IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 29, 2020, accepted July 30, 2020, date of publication August 4, 2020, date of current version August 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014051

Modeling Trust-Aware Recommendations With Temporal Dynamics in Social Networks

ADEEL AHMED[1], (Graduate Student Member, IEEE), KHALID SALEEM[1], (Member, IEEE), UMER RASHID[1], AND ABDULLAH BAZ[2], (Senior Member, IEEE)

[1]Department of Computer Sciences, Quaid-i-Azam University, Islamabad 45320, Pakistan
[2]Department of Computer Engineering, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21421, Saudi Arabia

Corresponding author: Adeel Ahmed (adeel.ahmed@cs.qau.edu.pk)

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by grant code 18-COM-1-01-0001.

ABSTRACT In recent years, user's trust has gained attention in recommender systems. Trust plays a vital role in the recommendation of online products. Trust is a dynamic feature which evolves with passage of time and varies from person to person. Trust-based cross domain recommender systems suggest items to the users usually by ratings, provided by similar users, often not available in the same domain. However, due to the sparse rating scores, recommender systems cannot generate up-to-the-mark recommendations. In this research, we solved a user cold start problem, mainly by modeling preference drift on a temporal basis. We tried to solve this problem by adopting one of the scenarios of cross domain of 'No Overlap' using cross domain information. In this work, we proposed a model called Trust Aware Cross Domain Temporal Recommendations (TrustCTR) that predict the rating of an item about an active user from the most recent time. We generated user features and item features by using latent factor model and trained the proposed model. We also introduced the concept of trust relevancy that shows the degree of trust, computed the trusted neighbors in target domain for an active user belonging to a source domain, and predicted the ratings of items for cold start users. We performed experiments on public datasets Ciao and Epinions and used these datasets in cross domain form such as the categories of Ciao as source domain and Epinions as the target domain. We selected five different domains, having a higher proportion of rating sparsity, for observing the performance of our approach using MAE, RMSE, and F-measure. Our approach is a viable solution of cold start problem and offers effective recommendations We also compared the model with state-of-the-art methods; the model generates satisfactory results.

INDEX TERMS Recommender systems, undirected graph, cross domain, ant colony system, social networks, matrix factorization, stochastic gradient descent, collaborative filtering.

I. INTRODUCTION

In the recent era, the information over the web is growing day by day and it is difficult for the users to find out the relevant information. The social networks and e-commerce applications such as Amazon, Alibaba, Epinions, Aliexpress, eBay, Walmart, and many others, shifted the information retrieval problem to filter out the personalized information. Recommender systems were developed in 90s to employ community opinions and identify the exciting content from a potentially overwhelming set of choices [51]. The recommender systems should have the ability to recommend a new item, based on (i) users' recommendations, and (ii) purchase history of other items belonging to the same category. In the case of a new user, since there is no history, recommender systems may depend upon the way that how new users behave and interact with the new items. The objective is to analyze items in which new users have possibly interested.

Collaborative-filtering methods estimate the ratings and recommend the items to users with similar preferences by considering previous recommendations [46]. The collaborative filtering techniques usually have lacks of information about user's behavior which limits their learning and predictive ability for generating quality recommendations [22].

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

The recommender systems are also suffered because of change in trust relationship among the users, with time [11]. Besides that, users' ratings about items by the recommender systems might be out-of-date and become noisy [12]. Matrix factorization techniques are better to compute the accuracy and popularity of preferences. They are based on latent factorized models [25]. However, these models have limitations; (i) they consider value of item features favored by all users [8], (ii) do not take into account the conditional preferences [35], and (iii) do not consider social effects on user preferences such that social relationships, social influence among peers and homophily effect [32], [75].

Data sparsity makes it difficult for collaborative filtering systems to generate accurate predictions [24]. The recommender systems usually provide recommendations in a particular domain facing sparsity problem, in addition to issues associated with new users and items [53]. In data sparsity, users can rate the few items or cannot assign a rating to them. It also refers to a new community problem, where ratings assigned to the items are low as compared to the number of items, so the useful recommendations are not possible [4].

Nowadays, the cold start problem is the key challenge faced by recommender systems [52]. In the cold start problem, the system cannot produce a recommended list of items to the users [56]. The cold start problem is associated with the situation, i.e., new user, item, and community [52], [80], [4]. This problem occurs when a new user arrives to the system and system do not have adequate information for performing similarity measures which degrades the quality of recommendations. In recent years, the cold start problem is addressed by employing hybrid models. Cold start recommender systems make use of auxiliary information, multimodal information, and side information to overcome the recommendation problems.

In a cross-domain scenario, trust-based recommenders recommend the items to the source users by exploiting target domains. Cross domain recommender systems introduce a new dimension to solve the mentioned problems; this can be done by transferring knowledge from a source domain to the target domain [8], [23]. Cross-domain recommender systems recommend the items to the users from items rating provided by similar users, which are not available in the same domain.

The more advanced types of recommender systems address cross-domain recommendation scenarios, where items related to source domains have been recommended in the target domains and vice versa [8]. In a cross-domain systems, two domains are considered distinct, and knowledge is transferred from a dense to sparse domain. Cross domain recommendation systems are based on four different scenarios; categorized into user overlap, item overlap, full overlap, and no overlap. In user overlap, some users are familiar with the source and target domains and have assigned ratings to items. In item overlap, some items are common in both domains and some user's rate items. In full overlap, all users and items are overlapped. In no overlap, there exists no common users and items across two domains [8].

According to sociology, trust is a set of expectations shared by all those involved in an interactions or networks [109]. The recommender systems based on trust are helpful in improvement of quality recommendations, solve data sparsity problem, and cold start problem.

In recommender systems, preferences are grouped into six main types; feature preferences [44], [77], feature value preferences [79], temporal dynamics [27], social influence [21], [38], [40], [76], [84], conditional preferences [35], user biases, and item biases [28]. Feature value preferences show the favorability of item values by users. Temporal dynamics refer to drift of user-items rating with time. Social influence is an effect of social communities on user preferences. Conditional preferences are the dependencies between the item features and their respective values. User biases and item biases give a higher rating to the items then other users and some items receive higher ratings than others, respectively. Ziegler *et al.* observed that trust and user preference similarity has a strong correlation [83]. If the two persons are more similar then there exists greater trust between them. For example, if people are interested in 'Book' on time at t_1 then they trust the experts in that domain, then after some time, say t_2, their preferences are changed and shifted to some other domain say 'Movies', then at that time, people trust experts in this domain. In the rating system, users can establish new trust relations and can add new ratings.

Our contributions are as under:

- In this paper, we propose a trust aware recommendation model using cross domain information for rating prediction, that solves a user cold start and sparsity problem in cross-domain scenario of 'No Overlap'.
- We propose a hybrid model called Trust-Aware Cross Domain Temporal Recommendations (TrustCTR) by integrating the neighborhood model and latent factor model with baseline estimates, and distance metric to understand the dynamics of user preferences in cross domain social networks.
- We improve the rating prediction process by relating the time with user preferences and learning the features of users, and items using proposed model.
- We propose a novel trust metric called Trust Relevancy that measures the degree of trust between two users by incorporating the trust information and the rating information and integrated this trust relevancy in rating prediction.
- We construct a trust network using implicit and explicit trust relationships w.r.t time and then find trustworthy neighbors for an active user, using Ant colony meta-heuristics.
- We train TrustCTR model by selecting the data on a time range t_1, and predict the ratings for test set from the most recent time that is t_2.
- We also investigated the applications for TrustCTR model such as trust prediction and rating prediction.
- The proposed model is evaluated on two public datasets that are Epinions and Ciao, using the categories of

these datasets in cross domain forms. We divide each domain in different views of users and is compare with well-known state-of-the-art methods and present improved results.

The main properties of the proposed model are listed as follows:

- Likewise, the latent factor model [25], TrustCTR also utilizes the user preferences but the latent factor model does not consider the trusted network and their influences whereas proposed model incorporates trust information, explicitly and implicitly, making it more effective.
- In [61], the neighborhood model has not incorporated the item characteristics and user preferences whereas the TrustCTR consider the user-item preferences, although both models take into account the trust information.
- The existing baseline estimates [26] and social recommendation model based on distance metric [74] have not included the time-sensitive information. Whereas the TrustCTR not only accounts for the time-sensitivity but it also integrates the distance metric with baseline estimates, that learns the data points (locations) closer to each other. The distance propagation shows that how users and items are located inside space by integrating distance metric learning and latent factor model. These locations of users and items are helpful in determining the users who have few ratings.
- TARS [3] is a trust-based recommendation framework based on ACO, that utilizes the rating values for creating an implicit trust graph. It does not consider the explicit trust relationship in generating recommendations that effect its performance when facing sparse data. While the TrustCTR incorporates explicit trust relationship among users from social networks for improving prediction process and implicit trust relationships in the form of cosine similarity and pheromones, this improving the ability of proposed model to tackle the sparsity issue, by choosing best neighbors.
- TCFACO [49] is a trust aware recommendation model that incorporated both explicit and implicit trust information with user-item preferences but it does not account for the dynamicity of trust and users' preferences. While the TrustCTR model has the capability for generating recommendations from most recent times and incorporated the up-to-date trust and rating information.
- One of the modules of TrustCTR is 'Discovering Neighbors' that finds the best neighbors for an active user using ACO. We choose ACO because it is dominant over genetic algorithms and simulated annealing approaches, as the convergence time of metaheuristics algorithm is faster than genetic algorithms and simulated annealing approaches. ACO has the capability of adapting the changes continuously when the graph changes dynamically. Ants can find a high-quality solution in a solution space and they share their knowledge in the form of pheromone as update strategy and solve the problem efficiently.

Following is the organization of the article. Section 2 discusses the background and the motivation behind our research. We describe the architecture for proposed model in section 3. Applications of TrustCTR Model are described in Section 4. The experimental setup is described in section 5. Evaluation and experimental results are discussed in Section 6 and finally, we describe conclusion in Section 7.

II. BACKGROUND AND MOTIVATION

In this section, we discuss the recommendation tasks and scenarios; give details of the state-of-the-art in trust-based and cross-domain recommendations; issues in existing recommender systems and motivation behind our research work.

A. RECOMMENDATION TASKS AND SCENARIOS

Recommendation task refers to the method required to give recommendations for user tasks. The recommendation tasks can be performed for both source and/or target domains. In recommendation scenarios, knowledge is transferred from source to target domain based on similarity between users and items. The recommendation systems are continuously evolving. The features related to items, users, and ratings are utilized in the development of recommender systems. Recommendations can be divided into three scenarios namely single, cross, and multi-domains (i) single domain recommendation considers the items belong to target domain, and are recommended to the users of target domain by learning knowledge or profile information available in source domain, (ii) cross domain recommendations consider items belong to source domain, are recommended to the target domain users or if items belong to target domain, are recommended to the users of source domain. (iii) multi-domain, recommends the items to the users of source domain from more than one target domain [8]. A domain can also be categorized into the system, data, and time scenarios; (i) in system domain scenario, two domains are considered as distinct if knowledge is transferred from dense domain to sparse domain; (ii) in the data domain, users interact with items in the form of ratings or likes/dislikes from multiple dimensions related to different data domains; (iii) in time domain rating matrix has timestamps, divided into time slices [70], [48]. Each time slice is called a temporal domain [34]. A domain can be defined at item/category, type, or attribute level (i) item level or category domain, majority of item characteristics are different, (ii) type level attributes of two items are different from each other and some attributes are similar, (iii) attribute level, two items are considered as an indifferent domain when they have different attribute values [58].

This research work is limited to cross domain scenario at item level (category) and temporal domains.

B. CROSS DOMAIN RECOMMENDER SYSTEMS

Xu *et al.* proposed a method for solving a cold start problem in cross-domain recommendation scenario [71]. The method predicted the rating of items for source users via target domain items using trust relations. Gao *et al.* developed a latent factor model based on clustering and improved the cross-domain recommendations to learn the patterns of rating using a non-negative tri-factorization matrix [16]. Cremonesi *et al.* solved the problem of sparsity by transfer learning from the source to the target domain to improve recommendation accuracy and observed that item-based approaches provide better recommendation quality, then user-based approaches [9]. Sahebi *et al.* addressed a cold start problem in a cross-domain scenario and found that cross-domain recommender systems give more accurate results as compared to multi-domain recommendation systems [53]. Kotkov *et al.* generated recommendations when items overlapping occur in cross domains at system level and observed that source domain could increase the performance of recommendations in the target domain when there is item overlap, but performance is decreased when items overlap occurs in different domains [29]. Moe *et al.* developed a context-based cross domain recommender system to recommend cosmetics products using contextual features of the customer [45]. Li *et al.* solved the problem of data sparsity by transferring patterns from auxiliary domain to other domains [33]. Abdollahi *et al.* proposed a cross-modal recommendation framework called Asymmetric Non-Negative Factorization (NMF), for solving a new item problem by utilizing multiple domains [1]. Elkahky *et al.* introduced a multi-view deep learning approach in which they modeled user preferences in cross-domain recommendation systems [15]. Zhuang *et al.* developed a framework known as TRACER that predicted the rating for cold-start users by transferring knowledge from multiple source domains [82]. Ma *et al.* developed a joint cross domain recommendation framework that incorporated transitive trust relations among users with context dependency for solving a data sparsity problem [37]. Li et al transferred the latent information through orthogonal mapping function between two domains via dual learning mechanism using autoencoder [95]. He *et al.* proposed a cross domain model by learning user and items latent information across domains for knowledge transfer and improved the prediction process by embedding the probabilistic modeling with Bayesian neural network [96]. Lin *et al.* given the concept of cross platform in social e-commerce systems and recommended the items to the active user when they shop the items via social media [97]. In 2019, Liu *et al.* proposed a cross domain recommendation model called Joint Spectral Convolutional Network and utilized the neighborhood information across domains and improved the recommendation process [98]. In 2018, Hu *et al.* developed a model called CoNet that used the hidden layers of neural network in cross-connections form and transfer the knowledge across domains for improving the performance of recommendation systems and overcome the issue of data sparsity [99]. Yuan *et al.* proposed a cross domain framework called deep domain adaptation model that extracts the features shared from two domains using autoencoder and then the rating predictor split the ratings of each domain [100].

C. TRUST-BASED RECOMMENDATION SYSTEMS

Lathia *et al.* proposed trust-based recommender systems by using the k-nearest neighborhood algorithm and the utility of rating information and trust information [31]. Goldback *et al.* have determined trust among users based on features extracting from profile similarity [17]. Dohetry *et al.* investigated trust-based recommendations using traditional algorithms and trust-based models [47]. Jamali *et al.* overcome the problem of cold-start users who are connected to a trusted network. They applied trust-based methods to generate item recommendations and suggested that few ratings give low precision because there exists weak trust when considering the rating of indirect neighbors [20]. Bedi *et al.*, proposed a model called TARS using an ant system to measure the dynamic trust between users using pheromones activity among ants and select the best neighborhoods [3]. Massa *et al.* proposed a model that used the trust metric to solve the data sparsity problem in recommendation systems [42]. Deng *et al.* applied a deep learning technique and developed a model called DLMF and solved the cold start problem in trust-aware recommendation systems [12]. Than *et al.* overcome the weaknesses of the collaborative filtering algorithms and proposed the architecture for the recommendation system that integrated the similarity with trust and reputation [65]. Carrasco *et al.* developed a trust-aware model using a direct/indirect trust relationship to overcome the problem of data sparsity [6]. Tang *et al.* performed a trust evaluation of product review sites and proposed a framework called eTrust by incorporating the user preferences that change with time [62]. Deng *et al.* developed a model called Relevant Trust Walker that solves the user cold-start problem [11]. Massa *et al.* presented the architecture for Ski mountaineering application that reported the users about snow conditions by incorporating trust [41]. Chen *et al.* proposed a recommended method to solve the cold start problem for new users of e-commerce by including trust/distrust network model [7]. Walter *et al.* introduced an ensemble model to integrate the trust of users and their friends to overwhelm the problem of data sparsity. They highlighted factors such as knowledge sparseness and network density, which affects the performance of recommender systems [67]. Zou *et al.* analyzed the review contents based on sentiments and items rating to find the purchasing actions on items of the user [84]. A trust-based recommender system was developed by incorporating Ant colony optimization and k-means [54]. Yuan *et al.* proposed a trust-aware recommendation model based on genetic algorithms that gives a better prediction, maximum coverage, and minimum maintenance cost [73]. Parvin *et al.* proposed a trust aware model called TCFACO and predicted the rating for cold-start users by using the ACO [49]. Rezaeimehr *et al.* proposed a model based on time-aware recommendations called TCARS. They first predicted initial ratings and then identify neighbors using community detection methods and overcome the challenges of sparsity [85]. In 2018, Yadhav *et al.* tried to solve the data sparsity problem and address the issue of how to prevent malicious attacks [86]. Pan *et al.* proposed a deep learning-based trust aware recommendation corelative denoising autoencoder. They used autoencoder to learn useful features from ratings and incorporated the social information [87]. In 2018, Parvin *et al.* integrated the trust and distrust relations and non-negative matrix factorization approach and solves the problem of cold start and data sparsity [88]. In work [89], Jiang *et al.* constructed a slope one algorithm for e-commerce recommendation systems by fusing user similarity and trust information. In 2020, AutoTrustRec is proposed by Bathla *et al.* that incorporated the directed and undirected trust information with autoencoder by learning the nonlinear relationship of neural networks present between the ratings and trust information [90]. Dong *et al.* describe the survey based on deep learning techniques for trust-based recommendations with three aspects such as explainable, social awareness, and robustness in recommender systems [91]. Son *et al.* proposed a trust aware model for social IoT. They incorporated implicit trust to overcome the limitations of explicit trust relation using environment of social IoT for prediction process [92]. Belkhadir *et al.* combined social trust information with social regularization by discovering an aggregate path from the multiple trust paths for generating recommendations [93]. Moradi *et al.* developed a trust based collaborative filtering model that incorporated trust information and predicted ratings about items [94]. Davoudi *et al.* proposed a social trust model for recommender systems and analyzed the impact of social ties, preference similarity, and different centrality measures on rating prediction. They observed that more similar users trust more on each other [101]. Deebak *et al.* developed community-based recommendation framework over IoT and solve the issues of cold start and data sparsity. The recommendations systems in IoT which contain applications related to big data allows to analyze a large amount of data to select suitable information. They learned the trusted neighborhood information for active users by finding similar preferences of target users within a community [102]. Lee *et al.* address a cold start problem by discovering the latent features between the trustor and trustees. They noticed that those trustors who follow the same trustee, having similar tastes and shared common features that improve the performance of social-based recommendations [103]. Ardissono proposed a LOCABAL + model that is a trust-based recommendation model by defining a multifaceted trust by users feedback and generated top N recommendations [108]. Noh *et al.* developed a trust aware model and studied the impact of power users available in social networks and exploited the trust cluster and observed that clusters with normal user provided better results [104]. Guo *et al.* developed a support vector regression model based on multiview clustering method to form clusters from user's similarity and trust information for solving cold start problem [107]. Sheugh *et al.* proposed a recommendation model to reduce the data sparsity issue and observed that with the increase of number of clusters, the coverage values decrease monotonically [105]. Zhang *et al.* developed a recommendation model by utilizing a two-layer neighborhood scheme for selecting trusted neighbors and improved recommendation accuracy [106].

