hat{R}_{i,j}|}{M}$$

$$RMSE = \sqrt{\frac{\sum_{i,j} (R_{i,j} - \hat{R}_{i,j})^2}{M}} \quad (2)$$



**FIGURE 2.** Evaluation results on filmtrust dataset.

where  $R_{i,j}$ ,  $\hat{R}_{i,j}$ , and  $M$  are the rate of the user  $i$  to item  $j$ , predicted rate for item  $j$  and the number of ratings.

### C. PERFORMANCE EVALUATION

To evaluate the performance, we compared CDFL with a number of methods including the state-of-the-art deep learning based recommender techniques. The models included in our comparison are listed as follows:

#### 1) BASELINE METHODS

- **UAvg/IAvg:** Such two methods are just average historical ratings of target users and target objects.
- **PMF** [15]: Probabilistic Matrix Factorization uses only the user-item rating matrix and models user and item latent factors into Gaussian distributions.
- **SoRec** [52]: Social Recommendation is a general framework proposed to alleviate the data sparsity by integrating the user-item rating matrix and social contextual information.
- **SoReg** [53]: Social Regularization maps information regarding the social network as regularized to restrict objective functions to matrix factorization.
- **SocialMF** [24]: This model contains trust information and propagation of trust information into the matrix factorization model.
- **TrustMF** [38]: This method adopts a matrix factorization approach that maps users into two low-dimensional spaces: trustee space and trustee space, by factoring trust networks depending on trust's directional properties.
- **TrustSVD** [2]: It is expanded on the SVD++ algorithm which implicitly includes the explicit and implicit influence of rated items by further integrating both the explicit and implicit impact of trusted users on the estimation of items for an active user.
- **SVD++** [31]: It extends the SVD-latent factor model that integrates implicit feedback into the model. In other words, both explicit and implicit influence of rated items have taken into account.

- **RSTE** [54]: A probabilistic matrix factorization framework which motivated by the fact that trusted users of a target user will affect the behavior of the target user.

