A security detection approach based on autonomy-oriented user sensor in social recommendation network

Shanshan Wan and Ying Liu

Abstract
User social network-based recommender system has achieved significant performance in current recommendation fields. However, the characteristic of openness brings great hidden dangers to the security of recommender systems. Shilling attackers can change the recommendations by foraging user relationships. Most shilling attack detection approaches depend on the explicit user historical data to locate shilling attackers. Some important features such as information propagation and social feedback of users in social networks have not been noticed. We propose a security detection method based on autonomy-oriented user sensor (AOUSD) to identify shilling attackers. Specifically, (1) the user is simulated as a social sensor with autonomous capabilities, (2) the user interaction model is built based on information propagation, information feedback and information disappearance mechanisms of social sensors, and a user dynamic knowledge graph is formed by considering the variable time function, (3) hierarchical clustering method is used to generate preliminary suspicious candidate groups and graph community detection clustering method is applied on the dynamic knowledge graph to detect the attackers. Then, AOUSD is first simulated on NetLogo and it is compared with other algorithms based on the Amazon data. The results prove the advantages of AOUSD in the efficiency and accuracy on shilling attack detection.

Keywords
Autonomous sensor, social recommendation network, dynamic knowledge graph, shilling attack, graph community detection

Date received: 28 September 2021; accepted: 3 February 2022
Handling Editor: Peio Lopez Iturri

Introduction
With the continuous development of Internet technology, especially the explosive growth of news, commodities and entertainment resources, people face serious information overload when turning to the Internet to search for the items they need.1,2 In order to provide users with information quickly and accurately, recommender system is widely used in various fields. Traditional recommendation methods mainly include collaborative filtering,3 content-based recommendation method,4,5 and hybrid recommendation method.6,7 With the development of social network, the social recommender system based on user social network is widely used in news, business, and other recommendation fields.8,9 To integrate social relations into the traditional recommender system can effectively improve the accuracy of the recommender system and alleviate the cold start problem of the recommender system.
development of Internet greatly facilitates people's life, but it also brings security problems. For example, the wireless body area network (WBAN) network is subject to major security and privacy threats, especially data collected in medical or healthcare applications. Due to the openness of the recommender system, the user can inject false scores and false social relationships into the system. Among them, the relationship shilling attacker changes the real user’s social relationships by establishing a large number of false relationships with other users, thus affecting the choice of real users, and the attackers will also spread the impact of the false relationships through the social network. There are two kinds of attacks: push attack affects the recommendation frequency by increasing the target items, while nuke attack achieves the purpose by reducing the target items. In order to improve the efficiency of stores, many businesses hire people to click farms to improve store scores and optimize commodity reviews. In 2001, Sony Pictures forged movie reviews in order to recommend newly released movies to users. These practices not only infringe on the interests of consumers, but also reduce the trust of all participants in the system over time.

In order to protect the security of recommendation network, shilling attack has been widely concerned by scholars since its emergence. At present, shilling attack detection methods can be divided into supervised, unsupervised, and semi-supervised. Supervised detection methods classify models by extracting various characteristics of users in social networks for training. Unsupervised detection methods mainly identify the abnormal points in the network by utilizing the topological relationship of social network combined with clustering method. Semi-supervised detection methods use both labeled data and unlabeled data to classify user data. 

The existing research on the shilling attack detection has achieved some results, but less attention is paid to user’s implicit social attributes, the dynamic changes of social relationships, and the information utilization of data is not comprehensive enough. This article proposes a security detection method based on autonomy-oriented user sensor (AOUSD). By establishing models for the individual user, user relationship, and user propagation in social recommendation network, the detection of user relationship attack is realized. This method simulates each user as a social sensor. A social sensor refers to an entity with the basic characteristics of autonomy, sociality, reactivity, and pre-action. According to the characteristics of social sensors, users in the recommendation network are modeled from the perspectives of activity, feedback, influence, and adaptability. The social relationships will change due to the autonomy attribute of the user sensors. Users can perceive the recommendation environment and give information feedback to the environment, which dynamically affects the surrounding environment. Then user relationships are modeled from the perspective of convergence, stability, and directionality. The influence of social sensors has the characteristics of timeliness and dynamics. After modeling user individual and user relationship, we study user information communication mode from the perspective of time interval, emergency, and attack influence.

Through the modeling of user individual, user relationship and user communication mode, the user relationship graph is established. Then, the factors which affect the user relationship, such as time and events, are added to form a dynamic user knowledge graph. The user group is classified by hierarchical clustering to obtain the preliminary suspicious candidate group. Finally, using the clustering method of graph community detection, the candidate groups are classified, and the results of shilling attack detection are obtained.

The main contributions of this article include the following: (1) users are defined as social sensors, and the definitions of user relationship and user interaction are more accurate and specific to simulate the social recommendation environment; (2) a dynamic knowledge graph model is proposed, which considers the impact of time and false user injection on the recommender system, making the detection algorithm more comprehensive and flexible; (3) hierarchical clustering and graph community detection methods are used to classify the graph structure data generated by dynamic knowledge graph, which improve the detection accuracy and reduce the damage of shilling attack.

The rest of this article is organized as follows: section “Related work” introduces the related research of dynamic knowledge graph and shilling attack detection. Section “Attack detection method based on dynamic knowledge graph” describes the proposed AOUSD in detail. In section “Experiment and analysis,” the experiment is conducted and the experimental results are analyzed to compare the effects of different detection models. Section “Conclusion and future work” summarizes the research of this article and introduces the future work.

Related work

This part summarizes the relevant literature of relational shilling attack detection, social sensors, dynamic knowledge graph, and graph community detection. The problems existing in the research of relational shilling attack detection are given in this section, as well as the motivation of this article.

Shilling attack detection

Shilling attack seriously affect the recommendation quality of the system. The relationship shilling attack model is specifically targeted at social networks. This type of attack mainly forges social information for the social network itself, thereby it is able to threaten the
core business of the social network. Therefore, how to protect the recommender system from being affected by shilling attack has become a research hotspot in the field of recommendation network security.

There are many existing detection methods for relational attack detection. For example, Tan et al., designed an unsupervised spam detection scheme called UNIK (a new UN-supervised social network spam detection scheme). This method is designed to defend against Sybil attacks. The author also points out that spammers cannot participate in social networks because it is difficult for spammers to persuade non-spammers to become their friends. However, this method is difficult to detect shilling attackers when the scope of private social networks and relational networks are small, and this method is more suitable for group and large-scale relational attacks. Ying et al. aimed to use the user relationship network formed by users of social networks to detect spam link attacks in social networks. This document uses the formation of abnormal user relationships to locate spam link attacks that occur in social networks. Zhongming et al. proposed an algorithm for locating large-scale e-commerce online supporters based on the user diagram model. However, this algorithm shows limited accuracy and efficiency.

The existing shilling attack detection methods rarely consider the dynamic changes of the users, the user relationship, and the relationship between users and environment. The utilization of information is not comprehensive enough. This article considers users as social sensors. Due to the characteristics of their own attributes and social attributes, the user relationships will change during social interaction, and it can dynamically make corresponding feedback to the surrounding environment. At the same time, users show autonomy, which means users can spontaneously exclude false users and do not have social activities with false users. After the influence ability of false users in relationship attack disappears, they can no longer affect the surrounding environment. Taking advantage of the huge difference between the influence ability of shilling attackers and real users, the accuracy and efficiency of shilling attack detection can be hopefully improved.

Social sensors

From the perspective of human identity as the perceiver of nature and society, social sensor regards human as a special intelligent sensor and studies the role of human as an agent in the network era. In 2004, Wang Fei-Yue proposed to carry out social computing research for network society, and put forward the idea and concept of social sensor network. As sensors, human individuals can move independently in the social and natural environment, and can perceive, interpret, integrate, and feedback information. Moreover, human groups can organize and spontaneously carry out social activities to achieve a certain balance and realize the self-organization of agents.