D. ISSUES AND MOTIVATION

Data sparsity is one of the main issues in generating recommendations, also referred to as a new community problem [4]. Users rate only a few items due to which the resultant user-item rating matrix may be sparse, which leads to poor quality of recommendations [84]. Another problem is the user and item cold start. The new item problem occurs when a new item is added to the recommender system without any ratings. As it does not have any ratings, it cannot be compared to other items, which leads to the item not being included in any recommendations. The new user problem occurs when a new user joins the recommender system. The new user has not provided any preferences in the form of ratings, and so there are no preferences to compare the new user to existing users or items and the recommender system not able to generate any recommendations [56]. Providing recommendations to users with limited history becomes a difficult problem for collaborative filtering models because of their learning and predictive ability due to data sparsity [22]. Locating people with similar neighbors is also a challenge since rating prediction depends upon these neighbors [84]. Trustworthiness is another issue, for example, fake feedback or noisy information for malicious purposes harm the prediction accuracy [11]. Nowadays, time sensitivity is one of the critical issues that occur in personalized recommendation systems. Trust relationship can change with time and ratings assigned by users, is also vary with time and may become noise or out of date [11], [12].

Based on the literature review and above-mentioned challenges, our motivation is to provide quality recommendations to a new user available at source domain from up-to-date rating profile of his/her neighbours present in target domain. We assumed that these recommendations are generated by incorporating trust from cross-domain information available in social networks by following a cross-domain scenario of 'No Overlap'.

III. PROPOSED MODEL ARCHITECTURE

Mostly the recommender systems cannot consider the temporal preferences and trust drifts [76]. We assumed that user and item biases, temporal dynamics, and social influence over features could be subjected to temporal preference drift. Thus, we addressed one major question in this research; ''How can we model the rating prediction while incorporating the influence of contextual factors such as time, preferences, and trust when there exists no common users or items across two domains, for solving user cold start problem?''.

We propose architecture for Trust Aware Cross Domain Temporal Recommendations (TrustCTR) model, as shown in Fig. 1. This architecture consists of four main modules

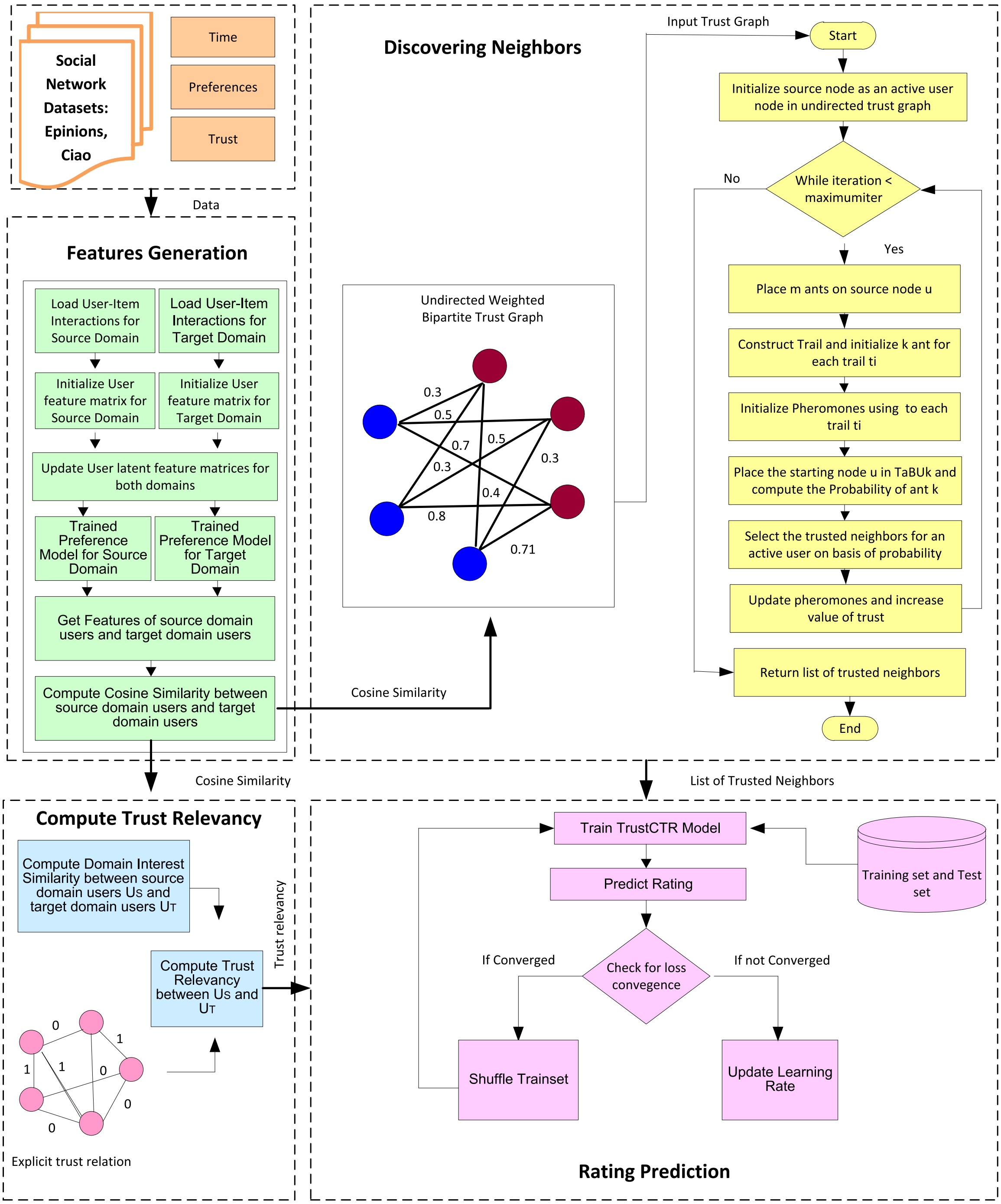
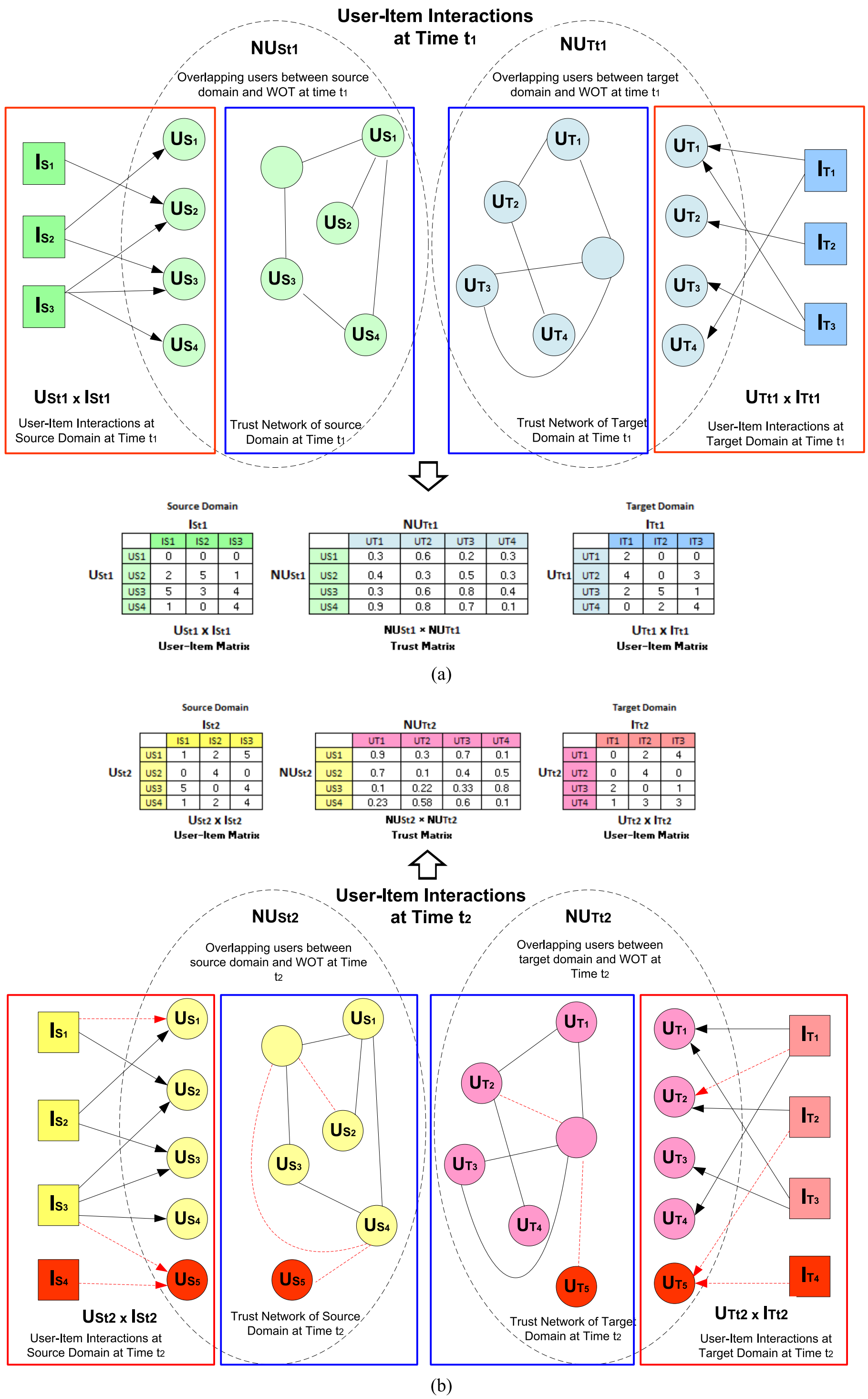


FIGURE 1. Proposed architecture for TrustCTR model.

responsible to generate features; compute trust relevancy, discover trusted neighbors, and predict rating.

A. PROBLEM FORMALIZATION

The rating system of product review sites that shows the user-item interactions in cross-domain that is source domain and target domain at time t_1 and time t_2, as shown in Fig. 2(a) and Fig. 2(b), respectively. Let $U_{S_{t_1}} = \{U_{S1}, U_{S2}, \ldots, U_{Sn}\}$ is the set of users at time t_1, $I_{S_{t_1}} = \{I_{S1}, I_{S2}, \ldots, I_{Sm}$ is the set of items at time t_1 in source domain. Also, in target domain $U_{T_{t_1}} = \{U_{T1}, U_{T2}, \ldots, U_{Tn}\}$ be the users at time t_1 and $I_{T_{t_1}} = \{I_{T1}, I_{T2}, \ldots, I_{Tm}\}$ is the number of items rated by users at time t_1 in the target domain. Similarly, the set of users and items exist in the source domain and target domain for time t_2. $[\mathrm{U_{St1}} \times \mathrm{I_{St1}}]$ and $[\mathrm{U_{Tt1}} \times I_{\mathrm{Tt1}}]$ are the user-item matrices for source domain and target domain, respectively.



USt1	IS1	IS2	IS3
US1	0	0	0
US2	2	5	1
US3	5	3	4
US4	1	0	4

NUSt1	UT1	UT2	UT3	UT4
US1	0.3	0.6	0.2	0.3
US2	0.4	0.3	0.5	0.3
US3	0.3	0.6	0.8	0.4
US4	0.9	0.8	0.7	0.1

UTt1	IT1	IT2	IT3
UT1	2	0	0
UT2	4	0	3
UT3	2	5	1
UT4	0	2	4

USt2	IS1	IS2	IS3
US1	1	2	5
US2	0	4	0
US3	5	0	4
US4	1	2	4

NUSt2	UT1	UT2	UT3	UT4
US1	0.9	0.3	0.7	0.1
US2	0.7	0.1	0.4	0.5
US3	0.1	0.22	0.33	0.8
US4	0.23	0.58	0.6	0.1

UTt2	IT1	IT2	IT3
UT1	0	2	4
UT2	0	4	0
UT3	2	0	1
UT4	1	3	3

FIGURE 2. (a). Example of Rating System at time t_1 in cross domain consists of user-item rating matrix and trust network of both source and target domains where the common users are extracted for building a trust matrix (trust network) at time t_1. (b). Example of Rating System at time t_2 in cross domain consists of user-item rating matrix and trust network of both source and target domains where the common users are extracted for building a trust matrix (trust network) at time t_2. In source domain at time t_2, there is a new user Us_5 assigned ratings to the new item Is_4 and existing item Is_3, and an existing user Us_1 assigned new rating to an existing item Is_1 (as shown by red dotted arrow lines). The user Us_5 established a new trust relation with an existing user Us_4 (as shown by red dotted lines) in trust network. Moreover, there exists some other new trust relationships among other existing users in a trust network. Similarly, in target domain at time t_2, there is a new user U_{T5} assigned ratings to the new item I_{T4} and existing item I_{T2} (as shown by red dotted arrow lines) and an existing user U_{T2} assigned new rating to an existing item IT_1. User U_{T5} established a new trust relation with other existing user (as shown by red dotted line) in a trust network.

For trust network at time t_1, we constructed an undirected trust graph at time t_1, since we assumed that there is no overlap between source domain and a target domain and there exists a joint factorization of source domain and target domain rating matrices. Therefore, we have $[\mathrm{NU}_{\mathrm{S}_{t_1}} \times \mathrm{NU}_{\mathrm{T}_{t_1}}]$ matrix, where $\mathrm{NU}_{\mathrm{S}_{t_1}}$ is the number of common users in source domain and WOT (Web of Trust), $\mathrm{NU}_{\mathrm{T}_{t_1}}$ be the number of common users in target domain and WOT, that is weighted trust matrix creation. Similarly, we have created the undirected trust graph from a trust network at time t_2 such that $[\mathrm{NU}_{\mathrm{S}_{t_2}} \times \mathrm{NU}_{\mathrm{T}_{t_2}}]$.

New users may be added and assigned a rating to the existing or new item at time t_2, thus $I_{S_{t_1}} \subseteq I_{S_{t_2}}$ and $I_{T_{t_1}} \subseteq I_{T_{t_2}}$, similarly, there are new trust relations that can be established, thus $U_{S_{t_1}} \subseteq U_{S_{t_2}}$ and $U_{T_{t_1}} \subseteq U_{T_{t_2}}$.

We can formulate a user cold start problem for a source user belonging to $U_{S_{t_2}}$ there exists some trustable neighbours from $U_{T_{t_2}}$ who have ratings related to a particular item in the target domain at time t_2. Then, for each active user $U_S \in U_{S_{t_2}}$, the goal is to predict the rating of item from target domain at time t_2, by incorporating trust. For training of our proposed model, we have used the rating system at t_1 as shown in Fig. 2(a) and for test set, we utilized the rating system to generate recommendations for the active user, at time t_2, as shown in Fig. 2(b).

B. GENERATE FEATURES

This module randomly initializes the user feature and item feature vectors. After initialization, d-dimensional user feature and item feature vectors are updated for both source and target domains. Algorithm 1 describes the procedure of feature generation for both source and target domains.

In recommendation systems, the large size rating matrix also increases the dimensionality of the user rating vector, so most of the users have assigned a few ratings or missing data scores. In this situation, we have adopted a technique of matrix factorization mainly to reduce dimensions in user rating data. Matrix factorization [44], [27], [18], [57] decomposes $M \times N$ rating matrix R into two latent feature matrices of users P and items Q with reduced dimensionality d such that both users and items are present in a single latent feature space and modeled the user-item interactions as inner products in latent feature space as given in equation (1). These two matrices are P_u and Q_i.

$$\mathrm{R} \approx P_u Q_i^T \tag{1}$$

where $P_u \in \mathrm{R}^{\mathrm{Mxd}}$ is the latent feature matrix of users and $Q_i \in \mathrm{R}^{\mathrm{Nxd}}$ is the latent feature matrix of items. For example, if we have a given matrix of user's rating that is described in Fig. 3, then the resultant matrix R gives the predicted or observed rating which is missing in the user-rating matrix by using these features. Initial values of P_u and Q_i are produced randomly or assigned manually. For example, the elements of P_u and Q_i are given in the form of a matrix. These elements of P_u are obtained randomly or assigned manually for each item i and user u, which shows that how much user is interested in items and Q_i having the degree to which item i keep those features, Q_i value is also generated randomly or assigned randomly. Then the predicted rating matrix is obtained by dot product of P_u and Q_i. After the initialization step, item feature and user feature vectors are updated using equation (4) and (5), respectively. So, after multiple iterations, we obtained accurate feature vectors for both source and target domains.