#### 2) DEEP LEARNING METHODS

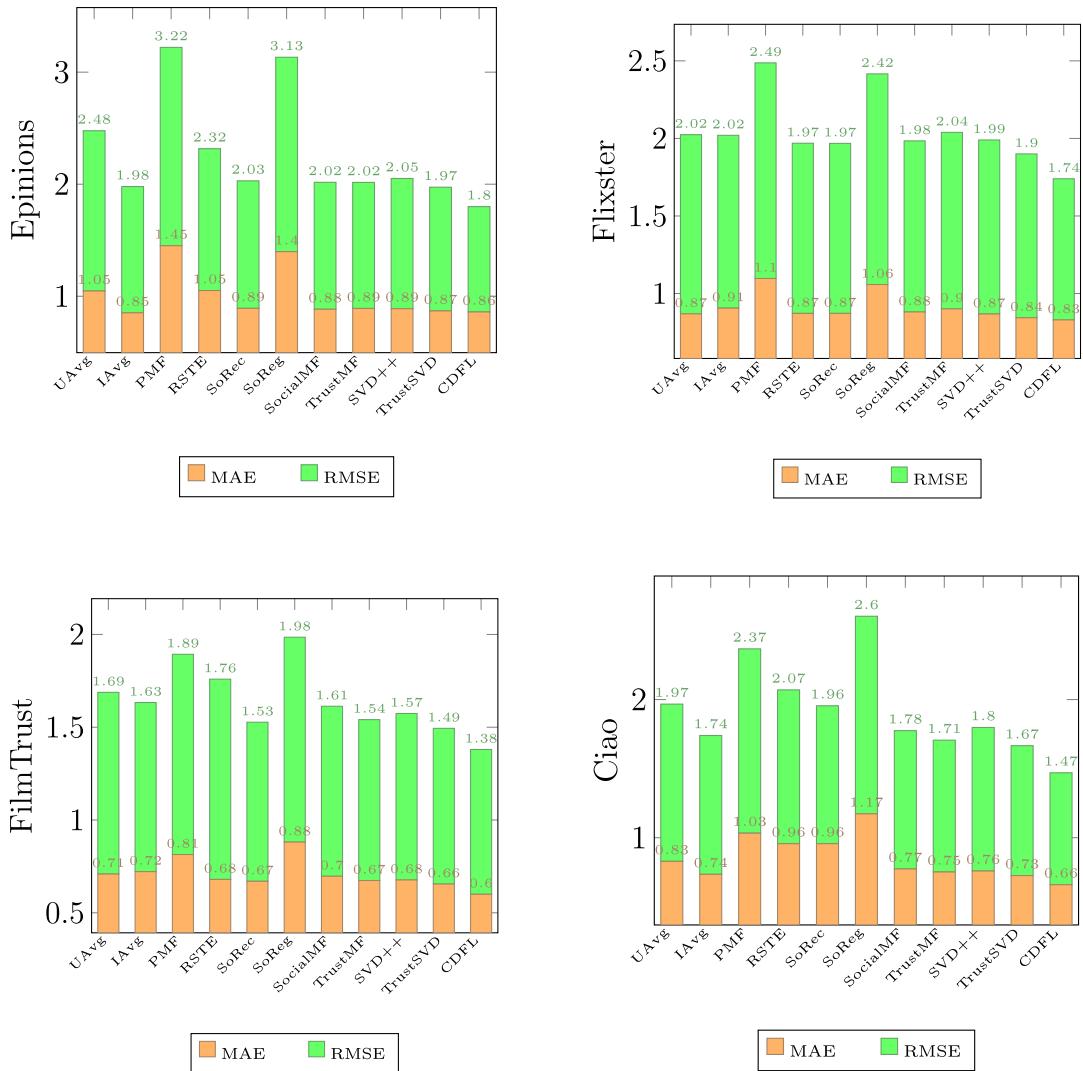
- **I-AutoRec/U-AutoRec** [48]: AutoRec takes partial user vectors  $r(u)$  or partial item vectors  $r(i)$  as input and is designed to recreate them in the output layer. It has two variants: item-based AutoRec (I-AutoRec) and User-based AutoRec (U-AutoRec), which refers to the two input types.
- **NNMF** [55]: Neural network matrix factorization (NNMF) replaces the inner product in a matrix factorization by a multi-layer feed-forward neural network. This function is learned by alternating between the neural network for fixed latent features and vice versa means optimizing the latent features for a fixed network.
- **NRR** [56]: This technique offered a multitask teaching approach for the estimation of ratings and the production of textual tips relevant to the user. The generated tips provide concise suggestions and anticipate the user's experience with and feelings about certain products consisting of two main parts: neural rating regression and abstractive tips generation. Neural rating regression performs representation learning for the user latent factor and item latent factor. Abstractive tips generation translates user latent factor and item latent factor into a fluent sequence of words using Gated Recurrent Unit (GRU) [57].

### D. CONFIGURATION

Each deep forest level comprises of four completely-random tree forests and four random forests consisting of 500 trees every level. For multi-grained scanning, Deep Forest uses feature windows with sizes of  $[d/16]$ ,  $[d/8]$ ,  $[d/4]$ .

### E. CASE STUDY

We apply the algorithms on Filmtrust dataset and evaluate their performance in terms of three metrics. Precision is the

**FIGURE 3.** Experimental results of the all users.

proportion of the  $K$  items recommended to the user  $u$ , defined as:

$$Precision = \frac{R_u^k}{K} \quad (3)$$

where  $R_u^k$  is the set of  $K$  items recommended to the user  $u$ . Recall reflects the proportion of relevant items recommended to the user over the set of the test items rated by the user  $u$  ( $I_u$ ):

$$Recall = \frac{R_u^k}{I_u} \quad (4)$$

nDCG<sub>u</sub> is the normalized relevance of  $K$  recommendations provided to user  $u$  based on its position in the recommendation list:

