# A Novel Shilling Attack Detection Method Based on T-Distribution over the Dynamic Time Intervals

Wanqiao Yuan[1], Yingyuan Xiao[1(✉)], Xu Jiao[1], Chenchen Sun[1], Wenguang Zheng[1], and Hongya Wang[2]

[1] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, Tianjin 300384, China
yyxiao@tjut.edu.cn

[2] College of Computer Science and Technology, Donghua University, Shanghai 201620, China

**Abstract.** The recommendation systems have become an important tool to solve the problem of information overload. However, the recommendation system is greatly fragile as it relies heavily on behavior data of users. It is very easy for a host of malicious merchants to inject shilling attacks in order to control the recommendation results. Some papers on shilling attack have proposed the detection methods, but they ignored experimental performance of injecting a small number of attacks and time overhead. To solve above issues, we propose a novel detection method of shilling attack based on T-distribution over dynamic time intervals. Firstly, we proposed Dynamic Time Intervals to divide the rating history of items into multiple time windows; secondly, the T-distribution is employed to calculate the similarity between windows, and the feature of T-distribution is obvious to detect small samples; thirdly, abnormal windows are identified by analyzing the T value, time difference and rating actions quantity of each window; fourthly, abnormal rating actions are detected by analyzing rating mean of abnormal windows. Extensive experiments are conducted. Comparing with similar shilling detection approaches, the experimental results demonstrate the effectiveness of the proposed method.

**Keywords:** Shilling attack · Dynamic Time Intervals · T-distribution · Recommendation system

## 1 Introduction

The recommendation systems [1, 2] have been extensively applied to various e-commerce websites to help users get rid of the trouble of information overload, but it heavily depends on user behavior data (e.g. ratings or clicks of users). Some malicious merchants [3] pour into a large amount of biased user rating profiles in the recommendation systems with the purpose of changing the recommending results for their own profits. These artificially biased user rating profiles are called attack profiles [4–7].


Y. Nah et al. (Eds.): DASFAA 2020 Workshops, LNCS 12115, pp. 225–240, 2020.
https://doi.org/10.1007/978-3-030-59413-8_19

The growing number of shilling attackers has been disturbing the recommendation results, which seriously affects the stability and accuracy of the recommendation systems. To solve the problem, Zhang et al. [8] construct a time series of rating to detect abnormal attack events according to the sample average and sample entropy in each window. But their fixed windows easily overflow attack events. Gao et al. [9, 10] propose two methods for detecting shilling attack. The former is based on fixed time intervals, and the latter adopts a dynamic partitioning method for time series. However, they neglect experimental performance of injecting a small amount of attack profiles.

To address the issue, this paper proposes a novel detection method of shilling attack based on T-distribution on Dynamic Time Intervals. Firstly, we analyze the difference between abnormal users and normal users based on shilling attack features [8–10] and present Dynamic Time Intervals algorithm; in addition, we use T-distribution to calculate the similarity between windows, which has significantly recognition capability of small sample; furthermore, we identify abnormal windows by analyzing the T value, time difference and rating actions quantity of each window; finally, abnormal rating actions are detected by comparing with rating mean of abnormal windows.

The main contributions of this paper are summarized as follows.

- We find that, analyzing statistical features, the incidence rate of injection attack events periods is tiny time intervals throughout entire lifecycle of the item.
- We proposed dynamic time intervals based on the existing shilling attack features to divide the rating history of items into multiple time windows.
- The T-distribution, whose feature is obvious to detect small samples, is employed to calculate the similarity between windows and we analyze the T value, time difference and rating actions quantity of each window to identify abnormal windows.
- Extensive experimental results demonstrate detection method of shilling attack based on T-distribution on dynamic time intervals is obvious to strengthen recommendation system.

The rest of this paper are organized as follows. Section 2 reviews the related work. Section 3 introduce T-distribution and define some notations and concepts used in this paper. Section 4 presents the proposed method. Section 5 evaluates our method through extensive experiments. Section 6 summarize this paper.