Algorithm 1 Features Generation

Input: (User-Item rating matrix 'R')
Output: (d-dimensional features vector of users and items)

```
P ← random(R) // randomly assign user features
Q ← random(R) // randomly assign item features
loss ← 0
for each entry in R do
user, item, rating ← entry
    u ← user [user]
    i ← item [item]
    estimate error according to equation (3).
    compute loss using equation (2).
      P ← P[u]
      Q ← Q[i]
      update item latent vector using equation (4).
      update user latent vector using equation (5).
end for
return user feature vector P and item feature vector Q
```

Algorithm 1 describes the process of learning latent feature vectors for users and items. Latent features are constructed for items and users of both source and target domains. To avoid over-fitting, a regularized model is used during the learning of feature vectors to minimize the squared error. Matrix factorization learns these latent features by reducing the following term that is a stochastic gradient algorithm [26],

$$\min_{Q,P} \sum_{(u,i)\epsilon k} (R_{ui} - Q_i^T P_u)^2 + \lambda \left(\|Q_i\|^2 + \|P_u\|^2 \right) \tag{2}$$

where, R_{ui} is the known ratings and learned parameter, λ controls the degree of regularization and can be deduced by cross-validation. Equation (2) is minimized by using stochastic gradient descent (SGD) and equation (3) is used for computing the error in true rating and the observed rating.

$$E_{ui} \leftarrow R_{ui} - Q_i^T P_u \tag{3}$$

Then SGD model adapts the parameters by a magnitude proportional to γ in opposite direction of the gradient:

$$Q_i \leftarrow Q_i + \gamma\,(E_{ui} P_u - \lambda Q_i) \tag{4}$$

$$P_u \leftarrow P_u + \gamma (E_{ui} Q_i - \lambda P_u \tag{5}$$

The output of the module'Generate Feature' gives d-dimensional feature vectors for users and items for both source and target domains.

$$
\begin{bmatrix} 5 & 3 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 1 & 1 & 0 & 5 \\ 1 & 0 & 0 & 4 \\ 0 & 1 & 5 & 4 \end{bmatrix}
\approx
\begin{bmatrix} -0.849 & 1.044 \\ -0.851 & 0.549 \\ 0.694 & -1.085 \\ 0.745 & -0.541 \\ 1.053 & 0.268 \end{bmatrix}
\times
\begin{bmatrix} -1.531 & 0.593 \\ -1.245 & -0.131 \\ 0.704 & 0.931 \\ 0.822 & -1.432 \end{bmatrix}
=
\begin{bmatrix} 1.919 & 0.920 & 0.374 & -2.193 \\ 1.628 & 0.988 & -0.088 & -1.486 \\ -1.706 & -0.722 & -0.522 & 2.124 \\ -1.461 & -0.857 & 0.021 & 1.387 \\ -1.453 & -1.346 & 0.991 & 0.482 \end{bmatrix}
$$

Rating matrix R **User feature matrix $\mathbf{P_u}$** **Item feature matrix $\mathbf{Q}_i$** **Predicted rating matrix $\mathbf{R} \approx \boldsymbol{P_u}\, \boldsymbol{Q_i}^{\boldsymbol{T}}$**

FIGURE 3. Example of matrix factorization.

C. COMPUTE TRUST RELEVANCY

The process of this module is computing trust relevancy between the users of the source domain and target domain. In social networks, a direct trust relationship between two users does not necessarily improve the prediction accuracy [11]. The active users present in the source domain and trusted users of target domain might differ in preferences, interests, and perception, therefore, we defined trust relevancy that shows the degree of trust present between two users.

We estimated the cosine similarity [59] between two users, i.e., the user of the source domain and user of the target domain by incorporating the user feature vectors of both domains. We computed the similarity using equation (6). Here $\vec{u}$ is the user feature vector of source domain and and $\vec{v}$ is the user feature vector available at target domain. Both of these feature vectors are obtained as an output of Algorithm 1.

$$sim\,(u_S, u_T) = cos(u_S, u_T) = \frac{\vec{u_S}.\vec{u_T}}{||\vec{u_S}||.|\,|\vec{u_T}\,||} \tag{6}$$

The domain interest similarity between pair of users for source and target domains shows that how many people belong to two different domains who trust with each other and they have more similar domain interests. So, we considered both domains for finding domain interest similarity. In equation (7), $fd_u\,(k)$ shows the probability that user u is interested in domain k. n_u represents the number of products rated by user u and $n_u(k)$ is number of products rated by user u, available from kth domain.

$$fd_u\,(k) = \frac{n_u(k)}{n_u} \tag{7}$$

For finding, domain interest similarity between the source and target users, we used Jensen Shannon Divergence[10] between users of two domains in the form of domain interest distributions that is fd_S,fd_T:

$$simf_{\,dist}\,(u_S, u_T) = \sqrt{D_{KL}\left(fd_S||m\right) + D_{KL}(fd_T||m)} \tag{8}$$

Here 'm' shows the average of two domain distributions and is computed using equation (9), as:

$$m = \frac{1}{2}\left(fd_S + fd_T\right) \tag{9}$$

Algorithm 2 Trust Relevancy

Input: (u_S, u_T, sim)
Output: (trust relevancy between $u_S$andu_T)
1: compute $\mathrm{fd_S}(k)$ according to equation (7).
2: compute $\mathrm{fd_T}(k)$ according to equation (7).
3: calculate m using equation (9).
4: estimate domain interest similarity i.e.
$\mathrm{f_{dist}}\,(u_S, u_T)$ betweenu_S and u_T using equation (8).
5: compute trust relevancy using equation (11).
6: **return**$tR\,(u_S, u_T)$

And D_{KL} is the Kullback-Leibler Divergence [30] and is computed as:

$$D_{KL}\left(fd_S||m\right) = \sum\nolimits_k fd_S\,(k) \log \frac{fd_S(k)}{m(k)} \tag{10}$$

Thus, we defined the concept of trust relevancy between the user u_S belongs to source domain and user u_T of target domain at time t_1 that is $tR^{\boldsymbol{t_1}}\,(u_S, u_T)$. It is the product of $simf^{\boldsymbol{t_1}}_{dist}\,(u_S, u_T)$, the domain interest similaritybetween user u_S and user u_T before time t_1, the trust relationship that exists between users u_S and u_T at time t_1, and $sim^{\boldsymbol{t_1}}\,(u_S, u_T)$, the cosine similarity between the two users before time t_1. The trust relationship shows the degree of trust that is provided by users, explicitly or can be calculated using some algorithms. In this work, we considered the degree of trust directly from the datasets. Algorithm 2 describes the steps involved in computing trust relevancy. We have formulated the equation (11), for trust relevancy, $0 \leq tR \leq 1$, as:

$$\begin{aligned} tR^{t_1}\,(u_S, u_T) = {} & simf^{t_1}_{dist}\,(u_S, u_T) \\ & \times trustRelation^{t_1}\,(u_S, u_T) \\ & \times sim^{t_1}\,(u_S, u_T) \end{aligned} \tag{11}$$

D. DISCOVERING NEIGHBORS

In this module, we find trusted neighbors of an active user using ant colony optimization in target domain. We presented a trust network in the form of a weighted undirected bipartite graph. We incorporated the implicit trust relationship among the users in the form of cosine similarity as trust weight computed using equation (6). This trust weight is updated in the form of a pheromone update strategy. For applying ACO, it is required to represent trust network as an undirected bipartite graph G = (V,E), where V represents the user(node), and E represents the undirected edge having initial trust value computed using equation (12). Fig. 4 shows the user's connectivity as an undirected weighted bipartite trust graph. This graph consists of six users; source user and target users. Each edge has the trust weight in the form of similarity value, between two nodes. Ants move back-and-forth in trust graph for constructing solutions [110].

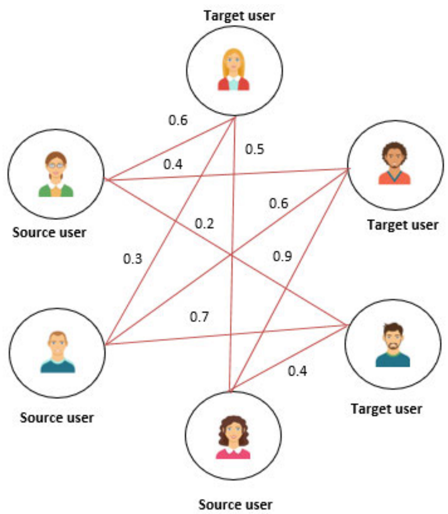


FIGURE 4. **Undirected weighted bipartite trust graph.**

We have considered the initial trust value for trust as initial pheromone value, which is computed by using modified equation (12). Algorithm 3 is derived from the ant colony system proposed by Dorigo *et al.* [14]. In ACO method, ants start search process in a solution space by investigating the population of ants to find the best quality solution. In this algorithm, the number of k ants are placed at active user node u in source domain towards users exist in target domain.

$$\tau_{xy}^{0} = \frac{\tau_{xy}}{n(u_x)\sum_{z\epsilon WOT_x} \tau_{xz}} \tag{12}$$

τ_{xy} is the trust value from user x to user y, nu_x is the number of users present in the web of trust for user x and $\mathrm{WOT_x}$ is the web of trust for user x. We utilized the initial pheromone value for initializing the trail t_i with respect to user trust for each ant k. Each k ant moves from current node x to neighbor node y by calculating the probability of ρ_{xy}^{k} of crossing the edge using the following equation (13),

$$\rho_{xy}^{k} = \frac{(\tau_{xy})^{\alpha}(n_{xy})^{\beta}}{\sum\limits_{z\epsilon N_x^k} (\tau_{xz})^{\alpha}(n_{xz})^{\beta}} \tag{13}$$

where N_x^k are the neighbors of ant k. The probability ρ_{xy}^{k} from node x to node y depends upon two parameters, one is pheromone level τ_{xy} and other is desirability of moves from node x to y, which is denoted by η_{xy}. α and β are used to control the influence of $\tau_{xy}\tau_{\mathrm{xy}}$ and η_{xy}. After calculating the probability, the ant k moves along with the edge that has highest probability. The ants stop their solution if there exist no more edges for traversing. This process is repeated up to a certain number of iterations when solution converges. In last iteration, each k ant preserves the record of trusted neighbors in $N_V(u)$. Only the best solution, globally updated after iteration is completed. The global pheromone level is computed using equation (14).

$$\tau_{xy} = (1-\rho)\,\tau_{xy} + \frac{\sum\limits_{k=1}^{m} \Delta\tau_{xy}^{k}}{L^k} \tag{14}$$

where m is number of ants and $\Delta\tau_{xy}^{k}$ is the amount of pheromone deposited by ant k at edge xy in one iteration. L^k is the length of trail t_i that the kth ant constructed. To find trusted neighbors we set different parameters in Algorithm 3 which are ants k=4, maxIter = 100, α = 3, β = 2, and ρ = 0.01. These parameter settings are chosen on the basis of convergence of proposed algorithm. Algorithm 3 describes the process that how we chose the quality raters for an active user available in the source domain, by incorporating trust between the users of the source and target domains.

Algorithm 3 Trusted Neighbors Discovery

Input: (trust matrix or undirected bipartite trust graph, ants, iteration, α, β, ρ, Q)
ants: number of ants, α: pheromone influence factor, β:local node influence factor, ρ: pheromone evaporation coefficient, Q:pheromone deposit factor, $TaBU_k(u)$ is a list that maintains the record of each visited node by ant k and prevents the ant k to visit the node again in each trail t_i.
Output:(list of trusted neighbors for an active user u)
1: Initialize $u \leftarrow sourcenode$
2: m $\leftarrow$ numofants
3: Place m ants on source node u.
4: Set iteration $\leftarrow$ 0
5: **while** iteration < maxIter **do**
6: Create a trail t_i by initializing k ant for each trail t_i
7: Generate random order of target trusted users in the trail for each ant k using Fisher-Yates shuffle.
8: Initialize pheromones to each trail t_i by using equation (12), at time t.
9: Place the starting node u in $TaBU_k(u)$, where k =1 to m.
10: Compute the probability of ant k in Graph G from active user node u to select node j using equation (13).
11: Select trustworthy neighbors for active user u on the basis of probability and store these into $N_V(u)$.
12: Update pheromones and increase the value of trust at time $t+1$ using equation (14).
13: **end while**
14: **return** list of trusted neighbors $N_V(u)$, for an active user u.

E. RATING PREDICTION

This module consists of two sub-modules; to train the TrustCTR model and rate the item for an active user (user belongs to source domain) as described in Algorithm 4 and Algorithm 5, respectively.

Algorithm 4 Trained TrustCTR Model

Input: (training set, test set)
Output: (trained model)
iteration ← 0
while iteration < maxIter **do**
loss ← 0
  for each entry inTraining set **do**
  user, item, rating ← entry
  u ← user [user]
  i ← item [item
  predictedvalue ← Rating Prediction (user, item)
  if predictedvalue ≠ Null **then**
    error ← rating– predictedvalue // compute using equation (26)
  else
    error ← rating–μ
  end if-else
  compute distance using equation (22).
  update b_u according to equation (24).
  update b_i according to equation (25).
  update H according to equatio. (29).
  update W using equation (30).
  update p_u using equation (27).
  update q_i using equation (28).
 end for
 iteration ← iteration + 1
 compute loss using equation (23).
 if isConverged(loss) **then**
   check for loss whether it is converged or not
   break
 end if
end while

A baseline model that is a latent factor model for rating prediction is discussed in [25], which depends on user preferences p_u^t and characteristics of an item q_i.

$$\hat{r}_{ui}^t = q_i^T p_u^t \tag{15}$$

Tang *et al.* presented a model for rating prediction for an item j of user i, by incorporating multifaceted trust in the neighborhood model [61].

$$\hat{r_{ij}^t} = \frac{\sum_{v \in N_i^t} \sum_{k=1}^{k} w_{ivk}^t q_j(k)\, r_{vj}}{\sum_{v \in N_i^t} \sum_{k=1}^{k} w_{ivk}^t q_j(k)} \tag{16}$$

Here in Eq. (16), N_i^t represents the group of people trusted by users i at time t at facet k. w_{ivk}^t is the trust strength between two user i and trusted users v at time t. r_{vj} is the decay rate and is calculated by using equation (17).

Algorithm 5 Rating Prediction

Input: (u,i,P,Q)
Output: (predicted rating)
if P contains user u and Q contains item i **then**
  compute distance using equation (22).
  discover trusted neighbors according to Algorithm 3.
  estimate rating using equation (20).
  return predict_rating
else
  returnμ
end if-els

$$\mathrm{r_{vj}} = \mathrm{e}^{-\eta_i(\mathrm{t}-\mathrm{t_{vj}})}\mathrm{r_{vj}^{t_{vj}}} \tag{17}$$

where value of $\eta_i \geq 0$ that controls the user-specific decay rate computed by using equation (17) for u_i and is learned from the data. The effect of u_v to u_i on rating given to item j at time t is the distance between t_{vj} and t that is formulated as $\mathrm{r_{vj}^{t_{vj}}}(\mathrm{t_{vj}} < \mathrm{t})$. It means that earlier ratings show the users earlier preferences and has fewer influences on current ratings when t and t_{vj} are so closed then the more impact of $\mathrm{r_{vj}^{t_{vj}}}$ on $\mathrm{r_{ij}^{t}}$ [13]. Equation (18) shows the baseline estimates that are used to obtain the interactions between users and items, which give different rating values [26]. These variations in rating values are related to users and items called user and item biases, respectively. These variations exist due to system tendencies such that some items received higher ratings than others and some received lower ratings than others. So, the recommendation system identifies the true portion of interactive values and for which first-order approximation of bias included in the rating r_{ui} is defined as:

$$\hat{r_{ui}} \leftarrow \mu + b_i + b_u \tag{18}$$

The predicted rating consists of average rating, item bias and user bias denoted by μ, b_i and b_u, respectively, where b_u and b_i shows the observed deviations of user u and item i, from the average.

Equation (19), gives us the distance metric described by Wu *et al.* which shows that the optimal distance metric can increase the performance of learning features [68], [69]. The distance metric is divided into two categories: first is the local distance metric, and the second is the global distance metric [72]. Matrix 'A' calculates the distance between two points x and y, where A $\geq$ 0 denotes that A is a positive definite matrix.

$$d_A^2(x, y) \leftarrow \|x_u - y_i\|_A^2 = (x - y)A\,(\text{x-y})^T \tag{19}$$

where x_u is the latent factor of user u and y_i is the latent factor of item i. 'A' is a positive semi-definite matrix, and $A \epsilon R^{kxk}$ and k is the dimensions of x and y, instead of learning a matrix 'A', model learns the $A = HH^T$ and H is not a positive semi-definite matrix and $H \epsilon R^{kxk}$.

The latent factor model is not considered the trusted network and their influences, and the neighborhood model has not incorporated the item characteristics and user preferences. Previous work [2], [25], [60] shows that by combining these two models, we can improve the rating prediction. We also integrated the baseline estimates [26] and distance metric [72] in our TrustCTR model because the distance metric learns the data points closer to each other. We have integrated the distance propagation in TrustCTR model likewise work presented in the literature [38] that used trust propagation and similarity propagation in collaborative filtering. So, we estimated the rating of item i about active user u at time t_1 as:

$$\hat{r}_{ui}^{t_1} = \alpha q_i^T p_u^{t_1} + (1-\alpha)\frac{\sum\limits_{v\in N_u^{t_1}} tR_{uvk}^{t_1} q_i r_{vi}}{\sum\limits_{v\in N_u^{t_1}} tR_{uvk}^{t_1} q_i} * \mu + b_u^{t_1} + b_i^{t_1} - \left\| p_u^{t_1} - q_i^{t_1} \right\|_A^2 \tag{20}$$

Here $\hat{r}_{ui}^{t_1}$ is the predicted or observed rating of item i about active user u, at time t_1, where $1 \leq t_1 \leq 10$. α is the learning parameter. Equation (16) represents the neighborhood model and is integrated into equation (20). $q_i^T p_u^{t_1}$ is the baseline model on the basis of user preferences $p_u^{t_1}$ and item characteristics q_i at time t_1. $N_u^{t_1}$ represents the group of people trusted by users u at time t_1. $tR_{uvk}^{t_1}$ is the trust relevancy between user u and trusted user v at time t_1 and is estimated using equation (11). r_{vj} is the decay rate and is calculated by using equation (21).

$$\mathrm{r}_{\mathrm{vi}} = \mathrm{e}^{-\eta_{\mathrm{i}}(\mathrm{t}_1 - \mathrm{t}1_{\mathrm{vi}})}\mathrm{r}_{\mathrm{vi}}^{\mathrm{t}_{1vi}} \tag{21}$$

We usually used the value of η_i= 0.4 that is the effect of u_v to u_i on rating given to item i at time t_1, is the distance between t_{1vi} and t_1 is formulated as $\mathrm{r}_{\mathrm{vi}}^{\mathrm{t1_{vi}}}(t_{1vi} < t_1)$. It means that more previous ratings show the users earlier preferences and has fewer influences on current ratings [13]. When t_1 and $t_{1\ vi}$ are so closed then the more impact of $\mathrm{r}_{\mathrm{vi}}^{\mathrm{t1_{vi}}}$ on $\mathrm{r}_{\mathrm{ui}}^{\mathrm{t1}}$. For decaying gradually, the influence of past rating we used the exponential function [13], [26].