In 2009, Wagner and others put forward the view of Human as Sensor, billions of human beings constitute a huge sensor network, which can perceive and express all aspects of social activities according to their prior knowledge and expressive ability. N Jabeur and Choksuriwong proposed an emergency event management model using Bayesian Belief Networks. The model is based on social sensors and domain expert knowledge. The model utilizes social sensors to uncover posterior knowledge from uncertain emergency events, and deep event understanding using Who, What, Where, When, Why, and How (5W1H).

Most of the existing applications based on social sensors are used to address the network security problems of emergencies in social media, predict information, and establish social graph considering the behavior characteristics of users. However, this concept is rarely used in the field of shilling attack detection, and the application of dynamic knowledge graph to the description of social sensors has not been proposed. This article takes users as social sensors and builds a dynamic knowledge graph of user relationship network, and applies it on shilling attack detection. Social sensor’s communication ability, perception ability of natural environment, and the autonomy ability are fully studied.

Dynamic knowledge graph

The knowledge graph aims to describe various entities or concepts existing in the real world and the relationship between them. A knowledge graph is a relational network that connects all kinds of information together, such as social networks. Because this relationship network changes over time, the relationship between entities and entities continues to change, in order to fully acquire knowledge, Jiang et al. first proposed a dynamic knowledge graph, adding a time dimension to the knowledge graph data, using time series analysis technology and graph similarity technology to analyze the change and trend of the graph structure over time, so as to grasp the key information. Jiang et al. found that the knowledge graph is time-sensitive and proposed a t-TransE model based on it.
Dasgupta et al.\textsuperscript{34} proposed HyTE model, which directly integrates time information in the process of knowledge graph representation. Xiaoli \textsuperscript{35} proposed a time interval aware dynamic knowledge graph representation method TDG2E Timespan-aware Dynamic knowledge Graph Embedding Evolution (TDG2E). This method cuts the dynamic knowledge graph into different static subknowledge graphs according to the time node, and then uses Gated Recurrent Unit (GRU) to process each static subknowledge graph to capture the dependency relationship, so as to model the structural evolution process of the dynamic knowledge graph.

Most of the existing knowledge graph representation methods are suitable for static knowledge graph, while dynamic knowledge graph contains time information. Most of the existing dynamic knowledge graph representation methods only consider the time information on the basis of static methods, but ignore the sudden and uncertainty in the process of event evolution and information reproduction and dissemination. The time window is usually flexible and changeable. Social sensors are autonomous, social, reactive, and proactive. They can feed back information in real-time in a social environment, and the relationship between people is constantly changing. The static knowledge graph can intuitively describe the relationship between users, but it cannot reflect the nature of social sensors, and the short-term dynamic relationship changes caused by reproduction and communication ability cannot be described.

In this article, a dynamic knowledge graph is proposed, which considers a dynamic sliding window related to the factors of time and events; thus, it can better describe the changes of social sensors in social activities, and the false relationships with the attacker attackers make a huge difference, and it can detect the attackers more accurately.

This article proposes a security detection method based on AOUSD in social recommendation networks. Users are modeled as social sensors to study the interpretability of their ability to information transmission and propagation. Considering the impact on user relationships over time and the occurrence of events, a flexible and dynamic knowledge graph is constructed. Through the autonomous behaviors of user social sensors, we build user dynamic knowledge graph, and the attackers can be easily detected and excluded by graph community detection.

**Attack detection method based on dynamic knowledge graph**

In order to make full use of users’ behavior characteristics and accurately describe the propagation of user relations in social recommender system, and further improve the accuracy of shilling attack detection, a shilling attack detection model based on dynamic knowledge graph is proposed.

The overall framework of the proposed AOUSD is shown in Figure 1, which is divided into two steps: (1) input external data, pre-process the data, and establish the user relationship knowledge graph. In which, the data is stored in the form of graph, and the graph data is generated according to the user preference and the relationship function between users. Considering the dynamic change of user relationship and propagation
over time and events, a dynamic knowledge graph is constructed. (2) through the characteristics of the disappearance of the communication ability of the shilling attackers and the shilling attackers’ inability to participate in social activities, hierarchical clustering is carried out to generate suspicious candidate groups. By using the clustering method of graph community detection, the suspicious candidate groups processed by the dynamic knowledge graph are classified to obtain different user groups, screen out the shilling attackers.

Social sensor-based user model

Social sensors can carry out data acquisition, data processing, and data feedback. Social sensors have the characteristics of autonomy, flexibility, and cooperation ability, and they show the ability of group cooperation to form a social sensor network. Social sensors and the environment present a two-way communication, which not only perceive environmental information, but also output information to the environment.

Due to the characteristics of social sensors, real users can respond to emergencies. Shilling attackers will have a certain impact on real users when injecting, and have a strong ability to reproduce and spread in a short time. However, shilling attackers do not have the nature of social sensors. After the communication power disappears, the behavior curve of attackers almost returns to zero and cannot carry out social activities. At the same time, according to the reality of a social system discussed by Wang et al., individuals with high credit will not easily imitate the behavior of individuals with low credit, that is, real users will not socialize with false users for a long time. In the real social process, some users have simple social relations, weak response to emergencies and small changes in behavior curve, which can be called “zombie users.” The behavior line diagram of three types of users is shown in Figure 2. The red line represents the influence propagation curve of shilling attackers, the blue line represents the behavior representation of real users, and the green line represents the behavior representation of zombie users. When the shilling attackers inject and emergencies occur, the real users react strongly and carry out active social activities, while the response of zombie users are relatively gentle. After the influence of the shilling attackers disappears, the shilling attackers cannot respond to emergencies, and the behavior curve approaches zero. At this time, the behavior curve of the shilling attackers is similar to that of zombie users. In the hierarchical clustering to dynamic knowledge graph, it may be uniformly classified into suspicious candidate groups.

During the classification of suspicious candidate groups, the attackers can give high and concentrated scores for specific targets, and will not give feedback in case of emergencies. History browsing and comment fields are relatively narrow, and browsing time is concentrated, which can be distinguished from zombie users.

The communication diagram of social sensor is shown in Figure 3. The innermost users are first affected by emergencies or their preference behavior changes. First, the user relationship changes with the relevant users in circle F1, and then the users in circle F2 generate information feedback. However, the influence of F2 is obviously much weaker than that of F1. The change in F3 is minimal. Due to the existence of real-time feedback ability of social sensors, although the influence will become weaker, its relationship will change immediately in case of emergencies. The relationship between the shilling attackers and other users in the social network is falsely established, and it will have an impact on the users who establish the false

![Figure 2. Three types of user behavior.](image-url)
relationship and achieve the purpose of the attack. However, in a short time, this impact will weaken or even disappear. In case of an emergency, the false user will not change accordingly, and the weight of the false relationship with the real user is always at the set value and will not change immediately. Moreover, due to the autonomy of social sensors, they will restore their self-judgment ability in a certain time after they are affected. That is, after the influence ability of the shilling attackers disappears, the communication ability will not be generated again without injecting new relationships, and it will not be affected by other social sensors. This is the most important basis to our proposed detection method.

The users in the social recommendation network are defined as social sensors. Because each user is an independent individual with different personal preferences and different views on events, hence, users present great heterogeneity in social environment. To better describe the user social sensors, the definitions of user attribute, user relationship, and user relationship propagation are given as follows.