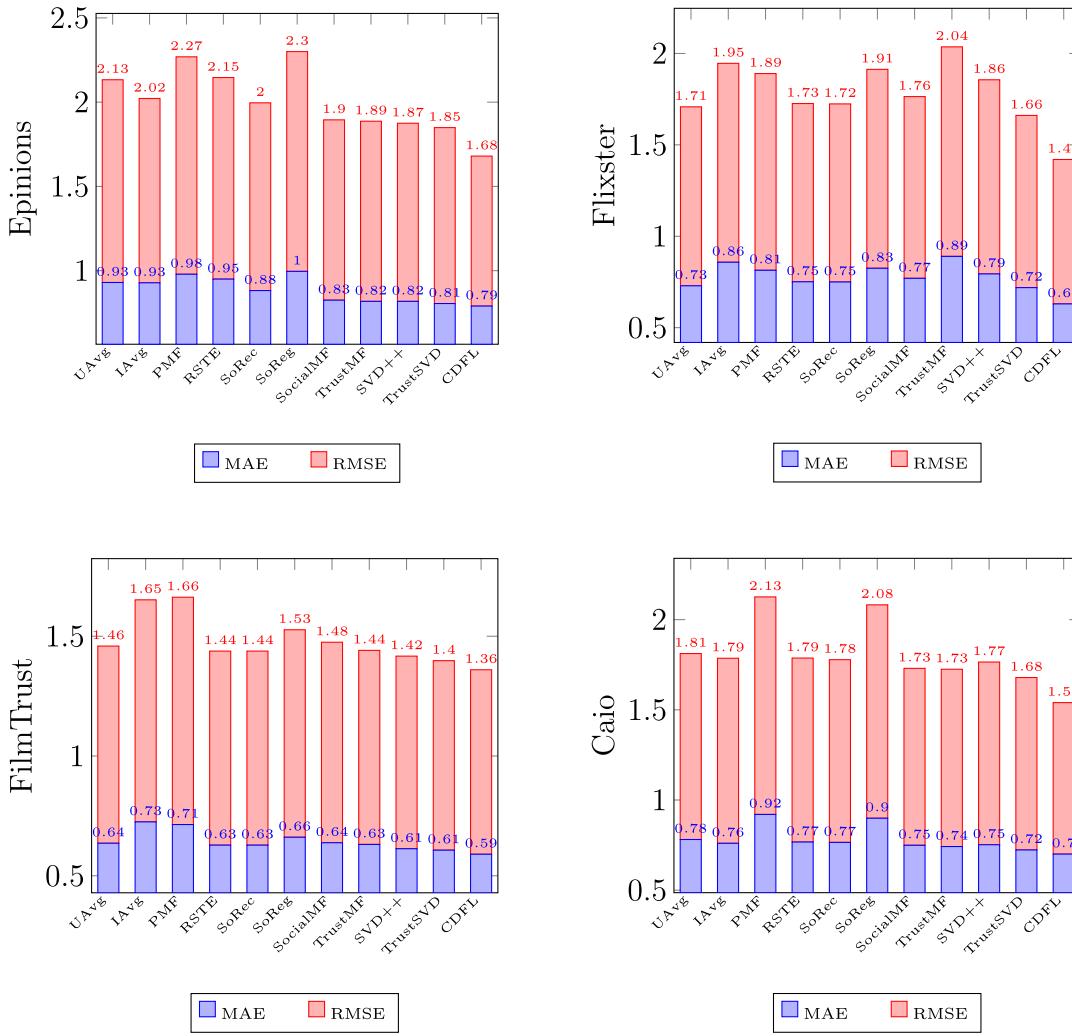
$$nDCG = \frac{DCG_u}{IDCG_u} \quad (5)$$

$IDCG_u$  is the  $DCG_u$  of ideal ranking order, that is, the actual rankings of the user in the test set.

$$DCG = \sum_{p=1}^K \frac{2^{r_{u,x_{u,p}}}}{\log_2^{(p+1)}} \quad (6)$$

$$nDCG = \sum_{p=1}^{I_u} \frac{2^{r_{u,y_{u,p}}}}{\log_2^{(p+1)}} \quad (7)$$

where  $x_{u,p}$  stand for the item recommended at the  $p$ -th position if items recommended to user  $u$  are sorted from higher to lower prediction ( $r_{u,i}$ ) and  $y_{u,p}$  stand for the item at the  $p$ -th position if test items rated by user  $u$  ( $I_u$ ) are sorted by its rating value ( $r_{u,i}$ ). The results indicate a significant improvement in quality of predictions. The quality of the recommendation is also substantially elevated. Fig. 2 contains the nDCG and the Precision & Recall values for Filmtrust dataset. In this



**FIGURE 4.** Experimental results of the Cold-Start users.

figure, we can observe that the results of nDCG show that the proposed approach CDFL outperforms baseline approaches. We can find that the I-AutoRec and NNMF approaches perform worse than CDFL but better than the other methods.