We modified the baseline estimate i.e. $\mu + b_u^{t_1} + b_i^{t_1}$ which is a part of equation (20), that shows the interactions between users and items where μ represents the average rating, $b_u^{t_1}$ is the user biases at time t_1 and $b_i^{t_1}$ is the item biases at time t_1. We used the Mahalanobis distance for computing distance between users and items using equation (22), as

$$\left\| p_u^{t_1} - q_i^{t_1} \right\|_A^2 = (p^{t_1} - q^{t_1})A\left(p^{t_1} - q^{t_1}\right)^T \tag{22}$$

where $p_u^{t_1}$ is the latent factor of user u and $q_i^{t_1}$ is the latent factor of item i. 'A' is a positive semi-definite matrix and $A\epsilon R^{kxk}$ and k is the dimension of p and q.

If there is a user u likes an item i and has friend f, then distance propagation model not only placed user u and item i close to each other but also find the close friend f that likes an item i of user u. This propagation of distance supports to overwhelmed a problem of data sparsity, for example, if user v has rated few items then the propagation distance recommends the items of user u likes, to user v. So, by using the distance metric we can discover the potential desired items and friends. The main reason for using distance metrics is to bring closer the data points and the data points that are discriminant in different sets with larger distances.

We integrated the baseline model with distance metric to minimize the distance between the user and the ratings given by the user to the items and their friends. For example, in baseline estimate, the users desired item may be distributed over a large area that fails to establish the meaning of latent factors, but after using distance metric with baseline estimates, the desired items for active users, are available in a smaller area, that captured fine-grained preferences.

If training data contains user and item then the rating for an item is predicted using equation (20), otherwise, we have considered the global mean. This global mean denoted by μ that is the average of users present in the training data and is incorporated to minimize the error between the true rating and observed rating.

The objective minimization function is achieved by computing the loss function that solves the following optimization problem using the gradient descent method as computed using equation (23).

$$\begin{aligned}\mathcal{L} = \min_{p_u^{t_1}\geq 0, q_i^{t_1}\geq 0, \eta\geq 0} \sum_{(u,i)\epsilon O_{t_1}} \Bigg(r_{ui}^{t_1} - \alpha q_i^T p_u^{t_1} \\ +(1-\alpha)\frac{\sum_{v\in N_u^{t_1}} tR_{uvk}^{t_1} q_i e^{-\eta(\mathrm{t}_1 - \mathrm{t}_1 vi)} r_{vi}^{t} vi}{\sum_{v\in N_u^{t_1}} s_{uvk}^{t_1} q_i} * \mu + b_u^{t_1} + b_i^{t_1} \\ - \left\| p_u^{t_1} - q_i^{t_1} \right\|_A^2 \Bigg)^2 + regB\left(\sum_{u=1}^{m} \left\| p_u^{t_1} \right\|^2 + \sum_{i=1}^{n} \left\| q_i^{t_1} \right\|^2\right) \\ +regU\left(\sum_{u=1}^{m} \left\| b_u^{t_1} \right\|^2\right) + regI\left(\sum_{i=1}^{n} \left\| b_i^{t_1} \right\|^2\right) \\ +\left(\alpha \sum_{u=1}^{m}\sum_{i=1}^{n}\left(1 - \left\| p_u^{t_1} - q_i^{t_1} \right\|_A^2\right)\right)\end{aligned} \tag{23}$$

We updated the user biases and item biases using the equation (24) and (25), respectively,

$$b_u^{t_1}[u] \leftarrow b_u^{t_1}[u] + \lambda * \left(e_{ui}^{t_1} - regU * b_u^{t_1}\right) \tag{24}$$

$$b_i^{t_1}[i] \leftarrow b_i^{t_1}[i] + \lambda * \left(e_{ui}^{t_1} - regI * b_i^{t_1}\right) \tag{25}$$

where λ is the learning rate or control the magnitudes of latent factors, $e_{ui}^{t_1}$ is the error between observed rating and predicted rating at time t_1 that is computed using equation (26). The *regU*, *regB,* and *regI* are the regularization parameters.

$$e_{ui}^{t_1} = r_{ui}^{t_1} - \hat{r}_{ui}^{t_1} \tag{26}$$

We updated the user latent features and item latent features by equation (27) and (28).

$$\begin{aligned} p_u^{t_1}[u] \leftarrow p_u^{t_1}[u] - \lambda((e_{ui}^{t_1} + \alpha\eta) * W\left(p_u^{t_1} - q_i^{t_1}\right)^T \\ \times (p_u^{t_1} - q_i^{t_1})) \end{aligned} \tag{27}$$

$$q_i^{t_1}[i] \leftarrow q_i^{t_1}[i] + \lambda((e_{ui}^{t_1} + \alpha\eta) * W\left(p_u^{t_1} - q_i^{t_1}\right)^T \times (p_u^{t_1} - q_i^{t_1})) \tag{28}$$

In equation (29), H is updated as

$$H \leftarrow H - \lambda((e_{ui}^{t_1} + \alpha\eta) * H\left(p_u^{t_1} - q_i^{t_1}\right)^T (p_u^{t_1} - q_i^{t_1})) \tag{29}$$

$H \epsilon R^{kxk}$ is learned with $A = HH^T$ but H is not a positive definite matrix. k is the dimension of p_u and q_i. α is the parameter that is the influence of user-item distance and η controls the influence of constraints.

$$W \leftarrow HH^T - H^T H \tag{30}$$

Stochastic gradient descent is an optimization technique that is used to estimates the error gradient by reading the current state of TrustCTR model [26]. So, during the training phase, the weights which are updated is called the learning rate. The value that we usually assign to learning rate is in the range between [0, 1] intervals; this learning rate controls how quickly the TrustCTR model is adjusted to the problem. The model converges too quickly to the optimum solution if the learning rate is large, and if the learning rate is too small, then the model is converged, too slowly.

F. TIME COMPLEXITY

In this section, we computed the time complexity of TrustCTR framework. We computed the time complexity concerning each module. Algorithm 1 that is used to generate user and item features for both source and target domains using matrix factorization. Line 1 to 3 takes constant time that is $O(1)$. Line 4 to 14 takes computational time that is $O(N)$. So, the overall complexity for 'Features Generation' module is $O(1+N) = O(N)$, where N denotes the input size of rating matrix 'R'.

The second module of proposed framework is 'Trust Relevancy' as described in Algorithm 2. First, we find the cosine similarity between users of source and target domains which takes $O(N^2)$. After that the domain interest similarity is computed and it takes $O(N^2)$ and then trust relation between two users is extracted from trust graph that is computed within $O(N)$ times. So, the time complexity of Algorithm 2 is given as, $O\left(N^2 + N^2 + N\right) = O\left(N^2\right)$, where N is the total number of nodes (users) present in a trust graph.

Algorithm 3 describes the process of finding neighbors using ACO, so a fully connected undirected graph with n users is created. Some parameters are initialized from Line 1 to 4 which takes $O(1)$. Line 5 is iterated up to $O(N)$ times. In Line 6, 'K' ants are initialized at trail so its computational time is $O(k)$. Then, the initial weight is assigned as pheromones value to each edge present among nodes (n users) and then probability is computed so Line 7 to 10 takes $O(n+N)$. After that we find the trusted neighbors and then the pheromones are updated, from Lines 11 to 14 that takes $O(N)$ time. The overall time complexity of Algorithm 3 is $O(1+N+k+n+N+N)$ that is $O(N)$.

The fourth module of proposed framework is 'Rating Prediction' as described in Algorithm 4. Line 1 is the simple initialization to iteration that computed in $O(1)$. Line 2 is the number of iterations that takes $O(m)$, line 4 assess each entry in the training set, it takes $O(m)$, so line 2 to 4 have complexity $O(m \times m)$. Moreover, line 5 to 14 involves simple assignment and initialization which takes constant time, $O(1)$ except line 8, that predicts the rating for an item i that takes $O(h)$ times. Line 15 to 21 computes the updating of parameters so it computed within a constant time that is $O(1)$ and line 22 to 27 executed in $O(1)$ times. We conglomerate overall time complexity of Algorithm 4 that is $O(1+(m \times m)+1+h+1+1)$ which is reduced to $O\left(m^2\right)$.

IV. APPLICATIONS OF TrustCTR MODEL

In this section, we analysed the role of TrustCTR model in product review applications, which can be helpful to enhance the performance of rating prediction and trust prediction.

A. TRUST PREDICTION

Guha *et al.* proposed a trust propagation model that discovered trust relationships based on known trust relations but these propagation models cannot predict the trust relationship for the cold start users or users who have little information [19]. Previous studies show that by incorporating trust information with rating similarity, better performance can be achieved than trust propagation model [5], [36]. Therefore, equation (31) can be utilized for trust prediction in the form of trust relevancy.

For recommending trust at time t_2, where $10 < t_2 \leq 11$, Given that the previous ratings $\{O_{t_1}\}$, and trust networks before time t_1, TrustCTR learns its parameters by solving the minimization problem using equation (23). Then the trust relevancy between user u_S and user u_T is calculated using equation (31). Therefore, the trust relevancy at t_2 between user u_S and user u_T can be computed as

$$tR^{t_2}(u_S, u_T) = simf_{dist}^{t_2}(u_S, u_T) \times trustRelation^{t_2}(u_S, u_T) \times sim^{t_2}(u_S, u_T) \tag{31}$$

B. RATING PREDICTION

Rating prediction is one of the significant tasks of recommendation systems for estimating rating about an active user u for item i based on previous rating history or profile information. In social networks, people rely on their close friends, sharing information with trusted friends, so trust networks are mainly used for improving rating prediction. However, people place trust differently with different people with different interests across domains, which shows that heterogeneous trust relations exist in various domains. Thus, in product review sites rating systems are dynamic where the preferences of users and interest of users change with time that shows the development of trust relations. Thus, TrustCTR models the trust development for the people who have similar domain interests and can be used for improving the performance of prediction.

Given that the earlier rating $\{O_{t_1}\}$, the objective of trust prediction is to estimate the ratings at time t_2 from test set$\{O_{t_2}\}$. Therefore, the rating of item i about active user u at time t_2 can be predicted as

$$\hat{r_{ui}^{t_2}} = \alpha q_i^T p_u^{t_2} + (1-\alpha)\frac{\sum_{v \in N_u^{t_2}} tR_{uvk}^{t_2} q_i r_{vi}}{\sum_{v \in N_u^{t_2}} tR_{uvk}^{t_2} q_i} * \mu + b_u^{t_2} + b_i^{t_2} - \| p_u^{t_2} - q_i^{t_2} \|_A^2 \quad (32)$$

For existing item i, the feature vector q_i does not evolve and for an active user u, b_i and n_i are independent on time and can be directly applied to time t_2.

C. COLD START PROBLEM

New users and items might be introduced at time t_2 in the source domain and target domain, so one of the challenges for TrustCTR model is how to address the cold start problem when there exists no rating information about items and users. Then in this situation, McPherson *et al.* show that the users who have more similar resemblance are more likely to trust each other [43]. So, for the new user, we first find the trusted neighbors in the target domain for the new user using Algorithm 3. Then compute the trust relevancy at tim t_2 e. If user u_S is a new user appears in the source domain and Nu_T contains his trusted neighbors available in target domain at time t_2, then preferences for new user u_S at time t_2 can be computed as:

$$p_S^{t_2} = \frac{\sum_{u_T \epsilon N_{u_T}}^{t2} tR^{\boldsymbol{t_2}}(u_S, u_T)\, p_T^{t2}}{\sum_{u_T \epsilon N_{u_T}}^{t2} tR^{\boldsymbol{t_2}}(u_S, u_T)} \quad (33)$$

where $tR^{\boldsymbol{t_2}}(u_S, u_T)$ is trust relevancy between user u_S in the source domain and user u_T in target domain at time t_2, and p_T^{t2} are the preferences for trusted neighbors in the target domain.

V. EXPERIMENTAL SETUP

We used the Ciao dataset published by Tang *et al.* [61] and Yu *et al.* [74]. Ciao contains the users that provide the reviews against items in different domains such as books, movies, cars, music, etc., and assigns the rating to these items in the range [1], [5]. In a social network, users express their trust with other users in the form of binary [0, 1]. If trust is present, the user assigns a value 1, otherwise 0. The trust statements are explicitly issued by users on the web of trust (WOT). For trust prediction and measuring the impact of social influence analysis, these trust relations have been used. We have also evaluated our results on another public dataset called Epinions which is a social network dataset [37], [41], [61], [74]. These datasets contain reviews about similar categories. We used these categories (item level) as a domain in the cross-domain scenario. We selected social network datasets 'Ciao' as a source domain and 'Epinions' as the target domain.

A. SETTING RANGE OF TIMESTAMPS

TrustCTR model predicted the ratings at time t_2 after training the model at time t_1. The rating matrix has timestamps, and these timestamps are divided into time slices, each time slice is referred to as a separate temporal domain, in both Epinions dataset and Ciao dataset. For Epinions dataset, in rating, we split the timestamps into eleven-time slices that is $T = \{T_1 \ldots . T_{11}\}$, where each T contains one-year data that is T_1 covers data before January 11, 2001, T_{11} covers data after January 11, 2010. T_2 covers data from January 12, 2001 to January 11, 2002 and so on. For training of TrustCTR model at time t_1 we choose the time t_1 whose range lies $t_1 = \{T_{1,} \ldots .T_{10}\}$ and for tests we set the time t_2 i.e., $t_1 = \{T_{11}\}$. Similarly, the same time slices are set for trust relations as given in Epinions dataset. For Ciao dataset, the whole dataset has timestamps and is divided into eleven-time slices where each time slice has range given in Ciao dataset and contains the preferences and trust information dated from 1st January 1970 to onwards.

B. DATASET STATISTICS

Since our experimental study is about four views of users, 'All User','Cold Start Users','Opinion User' and'Heavy Raters' related to the source and target domains. We selected the categories of Ciao dataset as a source domain and the categories available in Epinions dataset as target domains as we are trying to solve the cold start problem in a cross-domain scenario. Table 1 and Table 2 shows five categories and their statistics about both source and target datasets, respectively. The average ratings per item and average ratings per user concerning each source domain and target domain are also given in Tables I and II, respectively.

C. COMPUTING RATING SPARSITY

Categories are chosen as a domain [8] based on rating sparsity [66] present in a rating matrix of both datasets, as shown in Fig. 5. Sparsity can be measured, quantitatively as

$$Sparsity\,(\%) = 100(1 - \frac{|R|}{|u|\,.\,|i|}) \quad (34)$$

where R is the total number of user-item interactions, u is the total number of unique users, and i is the total number of unique items present in a dataset. We have chosen these categories as these categories have high rate of data sparsity. We aimed to choose categories from source domain that has high sparsity rate and categories from the target domain that has a dense data matrix, but we did not find such a combination. But despite a high rate of rating sparsity that is more than 99% our TrustCTR model performed well.

D. TRUST NETWORKS

Table 3 shows the statistics about trust network related to Ciao and Epinions datasets. A $user_1$ trusts $user_2$ if there exists 1 that shows the trust relationship and if the relationship is 0 then it means that $user_1$ does not trust $user_2$. We incorporated the direct trust relationship in TrustCTR model.

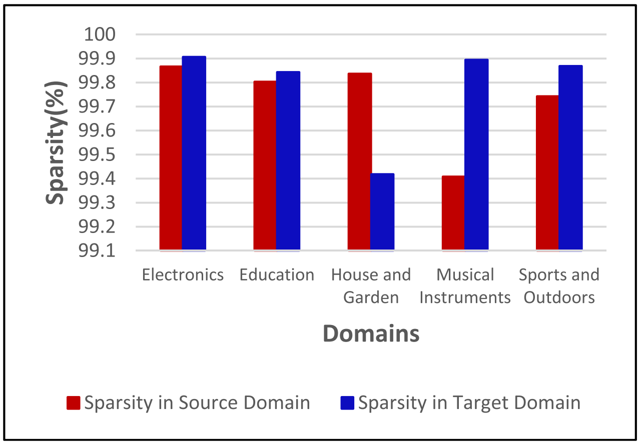


FIGURE 5. Presence of rating sparsity (%) in source and target domains.

TABLE 1. Statistics of CIAO dataset as source domain.

Source Domains	Users	Items	Ratings	Avg. Ratings per Item	Avg. Ratings per User
Electronics	1031	1525	2090	1.37	2.02
Education	1018	1219	2438	2.0	2.39
House and Garden	1114	2392	4344	1.81	3.89
Musical Instruments	214	362	458	1.26	2.14
Sports and Outdoors	477	863	1057	1.22	2.21

TABLE 2. Statistics of epinions dataset as target domain.

Target Domains	Users	Items	Ratings	Avg. Ratings per Item	Avg. Ratings per User
Electronics	3340	5809	13273	2.28	3.97
Education	2706	2794	6968	2.49	2.57
House and Garden	2435	6578	14279	2.17	5.86
Musical Instruments	305	876	1551	1.77	5.08
Sports and Outdoors	1841	3975	7653	1.92	4.15

Trust relationships among the users usually follow a power-law distribution. The degree is the number of edges that one node has with other nodes. Degree distribution shows the fraction of nodes in a social network with degree k. We used a power-law distribution that is commonly in social networks for computing degree distribution among the trustors and trustees for both trust networks of Ciao and Epinions as shown in Fig. 6 and 7, respectively.

E. DATA SAMPLING: TRAINING SET AND TEST SET

We trained TrustCTR model by selecting the data on time range t_1, after training, we predict the ratings for test set data that was available on time t_2. The new trust relationship is established among users of the source domain and target domain at time t_2. Besides, the user-item interactions are also improved in terms of their preferences, such as drift at their interests over time. The train set and test set sampling that we performed in our experiments for the cross-domain scenario of 'No Overlap', are described below:

TABLE 3. Statistics about trust network of ciao and epinions datasets.

	Ciao	Epinions
Nodes	2342	8518
Edges	57544	300091
Avg Degree	24.57	28.84
Network Diameter	11	6
Density	0.01	0.004
Modularity	0.47	0.38
Avg Clustering Coefficient	0.165	0.219

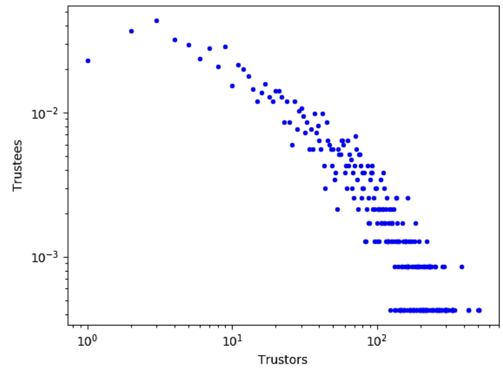


FIGURE 6. Degree distribution of trust network of ciao.