**Definition 1: user node.** The user is represented by \( r \), and the user group is represented by \( R \), that is

\[
R = \{ R_1, R_2, \ldots, R_n \} \tag{1}
\]

\[
R_i = \{ r_{i1}, r_{i2}, r_{i3}, \ldots, r_{in} \} \tag{2}
\]

\[
r_{ij} = \langle Ia | Sa \rangle \tag{3}
\]

where \( R \) represents the entire user group, including \( n \) small social groups, \( R_i \) represents each small social group, including \( m \) individual users, and \( r \) represents a single user, that is, a social sensor. The definition parameters of each individual user \( r_{ij} \) are shown in formula (3), which respectively represent <individual attributes/social attributes>. Attributes are the main source of one-to-many relationships and many-to-one relationships. Individual attributes \( Ia \) include \( he \)-height, \( wi \)-weight, \( ge \)-gender, \( ag \)-age, and \( pr \)-profession attributes, which are obtained when creating a user profile. Social attributes \( Sa \) include \( ac \)-activity, \( fe \)-feedback, \( in \)-influence, and \( ad \)-adaptability. The above parameters are normalized, and the value area is \([0, 5]\). Calculated as follows

\[
Ia = w_1 \times he + w_2 \times wi + w_3 \times ge + w_4 \times ag + w_5 \times pr \tag{4}
\]

\[
Sa = w_6 \times ac + w_7 \times fe + w_8 \times in + w_9 \times ad \tag{5}
\]

where, \( w_i \) is the weight ratio, \( \sum_{i=1}^{5} w_i = 1 \), \( \sum_{i=6}^{9} w_i = 1 \), and the size of the weight should be adjusted according to the needs of the relevant field.

Among them, individual attributes are obtained by users filling in information. Because users have doubts about the security of network information and refuse to fill in information, it will cause incomplete information acquisition, so the acquisition of social attributes is particularly important. The parameters of social attributes can be obtained through the user’s browsing history, the number of comments, and attention. The measurement of activity is realized through the user’s access to the project, browsing records, number of comments, number of clicks, and so on. The measurement indicators of feedback include the user’s response to social events (being asked, being concerned, etc.), the immediate attention to new projects, the user’s output information about the surrounding environment, and so on. The measurement of influence is measured by the convergence of other users to this user. Weak convergence includes the number of comments and follow-up degree of other users to this user, and strong convergence includes the proportion of other users taking the same social behavior as this user, such as buying the same goods, forwarding the same comments, and following the same users. The measurement of adaptability is realized by the length of time users accept emergencies. Generally, the behavior curve of users tends to be a smooth straight line. When emergencies occur, the behavior curve is abnormal, and the smooth length of time reflects the adaptability of users. Adaptability also shows the interaction ability of social sensors to the environment. Users and the surrounding environment affect, change, and adapt to each other.

The performance of social sensors is reflected in daily social interactions. For example, when a breaking news event occurs, a user \( r_{ij} \) frequently browses related information and expresses opinions on the matter, and the user \( r_{ij} \) is considered to be highly active. When shopping on a shopping platform, you will encounter the experience of being asked about the product you have bought. When you publish your product views, it is a reflection of feedback. Users with a large number of fans in Weibo have a greater influence. When reposting
events and expressing their opinions, other users will follow this user, their opinions will change to a certain extent, and the interpersonal relationship between fans will change. When public opinion occurs, users resume normal social activities in a short time because of the adaptability of social sensors.

Definition 2: user personal preference function. User historical visit items are described as $UH = \{u_1, u_2, u_3, \ldots, u_i, \ldots, u_n\}$, $i \in [1, n]$, where $n$ means that there are $n$ small social groups, which $u_i$ means the visit set of the $i$th group $u_i = \{u_{i1}, u_{i2}, u_{i3}, \ldots, u_{in}\}$, and $m$ means that there are $m$ individual users in the $i$th group. $u_{im}$ represents the user’s $r_{im}$ access list, which is shown in Table 1.

$IC^i, NO^i, NI^i, SN^i, NC^i, TC^i, RO^i$ respectively represent the normalized value of the above parameters, and the value range is $[0, 5]$. For any user $r_{ij}$, the user’s personal preference function is defined as $Hob$, namely

\[
Hob_{r_{ij}} = \langle CO, ST, DI \rangle \tag{6}
\]

\[
CO = \exp\{ad + Ia + NO^i\} \tag{7}
\]

\[
ST = \exp\{ae + pr + Ia + Sa + SN^i + TC^i + RO^i\} \tag{8}
\]

\[
DI = \exp\{fe + Sa + NI^i + NC^i\} \tag{9}
\]

\[
Hob_{r_{ij}} = \gamma (w_{i10} \times CO + w_{i11} \times ST + w_{i12} \times DI) \tag{10}
\]

where $Hob_{r_{ij}}$ is the calculation of a personal preference function, and the subscript represents the personal preference of the $r_{ij}$ user. $CO$ means the convergence, $ST$ means the stability, and $DI$ means the directionality. Take the exponential function of the parameters for better comparison calculation. Where $\sum_{x=10}^{12} w_{ix} = 1$. $\gamma$ are the rating parameters, and the user preference is evaluated in a unified order, and the value range is $[0,5]$. Convergence is a phenomenon in which users refer to each other’s preferences. For example, when a good friend makes a product purchase, they may refer to each other’s opinions and purchase together. This is different from the nature of the social sensor itself. This emphasizes the mutual follow relationship between users, which is a one-to-one relationship description. Stability means that the social relationship of real users will not change easily. In the event of an emergency, the user relationship will change slightly, but two users with high intimacy will not suddenly break their relationship, and the relationship network is relatively stable. Directionality refers to the direction of relationship between users. User $r_{11}$ affects user $r_{12}$, but $r_{12}$ has no effect on $r_{11}$. This is a one-way relationship. On the contrary, $r_{12}$ also affects $r_{11}$, which is a two-way relationship.

In the social environment, user relationship is a complex network structure, rather than the simple false user relationship established by the attackers when injecting. Therefore, it is necessary to establish a weighted knowledge graph structure by calculating the user relationship description function. The weight here is obtained by function calculation considering the above factors. In the triple relationship of knowledge graph, the weight of each edge represents the closeness of the relationship between users.

Social sensors-based user relationship

Social sensors can interact with data, information, and behavior. In this article, users in social networks are simulated as social sensors, and the relationship between them is modeled to form a knowledge graph relationship network with weight function.

In the recommender system, the user community will change due to emergencies and other factors. For example, user $r_{11}$, originally belonged to user group $R_1$, but due to a Weibo public opinion event, following a blogger, this user $r_{11}$, is owned by the blogger. The fan group $R_2$ of will establish a certain social relationship, and this user change will be denoted as $r_{11}^{11}$. Figure 4 shows the weight function change network of the knowledge graph, which is the corresponding change of the knowledge graph for emergencies and over time, and its weight changes in different sizes according to the closeness of the user relationship.

During injection, the shilling attackers sometimes inject the user profile and sometimes falsely construct the user relationship. However, due to the fact that the false users do not have the nature of social sensors, the user relationship weight function will not change and will only maintain the set value at the time of injection, the propagation force will disappear in a short time and will no longer make a two-way response to the environment.
Definition 3: short-term static relation function. Define the user relationship weight function that changes due to factors such as emergencies and time as \( MW_{TI} \), namely

\[
MW_{TI_v} = \left\langle Hob_{\gamma_v}, REC, CDL, SIM, IFL \right\rangle
\]  

(11)

where \( T_{I_v} \) is defined as follows

\[
T_{I_v} = (T_v + (j - 1) \times d, T_v + j \times d)
\]  

(12)

\[
d = \frac{K}{e + \Delta n + \Delta r}
\]  

(13)

Specifically

\[
REC = \frac{1}{|pr - pr|} + \frac{1}{|Hob_{\gamma_v} - Hob_{\gamma_a}|}
\]  

(14)

\[
CDL = \begin{cases} 
\frac{NC' + NC'}{2} + \frac{ST + ST}{2} \times \frac{1}{|ST - ST|}, & \text{if } |NC' - NC'| \in [0, 1], |ST - ST| > 1 \\
\frac{NC' + NC'}{2} \times \frac{1}{|NC' - NC'|}, & \text{if } |NC' - NC'| > 1, |ST - ST| \in [0, 1] \\
\frac{NC' + NC'}{2} + \frac{ST + ST}{2} \times \frac{1}{|NC' - NC'| \times |ST - ST|}, & \text{if } |NC' - NC'| \in [0, 1], |ST - ST| \in [0, 1]
\end{cases}
\]  