## F. ANALYSIS

Figures 3 and 4 show the experimental results, referring to All and Cold-start users, separately. In *All* view, all users will be used as the test set. The *Cold-Start* view specifies that the test set will only include users who rate less than five items [2], [38].

Note that Figure 3 indicates the percentage of improvements compared with the TrustSVD (the best approach among other competitors). Our results demonstrate strong performance being achieved across all datasets. We particularly note that CDFL approach is competitive with the results reported for the other state-of-the-art models, even exceeding its performance on all datasets for both views. We believe that these advantages arise from the fact that CDFL approach

implicitly enables each node to have access to the structural properties of the whole graph whereas the other methods don't use any auxiliary information. As the amount of data increases, the efficiency of deep learning-based methods significantly outperforms most classic machine learning algorithms. On large datasets, such as those we have considered in this work (Epinions and Flixster), better performance of the proposed method over others is more pronounced and the improvement tends to be more obvious. The percentages of increases in Epinions, despite its low density(0.0118%) are 5% and 11% for MAE and RMSE respectively, compared with the best approach among competitors(NRR) and rest of datasets consistently outperform all the baseline methods.

One crucial issue is to enhance the process of representation of features. Current CDFL implementation takes the simplest vector form. Due to the high-dimensional nature of the network or actual feature vectors, such a small number of augmented features would quickly drown out. It is remarkable that the bigger is the dataset, the higher would

be the improvement obtained by the proposed method. This way, we can claim that the proposed CDFL architecture is scalable in the sense that it works better when the dataset becomes larger. Constructing larger deep forests contributes to enhanced generalization efficiency in practice, while computing facilities are critical for the training of larger models.

CDFL uses a decision tree ensemble approach. Ensemble methods are some sort of effective machine learning methods that combine multiple learners for the same task. In addition, some studies [58] show that the performance can be even better than simply using deep learning methods by using ensemble methods such as random forests facilitated by deep learning representations. Besides, one of the most common ensemble methods is the random forest, which has been widely applied to different tasks [59]. The use of CDFL not only helps promoting diversity, but also offers an ability to manipulate unlabeled data. The growing of trees does not need labels, while label information is required only for the annotation of leaf nodes.

## V. CONCLUSION

In this paper, we proposed a new recommendation algorithm based on deep learning technology. We showed that high performance can be attained through joint deep representation learning and collaborative filtering. To the best of our knowledge, CDFL is the first deep forest model to narrow the gap between state-of-the-art deep learning models and RS. We introduced a representational deep learning approach and demonstrated how much they are more effective in the context of CF. A series of experimental results based on four real public datasets showed that our approach performs better than state-of-the-art methods. The proposed method also outperformed others in the case when only cold-start users were considered, demonstrating its power in dealing with the cold-start problem.

