Journal of King Saud University - Computer and Information Sciences 36 (2024) 101964

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University - Computer and Information Sciences

journal homepage: www.sciencedirect.com

Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis

H. Hamidi *, R. Moradi

Department of Industrial Engineering, Information Technology Group, K. N. Toosi University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Shilling attacks
Robust recommender systems
Dynamic recommender system
Social networks
Collaborative filtering

ABSTRACT

Collaborative filtering recommender systems type have been increasingly used in e-commerce sites both to facilitate users' decision-making and increase sales. On the one hand, the open and interactive nature of recommender systems makes them vulnerable to shilling attacks, and on the other hand, given that most recommender systems are used in dynamic environments, and this issue generates incremental data over time. One of the main obstacles of this type of recommender systems is the inability to model the dynamic behavior of users and the incremental flow of data, as well as the vulnerability to shilling attacks. Therefore, classic models do not have the necessary ability to provide suitable recommendations to the user and also to detect shilling attacks dynamically. The objective of this study was to address this gap by designing a dynamic and robust recommender system model against shilling attacks. This model was based on item context, trust, users' rating and users' rating time, and it benefits from the social networks analysis of users and items to suggest @N top items to the target user, and had a dynamic and increasing property over time and robust against shilling attacks. Finally, to assess the performance and robustness of the recommender system to shilling attacks, 4850 different tests have been performed without shilling attacks and under three average, random, and bandwagon attacks. To validate the proposed model, the results of the tests have been compared with similar methods such as TRACCF, TOTAR and T&TRS using the evaluation criteria of Precision, Recall, F1, MAE and RMSE.The results depicted that the proposed method, due to finding users and similar items in communities created by social networks, causes a reduction in the number of predicted items that are not liked by the user (FP) and the number of unpredicted items that are liked by the user (FN) and finally the F1 criterion (which is a combination of Precision and Recall criteria) performs better than the comparison methods. Also, this method is robust against the shilling attacks by detecting fake profiles and ignoring them in the recommendation process, and the evaluation criteria before and after shilling attacks show this.

1. Introduction

Currently, collaborative filtering recommender systems type are referred to as the most widely used attitude in recommender systems. The systems are commonly used in electronic commerce sites to help users select the right items and increase sales (Jia and Liu, 2015). When a recommender system by a collaborative filtering approach wants to recommend an item to a user, it does so based on the similarity and trust between two users.

In most recommender system algorithms based on collaborative filtering, the trust between two users is used to recommend all items, if the trust is based on context (Cerutti et al., 2015; Bo et al., 2017). According to the similarity and trust, in practice it is also observed that people's tastes are not the same in all contexts (Yang and Niu, 2023). For example, two users may have the same taste in digital devices, but not the same taste in cosmetics. Therefore, the simultaneous use of item context and trust can lead to an increase in the precision of the recommendations of the recommender system.

Due to the fact that there is no obstacle for people to post comments about items on websites and online stores, this position has given the strength of recommender systems (Si and Li, 2020). But the important point is the vulnerability of these systems against the registration of opinions of profit-seeking people. This vulnerability is known as shilling attack. Shilling attack is carried out by profiteers with the aim of increasing the rating of their own item or lowering the rating of competing items (Mobasher et al., 2007a; Burke et al., 2015).

* Corresponding author.
E-mail addresses: h_hamidi@kntu.ac.ir (H. Hamidi), ra.moradi@email.kntu.ac.ir (R. Moradi).

https://doi.org/10.1016/j.jksuci.2024.101964
Received 19 November 2023; Received in revised form 9 January 2024; Accepted 1 February 2024
Available online 13 February 2024

A shilling attack causes a change in the results of the recommender system. For that reason, it is necessary to detect shilling attacks and improve recommender systems in order to neutralize the effects of attacks in the systems. Since shilling attacks are done by the adversary with the same rating patterns and at limited intervals, to detect fake profiles and reduce the effect of shilling attacks, using the rating matrix and rating time can lead to reducing the effects of shilling attacks.

On the other hand, most methods for detecting shilling attacks and robust recommender systems have been designed with the assumption of static data (user rating and fake profile) and available data, but the important issue is the dynamics of data over time and the incremental data-flow in practice (Kumar et al., 2019; Zhang et al., 2017a). This incremental data-flow cannot be modeled by classical algorithms because the system performance reduces in reality. For this purpose, the use of social networks between items and users in order to communities detection with the ability to manage the increasing data-flow can compensate for this inability and increase the precision of the recommender system.

To fill the gaps, this study proposed a dynamic and robust recommender system model against shilling attacks based on item context, trust, user rating, and user rating time using social networks. The article is organized as follows. section 2 introduced the basic concepts of trust and shillings attack. In section 3, the related works includes the topic of robust recommender systems and related works in the field of providing recommendations in the recommender system are reviewed. section 4 presented the proposed method. in section 5, experimental evaluation has been done for the results. Finally, in section 6, conclusion is described.

2. Basic concepts

In this section, the examination of the concept of trust and its characteristics is initiated. Subsequently, brief explanations are given for the common definitions and terms related to shilling attacks, which have been employed in this research.

Trust: in recommender systems, trust between two users is a relationship by having similar preferences for items rated by users (Lathia et al., 2008; Shambour and Lu, 2012). Researchers have cased four important characteristics of similarity and trust as follows (Cerutti et al., 2015; Bo et al., 2017).

-Asymmetry/ subjectivity: according to Eq. (1) for two users u_1 and user u_2 we have:

$$Trust(u_1, u_2) \neq Trust(u_2, u_1) \tag{1}$$

-Transitivity: if user u_1 trusts in user u_2 and user u_2 trusts in user u_3, then user u_1 trusts in user u_3.

-Dynamics / temporality: users' trust and tastes are related to their past interactions and may change over time.

-Context dependence: there is trust and similarity of users in certain context.

In recommender systems, a person or a group of people who intend to attack the recommender system in order to change its result (Moradi and Hamidi, 2023). is called an **Adversary**. The adversary tries to change the result of the system's recommendations by registering fake feedback in the recommender system. This adversary action is called a **shilling attack**. In a recommender system, the set of user rating behaviors to the items in the system is recorded under the title of **profile**. In order to create a shilling attack, the adversary creates fake users and ratings items and injects **fake profiles** into the system. adversary injects fake profiles into the system to change the result of the recommender system. Each adversary typically performs an action with an intention referred to as the **attack intent**. Usually, the adversary targets push or nuke. In the push, the adversary injects fake profiles into the system in a way that increases the probability of one or more items being popular. In the nuke, the adversary injects fake profiles to reduce the popularity of the competing item(s). Another intent of the shilling attack is random sabotage (Burke et al., 2015) by disrupting recommendation algorithms and in order to reduce users' trust in the recommender system. The injection of fake profiles by an adversary into the recommender system is carried out using specific methods and models. Subsequently, terminologies related to this subject are introduced.

- Filler size: the number of items in the recommender system that are rated by fake profiles. Usually, the filler size is considered between 1 and 20 % of the total items.
- Attack size: the number of fake profiles in a shilling attack, which is usually between 1 and 15 % of the items in the recommender system.
- Attack cost: Injecting a shilling attack for the adversary has a knowledge cost and a deployment cost. The knowledge cost is related to the effort to obtain information from the recommender system and deployment cost is related to the effort to inject fake profiles (Burke et al., 2015). The best attack from the adversary's point of view is the attack with the greatest effect on changing the result of the recommender system with the lowest knowledge cost and deployment cost.
- Target item: the item or items that are attacked by the adversary (O'Mahony et al., 2005).
- Model of fake profiles and types of shilling attacks: as shown in Figs. 1 and 2, if we consider the fake profile injected by the adversary with the push or nuke as an n-dimensional vector of ratings, then this vector contains the item or a collection of target items i_t along with a rating function γ allocating a rating value and rates the target item(s) as maximum or minimum according to the attack intent. In some shilling attacks, a collection of items with specific features are also rated by the adversary, which is considered I_S. In order to normalize and obfuscate fake profiles to increase the difficulty of detection, a collection of items I_F is rated with a function σ. $I_\varnothing$ is the collection of items that are not rated. Different models of shilling attacks are created using the selection method and rating strategies to the above collections (Bhaumik et al., 2006; Mobasher et al.,2007b). The four famous attack models of random, average, bandwagon and reverse bandwagon in collaborative filtering recommender systems type are described according to Fig. 3 (Si and Li, 2020; Moradi and Hamidi, 2023; Rezaimehr and Dadkhah, 2021).

3. Literature review

Given that the proposed model consists of two parts, including robust against shilling attacks and the recommendation algorithm, in the literature review section of the research, the issue of robust recommender systems is first discussed, and then related works in the field of providing recommendations in the recommender system in the form of a table is checked in general. In the following, the robust recommender systems will be discussed.

Recommender systems are commonly used in electronic commerce sites, and registering and rating these items are free for the public. Due to the issue of open and interactive nature of the systems, although it has caused the strength of recommender systems, but it has caused their vulnerability, too (O'Mahony et al., 2004; Mobasher et al., 2006a).

Adverse behavior against the recommender system was first proposed in a study by Dellarocas(Dellarocas, 2000). Changes in recommender system reports through fraudulent behavior such as fake item rating have been discussed in this study. This study presents a collection of solutions to eliminate adverse behavior against the recommender system, which can eliminate the negative impacts of fake ratings and the strength of the recommender system.