(15)

\[
SIM = \frac{1}{|SN' - SN'|} + \frac{1}{|Ia - Ia|}
\]  

(16)

\[
IFL = \begin{cases} 
\frac{IC' + IC'}{2} + \frac{RO' + RO'}{2} \times \frac{1}{|RO' - RO'|}, & \text{if } |IC' - IC'| \in [0, 1], |RO' - RO'| > 1 \\
\frac{IC' + IC'}{2} \times \frac{1}{|IC' - IC'|}, & \text{if } |IC' - IC'| > 1, |RO' - RO'| \in [0, 1] \\
\left(\frac{IC' + IC'}{2} + \frac{RO' + RO'}{2}\right) \times \frac{1}{|RO' - RO'| \times |IC' - IC'|}, & \text{if } |IC' - IC'| \in [0, 1], |RO' - RO'| \in [0, 1]
\end{cases}
\]  

(17)

\( MW_{TI_v} \) represents the weight of the multi-factor weight in the \( T_{I_v} \) time period, and \( v \) is an arbitrary time sequence number. The size of the \( MW_{TI_v} \) function value is related to the \( <Hob_{\gamma_v}, REC, CDL, SIM, IFL> \) factor, and respectively represents the \( Hob_{\gamma_v} \)-user preference function, \( REC \)-relevance, \( CDL \)-credibility, \( SIM \)-similarity, and \( IFL \)-influence corresponding to the user at time \( T_{I_v} \). The dynamic time interval function is defined, and the interval width \( d \) can be changed. In a certain period, \( e \) represents the number of comments, \( \Delta n \) represents the change in the number of comments, \( \Delta r \) represents the relationship change, and \( K \) is the calculation.
parameter. Change the detection frequency as needed to improve the accuracy of the system.

The degree of relevance is the closeness of the relationship between users, such as leaving messages, comments, and following each other, and discussing when emergencies occur. It is believed that the degree of relevance between this user group is relatively high. Credibility is a form of user influence. Wang et al. proposed an evolutionary game model to encourage cooperation between nodes, which effectively combines the credit-based incentive method with the evolutionary game model. Experiments show that the larger the credit function, the lower the success rate of abnormal nodes in transmitting information. In this article, trust function is used to detect shilling attack. Similarity is the degree of similarity in behaviors such as preferences among users. According to research, the similarity between good friends is relatively high, their preferences are similar, and their responses to emergencies are similar. Influence is the user’s ability to influence when the time is $T_i$. When an emergency occurs, the influence of negative news and other factors has a huge change, and it is necessary to detect the influence of the user at each moment.

The relevance, credibility, similarity, and influence of users will change with the change of events and time. With the development of microblog and other online communication platforms, the influence of network public opinion is becoming greater and greater. Social sensors will not only change the feedback information when events occur, but also change preferences over time. Therefore, a variable time window is added to this detection method to make the use of database information more comprehensive and reliable, and the accuracy of the results is guaranteed.

**Definition 4: dynamic relation function.** For any user $r_i \in R_i$, the user relationship change function is defined as $URC$, namely

$$URC_{r_i} = |\exp(MW_{T_i-1}) - \exp(MW_{T_i})|$$

The value of $URC$ is related to the short-term static relationship function. Calculate the $MW$ value at $T_{i-1}$ and $T_i$ as $MW_{T_{i-1}}$ and $MW_{T_i}$ respectively. The obtained change value is the dynamic relationship of $URC$ at $T_i$. The greater the relationship change, the more active the user and the reaction to the event strong, reflecting the feedback of social sensors to information, the closer the $URC$ is to 0, it means that the user relationship remains unchanged for a long time. For the $URC$ ratings of users in different time periods, the user groups are clustered according to their levels. The attackers will no longer react to any influencing factors for a long time after the influence has disappeared. Its $URC$ is close to 0 and remains unchanged, unable to participate in social activities, and can be classified as a suspicious candidate group.

**Definition 5: N-hop relationship tightness.** The closeness of the relationship between the target user and the first hop user is defined as the one-hop proximity $F_1$, namely

$$F_1 = CI(r_{i1}; r_{i1}^1, \ldots, r_{i1}^1)$$

where $F_1$ indicates the closeness of the relationship between user $r_i$ and the first hop, and $r_{i1}^1, \ldots, r_{i1}^1$ indicates the first-tier users related to $r_i$. By analogy, we can define the closeness of the N-hop user relationship, namely

$$F_n = CI(r_{i1}; r_{i2}, \ldots, r_{in})$$

It represents the closeness of the relationship between user $r_n$ and the $N$th level, and $r_{i1}^1, \ldots, r_{in}^1$ represents the $N$th level user related to $r_n$. The larger the value, the closer the relationship. As the level expands, the value of $F_n$ gradually approaches zero.

When describing the attributes of the user and the user relationship function, the amount of calculation can be reduced according to the definition of the hop count. When the hop count of the $n-1$ step user $F_{n-1}$ has approached 0, there is no need to calculate the relationship of the $n$th user because the closeness is already not worthy of reference, such operations can reduce the amount of calculation and improve the efficiency of the algorithm.

**Dynamic knowledge graph**

After shilling attackers enter the social network, they will establish false relationships to attack the recommender system. But the shilling attackers do not have the nature of the social sensor, and they are unable to participate in complex social activities. Hence, their influence propagation will disappear in a short time, and they will not respond to emergencies. The user relationship function value of attackers is artificially injected and it is a fixed value. As shown in Figure 5, the red users are shilling attackers, and the relationships established with the users are artificially set. Since the behaviors of zombie users and attackers show similarity after the spreading power disappears, they will be unified into the suspicious candidate group in the process of knowledge graph.

Under normal circumstances, the knowledge graph $G$ is considered static, and $G$ includes triples of $(h, r, t) i \in [1, N]$, where $N$ represents the number of knowledge in $G$, and $h, r, t$ represents the head entity and tail entity connected by the relationship. In this article,
relationship refers to the interpersonal communication between users. Time series is added to the static knowledge graph, that is, time dimension information is added to the triple. At this time, the dynamic knowledge graph can be expressed as four tuples \((h_i, r_i, t_i, t'_i)\) graph structure. Compared with the static knowledge graph, the user relationship of the dynamic knowledge graph will change. In the static knowledge graph, the relationship between users is only determined by colleagues or friends, while in the dynamic knowledge graph, the relationship between users will change over time. For example, the relationship between colleagues disappears due to job transfer and the relationship between friends is broken due to emergencies, and so on. Using the characteristics of dynamic knowledge graph, this article considers the factors of time and emergencies. When there is no emergency, the fixed time window is used to detect the user relationship. When an emergency occurs, the width of the time window changes and the frequency of user relationship calculation increases, that is, the user relationship is extracted many times in a short time. The collection of calculation increases, that is, the user relationship changes and the frequency of user relationship calculation increases. When an emergency occurs, the time window is used to detect the user relationship. The use of user relationship hops to simplify the calculation of the algorithm and reduce the cost of calculation time. Data utilization is relatively comprehensive, forming a knowledge graph that changes dynamically over time, which is real-time. In the event of emergencies the user's authenticity can be judged by the changes in the values of the above functions, and suspicious candidate groups can be obtained to ensure the accuracy of the system.