## REFERENCES

- [1] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Rec. Syst. (RecSys)*, New York, NY, USA, 2007, pp. 17–24.
- [2] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.
- [3] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf. (WWW)*. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2018, pp. 689–698.
- [4] R. Mu, "A survey of recommender systems based on deep learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018.
- [5] N. Joorabloo, M. Jalili, and Y. Ren, "Improved collaborative filtering recommendation through similarity prediction," *IEEE Access*, vol. 8, pp. 202122–202132, 2020.
- [6] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 73–105.
- [7] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical Poisson factorization," in *Proc. 31st Conf. Uncertainty Artif. Intell. (UAI)*, Arlington, VA, USA, 2015, pp. 326–335.
- [8] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 56–69, Jan. 2004.
- [9] S. Meyffret, L. Médini, and F. Laforest, "Confidence on collaborative filtering and trust-based recommendations," in *E-Commerce and Web Technologies*, C. Huemer and P. Lops, Eds. Berlin, Germany: Springer, 2013, pp. 162–173.
- [10] Y. Jiang, H. Chen, and B. Yang, "Deep social collaborative filtering by trust," in *Proc. Int. Conf. Big Data Technol. (ICBDT)*, New York, NY, USA, 2018, pp. 52–56.
- [11] M. Wang, Z. Wu, X. Sun, G. Feng, and B. Zhang, "Trust-aware collaborative filtering with a denoising autoencoder," *Neural Process. Lett.*, vol. 49, no. 2, pp. 835–849, Apr. 2019.
- [12] F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, "TCARS: Time-and community-aware recommendation system," *Future Gener. Comput. Syst.*, vol. 78, pp. 419–429, Jan. 2018.
- [13] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.
- [14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.
- [15] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
- [16] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 880–887.
- [17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*. [Online]. Available: <http://arxiv.org/abs/1205.2618>
- [18] S. Molaei, H. Zare, and H. Veisi, "Deep learning approach on information diffusion in heterogeneous networks," *Knowl.-Based Syst.*, vol. 189, 2020, Art. no. 105153, doi: [10.1016/j.knosys.2019.105153](https://doi.org/10.1016/j.knosys.2019.105153).
- [19] K. Georgiev and P. Nakov, "A non-iid framework for collaborative filtering with restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1148–1156.
- [20] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1235–1244.
- [21] 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.
- [22] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel Bayesian similarity measure for recommender systems," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2013, pp. 2619–2625.
- [23] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "ETAF: An extended trust antecedents framework for trust prediction," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 540–547.
- [24] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Rec. Syst. (RecSys)*, New York, NY, USA, 2010, pp. 135–142.
- [25] H. Zare, M. A. N. Pour, and P. Moradi, "Enhanced recommender system using predictive network approach," *Phys. A, Stat. Mech. Appl.*, vol. 520, pp. 322–337, Apr. 2019.
- [26] S. Sedhain, S. Sanner, D. Braziunas, L. Xie, and J. Christensen, "Social collaborative filtering for cold-start recommendations," in *Proc. 8th ACM Conf. Rec. Syst. (RecSys)*, 2014, pp. 345–348.
- [27] S. Sedhain, S. Sanner, L. Xie, R. Kidd, K.-N. Tran, and P. Christen, "Social affinity filtering: Recommendation through fine-grained analysis of user interactions and activities," in *Proc. 1st ACM Conf. Online Social Netw.*, 2013, pp. 51–62.
- [28] G. Alexandridis, G. Siolas, and A. Stafylopatis, "Improving social recommendations by applying a personalized item clustering policy," in *Proc. RSWeb@ RecSys*, 2013, pp. 1–7.
- [29] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: A survey," *IEEE Access*, vol. 6, pp. 74003–74024, 2018.
- [30] G. Alexandridis, G. Siolas, and A. Stafylopatis, "Enhancing social collaborative filtering through the application of non-negative matrix factorization and exponential random graph models," *Data Mining Knowl. Discovery*, vol. 31, no. 4, pp. 1031–1059, Jul. 2017.
- [31] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2008, pp. 426–434.
- [32] Y. Qian, Y. Zhang, X. Ma, H. Yu, and L. Peng, "EARS: Emotion-aware recommender system based on hybrid information fusion," *Inf. Fusion*, vol. 46, pp. 141–146, Mar. 2019.