## 2 Related Work

### 2.1 Attack Profile and Attack Model

The shilling attack [3] main includes push and nuke attacks. The push attacks can enable the recommendation systems to make it easier to recommend target items, and the nuke attacks are to make it more difficult to recommend target items. To avoid being detected, many attack models have been introduced to disguise themselves. Several common attack models are proposed here. Attack profiles contain filler items, selected items, unrated items and target items [4]. Selected items are based on particular needs of the fake users, filler items are randomly selected items to disguise normal users, unrated items are those items with no ratings in the profiles, and target items are the items that attackers attempt

to promote or demote. The selected items and filler items strengthened the power of the shilling attack.

These basic elements in attack profiles make up different types of attack models. The three most common attack models are random attack, average attack, and popular attack. The random attack is to randomly select some items from the system for random ratings. Its advantage is that requires little cost but gains great benefits. The average attack is that items of random attack are assigned the corresponding average ratings. Popular attack is an extension of random attack, on this basis, the most popular item in the field is selected to have the highest rating. The model structure is shown in Table 1. In addition, in order to avoid detection, the attacker also adds obfuscated attack, which is more difficult to detect [5–7].

**Table 1.** Three classical attack models.

| Attack type | Push attack | Nuke attack |
|---|---|---|
| Random | $I_S = \Phi,\ I_F = r_{ran},\ I_t = r_{max}$ | $I_S = \Phi,\ I_F = I_{ran},\ I_t = r_{min}$ |
| Average | $I_S = \Phi,\ I_F = r_{avg},\ I_t = r_{max}$ | $I_S = \Phi,\ I_F = I_{ran},\ I_t = r_{min}$ |
| Bandwagon | $I_S = r_{max},\ I_F = r_{ran},\ I_t = r_{max}$ | $I_S = r_{max}, I_F = r_{ran}, I_t = r_{min}$ |

### 2.2 Shilling Attack Detection Methods

For robustness of the recommendation systems, many experts have researched some methods to detect shilling attacks. Chirita et al. [11] the earliest proposed the attribute RDMA (Rating Deviation from Mean Agreement), which characterizes the difference of user rating vector, and identified user profiles by average similarity and RDMA. Burke et al. [12] also defined some detection features about user profiles to classify normal and abnormal user. However, their classifications are not obvious. Later, more shilling attack detection methods have been proposed [13]. These methods are mainly divided into four categories [14], including supervised learning model, unsupervised learning model, semi-supervised learning model and statistical analysis method.

- **Supervised learning model:** Through training labeled data, classifier parameters are adjusted to identify attack profiles. Li et al. [15] proposed detection attacks method based on user selecting patterns. Firstly, the method extracted features through user selecting patterns; and then, the method constructed classification based on these features to attack profiles. Fan et al. [16] proposed the method through constructing Bayesian model and analyzing user potential features. However, these methods can only apply to existing specific attack models and need the suitable dataset.
- **Unsupervised learning model:** These models do not require training set. Attack profiles are identified by clustering based on user attribute. Yang et al. [17] proposed unsupervised method of detection attack profiles. The method includes three phases. Firstly, user undirected graph was constructed based on original data and the method

calculated similarity between users by graph mining method to reduce graph; Secondly, partly normal users were excluded by analyzing difference between users; Thirdly, target items were analyzed to detect attack profiles. Zhang et al. [18] proposed methods combination of hidden Markov and hierarchical clustering. Firstly, the method established user rating history model based on hidden Markov and analyzed user preferences to calculate suspicious degree; secondly, hierarchical clustering was employed to gain the attacker set. However, their experiments need strong preliminary knowledge and certain assumptions.

- **Semi-supervised learning model:** The learning methods combine supervised and unsupervised learning. They use a large number of unlabeled and labeled data for pattern recognition. Wu et al. [19] proposed hybrid attack detection method. Firstly, MC-Relief was employed to select shilling attack attribute; Secondly, Semi-supervised Naive Bayes model was used to identify abnormal users. Zhang et al. [20] proposed PSGD to identify abnormal user groups. PSGD uses labeled abnormal groups and unlabeled groups to detect attackers. However, their expensive calculational overhead and complexity are troublesome.
- **Statistical analysis method:** These methods analyze and survey obtained data and information and establish mathematical models to identify abnormal events. These kinds of methods have universality because of the view of items. Zhang et al. [8] proposed a time series detection method based on the sample mean and sample entropy according to the rapidity and purpose of the attacker. However, the time window of this method is fixed, and attack events easily overflow the window. Gao et al. [9, 10] proposed dynamic dividing for time series based on significant points, and then used $\chi^2$ to identify abnormal time window. However, $\chi^2$ heavily depended on the population variance, and experiment performance is not obvious.