In a study by O'Mahony et al. (O'Mahony et al., 2002) attacking recommender systems, the vulnerability of recommender systems and the strength of recommender systems against shilling attacks were raised for the first time.

After introducing shilling attacks, the detection of fake profiles was raised based on different views and concepts involved, leading to the presentation of many algorithms in the field of detecting shilling attacks.

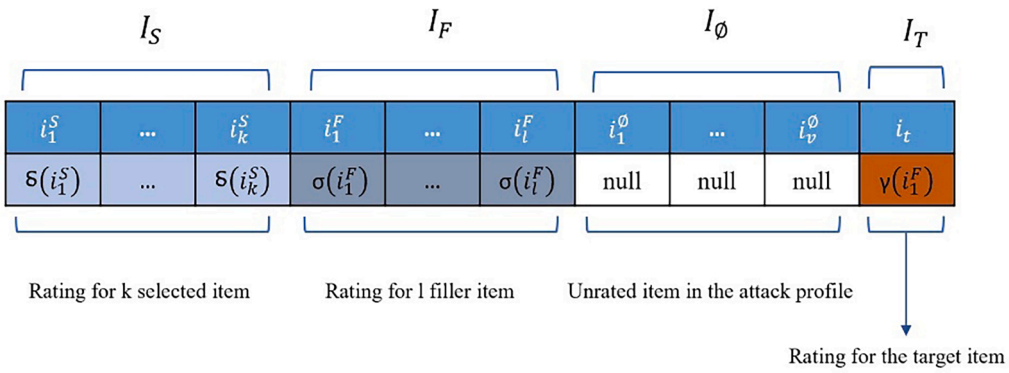

Fig. 1. Fake profiles with a target item. This diagram represents the vector of a fake profile with a target item i_t, k selected item I_S, l filler item I_F and v unrated item..$I_\emptyset$

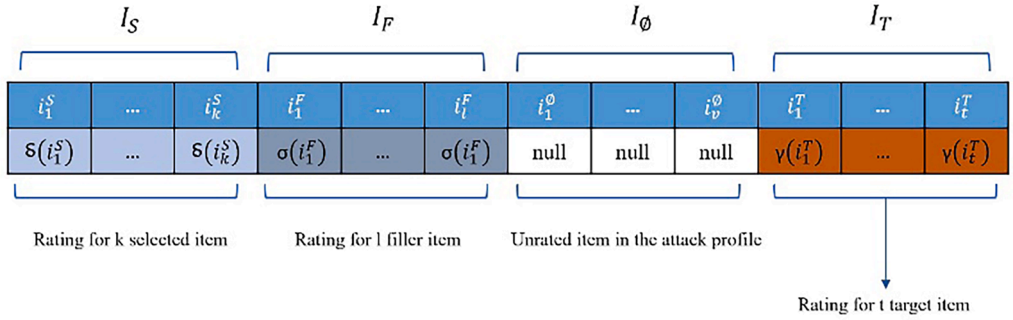

Fig. 2. Fake profiles with multiple target items. This diagram represents the vector of a fake profile with t target item I_t, k selected item I_S, l filler item I_F and v unrated item..$I_\emptyset$

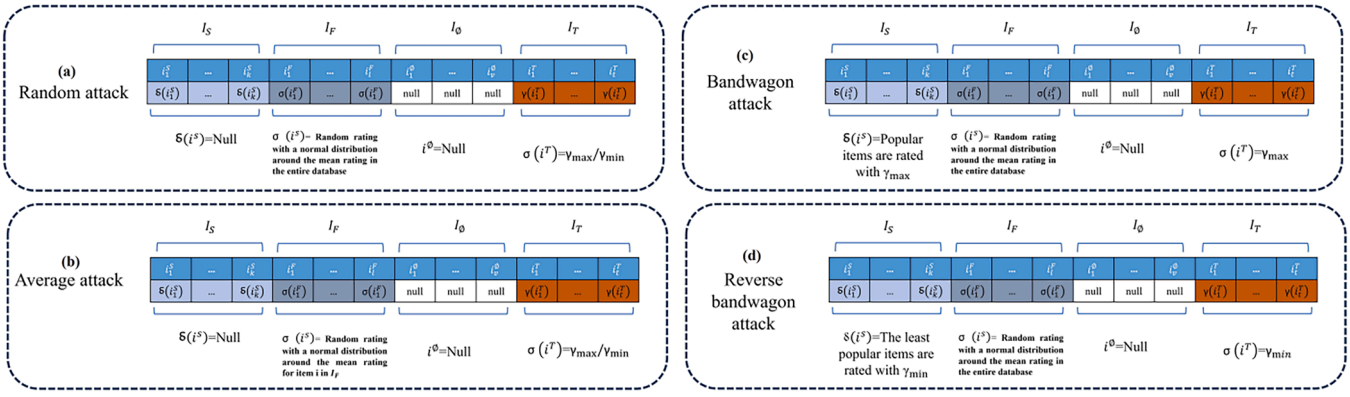

Fig. 3. Random, average, bandwagon and reverse bandwagon shilling attacks. Figure a shows the fake profile vector for the random attack, figure b shows the fake profile vector for the average attack, figure c shows the fake profile vector for the bandwagon attack, and figure d shows the fake profile vector for the reverse bandwagon attack.

But there are other methods to maintain the stability of the recommender system and prevent the negative impacts of fake profiles. One of the most efficient methods is to develop resistant algorithms against shilling attacks. In this section, we will briefly describe previous works on resistant algorithms in recommender systems. In a study by Mehta and Nejdl a recommender system by a new collaborative filtering approach using SVD was presented. The algorithm presented in this study had high precision (Mehta and Nejdl, 2008). In a study by Mehta et al. a robust design was presented using Robust Matrix Factorization (RMF) based on M−estimators (Mehta et al., 2007). In a study by Zhang and Sun using least median squares estimator, a robust method was presented for collaborative filtering recommender systems type (Zhang and Sun, 2014). Zhang et al. proposed a robust approach for collaborative filtering recommender systems based on non-negative matrix factorization (NMF) and R1-norm (Zhang et al., 2017b). Cheng and Hurley proposed least squares based MF (LSMF) and least trimmed squares based MF (LTSMF) method sensitive to far distances (Cheng and Hurley, 2010). Bilge et al. addressed four robust privacy protection methods proposed in collaborative filtering recommender systems against several types of shilling attacks. The empirical tests of this study showed that DWT and K-Means clustering methods are vulnerable to shilling attacks, and SVD-based and item-based methods are robust to shilling attacks (Bilge et al., 2014). Yu et al. using the kernel mapping of the rating matrix and kernel distance and the nearest neighbor model, presented a robust recommender system against shilling attacks, reducing the negative impacts of the attack (Yu et al., 2017).

Yi and Zhang proposed a resistant recommendation algorithm based on suspicious users and multidimensional trust. They presented a multidimensional trust model by user profile features to identify suspicious users and establish implicit trust relationship. The empirical results showed that it performed well in terms of precision and strength (Yi and Zhang, 2016). Mobasher et al. in order to strengthen the recommender system of collaborative filtering, presented a method based on Probabilistic Latent Semantic Analysis (PLSA). In this study, high strength of this method against shilling attacks is proved (Mobasher et al., 2006b).

For resistant algorithms against shilling attacks, the use of trust models has increased in recent years in order to overcome some of the limitations of traditional recommender systems (O'DONOVAN and Smyth, 2006). In a study by Zhang, by the neighbors of a user, calculated the user trust, and then defined the item trust. Next, they calculated the topic trust based on the average rating of the item trust belonging to the same topic. Finally, they combined the trust model with the traditional recommender system model (Zhang and Sheng-hua, 2007).

Gao et al. using rating similarity, interest similarity and linear dependence, presented a robust item-based collaborative filtering recommender system (Gao et al., 2014). Jia et al. presented a resistant algorithm against shilling attacks using the multidimensional trust model. In this approach, the user validity model is considered according to three dimension of item recommendation reliability, rating similarity, and user reliability (Jia et al., 2013). Turk and Bilge presented a robust multi-criteria collaborative filtering recommender system against shilling attacks (Turk and Bilge, 2018).

Alonso et al. proposed a robust innovative method based on matrix factorization to neutralize shilling attacks. In this method, first, the reliability value associated with each user's prediction for an item is obtained. Then, it avoids recommending shilling items to users by monitoring unusual confidence changes in item prediction. The empirical results showed that the method proposed can neutralize most of the shilling attacks (Alonso et al., 2019).

In a study by Yang and Niu instead of using the user trust values, used the trust genre, which are actually different in terms of the item genre, and it takes into account both the value of trust and the validity of the user. The empirical results showed superior and comparable genre trust for defending against different types of shilling attacks (Yang and Niu, 2023).

Rezaimehr and Dadkhah proposed a robust recommender system against shilling attacks using time and trust. The proposed recommender system based on a new community detection algorithm improves user clustering performance to identify fake users. The empirical results showed high precision of the proposed method against average attacks and random attacks (Rezaimehr and Dadkhah, 2023).