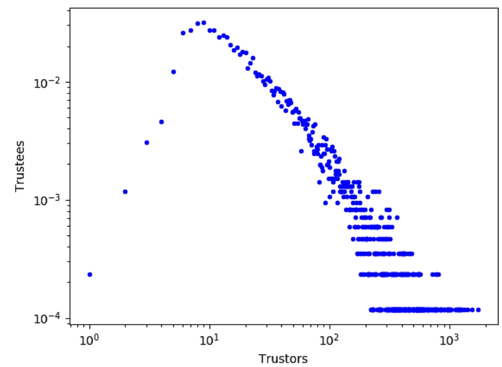


FIGURE 7. Degree distribution of trust network of epinions.

1) TRAINING AND TEST SET FOR ALL USERS, COLD START USERS, HEAVY RATERS AND OPINION USERS

For our experiments, we constructed a trainset and testset for All Users, as described in Table 4. The training set is built by including user-item interactions at time t_1 and trust relations available at time t_1. In the testset, we have found less user-item interactions as compared to the training set since at time t_2 the range of time-slice that we assumed, is just one year. We also constructed a trainset and testset for Cold Start Users, Heavy Raters, and Opinion Users as described in Table 5, Table 6, and Table 7, respectively. The training set is constructed by including user-item interactions and trust

TABLE 4. Training set at time t_1 and test set at time t_2 for all users.

	Users	Items	Ratings
Electronics Training set	3355	5639	12946
Electronics Test set	20	28	33
Education Training set	2709	2764	6928
Education Test set	6	6	6
Home and Garden Training set	2472	6216	13521
Home and Garden Test set	16	52	61
Musical Instruments Training set	294	802	1410
Musical instruments Test set	2	8	8
Sport and Outdoors Training set	1844	3706	7335
Sports and Outdoors Test set	4	11	11

TABLE 5. Training set at time t_1 and test set at time t_2 for cold start users.

	Users	Items	Ratings
Electronics Training set	795	765	945
Electronics Test set	425	397	469
Education Training set	1496	1220	1950
Education Test set	510	537	647
House and Garden Training set	1187	973	1449
House and Garden Test set	469	408	499
Musical Instruments Training set	138	152	167
Musical Instruments Test set	89	90	106
Sports and Outdoors Training set	812	752	1020
Sports and Outdoors Test set	415	430	539

TABLE 6. Training set at time t_1 and test set at time t_2 for heavy raters.

	Users	Items	Ratings
Electronics Training set	192	2525	3574
Electronics Test set	10	98	99
Education Training set	61	972	1343
Education Test set	9	45	45
House and Garden Training set	183	3818	5769
House and Garden Test set	48	799	899
Musical Instruments Training set	23	493	677
Musical Instruments Test set	2	10	10
Sports and Outdoors Training set	110	1800	2447
Sports and Outdoors Test set	7	84	87

TABLE 7. Training set at time t_1 and test set at time t_2 for opinion users.

	Users	Items	Ratings
Electronics Training set	3266	5304	11899
Electronics Test set	142	281	392
Education Training set	2676	2696	6639
Education Test set	62	113	118
House and Garden Training set	2368	5629	11264
House and Garden Test set	235	1097	1341
Musical Instruments Training set	280	728	1100
Musical Instruments Test set	28	62	67
Sports and Outdoors Training set	1772	3310	6137
Sports and Outdoors Test set	113	249	290

relations available at time t_1. The test set contains user-item interactions and trust relations at time t_2.

2) DOMAIN PARTITIONING

We evaluated proposed model on different aspects and split each domain into different views of users including:

All Users: A set of users that assigned ratings to the items in the range [1-5].

Cold Start Users: Some users express a lot of ratings, whereas some express a few ratings. We considered cold-start users are those who have assigned ratings for less than 3 items in each domain of our experimental study. From Table 8, it is observed that there are more than 14% of users who are cold start related to five domains of each dataset.

TABLE 8. Ratio of cold start users, heavy raters and opinion users.

	% of Cold Start Users in		% of Heavy Raters in		% of Opinion Users in	
Domain	Ciao	Epinions	Ciao	Epinions	Ciao	Epinions
Electronics	38	14	0.6	2.11	5.49	8.04
Education	34	28	0.95	0.89	5.67	5.99
House and Garden	24	17	1.75	1.44	6.2	5.53
Musical Instruments	38	58	1.06	1.42	5.73	3.57
Sports and Outdoors	38	15	1.12	1.4	5.48	5.81

Heavy Raters: A set of users who rated higher than 10 items. The ratio of Heavy Raters is less than the ratio of Cold Start Users and Opinion Users related to each domain as given in Table 8.

Opinion Users: These are the set of users that assigned more than 3 ratings to the items and the standard deviation of these ratings is higher than 0.48. Table 8, shows the ratio of Opinion Users to each domain.

VI. EVALUATION AND EXPERIMENTAL RESULTS

This section describes the results obtained by using TrustCTR model, and also discusses the evaluation obtained by comparing the results of TrustCTR model with state-of-the-art methods.

A. EVALUATION METRICS

For evaluation, we used Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and F-measure. The RMSE metric shows the closeness of predictions to outcomes [71], as described in equation (35). RMSE can be computed as

$$RMSE = \sqrt{\frac{\sum_{u,i}(r_{ui} - \hat{r}_{ui})^2}{N}} \tag{35}$$

r_{ui} is the actual rating and $\hat{r}_{ui}$ is predicted rating of an item i about user u. N is the total number of items under evaluation. The MAE measure is defined in the work [64] and can be computed as

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |r_{ui} - \hat{r}_{ui}| \tag{36}$$

Some recommendation systems may not predict all ratings available in test data, this occurs due to the high ratio of sparsity in the dataset. Therefore, we used a coverage metric to compute < user, item > prediction for this pair.

$$coverage = \frac{n^{predict}}{n^{actual}} \times 100\% \tag{37}$$

Here $n^{predict}$ is the number of observed ratings and n^{actual} is the number of actual ratings. We also computed the F-measure by combining RMSE and coverage into a single measure. So, to define F-measure, first, we converted RMSE into precision within range [0,1].

$$precision = 1 - \frac{RMSE}{4} \tag{38}$$

In equation (38), we set the maximum possible error i.e. 4 because rating values lie in between the range [1,5]. Therefore, F-measure [11] is defined as

$$F\_measure = \frac{2 \times precision \times coverage}{precision + coverage} \tag{39}$$

B. EVALUATION USING STATE OF THE ART METHODS

Since TrustCTR is a hybrid model, so we evaluated the performance of the proposed model with state-of-the-art methods, and we divide the state-of-the-art methods into three categories: Rating based methods, Trust based models and Cross domain model. These state-of-the-art methods are implemented using Python 3.6 and MATLAB, and the proposed model is implemented using Python 3.6.

1) RATING BASED METHODS

UserCF, ItemCF, BasicMF, PMF, SVD, and SVD ++ are rating based methods.

UserCF: UserCF is the collaborative filtering method that predicts the ratings based on the user's similarity [50].

ItemCF: It estimates the prediction about an item for a user based on the item's similarity [55].

BasicMF: It is based on user and item characteristics and predicts the ratings of an item about a user [25].

PMF: PMF is based on probabilistic matrix factorization and is discussed in [44].

SVD: This method integrates neighborhood model and matrix factorization, which improves the prediction accuracy [26].

SVD ++: SVD ++ combines the basis of the neighborhood model and factorization model [27].

2) TRUST BASED MODELS

SocialMF, SocialRec, SocialFD, RSTE, SoReg, LOCABAL, and CUNE-MF are trust-based models.

SocialMF: The SocialMF incorporates trust propagation technique into matrix factorization for solving a cold start problem in social recommender systems [21].

SoRec: It solves the data sparsity problem that occurs in trust-based social recommendation [38].

SocialFD: SocialFD maps a user's trust and items in low dimensional latent factor space by combining the distance metric and matrix factorization model [74].

RSTE: Ma et al. proposed a trust aware recommendation model called RSTE that is a hybrid model of matrix factorization and social network-based approach [39].

SoReg: This model performs matrix factorization with social regularization by improving prediction accuracy using the social trust information of trusted friends and solves the trust-aware recommendation problem [40].

LOCABAL: This method integrates the local and global context related to social media for generating recommendations [63].

CUNE-MF: It combines the bayesian personalized ranking and matrix factorization [78].

3) CROSS DOMAIN MODEL

We compared proposed model with one of the recently used cross-domain model called TRACER as it overcomes the limitations of codebook transfer [33].

TRACER: Zhuang et al. proposed a model called TRACER that predicted ratings for target users by adopting the concept of transfer learning from multiple source domains into a target domain [82].

C. PARAMETER ANALYSIS FOR STATE-OF-THE-ART METHODS AND TrustCTR MODEL

In our experiments, Table 9 described the parameter settings as we used these parameters for state-of-the-art methods with 100 iterations and 10 reduced dimensions. Parameters settings for TrustCTR model are determined after performing some experiments based on parameters α, regU, and reduced dimensions. We performed experiments to study the impact of parameters in two aspects of users that is for All Users and Cold Start Users, as our focus is on solving a new user cold-start problem.

TABLE 9. Parameters used in state of the art methods.

Methods	Learning rate (λ) (min to max)	Alpha (α)	regU	regI	regB	regS
User CF	-	-	-	-	-	-
Item CF	-	-	-	-	-	-
BasicMF	0.03 to 1	-	0.050	0.050	0.100	-
SVD	0.05 to 1	-	0.010	0.020	0.020	-
PMF	0.02 to 1	-	0.010	0.010	0.100	-
SVD++	0.05 to 1	-	0.050	0.050	0.100	-
RSTE	0.001 to 1	0.6	0.001	0.001	0.100	0.100
SoRec	0.1 to 1	-	0.050	0.050	0.100	0.100
SoReg	0.05 to 1	0.1	0.020	0.020	0.100	0.020
Social MF	0.05 to 1	-	0.050	0.050	0.100	0.100
LOCABAL	0.03 to 1	0.400	0.010	0.010	0.010	0.10
Social FD	0.01 to 1	0.3	0.005	0.005	0.010	0.100
CUNE MF	0.05 to 0.1	-	0.010	0.010	0.010	-
TRACER	0.80	1.0	-	-	0.00001	-

1) IMPACT OF DIMENSIONALITY

The purpose of the training phase for TrustCTR model is to reduce the dimensions of the user-item rating matrix for extracting the features and models the user-item characteristics. Therefore, we studied the impact of dimensionality using

TrustCTR to find the best parameters for All users as shown in Fig. 8 and for Cold Start users as shown in Fig. 9. For All users, we set $\alpha = 0.3$, $\lambda = 0.03$ to 1, regU = 0.050, regI = 0.050, regB = 0.10 with k =40 reduced dimensions for Electronics, k =30 for Education, k =40 for House and Garden, k =10 for Musical Instruments and k =20 for Sports and Education domain. For Cold Start Users, we considered same parameter settings for α, λ, regU, regI, regB with k =10 as the best parametric value over all domains.

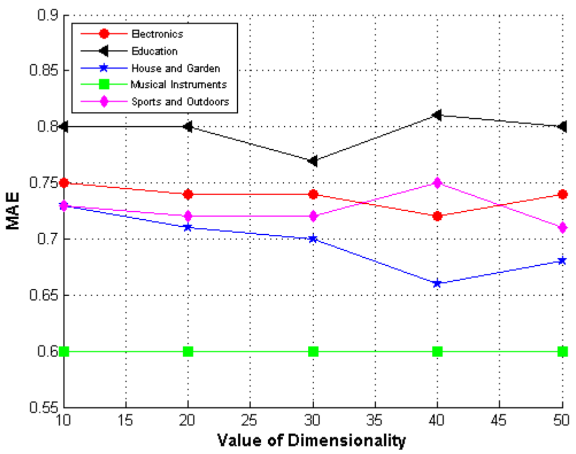


FIGURE 8. Impact of dimensionality for all users over various domains using TrustCTR.

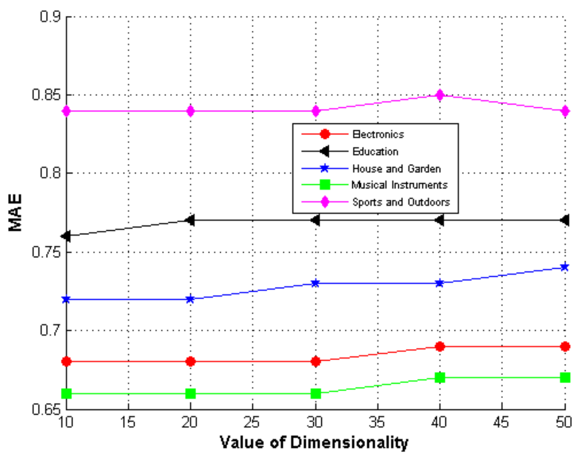


FIGURE 9. Impact of dimensionality for cold start users over various domains using TrustCTR.

The prediction accuracy in terms of MAE for cold-start users, is not improved in most of the domains when the dimensionality is increased as shown in Fig. 9. Moreover, for All users when the dimensions are increased then prediction accuracy is improved in most of the domains as shown in Fig. 8.

It means that the larger dimension may lead to model overfitting that turns out in degrading the accuracy. Another intuition is that when the dimensionality is small then the prediction accuracy is improved [12].

2) IMPACT OF α

The α is the learning parameter that controls the influence of users' trust on their friends and employs the user's characteristics. It means that by fusing the user's preferences information with favors of their trusted friends improves the prediction accuracy. The impact of incrementing the value of α from 0.1 to 1.0 for cold start users is same when we increment the value except House and Garden Domain. For All Users, the prediction accuracy remained same on the domain of Sports and Outdoors and Musical Instruments, and for Electronics domain the MAE value is not improved and for other domains the prediction accuracy is improved. We performed a sensitivity analysis of α using TrustCTR model by adjusting the value of α from 0.1 to 1.0 and find the best values of α on various domains that help improve the prediction accuracy. We set $\lambda = 0.03$ to 1, regU = 0.050, regI = 0.050, regB = 0.10 with k =10 reduced dimensions. From Fig. 10, it is observed that for All users, we obtained better MAE value when $\alpha = 0.1$ for Electronics domain, $\alpha = 0.4$ for Education, $\alpha = 0.4$ for House and Garden, $\alpha = 0.1$ for Musical Instruments and $\alpha = 0.4$ for Sports and Outdoors. Fig. 11 shows the impact of α on Cold Start Users on various domains. The value of $\alpha = 0.9$ shows the best result on House and Garden domain whereas for all other domains α is set to 0.3.

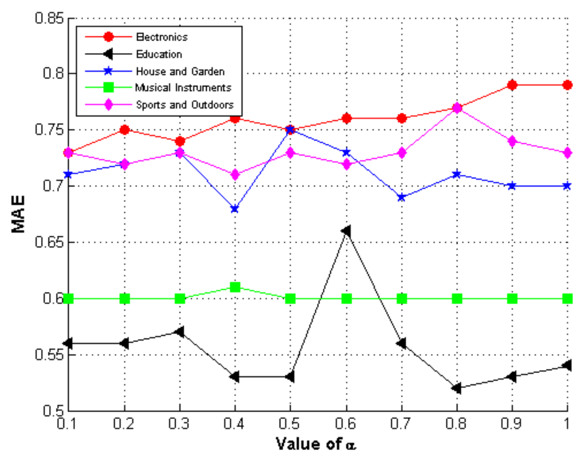


FIGURE 10. MAE results for varying number of α for all users over various domains using TrustCTR.

3) IMPACT OF REGULARIZATION PARAMETER

For observing the impact of the regularization parameter 'regU' that controls the effect of social network on a user's behavior, we set the value of regU from 0.06 to 0.10. Larger values of regU shows the more influence of social networks on users' behavior whereas the smaller values of regU lead to a model close to the baselines. An extremely small value of regU have a weak impact on social regularization terms. We observed that in most of the domains, the regU = 0.06 has given the best MAE prediction accuracy for both views of All Users and Cold Start Users as shown in Fig. 12 and Fig. 13, respectively. With the increase in the value of regU, there is slight improvement in MAE results. Along with the

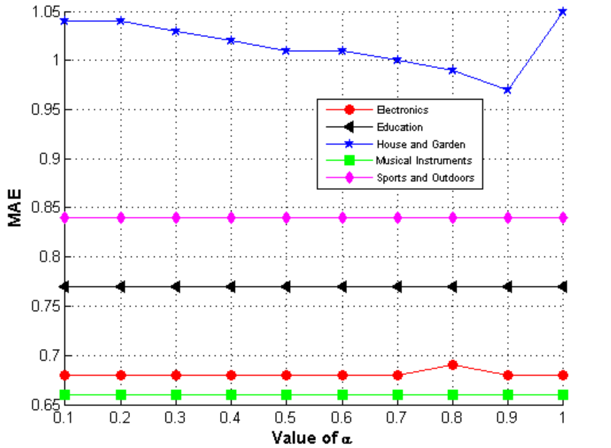


FIGURE 11. MAE results for varying number of α for Cold Start Users over various domains using TrustCTR.

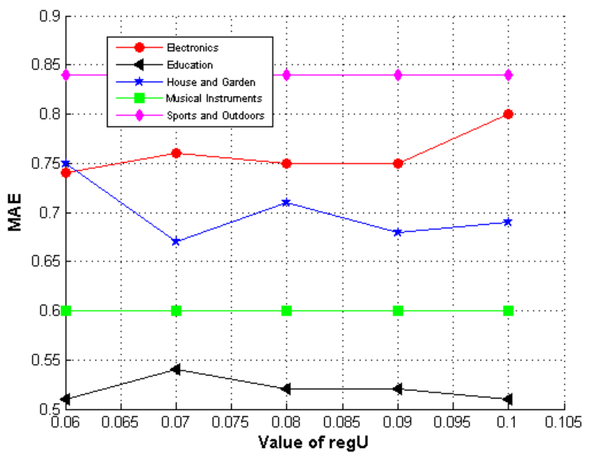


FIGURE 12. Effect of regularization parameter regU for all users over various domains using TrustCTR.