## 3 Preliminaries

T-distribution was first proposed by William Sealy Gosset [21]. T-distribution is dramatically significant for statistics researches to tackle several practical problems through Interval Estimation and Hypothesis Testing, which has significant applications in the product life, fishery, agriculture domain. Compared with the standard normal distribution curve, T-distribution has a lower middle of the curve, and a higher tail of the curve. This feature of T-distribution is significant for us to detect samples injected with a small number of attacks.

In order to facilitate the description below, we have the following notations and definitions:

- $I$: the set of the entire items.
- $U$: the set *of* the entire users.
- $H'$: the set of rating actions. Specifically, each rating action $h \in H'$ is represented as $h = \langle h.i,\ h.u,\ h.r,\ h.t \rangle$, where $h.i \in I$ refers to an item, $h.u \in U$ denotes the user that gives $h.i$ a rating, $h.r$ is the rating, and $h.t$ is the time of rating.

**Definition 1 (Rating History).** For each item $i_k \in I$, a rating history $H_k$ of item $i_k$ is a sequence of rating records formatted as $H_k = h_1 \overset{h_2.t-h_1.t}{\rightarrow} h_2 \overset{h_3.t-h_2.t}{\rightarrow} \ldots \overset{h_n.t-h_{n-1}.t}{\rightarrow} h_n$, where $h_j \in H'$, $0 < j < n$, $h_{j+1}.t > h_j.t$, and $\nexists h_{j'} \in H'$, $s.t. h_j.t < h_{j'}.t < h_{j+1}.t$.

**Definition 2 (Item-ratings Time Gaps Series).** For each rating history $H_k$, an item-ratings time gaps series is $IRTGS_k = \{(midT_1, gap_1), \ldots, (midT_{n-1}, gap_{n-1})\}$, which corresponds to a time middle series $midT = \{midT_1, midT_2, \ldots, midT_{n-1}\}$, and a time gap series $gap = \{gap_1, gap_2, \ldots, gap_{n-1}\}$. $midT_j = \big(h_{j+1}.t - h_j.t\big)/2$, refers to the median of the adjacent timestamps between ratings, and $gap_j = h_{j+1}.t - h_j.t$, refers to the adjacent timestamps gap between ratings.

**Definition 3 (Time Window).** Suppose $H_k$ is divided into m time intervals, time window series of $H_k$ is $W_k = \{w_1, w_2, \ldots, w_m\}$, each time window $w_x \in W_k$ corresponds to all $h \in H_k$ of the $x$ th time interval. Besides, $w_1 \cup w_2 \cup \ldots \cup w_m = H_k$, $w_1 \cap w_2 \cap \ldots \cap w_m = \emptyset$.

## 4 The Proposed Method

In order to gain more benefits, attackers inject a number of high (push attacks) and low ratings (nuke attacks) into the target item, in order to the recommendation system easier or harder to recommend the item [8]. Therefore, a common characteristic is that a number of abnormal rating actions will be injected into the target item in a short time interval [10]. That is to say, high rating quantity or low rating quantity will be significantly increased in the attack time interval. According to the shilling attack characteristics mentioned above, we propose TDTI (T-distribution to detect abnormal rating actions on the Dynamic Time Intervals) method.

Our method is mainly based on two features of the attack as preconditions:

1) Attackers inject a number of abnormal rating actions in a short time in order to save costs, and attacks must be very dense [8–10].
2) The incidence rate of injection periods must be tiny time intervals throughout entire lifecycle of the item.

The general idea of the proposed method is as follows: Firstly, we divide the rating history of item into time windows series by DTI (Dynamic Time Intervals); Secondly, the T-distribution is employed to calculate the similarity between windows. Thirdly, Abnormal windows are identified by analyzing the T value, time difference and rating actions quantity of each window; To be more precise, we calculate the ratings mean of abnormal windows, exclude the rating actions less than the mean (push) or the rating actions more than the mean (nuke), and the rest are the abnormal rating actions in attack profiles.