After literature review on robust recommender systems, the related works in the field of providing recommendations in the recommender system will be discussed. Given that the proposed model uses item context, trust, rating matrix, rating time, graph and social networks, the works related to these features are shown in Table 1. As shown in Table 1, (Ahmadian et al., 2022), (Ahmadian et al., 2020), (Wahab et al., 2022) and (Birtolo and Ronca, 2013) used trust, (Koren et al., 2021), (Rezaeimehr et al., 2018), (Moradi et al., 2016) and (Daneshmand et al., 2015) used time, (Jiang et al., 2022) and (Al-Ghobari et al., 2021) used graph and community detection. Also, (Feng et al., 2015) and (Rezaimehr and Dadkhah, 2023) used the combination of these features.

One of the important gaps of the reviewed recommender systems is their inability to model the dynamic behavior of users and the incremental flow of data. Recommender systems are used in practice in dynamic environments and data is incrementally entered into the system. Another gap in trust-related research is not considering context dependency. The simultaneous use of the features of time, trust, item content and users' rating matrix in social networks can lead to finding users and items with a strong connection to the target user, and this will reduce system error in providing recommendations to users. In order to address these gaps, this research presents a dynamic and robust recommender system model. Also, in this table, for a better comparison, the general features of the proposed method are also mentioned at the end of the table.

4. Proposed method

In this section, the details of the proposed method are reviewed. As shown in Fig. 4, the proposed method consists of four stages: injecting shilling attacks, creating social networks of users, communities detecting and recommending users. At the first stage, fake profiles are injected into the system to simulate a shilling attack. At the second stage, the social networks of users and items are formed based on the use of rating time information, the matrix of users' rating of items, trust between users and the context of items. At the third stage, communities in social networks, users and items are detected by managing the incremental data-flow over time. Finally, at the fourth stage, by detecting and ignoring fake profiles in the neighborhood of users, the @N top item is

Table 1
General features of related works and the proposed method.

Authors and year	Reference	Approach						DataSet
		Trust	Time	Graph	Context	Dynamics of rating and relations	Robust against attacks	
Ahmadian et al (2022)	(Ahmadian et al., 2022)	✓						Epinions Flixster
Ahmadian et al (2020)	(Ahmadian et al., 2020)	✓						FilmTrust Flixster Epinions
Wahab et al (2022)	(Wahab et al., 2022)	✓						MovieLens 1 M Epinions
Birtolo & Ronca (2013)	(Birtolo and Ronca, 2013)	✓						Epinions MovieLens Jester Poste Italiane datasets
Koren et al (2021)	(Koren et al., 2021)		✓					Tmall CIKM
Rezaimehr et al (2018)	(Rezaeimehr et al., 2018)		✓					MovieLens-100 K MovieLens-1 M
Moradi et al (2016)	(Moradi et al., 2016)		✓					Movielense-1G
Daneshmand et al (2015)	(Daneshmand et al., 2015)		✓					Netflix Movielens
Jiang et al (2022)	(Jiang et al., 2022)			✓				Twitter
Al-Gorabi et al (2021)	(Al-Ghobari et al., 2021)			✓				Google locations
Feng et al (2015)	(Feng et al., 2015		✓	✓				MovieLens Netflix
Rezaimehr & Dadkhah (2023)	(Rezaimehr and Dadkhah, 2023)	✓	✓	✓			✓	Epinion
Proposed method	–	✓	✓	✓	✓	✓	✓	Epinion

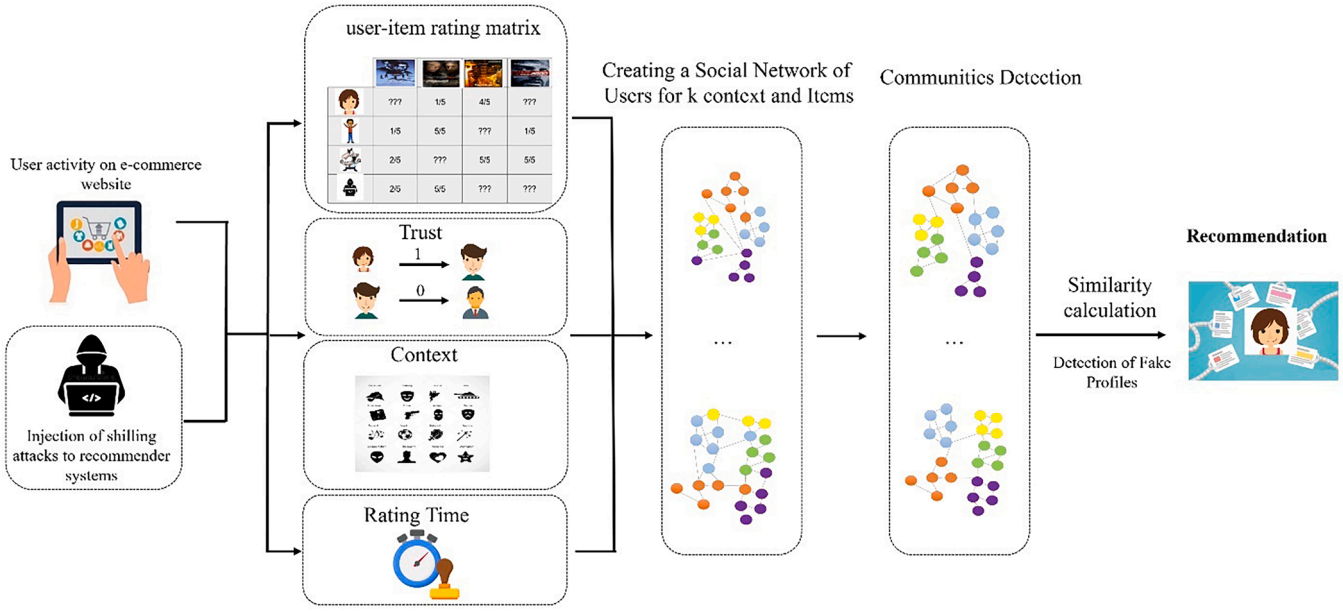

Fig. 4. Proposed method. Each box with dashed dots represents a part of the proposed method. The proposed method includes injection of shilling attack, creating social networks of users and items) using user rating matrix, trust, context and rating time(, community detection and finally providing recommendations to the target user.

recommended to the target user. Table 2 describes the action and objectives for the proposed method.

In the following, each stage of the proposed method is reviewed in detail.

4.1. Injection of shilling attacks

In order to simulate a shilling attack in the recommender system, it is necessary to inject fake profiles into the system. Given that no dataset containing shilling attacks is available, mistakes in injecting attacks into the system can interfere with the correct evaluation of the proposed method. For this reason, this stage is very important and the evaluation of the proposed method depends on the correct implementation of this stage.

At this stage, the dataset is attacked using the average attack, random attack and bandwagon attack whose model is explained in Fig. 3. Finally, a collection of fake profiles and normal profiles are mixed and used as input for the next stage.

Table 2
Actions and objectives of the proposed method.

stage		action	objective
Injecting shilling attacks		Injecting fake profiles to the recommender system by model of fake profiles and types of shilling attacks	simulating shilling attacks and creating dataset including normal and fake profiles to do the experiments
Creating social networks of users and items	Using rating time data	Calculating Aggregation Index of User Rating Timestamp and Relative Aggregation Index of User Rating Timestamp	Adding suspected to be a fake profile and normal profile tags to users
	Using user rating matrix data	Calculating the similarity between users using the Pearson correlation	Creating relationships between users and items
	Using trust of users	Calculating the trust between users	Improving the calculation of the similarity of users
	Using item context	Calculating the weight of each edge in k context	1. Considering context dependence for the similarity and trust 2. Reducing the effects of shilling attacks
Communities detecting		Incremental detecting communities	Managing incremental data-flow
Recommendation to users		Calculating of similarity of users in each community and calculating of item similarity	Recommending @N top items to the target user

4.2. Creation of social networks of users and items

Social network is actually a social structure by which a number of social actors are connected to each other with binary social relations. Contrary to public opinion, social networks are not necessarily virtual social networks such as Facebook, Twitter or Instagram. In social networks, social actors can be of different genders other than humans. For example, the ants of a colony form a social network in which the ants are social actors and the relationships governing the colony's social relationships or a group of computers in a network can also form a social network.

Studies have shown that there are simple rules and models that seem very complex in the structure of social networks. The rules and models can enable us to explore knowledge about the network. The emergence of graph theory was a turning point for social networks. Because the graph can model social networks in the form of vertices (for social actors) and edges (for social relationships) (Ajzen, 1991). At this stage, a social network is created for users and items. In the following, the process of creating a social network of users and items is discussed.

4.2.1. Users rating time

Given that the deployment cost is important for shilling attacks and it is done with the intention of gaining illegal benefits during a limited period of time, fake profiles rate items during a short period of time. accordingly, the time of rating items by fake profiles is significantly less compared to normal profiles and follows certain models (Zhang et al., 2006; Oestreicher-Singer and Sundararajan, 2012). The users' rating time can largely detect fake profiles and make robust recommender

system against these sentences. In this article, the RAIURT is used. This index is calculated as follows (Eqs. (2)-(5)).