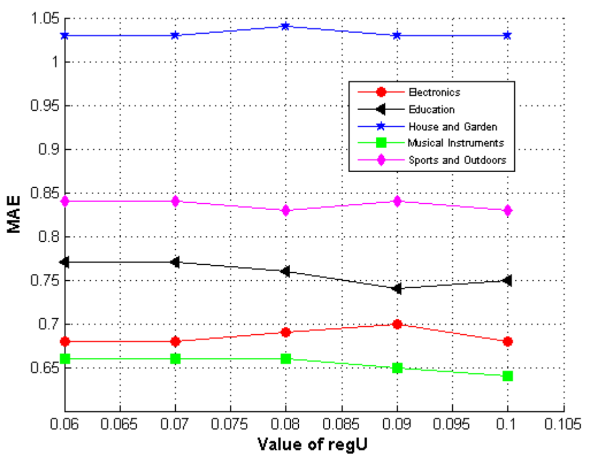


FIGURE 13. Effect of regularization parameter regU for cold start users over various domains using TrustCTR.

parameter settings of regU, we also set $\alpha = 0.3$, $\lambda = 0.03$ to 1, regI = 0.050, regB = 0.10 with k =10 reduced dimensions.

D. EXPERIMENTAL RESULTS

We computed the performance of TrustCTR model using MAE, RMSE, and F-measure and discussed the evaluation on different domains for all users, cold start users, heavy raters, and opinion users. We chose the best parameters based on the analysis of parameters as performed in section VI-C. The performance of TrustCTR model is evaluated for the view of All Users, Heavy Raters and Opinion Users, by setting the value of $\lambda = 0.03$ to 1,regU = 0.05, regI = 0.050, regB = 0.10, k = 10 and $\alpha = 0.1$ for Electronics domain, $\alpha = 0.4$ for Education, $\alpha = 0.4$ for House and Garden, $\alpha = 0.1$ for Musical Instruments and $\alpha = 0.4$ for Sports and Outdoors.

For Cold Start Users, the performance of proposed model is evaluated using the same parameters except for the value of α that is 0.9 for House and Garden domain, and for all other domains, α is set to 0.3.

1) EVALUATION OF TRUSTCTR MODEL OVER DOMAIN OF ELECTRONICS

In Table 10, it is observed that our TrustCTR model gives better results on the domain of Electronics as compared to other state-of-the-art methods in terms of MAE and RMSE. F-measure also outperforms other methods such as UserCF, BasicMF, PMF,RSTE,SoRec, SoReg,SocialMF, CUNEMF, LOCABAL, and TRACER, and have some of the results equivalent to ItemCF, SVD, SVD ++, and SocialFD. The experiments conducted for All Users, proposed model converged after 63 iterations out of 100 iterations, whereas other state-of-the-art methods are converged after 90 iterations. Since TrustCTR model solves the new user cold-start problem that is if a new user comes in the source domain and has no profile history then proposed model can estimate predictions and generate recommendations for a new user from trusted friends available in the target domain. In Table 10, TrustCTR model given improved results on Electronics domain as compared to baselines in terms of MAE and RMSE. Accuracy through F-measure for TrustCTR found better than UserCF, BasicMF, PMF, SVD ++, RSTE, SoReg, SoRec, SocialMF, LOCABAL, CUNEMF and TRACER but found equivalent to SocialFD, SVD and ItemCF. In this experiment, TrustCTR model is converged after 57 iterations for Cold Start Users. Similarly, for Heavy Raters and Opinion users, we got improved results as compared to other baselines methods, as described in Table 10. The performance of proposed model is also evaluated by computing the mean improvement value of MAE with respect to different views of users. It is revealing that improvement rate is 12.62%, 9.31%,7.59%, and 15.22% for All Users, Cold Start Users, Heavy Raters, and Opinion Users, respectively, as reported in Table 10. It is observed that the improvement rate varies with different views of users and the main improvement found is 15.22% in Opinion Users.

TABLE 10. Performance of TrustCTR on electronics domain with different views of users.

Methods	Error Metrics	All Users	Cold Start Users	Heavy Raters	Opinion Users
UserCF	MAE	0.80	0.74	0.81	0.85
	RMSE	1.21	0.97	0.98	1.11
	F-measure	0.82	0.86	0.86	0.83
ItemCF	MAE	0.79	0.71	0.80	0.81
	RMSE	1.13	0.92	0.96	1.01
	F-measure	0.84	0.87	0.86	0.85
BasicMF	MAE	0.81	0.91	0.91	0.86
	RMSE	1.23	0.97	1.06	0.98
	F-measure	0.82	0.86	0.84	0.86
SVD	MAE	0.79	0.78	0.85	0.89
	RMSE	1.10	0.94	1.01	1.12
	F-measure	0.84	0.87	0.85	0.83
PMF	MAE	0.82	0.73	0.85	0.88
	RMSE	1.24	1.02	1.01	1.07
	F-measure	0.82	0.85	0.85	0.84
SVD++	MAE	0.76	0.73	0.92	0.84
	RMSE	1.10	0.98	1.16	1.05
	F-measure	0.84	0.86	0.83	0.84
RSTE	MAE	0.98	0.79	0.89	0.96
	RMSE	1.33	1.08	1.18	1.26
	F-measure	0.80	0.84	0.82	0.81
SoRec	MAE	0.76	0.71	0.81	0.82
	RMSE	1.23	0.99	0.99	1.03
	F-measure	0.82	0.86	0.85	0.85
SoReg	MAE	0.86	0.71	0.85	0.84
	RMSE	1.25	0.97	1.04	1.06
	F-measure	0.81	0.86	0.85	0.84
SocialMF	MAE	0.84	0.72	0.82	0.88
	RMSE	1.24	1.00	1.05	1.09
	F-measure	0.82	0.86	0.84	0.84
LOCABAL	MAE	0.92	0.71	0.85	0.98
	RMSE	1.21	0.99	1.13	1.28
	F-measure	0.82	0.86	0.83	0.80
SocialFD	MAE	0.79	0.73	0.84	0.89
	RMSE	1.11	0.95	1.08	1.21
	F-measure	0.84	0.87	0.84	0.82
CUNE MF	MAE	0.80	0.78	0.84	0.80
	RMSE	1.22	0.98	0.98	1.02
	F-measure	0.82	0.86	0.86	0.85
TRACER	MAE	0.79	0.81	0.86	0.96
	RMSE	1.15	1.05	1.17	1.26
	F-measure	0.83	0.85	0.82	0.81
TrustCTR	MAE	0.73	0.69	0.79	0.76
	RMSE	1.09	0.91	0.92	0.94
	F-measure	0.84	0.87	0.87	0.86
Improve		**12.62%**	**9.31%**	**7.59%**	**15.22%**

2) EVALUATION OF TRUSTCTR MODEL OVER DOMAIN OF EDUCATION

In Table 11, we can see that TrustCTR has out class results as compared to all baselines on Education domain. All of the three metrics presented better results. In the Education domain, our model is converged after 62 iterations for All Users. For Cold Start Users, TrustCTR model has given improved results and model is converged after 59 iterations. Similarly, for Heavy Raters and Opinion users, we obtained the improved results as compared to other baselines methods, as described in Table 11.

The performance of proposed model is also evaluated by computing the mean improvement value to different views of users. It is revealing that improvement rate is 111.67%, 4.45%,33.82%, and 12.45% for All Users, Cold Start Users, Heavy Raters, and Opinion Users, respectively, as reported in Table 11. The higher improvement rate is found in the view of All Users.

TABLE 11. Performance of TrustCTR on education domain with different views of users.

Methods	Error Metrics	All Users	Cold Start Users	Heavy Raters	Opinion Users
UserCF	MAE	1.12	0.79	0.63	0.77
	RMSE	1.29	1.02	0.92	0.99
	F-measure	0.81	0.85	0.87	0.85
ItemCF	MAE	1.18	0.78	0.63	0.77
	RMSE	1.31	1.07	0.92	0.99
	F-measure	0.8	0.85	0.87	0.85
BasicMF	MAE	0.87	0.78	0.67	0.81
	RMSE	1.19	1.07	0.95	1.05
	F-measure	0.83	0.85	0.86	0.84
SVD	MAE	1.18	0.81	0.71	0.79
	RMSE	1.31	1.01	0.99	1.08
	F-measure	0.8	0.84	0.85	0.84
PMF	MAE	1.13	0.78	0.60	0.81
	RMSE	1.29	1.07	0.85	0.99
	F-measure	0.81	0.85	0.88	0.85
SVD++	MAE	1.18	0.81	0.68	0.75
	RMSE	1.31	1.02	0.86	1.01
	F-measure	0.8	0.85	0.87	0.85
RSTE	MAE	1.52	0.84	0.65	0.93
	RMSE	1.83	1.11	0.89	1.20
	F-measure	0.70	0.84	0.87	0.82
SoRec	MAE	1.03	0.78	0.83	0.87
	RMSE	1.24	1.07	1.06	1.17
	F-measure	0.82	0.85	0.84	0.82
SoReg	MAE	1.01	0.78	0.67	0.82
	RMSE	1.23	1.07	0.97	1.14
	F-measure	0.82	0.85	0.86	0.83
SocialMF	MAE	0.9	0.78	0.59	0.76
	RMSE	1.19	1.07	0.87	0.98
	F-measure	0.83	0.85	0.87	0.86
LOCABAL	MAE	1.34	0.90	0.88	0.91
	RMSE	1.65	1.23	1.19	1.22
	F-measure	0.74	0.82	0.82	0.82
SocialFD	MAE	0.99	0.85	0.83	0.89
	RMSE	1.04	1.06	1.02	1.24
	F-measure	0.85	0.85	0.85	0.81
CUNE MF	MAE	1.03	0.78	0.75	0.91
	RMSE	1.24	1.07	0.98	1.23
	F-measure	0.82	0.85	0.86	0.81
TRACER	MAE	0.93	0.80	0.81	0.86
	RMSE	1.3	1.24	1.02	1.12
	F-measure	0.81	0.82	0.85	0.83
TrustCTR	MAE	0.52	0.77	0.53	0.74
	RMSE	0.63	0.99	0.82	0.97
	F-measure	0.91	0.86	0.88	0.86
Improve		**111.67%**	**4.45%**	**33.82%**	**12.45%**

3) EVALUATION OF TRUSTCTR MODEL OVER DOMAIN OF HOUSE AND GARDEN

In Table 12, it is observed that our TrustCTR model given better results on House and Garden domain as compared to other states of the art methods in terms of MAE and RMSE. A metric that is F-measure outperformed as compared to some methods but there exist equivalent results of TrustCTR with UserCF, BasicMF, PMF, SoRec, SoReg, SocialMF and CUNEMF. For Cold Start Users, all of the three metrics given improved results. In an experiment for All Users and Cold-Start Users, TrustCTR model is converged after 66 and 57 iterations, respectively. Similarly, for Heavy Raters and Opinion users, we get the improved results as compared to the state-of-the-art methods.

The mean improvement values of MAE with respect to different views of users are 20.12%, 60.71%,9.49%, and 14.19% for All Users, Cold Start Users, Heavy Raters, and Opinion Users, respectively, as reported in Table 12. The Cold Start

TABLE 12. Performance of TrustCTR on house and garden domain with different views of users.

Methods	Error Metrics	All Users	Cold Start Users	Heavy Raters	Opinion Users
UserCF	MAE	0.79	1.05	0.74	0.79
	RMSE	1.13	1.48	0.94	1.01
	F-measure	0.84	0.77	0.86	0.85
ItemCF	MAE	0.94	0.91	0.74	0.81
	RMSE	1.44	1.23	0.94	1.05
	F-measure	0.78	0.82	0.86	0.84
BasicMF	MAE	0.8	0.98	0.76	0.79
	RMSE	1.12	1.38	0.97	1.00
	F-measure	0.84	0.79	0.86	0.85
SVD	MAE	0.92	0.96	0.78	0.85
	RMSE	1.43	1.25	0.96	1.09
	F-measure	0.78	0.81	0.86	0.84
PMF	MAE	0.72	0.99	0.75	0.84
	RMSE	1.09	1.39	0.96	1.11
	F-measure	0.84	0.79	0.86	0.83
SVD++	MAE	0.86	0.96	0.92	0.89
	RMSE	1.40	1.26	1.10	1.16
	F-measure	0.79	0.81	0.84	0.83
RSTE	MAE	1.2	1.06	0.86	0.96
	RMSE	1.61	1.38	1.11	1.25
	F-measure	0.75	0.79	0.83	0.81
SoRec	MAE	0.73	0.98	0.78	0.86
	RMSE	1.10	1.38	1.01	1.21
	F-measure	0.84	0.79	0.85	0.82
SoReg	MAE	0.75	0.97	0.82	0.79
	RMSE	1.11	1.37	0.99	1.00
	F-measure	0.84	0.79	0.85	0.85
SocialMF	MAE	0.76	1.00	0.84	0.86
	RMSE	1.12	1.39	1.08	1.16
	F-measure	0.84	0.79	0.84	0.83
LOCABAL	MAE	0.92	1.06	0.82	0.96
	RMSE	1.28	1.45	1.12	1.30
	F-measure	0.81	0.78	0.83	0.80
SocialFD	MAE	0.87	0.97	0.77	0.79
	RMSE	1.23	1.26	1.02	1.00
	F-measure	0.82	0.81	0.85	0.85
CUNE MF	MAE	0.74	0.99	0.86	0.96
	RMSE	1.09	1.39	1.15	1.31
	F-measure	0.84	0.79	0.84	0.80
TRACER	MAE	0.94	0.62	0.75	0.88
	RMSE	1.35	1.1	1.05	1.16
	F-measure	0.80	0.84	0.83	0.83
TrustCTR	MAE	0.71	0.60	0.73	0.77
	RMSE	1.08	1.01	0.92	0.99
	F-measure	0.84	0.86	0.87	0.85
Improve		**20.12%**	**60.71%**	**9.49%**	**14.19%**

Users have a high improvement rate as compared to other views of users.

4) EVALUATION OF TRUSTCTR MODEL OVER DOMAIN OF MUSICAL INSTRUMENTS

For the domain of Musical Instruments, we have found improved results in terms of MAE and RMSE to the state-of-the-art algorithms, as described in Table 13. But in terms of F-measure TrustCTR model given very close results as compared to baselines. In this study, TrustCTR model is converged after 44 iterations for All Users. Evaluation of TrustCTR model for Cold-Start Users on the domain of Musical Instruments, the performance of TrustCTR is better in terms of MAE, RMSE, and F-measure, and the model is converged after 68 iterations. Similarly, for Heavy Raters and Opinion users, we obtained the improved results as described in Table 13.

TABLE 13. Performance of TrustCTR on musical instruments domain with different views of users.

Methods	Error Metrics	All Users	Cold Start Users	Heavy Raters	Opinion Users
UserCF	MAE	0.66	0.69	0.60	0.81
	RMSE	0.76	0.84	0.80	1.01
	F-measure	0.90	0.88	0.88	0.85
ItemCF	MAE	0.64	0.67	0.61	0.81
	RMSE	0.71	0.84	0.78	1.01
	F-measure	0.90	0.88	0.89	0.85
BasicMF	MAE	0.63	0.67	0.65	0.81
	RMSE	0.75	0.84	0.83	1.01
	F-measure	0.90	0.88	0.88	0.85
SVD	MAE	0.66	0.68	0.59	0.84
	RMSE	0.72	0.81	0.72	1.08
	F-measure	0.90	0.89	0.90	0.84
PMF	MAE	0.69	0.67	0.58	0.79
	RMSE	0.78	0.84	0.81	0.98
	F-measure	0.89	0.88	0.88	0.86
SVD++	MAE	0.76	0.74	0.85	0.77
	RMSE	0.82	0.89	1.02	0.95
	F-measure	0.89	0.87	0.85	0.86
RSTE	MAE	1.1	0.88	0.67	0.82
	RMSE	1.37	1.2	0.82	1.05
	F-measure	0.79	0.82	0.88	0.84
SoRec	MAE	0.66	0.67	0.61	0.89
	RMSE	0.76	0.84	0.85	1.19
	F-measure	0.90	0.88	0.88	0.82
SoReg	MAE	0.67	0.67	0.74	0.79
	RMSE	0.76	0.84	0.98	1.03
	F-measure	0.90	0.88	0.86	0.85
SocialMF	MAE	0.63	0.67	0.71	0.89
	RMSE	0.75	0.84	0.91	1.18
	F-measure	0.90	0.88	0.87	082
LOCABAL	MAE	0.85	0.73	0.67	0.78
	RMSE	1.13	0.87	0.88	0.99
	F-measure	0.84	0.88	0.87	0.85
SocialFD	MAE	0.63	0.79	0.75	0.85
	RMSE	0.7	0.87	0.93	1.07
	F-measure	0.90	0.88	0.86	0.84
CUNE MF	MAE	0.65	0.67	0.71	0.76
	RMSE	0.76	0.84	0.92	0.98
	F-measure	0.90	0.88	0.87	0.86
TRACER	MAE	0.65	0.69	0.65	0.75
	RMSE	0.87	0.92	0.84	0.98
	F-measure	0.88	0.87	0.88	0.86
TrustCTR	MAE	0.60	0.66	0.50	0.74
	RMSE	0.70	0.79	0.71	0.93
	F-measure	0.90	0.89	0.90	0.86
Improve		**17.61%**	**7.03%**	**34.14%**	**9.65%**

We also computed the mean improvement value for different views of users. It is revealing that improvement rate is 17.61%, 7.03%,34.14%, and 9.65% for All Users, Cold Start Users, Heavy Raters, and Opinion Users, respectively, as reported in Table 13. It is observed that the improvement rate varies with different views of users and the major improvement found is 34.14% for Heavy Raters.

5) EVALUATION OF TRUSTCTR MODEL OVER DOMAIN OF SPORTS AND OUTDOORS

In Sports and Outdoors domain, MAE and RMSE metrics performance are better than all state-of-the-art methods. F-measure metric is shown improved accuracy than UserCF, ItemCF, BasicMF, SVD, PMF, SVD ++, RSTE, SocialMF, LOCABAL, CUNEMF and TRACER; and for remaining baseline methods, TrustCTR model shows equivalent results

TABLE 14. Performance of TrustCTR on sports and outdoors domain with different views of users.