- [33] T. Duricic, E. Lacic, D. Kowald, and E. Lex, "Trust-based collaborative filtering: Tackling the cold start problem using regular equivalence," in *Proc. 12th ACM Conf. Rec. Syst. (RecSys)*, New York, NY, USA, 2018, pp. 446–450.
- [34] A. Davoudi and M. Chatterjee, "Social trust model for rating prediction in recommender systems: Effects of similarity, centrality, and social ties," *Online Social Netw. Media*, vol. 7, pp. 1–11, Sep. 2018.
- [35] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," *Eng. Appl. Artif. Intell.*, vol. 46, pp. 58–66, Nov. 2015.
- [36] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 791–798.
- [37] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6655–6659.
- [38] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017.
- [39] E. D. S. da Silva, H. Langseth, and H. Ramampiaro, "Content-based social recommendation with Poisson matrix factorization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2017, pp. 530–546.
- [40] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke, "Social collaborative viewpoint regression with explainable recommendations," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 485–494.
- [41] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW)*. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.
- [42] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, Feb. 2016, pp. 153–162.
- [43] F. Vasile, E. Smirnova, and A. Conneau, "Meta-Prod2Vec: Product embeddings using side-information for recommendation," in *Proc. 10th ACM Conf. Rec. Syst.*, Sep. 2016, pp. 225–232.
- [44] J. B. Vuuren, M. Larson, and A. P. de Vries, "Exploring deep space: Learning personalized ranking in a semantic space," in *Proc. 1st Workshop Deep Learn. Rec. Syst.*, 2016, pp. 23–28.
- [45] S. Yakhchi, A. Beheshti, S.-M. Ghafari, M. A. Orgun, and G. Liu, "Towards a deep attention-based sequential recommender system," *IEEE Access*, vol. 8, pp. 178073–178084, 2020.
- [46] K. Li, X. Zhou, F. Lin, W. Zeng, and G. Alterovitz, "Deep probabilistic matrix factorization framework for online collaborative filtering," *IEEE Access*, vol. 7, pp. 56117–56128, 2019.
- [47] S. Wang, C. Huang, J. Li, Y. Yuan, and F.-Y. Wang, "Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts," *IEEE Access*, vol. 7, pp. 136951–136961, 2019.
- [48] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 111–112.
- [49] Y. Tay, L. Anh Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 729–739.
- [50] Z.-H. Zhou and J. Feng, "Deep forest," *Nat. Sci. Rev.*, vol. 6, no. 1, pp. 74–86, 2019.
- [51] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*. R. Meersman and Z. Tari, Eds. Berlin, Germany: Springer, 2004, pp. 492–508.
- [52] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, 2008, pp. 931–940.
- [53] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, 2011, pp. 287–296.
- [54] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2009, pp. 203–210.
- [55] G. K. Dziugaite and D. M. Roy, "Neural network matrix factorization," 2015, *arXiv:1511.06443*. [Online]. Available: <http://arxiv.org/abs/1511.06443>
- [56] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam, "Neural rating regression with abstractive tips generation for recommendation," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*. New York, NY, USA: Association for Computing Machinery, 2017, p. 345–354.
- [57] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [58] P. Kotschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1467–1475.
- [59] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.



**SOHEILA MOLAEI** received the M.Sc. degree in information technology from the University of Tehran, Tehran, Iran, where she is currently pursuing the Ph.D. degree. She is working on graph structured data, representation learning, and their applications in mining of large social networks.



**AMIRHOSSEIN HAVVAEI** received the M.Sc. degree in networked systems engineering from the University of Tehran, Tehran, Iran, in 2020. He is currently a Research Associate with the University of Tehran, where he works on recommender systems and social networks.



**HADI ZARE** received the Ph.D. degree from the Department of Computer Science, Amirkabir University of Technology, Tehran, Iran, in 2012. He is currently an Assistant Professor with the Faculty of New Sciences and Technologies, University of Tehran, Tehran. His current research interests include probabilistic graphical models, dimension reduction techniques, and discovering and summarizing big data structures, such as social networks.



**MAHDİ JALILI** (Senior Member, IEEE) received the B.S. degree in electrical engineering from Tehran Polytechnic, Tehran, Iran, in 2001, the M.S. degree in electrical engineering from the University of Tehran, Tehran, in 2004, and the Ph.D. degree in synchronization in dynamical networks from the Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland. He then joined the Sharif University of Technology as an Assistant Professor. He is currently a Senior Lecturer with the School of Engineering, RMIT University, Melbourne, VIC, Australia. His research interests include network science, dynamical systems, social networks analysis and mining, and data analytics.

Dr. Jalili was a recipient of the Australian Research Council DECRA Fellowship and the RMIT Vice-Chancellor Research Fellowship. He is an Associate Editor of the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING and an Editorial Board Member of the *Mathematical Problems in Engineering* and *Complex Adaptive Systems Modeling*.