- Set of User Rating Timestamp (SURT): This set of contains user rating timestamps for items that are sorted in descending order, the higher the serial number of an element in this set, the closer the rating timestamp of the element is to the current time. In this regard, the value of u is allocated to the specific user and n indicates the items rated by this user.

$$SURT_u = \{t_1, t_2, t_3, \cdots, t_n\} \tag{2}$$

- Maximum Interval of User Rating Timestamp (MIURT): this value is equal to the subtraction of the first and last element of the SURT collection (n refers to the elements in the user's SURT)

$$MIURT_u = SURT_n - SURT_1 \tag{3}$$

- Aggregation Index of User Rating Timestamp (AIURT): this value is equal to the maximum user rating timestamp interval compared to the total number of rated items. (u refers to a specific user).

$$AIURT_u = \frac{MIURT_u}{N_u} \tag{4}$$

- Relative Aggregation Index of User Rating Timestamp (RAIURT): this value is used to describe the relative aggregation of rating timestamp. ($\overline{MIURT}$ refers to the average of MIUTR and $\overline{\mathrm{N}}$ refers to the average of all user-rated items)

$$\mathrm{RAIURT_u} = \frac{|\mathrm{MIURT_u} - \overline{\mathrm{MIURT}}|}{|\mathrm{N_u} - \overline{\mathrm{N}}|} \tag{5}$$

If the $\mathrm{AIURT_u}$ is lower than a threshold limit t that is determined empirically and with the help of $\mathrm{RAIURT_u}$, then that user receives a tag suspected to be a fake profile and other profiles receive a normal profile tag. It should be noted that in the process of calculating the similarity, users with a tag suspected to be a fake profile will not be used for users with a normal profile tag, but users with a normal profile tag will be used to calculate the similarity for users with a tag suspected to be a fake profile. The reason for this is that unlike similar methods, not only the user with a tag suspected to be a fake profile will not be removed from the system (due to the possibility of wrong tagging due to user rating time), but also the recommended recommender system will recommend popular items to these types of profiles.

4.2.2. User rating matrix

After all the profiles have received a tag suspected to be a fake profile or a normal profile, we calculate the similarity between users and items.

At this stage, first, we calculate the similarity between users using the Pearson correlation. The Pearson correlation to calculate the similarity between two users a and b is obtained from Eq. (6):

$$sim(a,b) = \frac{\sum_{p\in P}(r_{a,p} - \overline{r}_a)(r_{b,p} - \overline{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p} - \overline{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p} - \overline{r}_b)^2}} \tag{6}$$

Where $r_{a,p}$ refers to the rating of user a to item p. Also, P refers to the collection of items rated by both users. $sim(a,b)$ can be between 1 (the highest similarity) and -1 (the lowest similarity). After calculating the similarity, we map the obtained value between 0 and 1.

4.2.3. Trust

Trust between users can be received as explicit and implicit. There are various methods for calculating implicit trust from users' ratings (Yang and Niu, 2023). In some datasets, there are explicit trust relationships in addition to user ratings. In this research, explicit trust relationship is also used. According to Eq. (7), explicit trust relationships are usually defined as follows:

$$Trust_{u,v} = \begin{cases} 1, trust \\ 0, distrust \end{cases} \tag{7}$$

4.2.4. Item context

As mentioned earlier, one of the features of trust is context dependence. Another feature of trust is transitivity. Based on transitivity, in the event that user u_1 trusts in user u_2 and user u_2 trusts in user u_3, then user u_1 trusts in user u_3. If we don't consider the context in our transportability, we may mistakenly consider user u_1 and user u_3 as the same. For example, suppose that user u_1 and user u_2 have the same taste in both game and book, and user u_2 and user u_3 have the same taste in music, so it cannot be concluded that user u_1 and user u_3 share the same taste in music. In this study, we also consider the context of each item as an input parameter. Including the context of the items can also reduce the effects of the shilling attack to some extent. In this study, the symbol K also indicates the number of different contexts of the items.

In the process of creating social networks of users, after calculating the similarity between users, we draw a weighted edge between both user u_1 and user u_2. The weight of each edge in k context is calculated by Eq. (8):

$$\mathrm{WeightedEdge}(u_1, u_2, k) = \frac{P_k}{P} * (Trust_{u1,u2} + sim(u1, u2)) \tag{8}$$

In this equation, P represents the set of items and P_k indicates the number of items which is rated by both users in context k.

Given that the weight of an edge between two users can have k different cases, in this study we draw k different social networks between users. Fig. 5 shows the difference between the edges of the same multiple users in social networks of different contexts.

When social networks of users are created in different contexts, a social network is also created between the items. The condition for establishing an edge between two items is that they have been rated by a user of the normal profile tag type. The weight of this edge is calculated by Eq. (9):

$$\mathrm{WeightedEdge}(i_1, i_2) = \frac{\sum_{u\in U}(1 - \frac{abs(r_u(i) - r_u(j))}{M})}{U1} \tag{9}$$

In this regard, the value of M is allocated to the maximum amount of points by a user.

4.3. Communities detection

The method (Zhao et al., 2019) has been used for community detection. This study proposed an incremental method for detecting communities by managing subgraphs. The idea of this method is to detect the communities in the initial case. After this stage, incremental dynamic changes are collected and analyzed, and then the communities are gradually updated. In this method, there is no need to determine the number of communities in advance. Fig. 6 shows the flowchart of the proposed method for detecting communities in dynamic networks.

At the beginning and at the moment t_0, the communities in social networks are detected. Over time and the introduction of new information into the network, incremental elements should be collected and analyzed in the network. Incremental update strategies are then planned based on the communities in the latest case.

Let's assume that during the time between moment t and t + 1 incremental elements ΔG^{t+1} have been added to the network in the form of different subgraphs. If subG is a subgraph of ΔG^{t+1}, V(subG) define the vertices in subG and E(subG) define the collection of edges in subG. As shown in Fig. 7, the relationship of each subG with the previously existing communities at time t, denoted by the symbol CS^t will be displayed in the following 4 cases:

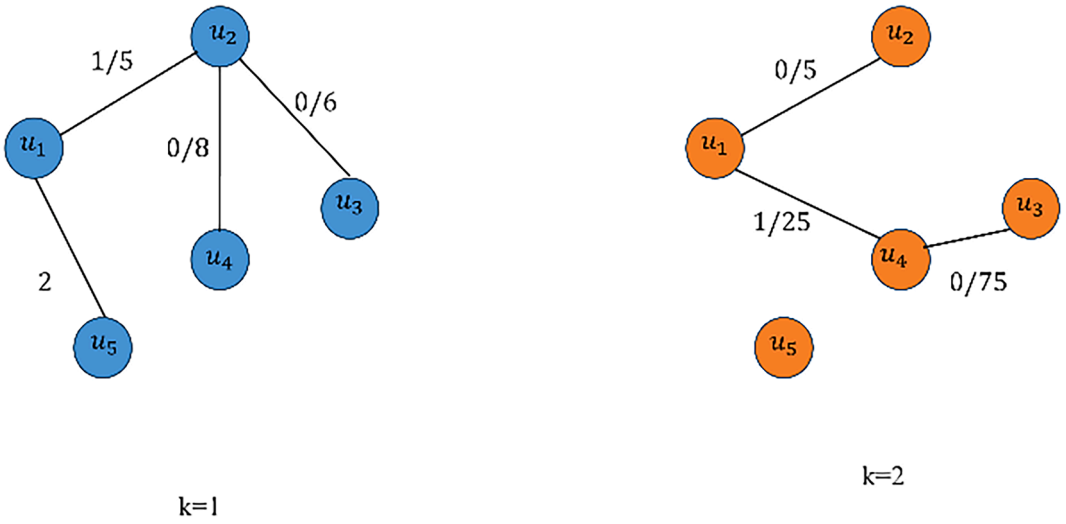

Fig. 5. Difference between the same users in social networks of different contexts. This figure shows that users in the social network of different contexts have different connections.

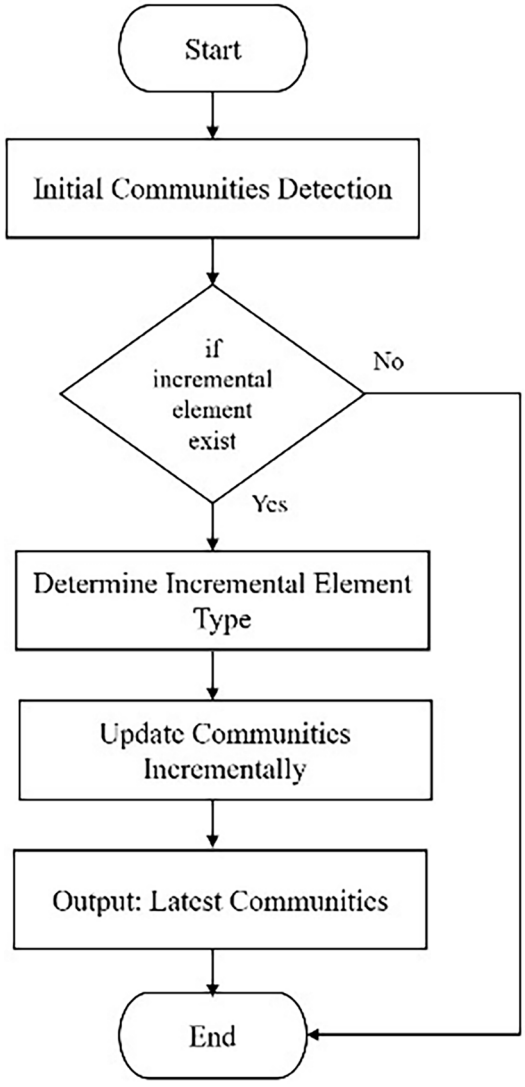

Fig. 6. Flowchart of the proposed method for communities detecting in dynamic networks. This figure shows how to communities detection as well as dynamically update communities with incremental elements.