Methods	Error Metrics	All Users	Cold Start Users	Heavy Raters	Opinion Users
UserCF	MAE	0.81	0.86	0.83	0.75
	RMSE	1.01	1.15	1.06	0.93
	F-measure	0.86	0.83	0.84	0.86
ItemCF	MAE	0.86	0.89	0.82	0.75
	RMSE	0.91	1.14	1.04	0.93
	F-measure	0.87	0.83	0.85	0.86
BasicMF	MAE	0.75	0.88	0.83	0.77
	RMSE	0.94	1.14	1.06	0.96
	F-measure	0.87	0.83	0.84	0.86
SVD	MAE	0.72	0.91	0.85	0.79
	RMSE	0.87	1.14	1.10	0.98
	F-measure	0.88	0.83	0.84	0.86
PMF	MAE	0.87	0.96	0.81	0.77
	RMSE	0.97	1.14	1.06	0.96
	F-measure	0.86	0.83	0.84	0.86
SVD++	MAE	1.09	0.89	0.78	0.84
	RMSE	1.49	1.16	0.99	1.05
	F-measure	0.77	0.83	0.85	0.84
RSTE	MAE	0.77	0.97	0.92	0.91
	RMSE	0.91	1.14	1.15	1.13
	F-measure	0.87	0.83	0.83	0.83
SoRec	MAE	0.75	0.95	0.77	0.81
	RMSE	0.81	1.14	1.02	0.99
	F-measure	0.89	0.83	0.85	0.85
SoReg	MAE	0.74	0.97	0.79	0.78
	RMSE	0.79	1.14	0.97	1.02
	F-measure	0.89	0.83	0.86	0.85
SocialMF	MAE	0.76	0.99	0.88	0.76
	RMSE	0.87	1.14	1.13	0.97
	F-measure	0.88	0.83	0.83	0.86
LOCABAL	MAE	0.86	0.86	0.87	0.75
	RMSE	1.18	1.2	1.16	0.96
	F-measure	0.83	0.82	0.83	0.86
SocialFD	MAE	0.75	0.85	0.91	0.92
	RMSE	0.79	1.11	1.24	1.15
	F-measure	0.89	0.84	0.81	0.83
CUNE MF	MAE	0.74	0.90	0.88	0.73
	RMSE	0.88	1.14	0.99	0.92
	F-measure	0.88	0.83	0.85	0.87
TRACER	MAE	0.77	0.88	0.98	1.04
	RMSE	1.19	1.33	1.15	1.41
	F-measure	0.83	0.80	0.83	0,78
TrustCTR	MAE	0.71	0.84	0.7662	0.7234
	RMSE	0.78	1.1	0.95	0.90
	F-measure	0.89	0.84	0.86	0.87
Improve		**13.07%**	**8.50%**	**12.03%**	**12.79%**

as shown in Table 14. In this study, TrustCTR model is converged after 69 iterations for All Users. The TrustCTR model shown improved results in terms of MAE and RMSE for Cold Start Users. With F-measure, TrustCTR has the same value as SocialFD, and from others, F-measure has slightly improved results. TrustCTR model is converged after 65 iterations for Cold-Start Users. For Heavy Raters and Opinion users, we also get the improved results.

The improvement rate is 13.07%, 8.50%,12.03%, and 12.79% for All Users, Cold Start Users, Heavy Raters, and Opinion Users, respectively, as reported in Table 14. We have found a higher rate of improvement for All Users that is 13.07%.

E. CONVERGENCE ANALYSIS

In this section, we compared the convergence property of proposed model to the most recently used rating based methods such as PMF [44], and SVD ++[27], trust aware models such as Social FD [74], and CUNEMF [78], and cross-domain recommender systems such as TRACER [82].

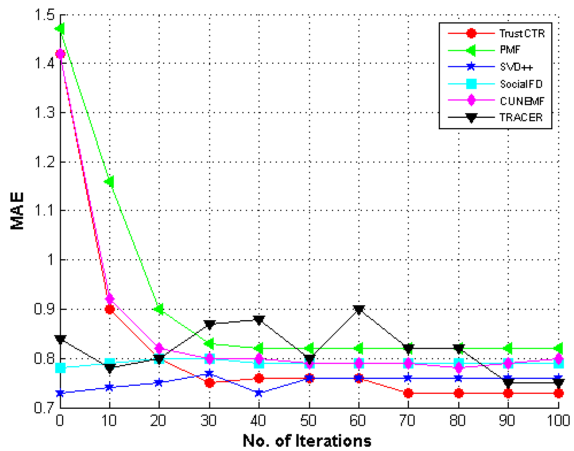


FIGURE 14. Convergence comparison of TrustCTR with baselines over electronics domain for all users.

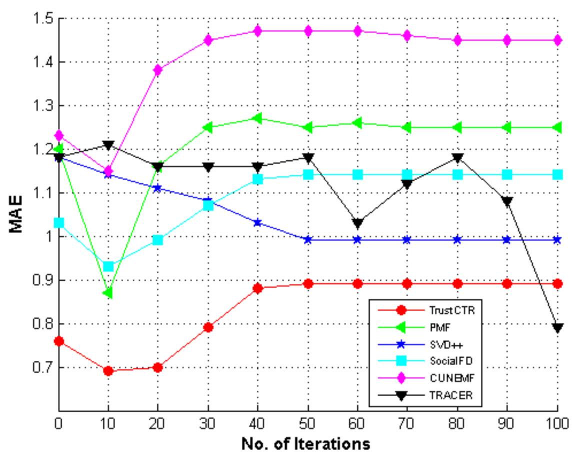


FIGURE 15. Convergence comparison of TrustCTR with baselines over education domain for all users.

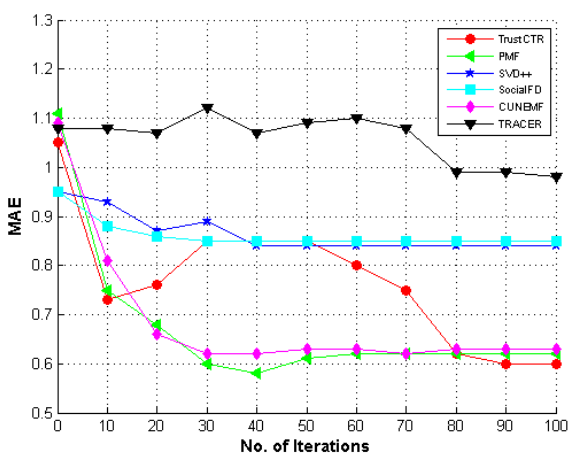


FIGURE 16. Convergence comparison of TrustCTR with baselines over house and garden domain for all users.

We performed the convergence analysis for All Users. Figure 14 shows the convergence analysis graph of the TrustCTR to other baselines in the domain of Electronics.

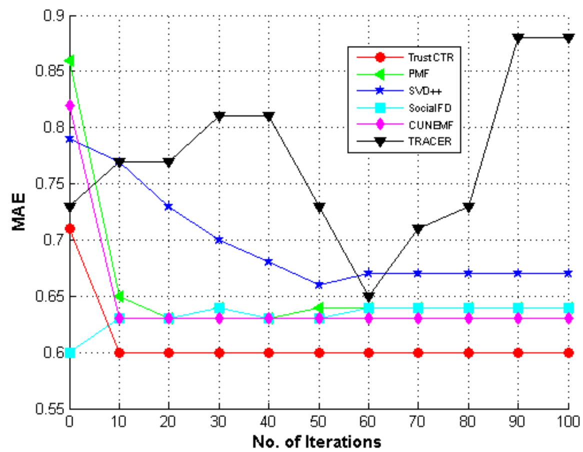


FIGURE 17. Convergence comparison of TrustCTR with baselines over musical instruments domain for all users.

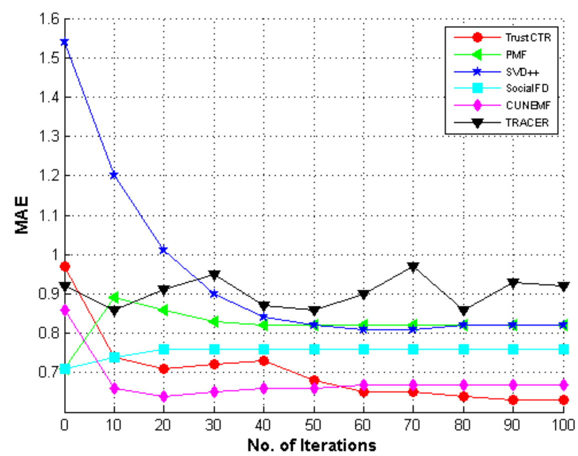


FIGURE 18. Convergence comparison of TrustCTR with baselines over sports and outdoors domain for all users.

It is observed that all methods are converged to lower values of MAE in later iterations, but the proposed model is converged early as compared to other methods. Since the proposed model utilizes the ACO search property and across domain information, so the convergence time and accuracy results of TrustCTR are better than other methods. Similarly, the convergence comparison of proposed model with other methods for the domains of Education, House and Garden, Musical Instruments, and Sports and Outdoors are shown in figure 15, 16, 17, and 18, respectively, which clearly shows that the proposed model converged early with lower MAE values as compared to state-of-the-art methods. Using the explicit and implicit trust relations as trust relevancy and importance of features learning across domains which help find neighbors, results in significant improvement of rating prediction process.

F. COMPARISON OF RESULTS

We compared proposed model with average MAE of rating-based methods, trust-based models and cross domain model concerning all view of users. From Table 15, 16, 17 and 18,

TABLE 15. Comparison of results in different domains for all users.

Domain	Rating based Methods Avg. MAE	Trust based Models Avg. MAE	Cross domain Model Avg. MAE	TrustC TR Model MAE
Electronics	**0.79**	0.85	**0.79**	**0.73**
House and Garden	**0.83**	0.85	0.94	**0.71**
Education	1.11	1.11	**0.93**	**0.52**
Musical Instruments	0.67	0.74	**0.65**	**0.60**
Sports and Outdoors	0.85	**0.76**	0.77	**0.71**

TABLE 16. Comparison of results in different domains for cold start users.

Domain	Rating based Methods Avg. MAE	Trust based Models Avg. MAE	Cross domain Model Avg. MAE	TrustC TR Model MAE
Electronics	0.77	**0.74**	0.81	**0.69**
House and Garden	0.97	1.00	**0.62**	**0.60**
Education	**0.79**	0.82	0.80	**0.77**
Musical Instruments	**0.68**	0.72	0.69	**0.66**
Sports and Outdoors	0.89	0.92	**0.88**	**0.84**

TABLE 17. Comparison of results in different domains for heavy raters.

Domain	Rating based Methods Avg. MAE	Trust based Models Avg. MAE	Cross domain Model Avg. MAE	TrustC TR Model MAE
Electronics	0.85	**0.84**	0.86	**0.79**
House and Garden	0.78	0.82	**0.75**	**0.73**
Education	**0.65**	0.74	0.81	**0.53**
Musical Instruments	**0.64**	0.69	0.65	**0.50**
Sports and Outdoors	**0.82**	0.86	0.98	**0.76**

TABLE 18. Comparison of results in different domains for opinion users.

Domain	Rating based Methods Avg. MAE	Trust based Models Avg. MAE	Cross domain Model Avg. MAE	TrustC TR Model MAE
Electronics	**0.85**	0.88	0.96	**0.76**
House and Garden	**0.82**	0.88	0.88	**0.77**
Education	**0.78**	0.87	0.86	**0.74**
Musical Instruments	0.80	0.82	**0.75**	**0.74**
Sports and Outdoors	**0.77**	0.80	1.04	**0.72**

it is obvious that TrustCTR has given improved results as compared with all three types of models in each domain. In comparing with rating-based methods, we observed that this significant improvement in proposed model is due to incorporating the rating of target domain based on user-item interactions with respect to time. In comparing with trust-based recommendations, the results are quite improved due to the use of explicit and implicit trust relationships across domains in the form of trust metric and the use of meta-heuristics of ACO for filtering trusted neighbors, in prediction process. Moreover, in comparing with cross domain model, the prediction process of proposed model is improved as we related the time with user preferences and learned the latent features for users and items with trust relations across two domains.

Cross-domain model and rating based models performed well on large and sparse datasets as compared to trust-based methods for All Users and Cold Start Users, as given in Table 15 and XVI, respectively. From Table 17, it is also noticeable that for Heavy Raters, rating based methods and trust-based models provided improved results in different domains as compared to cross-domain models. For Opinion Users, rating-based methods provided better results in all domains except Musical Instruments.

VII. CONCLUSION

Nowadays, recommender systems are facing cold start and data sparsity problems. Collaborative filtering techniques have been failed and are unable to generate quality recommendations; however, hybrid models provide better recommendations as compared to traditional collaborative filtering models. Recently, recommender systems based on trust are beneficial in the improvement of quality recommendations and solve the problem of data sparsity and cold start. In a cross-domain scenario, trust-based recommendations recommended the items to the source users from the target domain. One of the challenges in trust-based recommender systems is that the trust relationship among users can change with time. Another challenge is that the user-item ratings vary with time and may become out of date. In our work, we also found data sparsity in cross domains about more than 99%. This was another challenged for proposed model that we did not find any dense rating system in both datasets. We addressed all these mentioned challenges of recommender systems. Notably, we proposed a model that solves a user cold-start problem in a cross-domain scenario of 'No Overlap' at category or item level domain by incorporating preferences. Our model predicts rating by integrating the neighborhood and latent factor models with baseline estimates and distance metrics to understand the dynamics of user preferences in cross-domain social networks. We also introduced the concept of trust relevancy that represents the trust degree among the trustors and trustees. For the quality of recommendations, we exploited the Ant colony system to find close neighbors of an active user. We have performed experiments on two public datasets and considered five cross domains. The evaluation gives satisfactory results. There are some interesting findings, our model overcomes the problem of data sparsity in the rating system and solves a user cold-start problem by generating recommendations from target domains to an active user by incorporating trust and preferences drift with temporal aspects. If there exists sparsity in source domain rating system and target domain rating system then TrustCTR model still has the capability to generate better recommendations than other baseline algorithms. We also investigated the applications of our model and found that it improves the rating prediction and trust prediction. In the future, for generating quality recommendations and enhancing the efficiency of recommendation systems, we will employ user-item ratings and distrust relations from multiple domains with time sensitive information.

REFERENCES

[1] B. Abdollahi and O. Nasraoui, "A cross-modal warm-up solution for the cold-start problem in collaborative filtering recommender systems," in *Proc. ACM Conf. Web Sci. (WebSci)*, 2014, pp. 257–258, doi: 10.1145/2615569.2615665.

[2] C.-M. Au Yeung and T. Iwata, "Strength of social influence in trust networks in product review sites," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2011, pp. 495–504, doi: 10.1145/1935826.1935899.

[3] P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1183–1190, Jan. 2012, doi: 10.1016/j.eswa.2011.07.124.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013, doi: 10.1016/j.knosys.2013.03.012.

[5] P. Borzymek, M. Sydow, and A. Wierzbicki, "Enriching trust prediction model in social network with user rating similarity," in *Proc. Int. Conf. Comput. Aspects Social Netw.*, Jun. 2009, pp. 40–47, doi: 10.1109/cason.2009.30.

[6] A. L. Carrasco, "Towards trust-aware recommendations in social networks," Ph.D. dissertation, Facultat d'Informàtica de Barcelona, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona, Spain, 2012.

[7] C. C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Inf. Sci.*, vol. 224, pp. 19–36, Mar. 2013, doi: 10.1016/j.ins.2012.10.037.

[8] P. Cremonesi, A. Tripodi, and R. Turrin, "Cross-domain recommender systems," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 496–503, doi: 10.1109/icdmw.2011.57.

[9] P. Cremonesi and M. Quadrana, "Cross-domain recommendations without overlapping data: Myth or reality?" in *Proc. 8th ACM Conf. Recommender Syst.-RecSys*, 2014, pp. 297–300, doi: 10.1145/2645710.2645769.

[10] I. Dagan, L. Lee, and F. Pereira, "Similarity-based methods for word sense disambiguation," in *Proc. 35th Annu. Meeting Assoc. Comput. Linguistics*, 1997, pp. 56–63, doi: 10.3115/976909.979625.

[11] S. Deng, L. Huang, and G. Xu, "Social network-based service recommendation with trust enhancement," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8075–8084, 2014, doi: 10.1016/j.eswa.2014.07.012.

[12] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017, doi: 10.1109/tnnls.2016.2514368.

[13] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2005, pp. 485–492, doi: 10.1145/1099554.1099689.

[14] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[15] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 278–288, doi: 10.1145/2736277.2741667.

[16] S. Gao, H. Luo, D. Chen, S. Li, P. Gallinari, and J. Guo, "Cross-domain recommendation via cluster-level latent factor model," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 161–176, doi: 10.1007/978-3-642-40991-2_11.

[17] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *ACM Trans. Web*, vol. 3, no. 4, pp. 1–33, Sep. 2009, doi: 10.1145/1594173.1594174.

[18] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, Jul. 2012, doi: 10.1109/tnnls.2012.2197827.

[19] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, "Propagation of trust and distrust," in *Proc. 13th Conf. World Wide Web (WWW)*, 2004, pp. 403–412, doi: 10.1145/988672.988727.

[20] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 397–406, doi: 10.1145/1557019.1557067.

[21] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst. (RecSys)*, 2010, pp. 135–142, doi: 10.1145/1864708.1864736.

[22] H. Jing, A.-C. Liang, S.-D. Lin, and Y. Tsao, "A transfer probabilistic collective factorization model to handle sparse data in collaborative filtering," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 250–259, doi: 10.1109/icdm.2014.68.

[23] M. M. Khan, R. Ibrahim, and I. Ghani, "Cross domain recommender systems: A systematic literature review," *ACM Comput. Surv.*, vol. 50, no. 3, pp. 1–34, Oct. 2017, doi: 10.1145/3073565.

[24] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: Issues, challenges, and research opportunities," in *Proc. Inf. Sci. Appl. (ICISA)*, 2016, pp. 1179–1189, doi: 10.1007/978-981-10-0557-2_112.

[25] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 426–434, doi: 10.1145/1401890.1401944.

[26] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[27] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 447–456, doi: 10.1145/1557019.1557072.

[28] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. New York, NY, USA: Springer, 2015, pp. 77–118, doi: 10.1007/.978-1-4899-7637-6_3.

[29] D. Kotkov, S. Wang, and J. Veijalainen, "Cross-domain recommendations with overlapping items," in *Proc. 12th Int. Conf. Web Inf. Syst. Technol.*, 2016, pp. 1–9, doi: 10.5220/0005851301310138.

[30] S. Kullback, *Information Theory and Statistics*. Chelmsford, MA, USA: Courier Corp., 1997, doi: 10.1007/978-3-642-04898-2_643.

[31] J. Wang, J. Yin, Y. Liu, and C. Huang, "Trust-based collaborative filtering," in *Proc. 8th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Jul. 2011, pp. 119–134, doi: 10.1109/fskd.2011.6020048.