**Graph community detection model**

Due to the similarity between zombie users and shilling attackers, it often leads to high false detection and missed detection, which affects the detection accuracy. In this article, the suspicious user candidate group is subdivided by graph community detection, and the shilling attack group and zombie user group are clustered. Graph community detection is a graph-based clustering method. Graph clustering is an unsupervised learning process of graph structure classification. The graph clustering aims to divide a graph into \(k\) disjoint subgraphs \(g_i = (v_i, e_i, w_i), (i = 1, 2, \ldots, k)\). In the dynamic knowledge graph constructed in this article, there are multiple graph groups to represent the relationship between users. The number of users (vertices) adjacent to the target user (vertices) is marked as \(k\) and defined as the degree of users (vertices). The modularity function is defined as the following formula

\[
M = \frac{1}{2L} \sum_{i,j=1}^{N} A_{ij}(r_i - k_{i,v}/2L) \delta(c_{r_i}, c_{r_j})
\]

where \(N\) represents the number of users, \(L\) represents the number of edges contained in the graph, which is the relationship between users, \(k_{r_i}\) represents the degree of users, \(A_{ij}\) is the weight on the domain of real number, \(e_i\) is the set of edges, \(v_i\) is the set of vertices and \(r_i\) is an edge weight function, which maps an edge to a weight on the domain of real number. For an unweighted graph, then the weight of each edge is 1. The graph clustering aims to divide a graph \(g = (v, e, w)\) into \(k\) disjoint subgraphs \(g_i = (v_i, e_i, w_i), (i = 1, 2, \ldots, k)\). In the dynamic knowledge graph constructed in this article, there are multiple graph groups to represent the relationship between users. The number of users (vertices) adjacent to the target user (vertices) is marked as \(k\) and defined as the degree of users (vertices). The modularity function is defined as the following formula

\[
M = \frac{1}{2L} \sum_{i,j=1}^{N} A_{ij}(r_i - k_{i,v}/2L) \delta(c_{r_i}, c_{r_j})
\]

where \(N\) represents the number of users, \(L\) represents the number of edges contained in the graph, which is the relationship between users, \(k_{r_i}\) represents the degree of users, \(A_{ij}\) is the weight on the domain of real number, \(e_i\) is the set of edges, \(v_i\) is the set of vertices and \(r_i\) is an edge weight function, which maps an edge to a weight on the domain of real number. For an unweighted graph, then the weight of each edge is 1.
of user \( r_{in} \), \( A_{r_{in}r_{jm}} \) is the value in the adjacency matrix. \( c_{rin} \) represents the clustering of user \( r_{in} \). \( M \) is a standard to measure the quality of the division of graph groups. The better results of the division, the greater the value of \( M \). \( \delta \) is the Krone function

\[
\delta_{ij} = \begin{cases} 
0 & \text{if } i \neq j \\
1 & \text{if } i = j 
\end{cases}
\] (23)

According to equation (23), if users \( r_{in} \) and \( r_{jm} \) belong to the same cluster, then \( \delta(c_{rin}, c_{rim}) \) returns 1, and if users \( r_{in} \) and \( r_{jm} \) do not belong to the same cluster, then \( \delta(c_{rin}, c_{rim}) \) returns 0. Take user \( r_{in}, r_{jm} \) randomly. If \( r_{in} \) and \( r_{jm} \) is the same cluster, \( \delta(c_{rin}, c_{rim}) \) returns 1. At the same time, the user \( r_{in}, r_{jm} \) performs fusion and calculates the \( M \) value. At this time, a modular difference \( \Delta M \) will be generated. Then, take the two clusters with the largest growth of \( \Delta M \) for fusion. If the user \( r_{in}, r_{jm} \) is not in the same cluster, \( \delta(c_{rin}, c_{rim}) \) returns 0, and \( M \) is also 0. Fusion cannot be carried out. Users (vertices) will continue to be randomly selected until all vertex users are divided into a single cluster group.

After the dynamic knowledge graph detects the data, the users are hierarchically clustered, the Euclidean distance is calculated according to the personal preference \( Hob_{r_{in}} \) and the relationship weight function \( MW_{T_{in}} \) between the users, and the clustering is performed to generate suspicious candidate groups, which may include zombie users and shilling attackers. The attackers use the graph community detection clustering method to cluster the user community again. Although zombie users are not active, they can participate in social activities and social networks exist, while fake users cannot perform social activities. The weight of the relationship function with real users is a fixed value, and only has a set relationship with individual users, and there is no social network. Figure 6 shows the specific clustering method. Since false users (the red users in Figure 6) are injected, some relationships are falsely established. After the influence disappears, the false relationships cannot change flexibly, and the false users do not have the attributes of social sensors \( Ia \) and \( Sa \). In this way, the user relationships can not be changed, and the false users are marked as suspicious candidate groups in the process of dynamic knowledge graph processing. In the clustering process of graph community detection, the relationships between the attackers and the group are relatively distant and do not have a complex network structure, which makes it impossible to cluster the attackers into the clustered group. Although zombie users are not active enough, they can engage in social activities and they show certain social activities. When emergencies occur, they will give certain feedback and can be clustered into groups. The groups of different colors in Figure 6 are the clustered user groups formed after the graph structure is clustered. After eliminating the shilling attackers, the accuracy of the recommendations can be improved.

**AOUSD algorithm description**

According to the definition of social sensor attributes and user weight function, an algorithm of shilling
Algorithm 1. Dynamic knowledge graph algorithm.

Input: user profile $U_P = \{z_1, z_2, z_3, \ldots, z_i\}$ data set; user history browsing data set $U_H = \{u_1, u_2, u_3, \ldots, u_n\}$.
Output: graph structure data packet $G_d$ and suspicious candidate group $S_g$.
1. Establish individual user attribute information $I_a$ according to $U_P$.
2. Extract the number of users’ followers, trackers, visits, comments, and so on according to $U_H$.
3. $i = 1, j = 1$.
4. For $r_i \in R_i$, do
5. Calculate user a $F_i$, hop function.
6. If $F_i \in [0, 0.3]$, then
7. Jump to Step 14.
8. Else do calculates the user’s personal preference function $Hob_r_i$.
9. Insert a dynamic time sliding window $T_i$.
10. Calculate short-term static relationship function $MW_{T_i}$.
11. Repeat Steps 8–10 to calculate the dynamic relationship change function $URC_{r_i}$.
12. $i = i + 1, j = j + 1$.
13. Return for.
14. Construct user relationship knowledge graph $G_i$.
15. Dynamic Knowledge Graph Fusion $G_d = G_1 \cup G_2 \cup G_3 \ldots G_f$.
16. Take the min($URC$) user group as the suspicious candidate group $S_g$.

Algorithm 2. Graph community detection algorithm.

Input: Dynamic Knowledge Graph Structure Data $G_d$.
Output: clustering user group $C_g$ and shilling attackers $S_g$.
1. Read $G_d$ data.
2. $i = 1, j = 1$.
3. For $r_i \in R_i$, do
4. $M = \frac{1}{P} \sum_{i=1}^{N} \left( A_{c_{ij}} \cdot \frac{k_{m_{ij}}}{k_{m_{j}+k_{m_{i}}}} \right) \delta (c_{ij}, c_{ij})$.
5. Randomly select two $c_{ij}, c_{ij}$ clusters to fuse.
6. Calculation Modularity Change $\Delta M$.
7. Sort the calculated $\Delta M$ from largest to smallest.
8. Fuse the first two $\Delta M$ clusters in the sorting.
9. Calculate new modularity $M_i$.
10. Keep repeating Steps 4–8.
11. Until all user $r_i$ is clustered.

Dynamic knowledge graph algorithm. Algorithm 1 is mainly composed of three parts. The first part (lines 1–2) extracts the characteristics of database user relationships. The second part (lines 3–13) constructs time and event windows, and calculates the relational function of dynamic changes. The third part (lines 14–16) constructs a dynamic knowledge graph and obtains the suspicious candidate group.

The time complexity of user relationship feature extraction is $O(|P|^2)$, the time complexity of construction time and event sliding window is $O(|P|^3)$, and the time complexity of dynamic knowledge graph construction and suspicious candidate group generation is $O(|P|^3)$. Therefore, the time complexity of Algorithm 1 is $O(|P|^2) + O(|P|^3) + O(|P|^2) \approx O(|P|^3)$.

Graph community detection algorithm. Algorithm 2 mainly includes two parts. The first part (lines 1–4) reads the data and calculates the modularity $M$ in the graph structure of the dynamic knowledge graph. The second part (lines 5–11) calculates the modularity change $\Delta M$, generates a clustered group, and eliminates the shilling attackers.