A. Complete Independent: If $V(subG) \cap V(CS_k^t) = \varnothing$ (if none of the vertices of the subgraph is in a community of CS^t), then that subgraph is complete independent.
B. Complete Contained: If $\forall v \in V(subG) \rightarrow v \in V(CS_k^t)$ (if all vertices of the subgraph have already appeared in a certain community of CS^t), then the subgraph is complete contained.
C. Mixed with New and Old Vertices: If a number of vertices in subG belong to one or more communities of CS^t and other vertices are new, then the subgraph is mixed.

d) Multiple Contained: If in subG all its vertices belong to several (more than one) communities of CS^t, then the subgraph is multiple contained.

After determining the case of incremental elements, each element updates communities with its own strategy. The strategy of the above four incremental elements is as follows:

A. Complete Independent: in this strategy, a new community is created in which subG is considered a new community at time (t + 1).
B. Complete Contained: In this strategy, the weights of the edges are updated first, and then some edges are removed. The objective of this strategy is the local modularity of the current community.
C. Mixed with New and Old Vertices: In this strategy, by calculating the subordinating strength, it is found that the new vertices will form a new community or be added to other communities.
D. Multiple Contained: In this strategy, by calculating the subordinating strength, it is found that the vertices belonging to each community will be added to their community or create a new community with other vertices.

Finally, the output of this method after completing the incremental data-flow, the last communities have been determined.

4.4. Recommendation to users

4.4.1. Calculation of similarity of users in each community

After creating social networks for each context and detecting the community, at this stage, similar k top users should be found for the target user. In social networks, each context of target user vertices have two cases compared to other vertices.

A. Similarity calculation for vertices that are directly connected with an edge: according to Eq. (10), the similarity between two users is equal to the weight of the edge.

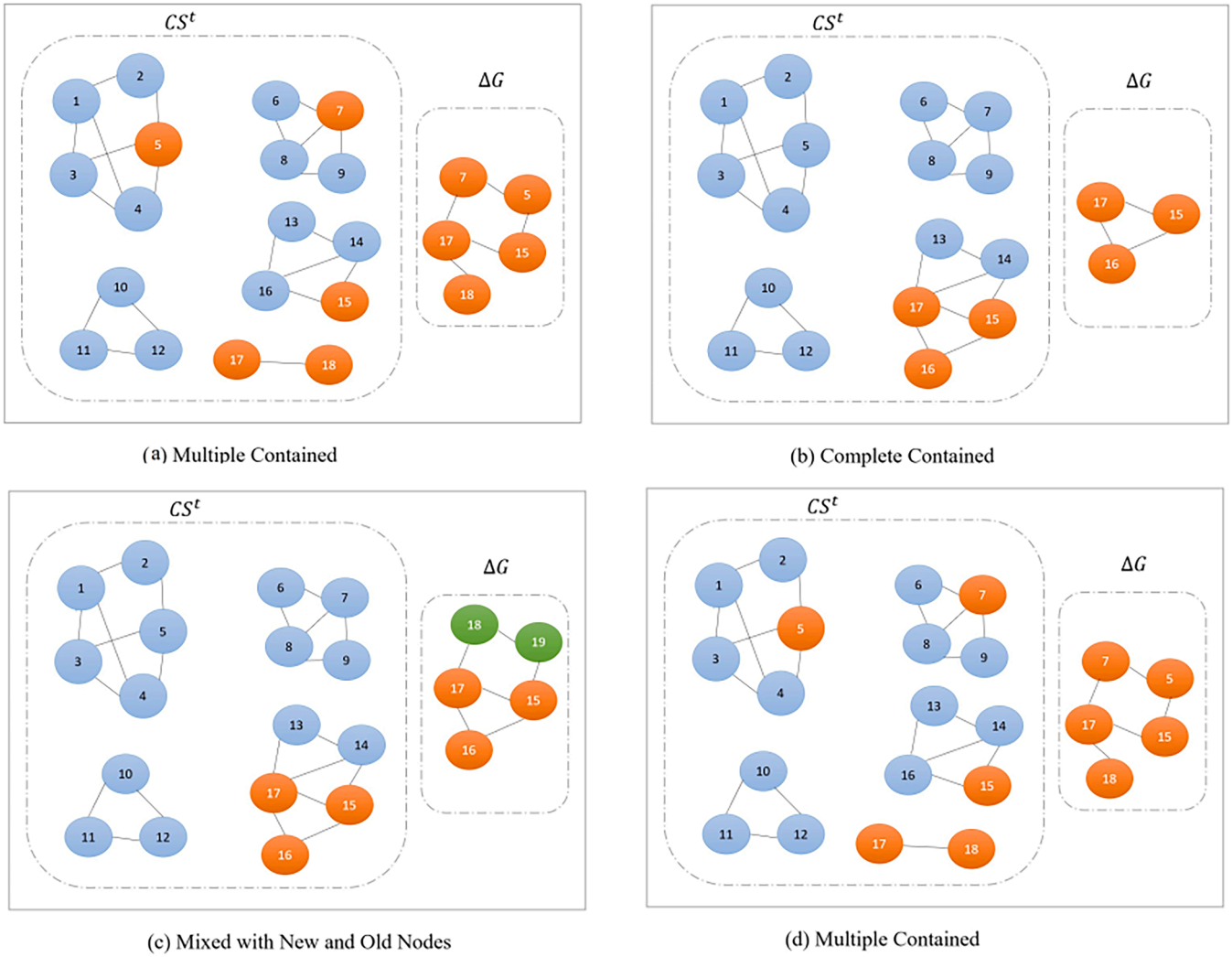

Fig. 7. Four types of incremental elements. This figure shows dealing with incremental elements that have been added to the network in the form of different subgraphs in the time period between moment t and t + 1.

$$\text{Sim\_du}(u_1, u_2, k) = \text{WeightedEdge}(u_1, u_2, k) \tag{10}$$

B. Calculation of similarity of vertices that are not directly connected: according to Eqs. (11) and (12), the parameters of the index of degree centrality and the length of the shortest path (Jalali and Hosseini, 2021) are used. In social networks, the index of degree centrality shows the degree of importance and influence of a user. The larger the index of degree centrality for a user is, the more people influence this user. The calculation of this index is that the degree of each node k_i is divided by N-1 (N define number of vertices in the network) (Kong et al., 2019).

$$C_D(i) = \frac{k_i}{N-1} \tag{11}$$

$$Sim\_\text{du}(u_1, u_2, k) = C_D(u_2)\text{*reverse shortest path}(u_1, u_2) \tag{12}$$

After calculating the similarity and selecting k top users, the items selected by the target user and k top users are placed into a candidate list.

4.4.2. Calculation of item similarity

To improve the recommendations process, the items that are inside the communities of these items are extracted through social networks of items and placed in the candidate list. There are two cases for calculating the similarity of items:

A. Calculation of similarity for vertices that are directly connected with an edge: according to Eq. (13), in this case, the similarity between two items is equal to the weight of the edge.

$$\text{sim\_di}(i_1, i_2) = \text{WeightedEdge}(i_1, i_2) \tag{13}$$

B. Calculation of similarity for vertices that are not directly connected: according to Eq. (14), in this case, we use the parameters of the index of degree centrality and the length of the shortest path.

$$Sim\_\text{di}(i_1, i_2) = C_D(i_2)\text{*reverse shortest path}(i_1, i_2) \tag{14}$$

4.4.3. Item rating prediction

After extracting the list of candidate items, the rating of the target user is calculated for each of the items in the candidate list. The user-based and item-based rating prediction is according to the following two cases:

A. User-based case is obtained from Eq. (15):

$$\text{pred}(a, p) = \overline{r_a} + \frac{\sum_{b \in \text{community(a)}} \text{sim\_du}_{(a,b,k)} \times \left(r_{b,p} - \overline{r_b}\right)}{\sum_{b \in \text{community(a)}} \text{sim\_du}_{(a,b,k)}} \tag{15}$$

B. Item-based case is obtained from Eq. (16):

$$\mathrm{pred}(a,\mathrm{p}) = \overline{\mathrm{r}_{\mathrm{u}}} + \frac{\sum_{\mathrm{i}\in \mathrm{community}(p)} \mathrm{Sim\_dg}(\mathrm{i},\mathrm{p}) \times \left(\mathrm{r}_{\mathrm{u,i}}\right)}{\sum_{\mathrm{i}\in \mathrm{community}(\mathrm{p})} \mathrm{Sim\_dg}(\mathrm{i},\mathrm{p})} \tag{16}$$

The final rating for the items is predicted by the target user as the average of case A and B. Finally, after rating all items, the @N top item is recommended to the user.

5. Empirical evaluation

5.1. Datasets

In this implementation, the Epinions dataset has been used to perform tests. The Epinions dataset is a real dataset made publicly available. As shown in Fig. 8, this dataset includes 922,267 rating records by 22,166 users for 296,277 items and explicit trust/distrust relationships between users. This dataset includes a rating file that contains users' rating of items and a trust file that shows trust relationships between users. Also, in this dataset, the contexts of the items are divided into 27 groups. To unify the dataset to evaluate the proposed method with the compared methods, the data of 943 users and 1682 items were used (Rezaimehr and Dadkhah, 2023).