[32] K. Lewis, M. Gonzalez, and J. Kaufman, "Social selection and peer influence in an online social network," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 1, pp. 68–72, Jan. 2012, doi: 10.1073/pnas.1109739109.

[33] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 2052–2057, doi: 10.5555/1661445.1661773.

[34] B. Li, X. Zhu, R. Li, C. Zhang, X. Xue, and X. Wu, "Cross-domain collaborative filtering over time," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 2293–2298, doi: 10.5555/2283696.2283780.

[35] W. Liu, C. Wu, B. Feng, and J. Liu, "Conditional preference in recommender systems," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 774–788, Feb. 2015, doi: 10.1016/j.eswa.2014.08.044.

[36] H. Liu, E.-P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. A. Kim, "Predicting trusts among users of online communities: An epinions case study," in *Proc. 9th ACM Conf. Electron. Commerce EC*, 2008, pp. 310–319, doi: 10.1145/1386790.1386838.

[37] G. Ma, Y. Wang, X. Zheng, and M. Wang, "Leveraging transitive trust relations to improve cross-domain recommendation," *IEEE Access*, vol. 6, pp. 38012–38025, 2018, doi: 10.1109/access.2018.2850706.

[38] 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, doi: 10.1145/1458082.1458205.

[39] 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)*, 2009, pp. 203–210, doi: 10.1145/1571941.1571978.

[40] 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)*, 2011, pp. 287–296, doi: 10.1145/1935826.1935877.

[41] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Recommender Syst. RecSys*, 2007, pp. 17–24, doi: 10.1145/1297231.1297235.

[42] P. Massa and P. Avesani, "Trust metrics in recommender systems," in *Computing With Social Trust*. London, U.K.: Springer, 2009, pp. 259–285, doi: 10.1007/978-1-84800-356-9_10.

[43] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociology*, vol. 27, no. 1, pp. 415–444, Aug. 2001, doi: 10.1146/annurev.soc.27.1.415.

[44] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264, doi: 10.5555/2981562.2981720.

[45] H. H. Moe and W. T. Aung, "Context-aware cross-domain based recommendation," in *Proc. Int. Conf. Adv. Eng. Technol.*, 2014, p. 29, doi: 10.15242./iie.e0314 035.

[46] M. Montaner, B. López, and J. L. De La Rosa, "A taxonomy of recommender agents on the Internet," *Artif. Intell. Rev.*, vol. 19, pp. 285–330, Jun. 2003, doi: 10.1023/A:1022850703159.

[47] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz, "Trust-based recommendation systems: An axiomatic approach," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, 2008, pp. 199–208, doi: 10.1145/1367497.1367525.

[48] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang, "Transfer learning to predict missing ratings via heterogeneous user feedbacks," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 2318–2323, doi: 10.5591/978-1-57735-516-8/IJCAI11-386.

[49] H. Parvin, P. Moradi, and S. Esmaeili, "TCFACO: Trust-aware collaborative filtering method based on ant colony optimization," *Expert Syst. Appl.*, vol. 118, pp. 152–168, Mar. 2019, doi: 10.1016/j.eswa.2018.09.045.

[50] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*, 1994, pp. 175–186, doi: 10.1145/192844.192905.

[51] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, no. 3, pp. 56–59, Mar. 1997, doi: 10.1007/978-3-319-29659-3.

[52] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Cham, Switzerland: Springer, 2011, pp. 1–35, doi: 10.1007/978-0-387-85820-3_1.

[53] S. Sahebi and P. Brusilovsky, "Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation," in *Proc. Int. Conf. User Modeling, Adaptation, Personalization*, 2013, pp. 289–295, doi: 10.1007/978-3-642-38844-6_25.

[54] N. Shahabi and F. Najian, "A new strategy in trust-based recommender system using K-means clustering," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 9, pp. 152–156, 2017, doi: 10.14569/ijacsa.2017.080922.

[55] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web (WWW)*, vol. 1, 2001, pp. 285–295, doi: 10.1145/371920.372071.

[56] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2002, pp. 253–260, doi: 10.1145/564376.564421.

[57] N. Sherif and G. Zhang, "Collaborative filtering using probabilistic matrix factorization and a Bayesian nonparametric model," in *Proc. IEEE 2nd Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2017, pp. 391–396, doi: 10.1109/icbda.2017.8078847.

[58] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, vol. 1, 2nd ed. Cham, Switzerland: Springer, 2015, p. 156, doi: 10.1007/978-1-4899-7637-6.

[59] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, Dec. 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.9319

[60] J. Tang, H. Gao, and H. Liu, "MTrust: Discerning multi-faceted trust in a connected world," in *Proc. 5th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2012, pp. 93–102, doi: 10.1145/2124295.2124309.

[61] J. Tang, H. Liu, H. Gao, and A. Das Sarmas, "ETrust: Understanding trust evolution in an online world," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 253–261, doi: 10.1145/2339530.2339574.

[62] J. Tang, H. Gao, A. Das Sarma, Y. Bi, and H. Liu, "Trust evolution: Modeling and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1724–1738, Jun. 2015, doi: 10.1109/TKDE.2014.2382576.

[63] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2712–2718, doi: 10.5555/254.0128.2540519.

[64] P. Tengkiattrakul, S. Maneeroj, and A. Takasu, "Applying ant-colony concepts to trust-based recommender systems," in *Proc. 18th Int. Conf. Inf. Integr. Web-based Appl. Services (iiWAS)*, 2016, pp. 34–41, doi: 10.1145/3011141.3011161.

[65] C. Than and S. Han, "Improving recommender systems by incorporating similarity, trust and reputation," *J. Internet Services Inf. Secur.*, vol. 4, no. 1, pp. 64–76, 2014, doi: 10.22667/JISIS.2014.02.31.064.

[66] I. F. Tobías, "Matrix factorization models for cross-domain recommendation: Addressing the cold start in collaborative filtering," Ph.D. dissertation, Departamento de Ingeniería Informática, Univ. Autónoma de Madrid, Madrid, Spain, 2017.

[67] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Auto. Agents Multi-Agent Syst.*, vol. 16, no. 1, pp. 57–74, Feb. 2008, doi: 10.1007/s10458-007-9021-x.

[68] J. Wang, Z. Deng, K.-S. Choi, Y. Jiang, X. Luo, F.-L. Chung, and S. Wang, "Distance metric learning for soft subspace clustering in composite kernel space," *Pattern Recognit.*, vol. 52, pp. 113–134, Apr. 2016, doi: 10.1016/j.patcog.2015.10.018.

[69] P. Wu, S. C. H. Hoi, P. Zhao, C. Miao, and Z.-Y. Liu, "Online multimodal distance metric learning with application to image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 454–467, Feb. 2016, doi: 10.1109/TKDE.2015.2477296.

[70] Q. Xu, E. Xiang, and Q. Yang, "Social-behavior transfer learning for recommendation systems," in *Proc. Workshop Social Web Mining*, 2011, pp. 1–9. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.231.3676&rep=rep1&type=pdf

[71] Z. Xu, F. Zhang, W. Wang, H. Liu, and X. Kong, "Exploiting trust and usage context for cross-domain recommendation," *IEEE Access*, vol. 4, pp. 2398–2407, 2016, doi: 10.1109/ACCESS.2016.2566658.

[72] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Michigan State Univ., Lansing, MI, USA, Tech. Rep. 2, 2006, vol. 2, no. 2, p. 4. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.91.4732

[73] W. Yuan and D. Guan, "Optimized trust-aware recommender system using genetic algorithm," *Neural Netw. World*, vol. 27, no. 1, pp. 77–94, 2017, doi: 10.14311/NNW.2017.27.004.

[74] J. Yu, M. Gao, W. Rong, Y. Song, and Q. Xiong, "A social recommender based on factorization and distance metric learning," *IEEE Access*, vol. 5, pp. 21557–21566, 2017, doi: 10.1109/ACCESS.2017.2762459.

[75] R. Zafarani, M. A. Abbasi, and H. Liu, *Social Media Mining: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2014, doi: 10.1017/CBO9781139088510.

[76] F. Zafari and I. Moser, "Modelling socially-influenced conditional preferences over feature values in recommender systems based on factorised collaborative filtering," *Expert Syst. Appl.*, vol. 87, pp. 98–117, Nov. 2017, doi: 10.1016/j.eswa.2017.05.058.

[77] F. Zafari, F. Nassiri-Mofakham, and A. Z. Hamadani, "Dopponent: A socially efficient preference model of opponent in bilateral multi issue negotiations," *J. Comput. Secur.*, vol. 1, no. 4, pp. 283–292, 2014.

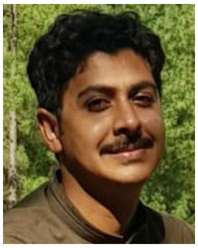
[78] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 381–389, doi: 10.1137/1.9781611974973.43.

[79] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 83–92, doi: 10.1145/2600428.2609579.

[80] Z. K. Zhang, C. Liu, Y. Zhang, and Z. Zhou, "Solving the cold-start problem in recommender systems with social tags," *EPL*, vol. 92, no. 2, p. 28002, 2010, doi: 10.1209/0295-5075/92/28002.

[81] H. Zhao, S. Wang, Q. Chen, and J. Cao, "Probabilistic matrix factorization based on similarity propagation and trust propagation for recommendation," in *Proc. IEEE Conf. Collaboration Internet Comput. (CIC)*, Oct. 2015, pp. 90–98, doi: 10.1109/CIC.2015.11.

[82] F. Zhuang, J. Zheng, J. Chen, X. Zhang, C. Shi, and Q. He, ''Transfer collaborative filtering from multiple sources via consensus regularization,'' *Neural Netw.*, vol. 108, pp. 287–295, Dec. 2018, doi: 10.1016/j.neunet.2018.08.022.
[83] C.-N. Ziegler and J. Golbeck, ''Investigating interactions of trust and interest similarity,'' *Decis. Support Syst.*, vol. 43, no. 2, pp. 460–475, Mar. 2007, doi: 10.1016/j.dss.2006.11.003.
[84] H. Zou, Z. Gong, N. Zhang, Q. Li, and Y. Rao, ''Adaptive ensemble with trust networks and collaborative recommendations,'' *Knowl. Inf. Syst.*, vol. 44, no. 3, pp. 663–688, Sep. 2015, doi: 10.1007/s10115-014-0782-7.
[85] 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, doi: 10.1016/j.future.2017.04.003.
[86] S. Yadav, K. Vikesh, S. Shreyam, and S. Nagpal, ''Trust aware recommender system using swarm intelligence,'' *J. Comput. Sci.* vol. 28, pp. 180–192, Sep. 2018, doi: 10.1016/j.jocs.2018.09.007.
[87] Y. Pan, F. He, and H. Yu, ''A correlative denoising autoencoder to model social influence for top-N recommender system,'' *Frontiers Comput. Sci.*, vol. 14, no. 3, Jun. 2020, Art. no. 143301, doi: 10.1007/s11704-019-8123-3.
[88] H. Parvin, P. Moradi, S. Esmaeili, and M. Jalili, ''An efficient recommender system by integrating non-negative matrix factorization with trust and distrust relationships,'' in *Proc. IEEE Data Sci. Workshop (DSW)*, Jun. 2018, pp. 135–139, doi: 10.1109/.dsw.2018.8439905.
[89] L. Jiang, Y. Cheng, L. Yang, J. Li, Y. Hongyang, and X. Wang, ''A trust-based collaborative filtering algorithm for E-commerce recommendation system,'' *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3023–3034, 2019, doi: 10.1007/.s12652-018-0928-7.
[90] G. Bathla, H. Aggarwal, and R. Rani, ''AutoTrustRec: Recommender system with social trust and deep learning using AutoEncoder,'' *Multimedia Tools Appl.*, pp. 1–16, Apr. 2020, doi: 10.1007/s11042-020-08932-4.
[91] M. Dong, F. Yuan, L. Yao, X. Wang, X. Xu, and L. Zhu, ''Trust in recommender systems: A deep learning perspective,'' 2020, *arXiv:2004.03774*. [Online]. Available: http://arxiv.org/abs/2004.03774
[92] J. Son, W. Choi, and S.-M. Choi, ''Trust information network in social Internet of Things using trust-aware recommender systems,'' *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 4, Apr. 2020, Art. no. 155014772090877, doi: 10.1177/1550147720908773.
[93] I. Belkhadir, E. D. Omar, and J. Boumhidi, ''An intelligent recommender system using social trust path for recommendations in Web-based social networks,'' *Procedia Comput. Sci.*, vol. 148, pp. 181–190, 2019, doi: 10.1016/j.procs.2019.01.035.
[94] P. Moradi and S. Ahmadian, ''A reliability-based recommendation method to improve trust-aware recommender systems,'' *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7386–7398, 2015, doi: 10.1016./j.eswa.2015.05.027.
[95] P. Li and A. Tuzhilin, ''DDTCDR: Deep dual transfer cross domain recommendation,'' in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 331–339, doi: 10.1145/3336191.3371793.
[96] J. He, R. Liu, F. Zhuang, F. Lin, C. Niu, and Q. He, ''A general cross-domain recommendation framework via Bayesian neural network,'' in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1001–1006, doi: 10.1109/ICDM.2018.00125.
[97] T.-H. Lin, C. Gao, and Y. Li, ''CROSS: Cross-platform recommendation for social E-Commerce,'' in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 515–524, doi: 10.1145/3331184.3331191.
[98] Z. Liu, L. Zheng, J. Zhang, J. Han, and P. S. Yu, ''JSCN: Joint spectral convolutional network for cross domain recommendation,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 850–859, doi: 10.1109/BigData47090.2019.9006266.
[99] G. Hu, Y. Zhang, and Q. Yang, ''CoNet: Collaborative cross networks for cross-domain recommendation,'' in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 667–676, doi: 10.1145/3269206.3271684.
[100] F. Yuan, L. Yao, and B. Benatallah, ''DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns,'' in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–7, doi: 10.24963/ijcai.2019/587.
[101] 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, doi: 10.1016/j.osnem.2018.05.001.
[102] B. D. Deebak and F. Al-Turjman, ''A novel community-based trust aware recommender systems for big data cloud service networks,'' *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102274, doi: 10.1016/j.scs.2020.102274.
[103] J. Lee, G. Noh, H. Oh, and C.-K. Kim, ''Trustor clustering with an improved recommender system based on social relationships,'' *Inform. Syst.*, vol. 77, pp. 118–128, Sep. 2018, doi: 10.1016/j.is.2018.05.008.
[104] G. Noh, H. Oh, and J. Lee, ''Power users are not always powerful: The effect of social trust clusters in recommender systems,'' *Inf. Sci.*, vol. 462, pp. 1–15, Sep. 2018, doi: 10.1016/j.ins.2018.05.058.
[105] L. Sheugh and S. H. Alizadeh, ''A novel 2D-graph clustering method based on trust and similarity measures to enhance accuracy and coverage in recommender systems,'' *Inf. Sci.*, vol. 432, pp. 210–230, Mar. 2018, doi: 10.1016/j.ins.2017.12.007.
[106] Z. Zhang, Y. Liu, J. Zhigan, and R. Zhang, ''A dynamic trust based two-layer neighbor selection scheme towards online recommender systems,'' *Neurocomputing*, vol. 285, pp. 94–103, Apr. 2018, doi: 10.1016/j.neucom.2017.12.063.
[107] G. Guo, J. Zhang, and N. Yorke-Smith, ''Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems,'' *Knowl.-Based Syst.*, vol. 74, pp. 14–27, Jan. 2015, doi: 10.1016/.j.knosys.2014.10.016.
[108] L. Ardissono and N. Mauro, ''A compositional model of multi-faceted trust for personalized item recommendation,'' *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112880, doi: 10.1016/j.eswa.2019.112880.
[109] L. G. Zucker, ''Production of trust: Institutional sources of economic structure,'' *Res. Organizational Behav.*, vol. 2, pp. 1840–1920, 1986.
[110] B. C. Mayer, J. Dressler, F. Harlow, G. Brault, and K. S. Candan, ''Replicating multi-quality Web applications using ACO and bipartite graphs,'' in *Proc. Int. Workshop Ant Colony Optim. Swarm Intell.*, 2006, pp. 270–281, doi: 10.1007/11839088_24.



ADEEL AHMED (Graduate Student Member, IEEE) received the M.Phil. degree in computer science from Quaid-i-Azam University, Islamabad, Pakistan, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He was with the software industry for some years. His research interests include swarm intelligence, machine learning, recommendation systems, social network analysis, and information visualization.

KHALID SALEEM (Member, IEEE) received the M.Sc. degree from Quaid-i-Azam University, Pakistan, in 1994, and the M.Phil. and Ph.D. degrees from the University of Montpellier 2, France, in 2005 and 2008, respectively, all in computer science. He was with the software industry for some years. He is currently an Assistant professor with Quaid-i-Azam University. He is the President of PAK-France Alumni Network. His research interests include machine learning, deep learning, schema matching and integration, data analytics, stegnography, cryptography, and mobile computing.

UMER RASHID received the B.S. (CS) degree from the University of Lahore, Pakistan, in 2005, and the M.Phil. and Ph.D. degrees from Quaid-i-Azam University, Islamabad, Pakistan, in 2008 and 2017, respectively, all in computer science. He has teaching and research experience in international organizations. He is currently serving as an Assistant Professor with Quaid-i-Azam University. His research interests include user-centered computing, multimedia information retrieval, and multimedia technology. His work is published in international journals and conferences.

ABDULLAH BAZ (Senior Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Umm Al-Qura University (UQU), in 2002, the M.Sc. degree in electrical and computer engineering from KAU, in 2007, and the M.Sc. degree in communication and signal processing and the Ph.D. degree in computer system design from Newcastle University, in 2009 and 2014, respectively. He was the Vice-Dean and then the Dean of the Deanship of Scientific Research with UQU, from 2014 to 2020. He is currently an Assistant Professor with the Computer Engineering Department, the Vice-Dean of DFMEA, the General Director of the Decision Support Center, and the Consultant of the University Vice Chancellor with UQU. His research interests include data science, ML, AI, VLSI design, EDA/CAD tools, intelligent transportation, computer systems and architecture, smart systems, and smart health. Since 2015, he has been serving as a Review Committee Member of the IEEE International Symposium on Circuits and Systems (ISCAS) and a member of the Technical Committee of the IEEE VLSI Systems and Applications. He served as a Reviewer for a number of journals, including the IEEE INTERNET OF THINGS, the *IET Computer Vision*, the *Artificial Intelligence Review*, and the *IET Circuits, Devices and Systems*.