The time complexity of computing graph structure modularity is $O(|P|^2)$, and the time complexity of computing modularity change $\Delta M$ is $O(|P|^3)$, so the time complexity of Algorithm 2 is $O(|P|^2) + O(|P|^3) \approx O(|P|^3)$.

Experiment and analysis

This part first introduces the simulation platform, experimental parameters, and interface settings. Then the simulation experiments are carried out on the detection of the shilling attackers, and the influence of the factors proposed in this paper on the detection is given. Finally, the experimental results are analyzed and compared with other models on the Amazon data set.

Experiment platform NetLogo

With NetLogo, the simulation world can be manipulated through instructions to observe the changes in the simulation world. As long as the properties and behavior rules of each subject input clear commands can make the simulation system run. Each agent in NetLogo is a visual, self-acting actor, which makes it possible to explore the connections between individual behavior at the micro-level and the macro-patterns. NetLogo can be applied to many fields of natural and social sciences, such as biology, medicine, physics, chemistry, computer science, as well as economics and social psychology. For example, the SIR Susceptible
Infected Removed (SIR) network virus spread model based on NetLogo shows the spread of viruses through the network.\textsuperscript{40}

**Simulation interface design**

In a social recommender system, each social sensor can change its state through mutual influence and changes. The main purposes of simulation modeling are follows: (1) observe the interpersonal network generated by the activities of social sensors in the social environment, as the autonomy of social sensors, (2) observe the effect of user weight function on the communication ability of shilling attackers, (3) based on the dynamic time window function, the influence of detection frequency on the propagation ability of shilling attackers and the structure of user relationship network is observed, and (4) observe the impact of the number of shilling attackers on the communication ability of users, and analyze the precision and recall of shilling attack detection methods. Figure 7 shows the visual interface of the simulation system. The simulation experiment environment is composed of three interfaces: button, world, and drawing. The view area includes the social sensor social network view and the drawing interface supporting the attackers’ communication ability.

The simulation world of the attackers’ communication ability is a network of “$xcor \ [-60, 60]$, $ycor \ [-60, 60]$.” The meanings of the design components are described in Table 2.

The blue nodes represent the real users of the social sensors. When an attacker is added to the social network (the initial red node), the attacker will have a certain spread influence when he/she joins the network. So that users who are closely related at the time of injection are affected and become abnormal users. Due to the adaptability $ad$ and feedback $fe$ in the social attribute $Sa$ of the social sensor, abnormal users will resume their normal preferences and social interactions after a certain period of time, which is represented by gray nodes.

Figure 8 shows the system interface for injecting false relations. The social sensors, which are the blue nodes, are set to 1000, and 15 shilling attackers are injected, which are shown as the red nodes. Users continue to establish social relationships and form social networks. It can also be observed that they are divided into many closely related user groups. Attackers are asked to establish false relationships with these user groups and injected into the user’s social network to achieve the purpose of the shilling attacker to influence the recommender system.

**Evaluation indicators**

From the perspective of shilling attack detection, the judgment of shilling attackers can be regarded as a classification problem. The commonly used metrics are precision, recall, and $F1$

\[
P = \frac{TP}{TP + FP} \quad (24)
\]

\[
R = \frac{TP}{TP + FN} \quad (25)
\]

\[
F1 = \frac{2 \times P \times R}{P + R} \quad (26)
\]

where $TP$ represents the number of real attacking users in the set of users judged to be attacking. $FP$ represents the number of real users who are misjudged in the set of real users judged to be attackers. $FN$ represents the number of hidden attacking users in the set which are judged as real users. The higher the value of $P$, the greater possibility it is that the found attacking user is indeed the real shilling attacker. The higher the value of $R$, the greater the possibility that the real shilling attacker will be detected. $F1$ metric is a more balanced index. The larger the $F1$ metric, the more accurate the detection result.

**Analysis of simulation**

**Influence of $MW_{TL}$ function on elapsed time.** Set the value of the user relationship weight function $MW_{TL}$ to 20%, 30%, 40%, and 50%, respectively, and obtain the line graph of the influence of $MW_{TL}$ on the propagation capability as shown in Figure 9. The turning point coordinates of the gray curve from the diagonal line to the horizontal line are $(142.6, 36.7)$, $(137.2, 46.3)$, $(110.5, 49.2)$, and $(82.2, 56.0)$. The coordinate meaning of the turning point is as follows: the horizontal axis represents the time when the attackers’ influence ability disappears, and the vertical axis represents the ability to resist the attackers. The abscissa value gradually decreases from 142.6 to 82.2, which indicates that the time for attackers to affect their ability gradually decreases. The value of ordinate gradually increases from 36.7 to 56.0, which indicates that the ability of users to resist attackers is gradually enhanced.

The CO-convergence of social sensors makes the ad-adaptability and attention relationship between users more stable when there are no emergencies. When the false relationship is injected by the attackers, the attackers cannot participate in the social activities in the normal mode, so it is identified and eliminated. The larger the user relationship function between real users is, the more active social activities are, and the attackers can be identified more quickly.

**Impact of dynamic time sliding window on communication ability.** The width of the time window should be flexible according to emergencies, changes in the number of
comments, and changes in relationships. The faster the detection frequency, the shorter the discovery time for the attackers, thus, the detection efficiency of the system can be improved. Some of the attackers injected in the same batch are detected and removed from the system, so the spreading influence of this batch of attackers will be weakened. As shown in Figure 10, the time intervals of emergencies are set to 7, 6, 5, 4, 3, and 2, respectively, that is, the smaller the time interval of emergencies, the higher the frequency of events. At this time, the $d$ of the width of the dynamic time window that the detection system automatically corresponds to decrease in sequence, and the detection frequency increases in sequence.

**Figure 7.** Visual interface diagram of simulation system: (a) system buttons, (b) system worlds, and (c) system drawing.
Table 3 shows the apex of the red curve indicating the transmission capability and the turning point of the gray curve indicating the elapsed time. It can be seen that as the detection frequency increases, the horizontal and vertical coordinates of the flat point of the gray curve decrease in turn. This means that the elapsed time of influence capacity is shortened. Given a fixed time $t = 500$, the elapsed rate of the attackers’ influence is calculated at each time interval, and it can be found that the faster the detection frequency, the higher the

| Species | Name | Meaning description |
|---------|------|---------------------|
| World   | The user’s social environment | Blue nodes represent users, Red nodes represent the attackers and affected users, Gray nodes indicate the restoration of real users |
| Button  | Set up | Initialize the world |
|         | Go once | Run the program once |
|         | Go | Run the program |
|         | Set up 2 | Initialize the shilling attackers |
|         | Spread | Spread influence |
|         | Redo layout | Arrange again |
|         | Resize nodes | Adjust node size |
| Slider  | Initial-attack-scale | Initial attack scale, number of shilling attackers injected |
|         | Unexpected-events-occur-interval | The time interval between emergencies |
|         | Shilling-attacker-spread-ability | The shilling attackers’ ability to spread |
|         | Social-frequency-of-real-users | Social frequency of real users |
|         | User-relationship-description-function | User relationship description function |
| Drawing | Red line chart (influence) | Response to the shilling attackers’ ability to spread |
|         | Gray line chart (resistant) | Social sensor resistance |
|         | Blue line chart (susceptible) | The number of unaffected real users |
| Monitor | Users | Detect the total number of social sensors at each moment |

Table 2. Simulation experiment interface information.

![Figure 8. Shilling attack injection interface.](image-url)
elapsed rate. The ordinate of the apex of the red curve decreases successively, and the transmission capacity decreases too. The higher the detection frequency, the faster the attackers will be detected and eliminated, and the elapsed time of the propagation ability will also decrease. Therefore, adding a variable time window to

Figure 9. Influence of $MW_{TI}$ function on elapsed time ($MW_{TI} = 20\%, 30\%, 40\%, 50\%$).

Figure 10. The influence of detection frequency on propagation capability (the time interval is 7, 6, 5, 4, 3, and 2).
the dynamic knowledge graph has a certain effect on the improvement of detection efficiency.