5.2. Empirical design

To evaluate the recommender system, 50 tests (5 tests of @N Top with 10 replications and recording the average) were performed. To evaluate its robustness against shilling attacks, 4800 different tests were performed in the form of two attack intents (push & nuke), three types of attacks (random, average and bandwagon), four attack sizes, four profile sizes with 10 replications and recording the average. Table 3 shows the data of empirical test design for robustness against shilling attacks.

5.3. Evaluation criteria

In this study, five evaluation criteria of recommender system algorithms have been used to compare the proposed method. In the algorithms of recommender systems, @N Top items are recommended to the target user. According to Table 4 the number of items predicted by the algorithm that are liked by the user is defined as True Positive (TP) and the number of items predicted by the algorithm that is not liked by the user is defined as False Positive (FP). Also, the number of items not predicted by the algorithm that is liked by the user is defined as False Negative (FN) and the number of items not predicted by the algorithm that is not liked by the user is defined as True Negative (TN). By

	userid	productid	categoryid	rating	helpfulness	timestamp
0	1	1	3	2	2	973234800
1	1	2	2	2	3	972025200
2	1	3	3	2	1	971593200
3	1	4	3	5	2	966668400
4	1	5	3	3	2	963471600
...	...	...	...	...	...	...
922262	22166	83922	5	4	2	1144479600
922263	22166	23442	5	4	2	1144479600
922264	22166	43538	5	5	2	1144393200
922265	22166	38711	3	4	3	1144220400
922266	22166	41790	5	3	1	1144220400

922267 rows × 6 columns

Fig. 8. Rating information for Epinions dataset. there are five columns and they are userid, productid, categoryid, rating, helpfulness, timestamp.

Table 3
Data of empirical test design for the robustness against shilling attacks.

Type	Options	Quntity
Target Item	Push & Nuke	2
@N top Item	{5,10,15,20,30}	5
Attack Type	Random, Average & Bandwagon	3
Attack Size	{20,25, 30, 35}	4
Profile Size	{20,25, 30, 35}	4
Repeat experiment	10	10
Total experiments		4800

Table 4
Confusion matrix visualization.

	Liked by the user	Not liked by the user
Predicted by the algorithm	TP	FP
Not predicted by the algorithm	FN	TN

obtaining these four values of the evaluation criteria used in this study, the precision, recall and F1 are calculated according to the Eqs. (17)-(19) as follows:

$$\mathrm{precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP}{TP+FP} \tag{17}$$

$$\mathrm{recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP}{TP+FN} \tag{18}$$

$$F1 = \frac{2{*}precision{*}recall}{precision + recall} \tag{19}$$

According to Eq. (20), MAE, which is equal to the mean absolute error and measures the closeness of the estimated predictions with the observed cases and can be used to deviate from the actual values, is calculated as follows:

$$MAE = \frac{\sum_{i=1}^{N}|r_i - p_i|}{N} \tag{20}$$

Where N is the total number of points predicted for the items and p_i and r_i are the values of the actual and predicted points for item i. Item i is actually the target item of shilling attacks.

Also, according to Eq. (21), RMSE is equal to the root mean square error and can be used to measure the precision of collaborative filtering algorithms and is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - p_i)^2} \tag{21}$$

5.4. Compared methods

The proposed method was compared with TRACCF (Birtolo and Ronca, 2013), (Feng et al., 2015) and T&TRS (Rezaimehr and Dadkhah, 2023) methods for the performance. The reason for selecting the TRACCF and TOTAR methods is to use the trust and graph and select the T&TRS method to use the time, trust and graph, which in addition to being close to the proposed method is the most recent research in the field of robust algorithms.

5.5. Evaluation results

To evaluate the robustness of the proposed method against shilling attacks, as shown in Table 5, the proposed method is compared in the case without injection of shilling attack and under 3 mean, random and bandwagon attacks in Precision, Recall, F1, MAE and RMSE. The evaluation results in the case without shilling attack showed that the

Table 5
Results of evaluating the robustness of the proposed method against shilling attacks.

Algorithm	Target item	Evaluation Criteria	Top5	Top10	Top15	Top20	Top30	MAE	RMSE
Proposed method		Precision	0.8917	0.9011	0.8814	0.8872	0.8814	0.7852	1.112
(Without attack)		Recall	0.7914	0.8216	0.8445	0.8583	0.8590		
		F1	0.8384	0.8594	0.8625	0.8724	0.8700		
Proposed method	Push	Precision	0.7537	0.7841	0.8192	0.8519	0.8748	2.195	3.2818
(With average attack)		Recall	0.7413	0.7015	0.6922	0.6887	0.6741		
		F1	0.7474	0.7404	0.7502	0.7616	0.7614		
	Nuke	Precision	0.8615	0.8354	0.8110	0.7963	0.7744	2.044	3.0074
		Recall	0.6835	0.7197	0.7218	0.7871	0.8119		
		F1	0.7622	0.7731	0.7637	0.7915	0.7926		
Proposed method	Push	Precision	0.8366	0.7425	0.8009	0.7177	0.7345	2.432	3.3819
(With random attack)		Recall	0.6833	0.6894	0.7055	0.7836	0.8254		
		F1	0.7521	0.7148	0.7501	0.7485	0.7772		
	Nuke	Precision	0.8441	0.7152	0.6926	0.7264	0.6915	2.255	3.0196
		Recall	0.7857	0.6992	0.7045	0.8133	0.8172		
		F1	0.8138	0.7070	0.6984	0.7672	0.7489		
Proposed method	Push	Precision	0.8363	0.6937	0.6815	0.7029	0.6840	2.8321	3.5754
(With bandwagon/reverse bandwagon attack)		Recall	0.7644	0.7619	0.7542	0.7480	0.7432		
		F1	0.7986	0.7261	0.7158	0.7246	0.7123		
	Nuke	Precision	0.8412	0.8073	0.7922	0.7864	0.7614	2.469	3.2937
		Recall	0.7219	0.7230	0.7333	0.7984	0.8422		
		F1	0.7769	0.7627	0.7615	0.7922	0.7997		

proposed method was at an acceptable level for 5, 10, 15, 20 and 30 top items in the Precision, Recall and F1.

Also, the results showed that the proposed method has suffered a slight reduction in the Precision, Recall and F1 after creating a shilling attack and injecting fake profiles, but the values are still at an acceptable level. F1 indicates the comprehensive effect of a model. So, based on the tests, it can be concluded that the proposed method is robust against average, random, and bandwagon attacks.

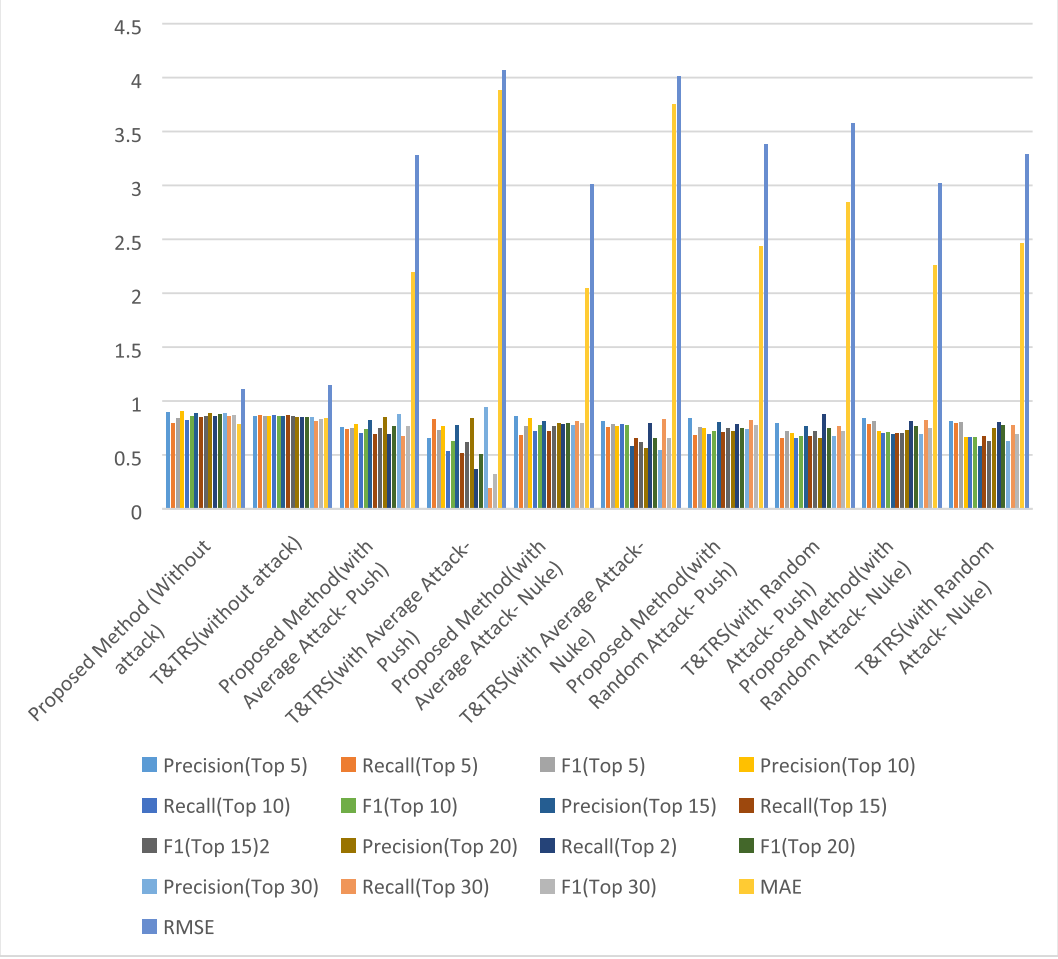

Fig. 9. Comparison of the proposed method with the T&TRS method. This figure shows the proposed method with the T&TRS method in the evaluation criteria of Precision, Recall, F1, MAE, and RMSE in the cases of without attack, with average attack-push, with average attack-nuke, with random attack-push and random attack-nuke.