**Effect of the number of attackers on the time of influence disappearance.** Figure 11 shows the influence of the number of attackers on the communication power. The gray broken line indicates that the affected real users have restored their original preferences. The actual users are set to 1000, and the number of attackers is 15, 30, 45, 60, 75, and 90. The starting point of the red curve represents the number of attackers detected, the highest point represents the strength of the propagation ability, and the point where the ordinate tends to zero represents the time when the propagation ability has elapsed. Table 4 shows the values of each coordinate value, accuracy rate, and recall rate. As the number of attackers increases, the ordinate of the highest point of the red curve, which is the peak point of influence, gradually increases from 14.21 to 30.3. This phenomenon indicates that the greater the number of attackers injected at one time, the greater the influence. The time point of influence disappeared gradually increased from 129 to 170. Taking the average value as $T = (129 + 144 + 141 + 162 + 153 + 170)/6 = 150$, it can be obtained that the number of attackers under the same conditions has little effect on the loss of influence. Since real users have the attributes of social sensors, after a period of social activities, an attacker who cannot conduct social activities is identified, and their influence disappears. After calculation, the value of $F1$ under various numbers of attackers is close to 1, indicating that the accuracy of the system is high.

After simulation, it is confirmed that the user weight function and the variable time interval window function have a certain effect on the influence ability and detection efficiency of the trust attack. User relationship weight function $MW_{TI}$, real users frequently socialize, and actively feedback information to each other. In this way, the false identity of the shilling attackers will be

| Time interval | Red vertex coordinates | Gray leveling point coordinates | Decay rate |
|---------------|------------------------|---------------------------------|------------|
| 7             | (43.0, 67.6)           | (282.1, 58.1)                  | 0.436      |
| 6             | (39.0, 63.6)           | (247.7, 50.8)                  | 0.505      |
| 5             | (38.0, 54.7)           | (207.0, 48.2)                  | 0.586      |
| 4             | (29.3, 45.3)           | (186.0, 46.6)                  | 0.628      |
| 3             | (23.6, 35.6)           | (144.1, 42.4)                  | 0.712      |
| 2             | (17.6, 24.9)           | (121.0, 34.8)                  | 0.758      |

**Figure 11.** The impact of the number of attackers on the spread (shilling attackers =15, 30, 45, 60, 75, and 90).
easily distinguished, and the real user will not follow the shilling attackers, nor will they have social interaction with the shilling attackers, and the influence of the shilling attackers will disappear. The larger the value of the user relationship weight function, the more interaction between users, and the faster the influence of the attackers will fade away. The width \( d \) in the dynamic time window function \( T_I \) is related to emergencies, changes in the number of comments, and changes in relationships. The more frequent emergencies, the \( d \) of the dynamic time window automatically decreases, which increases the frequency of detection. After detecting some of the attackers, the recommender system will be optimized, which will weaken the influence of the attackers, speed up the elapsed time, and better protect the social recommender system.

**Model comparison based on the Amazon data set**

The Amazon data set is used as the scoring database of the recommender system for experiments. The comparison detection algorithms include a group attack detection method based on graph embedding (GAGE),\(^{41}\) a DeepWalk detection method which takes a graph as input and produces a latent representation as an output (GADW),\(^{42}\) and a novel technique for identifying group attack profiles (DeR-TIA).\(^{43}\) GAGE is a group attack detection method based on graph embedding. First, the user relationship graph is constructed by analyzing the user’s scoring behavior, and the low-dimensional vector representation of nodes is obtained by graph embedding method, and then the attack group is identified by twice clustering. GADW is a new method to learn the potential representation of vertices in networks. These potential representations encode social relations in a continuous vector space. Social representation is the potential feature of capturing the vertices of domain similarity and community membership. DeR-TIA is a two-stage method for detecting group shilling attacks. First, Rating Deviation from Mean Agreement (RDMA) and Degree of Similarity with Top Neighbors (DegSim) metrics and K-means clustering algorithm are used to obtain a group of suspicious users, and then target item analysis method is used to identify a group of attacking users. Figure 12 shows the comparison of the precision rate, recall rate, and F1 parameters of each model. The experiments in this study were performed in a desktop computer which has 8 GB RAM, Intel Core i5-8250 CPU, Nvidia RTX 2070 GPUs. The operating system is Windows 10, 64-bit, and the software we used is Python 3.6. The data used is the Amazon data set which is open source on the Internet. Its website is https://nijianmo.github.io/amazon/index.html.

| Number of shilling attackers | Peak point of influence | Time point of influence | FP | FN | P  | R  | F1 |
|-----------------------------|-------------------------|------------------------|----|----|----|----|----|
| 15                          | 14.21                   | 129                    | 2  | 1  | 0.875 | 0.933 | 0.903 |
| 30                          | 22.65                   | 144                    | 1  | 2  | 0.965 | 0.933 | 0.949 |
| 45                          | 24.4                    | 141                    | 6  | 1  | 0.880 | 0.978 | 0.926 |
| 60                          | 26.9                    | 162                    | 4  | 6  | 0.931 | 0.900 | 0.915 |
| 75                          | 28.4                    | 153                    | 2  | 1  | 0.973 | 0.986 | 0.979 |
| 90                          | 30.3                    | 170                    | 2  | 4  | 0.977 | 0.956 | 0.966 |

**Figure 12.** Performance comparison of different approaches.
It can be concluded that AOUSD model is better than GAGE, GADW, and DeR-TIA models in accuracy, recall, and $F_1$. This is mainly due to considering the nature of users’ autonomous sensors, adding time variable factors, making information collection more flexible, and using twice clustering to carefully classify the user groups to obtain the user groups.

Using the NetLogo simulation, it is proved that the change of user relationship weight function and dynamic time window function have a certain impact on the transmission and propagation ability of trust attackers. Further by comparing with other detection methods, AOUSD shows efficiency and effectiveness on shilling attack detection. Hence, it can be concluded that the factor direction considered in the proposed security detection method based on user autonomous sensor is effective, and the detection efficiency of trust attack is considerable.

**Conclusion and future work**

The existence of shilling attackers affects the security of recommendation network and causes bad user experience. In order to improve the accuracy and flexibility of shilling attack detection, a security detection method based on AOUSD is proposed. Users are modeled as social sensors with autonomy characteristic, and a dynamic knowledge graph is formed by adding variable time and event sliding windows. After forming the graph structure, graph community detection is used for clustering, which reduces the misjudgment rate of shilling attackers. The experiments show that the false relationship shilling attackers can be quickly detected through the introduction of social sensors, user relationship function, time window, and other factors. The proposed AOUSD has the following characteristics:

1. The intrinsic attributes of social sensors are used to build a knowledge graph, which reduces the dependence on user historical scores and other information.
2. The sliding window of variable time and events is added to the static knowledge graph, which makes the network map change with time, more real-time and comprehensive information utilization.
3. The following characteristics of the shilling attackers are used to detect: the shilling attackers cannot participate in social activities, and the communication ability of the shilling attackers disappears with time and events.
4. The dynamic knowledge graph is a white box, which increases the interpretability of the detection system and facilitates the optimal design of the detection system.

In the future work, we will focus on how to build a user autonomy model to better identify shilling attackers when shilling attackers have more complex behavior curves. In addition, we will use graph convolution to realize the shilling attack detection based on dynamic knowledge graph to improve the detection accuracy.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors wish to acknowledge the support of the National Natural Science Foundation of China (no. 61902016, no. 61871020), the Beijing Municipal High-Level Innovation Group Construction Project (no. IDHT20190506), the Natural Science Foundation of Beijing Municipality (no. 4204093), and the National Key R&D Program of China (no. 2018YFC0807806-2).