Also, as shown in Fig. 9, the proposed method is compared with the T&TRS method (which is the most recent research in the field of resistant algorithms) in terms of robustness. The evaluation results showed that the proposed method under the shilling attack has an acceptable performance in the Precision and Recall and a better performance than the T&TRS method in F1, MAE and RMSE. According to the data and their interpretation, it can be concluded that the use of item context, trust, rating matrix and user rating time can reduce the effect of shilling attack to a great extent.

This section shows the evaluation results of the proposed method with the compared methods for the performance of the proposed method. The proposed method was compared with other methods by the evaluation criteria of Precision, Recall and F1 for 5, 10, 15, 20 and 30 top items without injecting the shilling attack, respectively. As shown in Table 6, the proposed method has an acceptable performance in the two criteria of Precision and Recall, and in the F1, which is actually the result of the two criteria of Precision and Recall and shows the comprehensive effect of a model, the proposed method has a better performance than other methods. In general, by increasing the number of items recommended to the user, F1 increases. Fig. 10 shows the performance of the proposed method for MAE and RMSE. The results show the reduction in errors in these two criteria in the proposed method compared to the TRACCF and T&TRS methods.

The time complexity for creating social networks of users for each context is equal to V^2. Where V represents the number of vertices (users). Also, community detection in the network is the maximum case equal to V^2 and the accumulation of overlapping communities is equal to C^2. Where C represents the number of communities. Since in reality the values of K (the number of context items) and C (the number of communities) are small compared to the number of users (K, C $<<$ N). Therefore, the time complexity of the proposed algorithm is equal to O (n^2) in the maximum case. Where n represents the number of users. Table 7 shows the difference between the proposed method in terms of time complexity and the compared methods.

5.6. Discussion

The results of the experiments of this research show that tagging profiles using two indices Aggregation Index of User Rating Timestamp and Relative Aggregation Index of User Rating Timestamp improves the detection of fake profiles and normal profiles because the shilling attacks investigated in this research by the adversary in the interval a limited time is injected into the system. However, if the adversary wants to inject shilling attacks into the system in a long period of time (for example, about a year), it can lead to a decrease in the accuracy of the proposed recommender system. Although in practice, this greatly increases the cost of the attack for the adversary.

Experiments show that the simultaneous use of the user rating matrix, trust and context to calculate the similarity between users and items and using its results in creating social networks of users and items has been able to reduce the number of predicted items that are not of interest to the user (FP) as well as the number of unexpected items that the user is interested in (FN) and ultimately improve the F1 criterion (which is a combination of Precision and Recall criteria) compared to the comparison methods.

Also, using the interests of similar users in communities detected in social networks of users and similar items in the social network of items due to finding users and items with a strong connection to the target user and reduces the system error in the two criteria MAE and RMSE compared to the comparison methods.

The ability to update communities with incremental elements dynamically and the results of the above indicate that the proposed method has the ability to suggest more suitable items to users with less error in dynamic environments.

Table 6
Comparison of the evaluation results of the proposed method with the compared methods.

Algorithm	Evaluation Criteria	Top5	Top10	Top15	Top20	Top30
TRACCF	Precision	**0.9462**	0.8981	**0.8981**	**0.8981**	0.5319
	Recall	0.5239	0.5399	0.5399	0.539	0.5481
	F1	0.6744	0.6744	0.6744	0.6744	0.5399
TOTAR	Precision	0.8550	0.8521	0.8416	0.8281	0.8237
	Recall	0.4356	0.5467	0.7322	0.7985	0.8134
	F1	0.5636	0.6660	0.7830	0.8130	0.81185
T&TRS	Precision	0.8502	0.8527	0.8543	0.8541	0.8543
	Recall	**0.8152**	**0.8509**	**0.8635**	**0.8657**	**0.8697**
	F1	0.8323	0.8518	0.8588	0.8599	0.8610
Proposed method	Precision	0.8917	**0.9011**	0.8814	0.8872	**0.8814**
	Recall	0.7914	0.8216	0.8445	0.8583	0.8590
	F1	**0.8384**	**0.8594**	**0.8625**	**0.8724**	**0.8700**

6. Conclusion

In collaborative filtering recommender systems type, the inability to model the dynamic behavior of users and the incremental data-flow, as well as the vulnerability to shilling attacks, are the challenges of these systems. In order to fill the gaps, this paper has presented a dynamic and robust recommender system against shilling attacks based on item context, trust, rating matrix and rating time using social network analysis.

First, the proposed recommender system separated the profiles in the system with a tag suspected to be a fake profile and a normal profile using the user rating time. Then, using the rating and trust matrix, social networks of users were created for each context and social network of items. The creation of social networks and communities detection in this article were dynamically and can manage the increasing data-flow over time. After creating social networks of users and items, we extracted the best items by the features of social networks, and simultaneously using the user-based and the item-based cases, we predict the target user's rating for the items. Finally, we offer @N top items to the user.

We performed 4850 different tests to evaluate the performance and robustness of the recommender system against shilling attacks and compared it with close methods such as TRACCF, TOTAR and T&TRS according to the evaluation criteria of Precision, Recall, F1, MAE and RMSE. The results of the empirical tests showed that the proposed method performed better than the compared methods in several evaluation criteria and in the comprehensive effect of the model, indicating the efficiency of the proposed method. It can be concluded that the use of item context, trust, rating matrix and user rating time can reduce the effect of shilling attack to a great extent. In future research, the use of demographic information as well as the friendship relationship in social networks is considered because by this information, the strength of recommender systems can be increased to some extent and the negative impacts of shilling attacks can be reduced.

CRediT authorship contribution statement

Hojatollah(Hodjat) Hamidi: Social Network Marketing, E-Health Cloud, Mobile health, Computation Intelligence for Energy Internet, Data Mining and Data Intelligence in Security and Privacy, Deep Learning. **Rasul Moradi:** Social Networks, Detecting Shilling Attacks, Recommender System, Neural Network.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

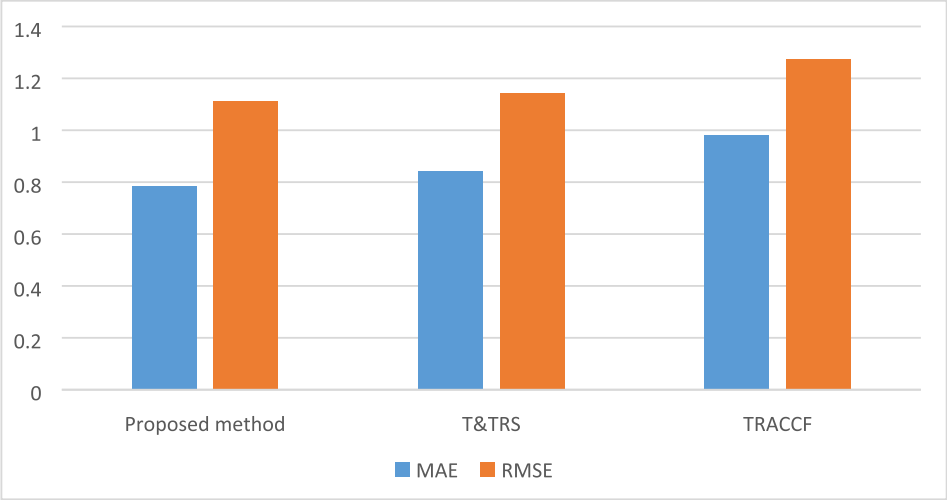

Fig. 10. Comparison of MAE and RMSE. This figure shows the two error measures of MAE and RMSE between the proposed method and the T&TRS and TRACFF methods. The proposed method has less error than other methods in these two measures.

Table 7
Comparison of the time complexity of the proposed method with the compared methods.

Algorithm	Time complexity
TRACCF	O(nm)
TOTAR	O(nm)
T&TRS	$O(n^2m)$
Proposed method	$O(n^2)$

the work reported in this paper.

References

Ahmadian, M., Ahmadi, M., Ahmadian, S., 2022. A reliable deep representation learning to improve trust-aware recommendation systems. Expert Syst. Appl. 197, 116697.

Ahmadian, S., Joorabloo, N., Jalili, M., Ren, Y., Meghdadi, M., Afsharchi, M., 2020. A social recommender system based on reliable implicit relationships. Knowl.-Based Syst. 192, 105371.

Ajzen, I., 1991. The theory of planned behavior. Organ. Behav. Hum. Decis. Process. 50 (2), 179–211.

Al-Ghobari, M., Muneer, A., Fati, S.M., 2021. Location-Aware Personalized Traveler Recommender System (LAPTA) Using Collaborative Filtering KNN. Computers, Materials & Continua 69 (2).

Alonso, S., Bobadilla, J., Ortega, F., Moya, R., 2019. Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. IEEE Access 7, 41782–41798.

Bhaumik, R., Williams, C., Mobasher, B., & Burke, R. (2006, July). Securing collaborative filtering against malicious attacks through anomaly detection. In Proceedings of the 4th workshop on intelligent techniques for web personalization (ITWP'06), Boston (Vol. 6, p. 10).

Bilge, A., Gunes, I., Polat, H., 2014. Robustness analysis of privacy-preserving model-based recommendation schemes. Expert Syst. Appl. 41 (8), 3671–3681.

Birtolo, C., Ronca, D., 2013. Advances in clustering collaborative filtering by means of fuzzy C-means and trust. Expert Syst. Appl. 40 (17), 6997–7009.

Bo, Z., Huan, Z., Meizi, L., Qin, Z., Jifeng, H., 2017. Trust traversal: a trust link detection scheme in social network. Comput. Netw. 120, 105–125.

Burke, R., O'Mahony, M.P., Hurley, N.J., 2015. Robust collaborative recommendation. Recommender Systems Handbook 961–995.

Cerutti, F., Kaplan, L.M., Norman, T.J., Oren, N., Toniolo, A., 2015. Subjective logic operators in trust assessment: an empirical study. Inf. Syst. Front. 17, 743–762.

Cheng, Z., Hurley, N., 2010. Robust collaborative recommendation by least trimmed squares matrix factorization. In: In 2010 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 2. IEEE, pp. 105–112.

Daneshmand, S.M., Javari, A., Abtahi, S.E., Jalili, M., 2015. A time-aware recommender system based on dependency network of items. Comput. J. 58 (9), 1955–1966.

Dellarocas, C. (2000, October). Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In Proceedings of the 2nd ACM Conference on Electronic Commerce (pp. 150-157).

Feng, H., Tian, J., Wang, H.J., Li, M., 2015. Personalized recommendations based on time-weighted overlapping community detection. Inf. Manag. 52 (7), 789–800.

Gao, M., Ling, B., Yuan, Q., Xiong, Q., & Yang, L. (2014). A robust collaborative filtering approach based on user relationships for recommendation systems. Mathematical Problems in Engineering, 2014.

Jalali, S., Hosseini, M., 2021. Social collaborative filtering using local dynamic overlapping community detection. J. Supercomput. 77, 11786–11806.

Jia, C.X., Liu, R.R., 2015. Improve the algorithmic performance of collaborative filtering by using the interevent time distribution of human behaviors. Physica A 436, 236–245.

Jia, D., Zhang, F., Liu, S., 2013. A Robust Collaborative Filtering Recommendation Algorithm Based on Multidimensional Trust Model. J. Softw. 8 (1), 11–18.

Jiang, L., Shi, L., Liu, L., Yao, J., Ali, M.E., 2022. User interest community detection on social media using collaborative filtering. Wirel. Netw 1–7.

Kong, X., Shi, Y., Yu, S., Liu, J., Xia, F., 2019. Academic social networks: Modeling, analysis, mining and applications. J. Netw. Comput. Appl. 132, 86–103.

Koren, Y., Rendle, S., Bell, R., 2021. Advances in collaborative filtering. Recommender Systems Handbook 91–142.

Kumar, R., Bhanodai, G., Pamula, R., 2019. Book search using social information, user profiles and query expansion with pseudo relevance feedback. Appl. Intell. 49, 2178–2200.

Lathia, N., Hailes, S., Capra, L., 2008. Trust-Based Collaborative Filtering. In: Karabulut, Y., Mitchell, J., Herrmann, P., Jensen, C.D. (Eds.), Trust Management II. IFIPTM 2008. IFIP – the International Federation for Information Processing, vol 263. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09428-1_8.

Mehta, B., & Nejdl, W. (2008, July). Attack resistant collaborative filtering. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 75-82).

Mehta, B., Hofmann, T., & Nejdl, W. (2007, October). Robust collaborative filtering. In Proceedings of the 2007 ACM conference on Recommender systems (pp. 49-56).

Mobasher, B., Burke, R., Bhaumik, R., Williams, C., 2007a. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology (TOIT) 7 (4).

Mobasher, B., Burke, R., Bhaumik, R., Sandvig, J.J., 2007b. Attacks and remedies in collaborative recommendation. IEEE Intell. Syst. 22 (3), 56–63.

Mobasher, B., Burke, R., Williams, C., & Bhaumik, R. (2006a). Analysis and detection of segment-focused attacks against collaborative recommendation. In Advances in Web Mining and Web Usage Analysis: 7th International Workshop on Knowledge Discovery on the Web, WebKDD 2005, Chicago, IL, USA, August 21, 2005. Revised Papers 7 (pp. 96-118). Springer Berlin Heidelberg.

Mobasher, B., Burke, R., & Sandvig, J. J. (2006b, July). Model-based collaborative filtering as a defense against profile injection attacks. In AAAI (Vol. 6, p. 1388).

Moradi, R., Hamidi, H., 2023. A New Mechanism for Detecting Shilling Attacks in Recommender Systems Based on Social Network Analysis and Gaussian Rough Neural Network with Emotional Learning. Int. J. Eng. 36 (2), 321–334.

Moradi, P., Rezaimehr, F., Ahmadian, S., Jalili, M., 2016. In: September). A Trust-Aware Recommender Algorithm Based on Users Overlapping Community Structure. IEEE, pp. 162–167.

O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. (2002). Promoting recommendations: An attack on collaborative filtering. In Database and Expert Systems Applications: 13th International Conference, DEXA 2002 Aix-en-Provence, France, September 2–6, 2002 Proceedings 13 (pp. 494-503). Springer Berlin Heidelberg.

O'DONOVAN, J. O. H. N., & Smyth, B. (2006). Mining trust values from recommendation errors. International Journal on Artificial Intelligence Tools, 15(06), 945-962.

Oestreicher-Singer, G., Sundararajan, A., 2012. Recommendation networks and the long tail of electronic commerce. MIS Q. 65–83.

O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. (2005, July). Recommender systems: Attack types and strategies. In AAAI (pp. 334-339).

O'Mahony, M., Hurley, N., Kushmerick, N., Silvestre, G., 2004. Collaborative recommendation: A robustness analysis. ACM Transactions on Internet Technology (TOIT) 4 (4), 344–377.
Rezaeimehr, F., Moradi, P., Ahmadian, S., Qader, N.N., Jalili, M., 2018. TCARS: Time- and community-aware recommendation system. Futur. Gener. Comput. Syst. 78, 419–429.
Rezaimehr, F., Dadkhah, C., 2021. A survey of attack detection approaches in collaborative filtering recommender systems. Artif. Intell. Rev. 54, 2011–2066.
Rezaimehr, F., Dadkhah, C., 2023. T&TRS: robust collaborative filtering recommender systems against attacks. Multimed. Tools Appl. 1–31.
Shambour, Q., Lu, J., 2012. A trust-semantic fusion-based recommendation approach for e-business applications. Decis. Support Syst. 54 (1), 768–780.
Si, M., Li, Q., 2020. Shilling attacks against collaborative recommender systems: a review. Artif. Intell. Rev. 53, 291–319.
Turk, A.M., Bilge, A., 2018. In: July). A Robust Multi-Criteria Collaborative Filtering Algorithm. IEEE, pp. 1–6.
Wahab, O.A., Rjoub, G., Bentahar, J., Cohen, R., 2022. Federated against the cold: A trust-based federated learning approach to counter the cold start problem in recommendation systems. Inf. Sci. 601, 189–206.
Yang, L., Niu, X., 2023. A genre trust model for defending shilling attacks in recommender systems. Complex & Intelligent Systems 9 (3), 2929–2942.
Yi, H., Zhang, F., 2016. Robust recommendation method based on suspicious users' measurement and multidimensional trust. J. Intell. Inf. Syst. 46 (2), 349–367.
Yu, H., Gao, R., Wang, K., Zhang, F., 2017. A novel robust recommendation method based on kernel matrix factorization. J. Intell. Fuzzy Syst. 32 (3), 2101–2109.
Zhang, F. G., & Sheng-hua, X. (2007, November). Analysis of trust-based e-commerce recommender systems under recommendation attacks. In The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007) (pp. 385-390). IEEE.
Zhang, S., Chakrabarti, A., Ford, J., Makedon, F., 2006. August). Attack detection in time series for recommender systems. In: In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 809–814.
Zhang, F., Sun, S., 2014. A Robust Collaborative Recommendation Algorithm Based on Least Median Squares Estimator. J. Comput. 9 (2), 308–314.
Zhang, F., Lu, Y., Chen, J., Liu, S., Ling, Z., 2017a. Robust collaborative filtering based on non-negative matrix factorization and R1-norm. Knowl.-Based Syst. 118, 177–190.
Zhang, Z., Xu, G., Zhang, P., Wang, Y., 2017b. Personalized recommendation algorithm for social networks based on comprehensive trust. Appl. Intell. 47 (3), 659–669.
Zhao, Z., Li, C., Zhang, X., Chiclana, F., Viedma, E.H., 2019. An incremental method to detect communities in dynamic evolving social networks. Knowl.-Based Syst. 163, 404–415.