**ORCID iD**

Shanshan Wan https://orcid.org/0000-0002-3421-4387

**References**

1. Warren J and Marz N. *Big Data: principles and best practices of scalable realtime data systems*. New York: Simon & Schuster, 2015.
2. Xiong H and Liu Z. A situation information integrated personalized travel package recommendation approach based on TD-LDA model. In: 2015 international conference on behavioral, economic and socio-cultural computing (BESC) Nanjing, China, 30 October–1 November 2015. New York: IEEE.
3. Wang X, Yu L, Ren K, et al. Dynamic attention deep model for article recommendation by learning human editors’ demonstration. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, Halifax, NS, Canada, 13–17 August 2017. New York: Association for Computing Machinery.
4. Oppermann M, Kincaid R and Munzner T. Viz recommender: computing text-based similarity in visualization repositories for content-based recommendations. *IEEE Trans Vis Comp Graph* 2020; 27(2): 495–505.
5. Wahyudi K, Latupapua J, Chandra R, et al. Hotel content-based recommendation system. *J Phys Conf Ser* 2020; 1485(1): 012017.
6. Tarus JK, Niu Z and Kalui D. A hybrid recommender system for e-learning based on context awareness and sequential pattern mining. *Soft Comput* 2018; 22(8): 2449–2461.
7. Wang X and Wang Y. Improving content-based and hybrid music recommendation using deep learning. In: *Proceedings of the 22nd ACM international conference on multimedia*, Orlando, FL, 3–7 November 2014. New York: Association for Computing Machinery.
8. Golbeck J. Generating predictive movie recommendations from trust in social networks. In: International conference on trust management, 2006. Berlin: Springer, https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.3337&rep=rep1&type=pdf

9. Ma H, Yang H, Lyu M, et al. SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM conference on information and knowledge management, Napa Valley, CA, 26–30 October 2008. New York: Association for Computing Machinery.

10. Chen CM, Xiang B, Wu TY, et al. An anonymous International Journal of Distributed Sensor Networks

11. Chirita PA, Nejdl W and Zamfir C. Preventing shilling attacks in online recommender systems. In: Proceedings of the 7th annual ACM international workshop on web information and data management, Bremen, 4 November 2005. New York: Association for Computing Machinery.

12. Zhang S, Ouyang Y, Ford J, et al. Analysis of a low-dimensional linear model under recommendation attacks. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, Seattle, WA, 6–11 August 2006. New York: Association for Computing Machinery.

13. Peng X and Selvachandran G. Pythagorean fuzzy set: state of the art and future directions. Artif Intell Rev 2019; 52(3): 1873–1927.

14. Yang Z, Xu L and Cai Z. Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. Knowl Based Syst 2016; 100: 74–88.

15. Bryan K, O’Mahony M and Cunningham P. Unsupervised retrieval of attack profiles in collaborative recommender systems. In: Proceedings of the 2008 ACM conference on recommender systems, Lausanne, 23–25 October 2008. New York: Association for Computing Machinery.

16. Yang Z, Cai Z and Guan X. Estimating user behavior toward detecting anomalous ratings in rating systems. Knowl Based Syst 2016; 111: 144–158.

17. Cao J, Wu Z, Mao B, et al. Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web 2013; 16(5–6): 729–748.

18. Zhang L, Wu Z and Cao J. Detecting spammer groups from product reviews: a partially supervised learning model. IEEE Access 2017; 6: 2559–2568.

19. Xingjun W and Qinghua G. Semi-supervised detection of shilling attack for recommender system based on characteristic index. Appl Res Comp 2018; 35(7): 2185–2188.

20. Lee K, Caverlee J and Webb S. Uncovering social spammers: social honeypots + machine learning. In: Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval, Geneva, 19–23 July 2010. New York: Association for Computing Machinery.

21. Gao H, Hu J, Wilson C, et al. Detecting and characterizing social spam campaigns. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, Melbourne, VIC, Australia, 1–30 November 2010. New York: Association for Computing Machinery.

22. Li Z, Zhang X, Shen H, et al. A semi-supervised framework for social spammer detection. In: Cao T, Lim EP, Zhou ZH, et al. (eds) Pacific-Asia conference on knowledge discovery and data mining. Cham: Springer, 2015, pp.177–188.

23. Yu J, Gao M, Rong W, et al. Hybrid attacks on model-based social recommender systems. Phys A Stat Mech Appl 2017; 483: 171–181.

24. Tan E, Guo L, Chen S, et al. UNIK: unsupervised social network spam detection. In: Proceedings of the 22nd ACM international conference on information & knowledge management, San Francisco, CA, 27 October–1 November 2013. New York: Association for Computing Machinery.

25. Ying X, Wu X and Barbará D. Spectrum based fraud detection in social networks. In: 2011 IEEE 27th international conference on data engineering, Hannover, 11–16 April 2011. New York: IEEE.

26. Zhongming H, Ke Y and Xusheng T. Sing the spectral analysis of weighted user relationship graphs to detect large-scale e-commerce navy groups. Chin J Comp 2017; 40(4): 939–954.

27. Wang F. Social computing: Concepts, contents, and methods. Int J Intell Syst 2004; 9(2):91–96.

28. Wagner DA, Dvorak D, Baroff L, et al. A control architecture for safe human-robotic interactions during lunar surface operations. Reston, VA: American Institute of Aeronautics and Astronautics, 2009.

29. Jabeur N, Karam R, Melchiori M, et al. A comprehensive reputation assessment framework for volunteered geographic information in crowdsensing applications. Pers Ubiquit Comput 2019; 23(5): 669–685.

30. Sahoh B and Choksurirawong A. A proof-of-concept and feasibility analysis of using social sensors in the context of causal machine learning-based emergency management. J Ambient Intell Humaniz Comput ahead of print 24 May 2021, DOI:10.1007/s12652-021-03317-3.

31. Luo C, Pang W, Wang Z, et al. Hete-CF: social-based collaborative filtering recommendation using heterogeneous relations. In: 2014 IEEE international conference on data mining, Shenzhen, China, 14–17 December 2014. New York: IEEE.

32. Zhao H, Yao Q, Li J, et al. Meta-graph based recommendation fusion over heterogeneous information networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, 13–17 August 2017. New York: Association for Computing Machinery.

33. Jiang T, Liu T, Ge T, et al. Encoding temporal information for time-aware link prediction. In: Proceedings of the 2016 conference on empirical methods in natural language processing, Austin, TX, 1–5 November 2016. New York: Association for Computing Machinery.

34. Dasgupta SS, Ray SN and Talukdar P. Hyte: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, 2018. Association for Computational Linguistics, https://aclanthology.org/D18-1225
35. Xiaoli T. *Research on dynamic knowledge graph representation algorithm and knowledge perception recommendation method*. MA Thesis, South China University of Technology, Guangzhou, China, 2020.

36. Wang EK, Chen CM, Yiu SM, et al. Incentive evolutionary game model for opportunistic social networks. *Future Gener Comput Syst* 2020; 102: 14–29.

37. Jayakrishnan R, Srinivasan R, Santhi C, et al. Advances in the application of the SWAT model for water resources management. *Hydrol Process* 2005; 19(3): 749–762.

38. Lai W, Huang X and Eades P. A framework of filtering, clustering and dynamic layout graphs for visualization. In: *ACSC’05 proceedings of the 28th Australasian conference on computer science*, Newcastle, NSW, Australia, 1 January 2005, vol. 38. Darlinghurst, NSW, Australia: Australian Computer Society.

39. Zhang J and Zhang J. Research on generation of urban rail transit network simulation environment based on NetLogo. *Technol Innov Appl* 2021; 3: 1–6.

40. Stonedahl F and Wilensky U. *NetLogo virus on a network model*. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, 2008.

41. Zhang F, Qu Y, Xu Y, et al. Graph embedding-based approach for detecting group shilling attacks in collaborative recommender systems. *Knowl Based Syst* 2020; 199: 105984.

42. Perozzi B, Al-Rfou R and Skiena S. DeepWalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, New York, 24–27 August 2014. New York: Association for Computing Machinery.

43. Zhou W, Koh Y, Wen J, et al. Detection of abnormal profiles on group attacks in recommender systems. In: *Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval*, Gold Coast, QLD, Australia, 6–11 July 2014. New York: Association for Computing Machinery.