In this section, we will introduce our TDTI method in detail, divided into two modules:

1) We design the DTI partitioning rating history of item into time window series.
2) T-distribution identifies abnormal rating actions.

### 4.1 Designing the DTI Partitioning Rating History

We aim to partition the rating history of item into time window series and ensure that the abnormal rating actions are divided into one time window in the same period. Based on feature 1 (a number of abnormal rating actions are injected system in a short time), the specific steps are as follows:

1) According to the rating history of an item, we calculate the corresponding IRTGS (Item-ratings Time Gaps Series).
2) Divide IRTGS into two subsequences according to the *gap* maximum in IRTGS.
3) Repeat step2 for the two subsequences again, and record the *midT* values of *gap* maximum in IRTGS as mark points, until the difference between maximum and minimum of *gap* is less than $\alpha$ ($\alpha$ denotes the threshold of difference between *gap* maximum and *gap* minimum in IRTGS and controls the end circulation of DTI). Because of attacks promptness, the $\alpha$ cannot be too large, otherwise the normal rating actions will be integrated into the abnormal window. Meanwhile, the $\alpha$ also cannot be too small, otherwise the abnormal rating actions will be divided into multiple windows. $\alpha$ is related to the rating characteristic and obtained by a large number of experiments.
4) Divide the rating history of the item based on the mark points recorded by step3.

**Algorithm 1**: DTI$(H, \alpha)$

**Input** : $H$ is a Rating history; $\alpha$ is the threshold of difference between $gap$ maximum and $gap$ minimum in $IRTGS$
**Output:** Time window series: $W\{w_1, w_2, \ldots, w_m\}$, $w_j \subset H$, $0 < j \leq m$

Initialize $gap$, $midT$, $W$, $w$ to four empty lists
**for** $i = 0, \ldots, n-1$ **do**
  $gap_i \leftarrow h_{i+1}.t - h_i.t$, $midT_i \leftarrow (h_{i+1}.t + h_i.t)/2$
  Add $gap_i$ in $gap$, Add $midT_i$ in $midT$
**end**
$S \leftarrow gap$, $T \leftarrow midT$
**PRH**$(S, T, \alpha)$
Add 0, $h_n.t + 1$ in $mark$
Sort $mark$ in ascend order
$l \leftarrow$ obtaining the length of $mark$
**for** $i = 0, \ldots, l-1$ **do**
  **for** $h_j \in H$ **do**
    **if** $mark_i < h_j.\mathrm{t} \leq mark_{i+1}$ **then**
      Add $h_j$ in $w$
  **end**
  Add $w$ in $W$
**end**
**return** $W$

Algorithm 1 describes the process of DTI, which divides the rating history into time window series. Algorithm 2 named PRH ($S$, $T$, $\alpha$) is a sub-algorithm of algorithm 1 and is called by Algorithm 1 at line 7. In the above Algorithm 2, the function GetMax ($gap$) is responsible for obtaining the maximum element of $gap$; the function GetMin ($gap$) returns the minimum element of $gap$; the function Extract($L, f, e$) extracts the elements of subscripts from $f$ to $e$ in the $L$, where $L$ denotes a list.

In order to illustrate our method more clearly, we randomly selected an item from MovieLens 100k dataset. The IRTGS of the item original data is shown in Fig. 1(a). We injected 50 attack profiles to show clear result, where a host of abnormal rating actions are shown in the red circle of Fig. 1(b). Figure 1(c) shows the mark points are recorded by DTI. We record the horizontal coordinates corresponding to the red vertical line as mark point set in order to divide the rating history. Figure 1(d) shows the cutting of rating history based on the mark points. According to mark points set, the rating history is divided into time window series, and attacks are divided into a time window, which contains tiny normal rating actions.

**Algorithm 2**: PRH($S, T, \alpha$)

**Input :** $S$ is a $gap$;
$T$ denotes a $midT$;
$\alpha$ is the threshold of difference between $gap$ maximum and $gap$ minimum in $IRTGS$

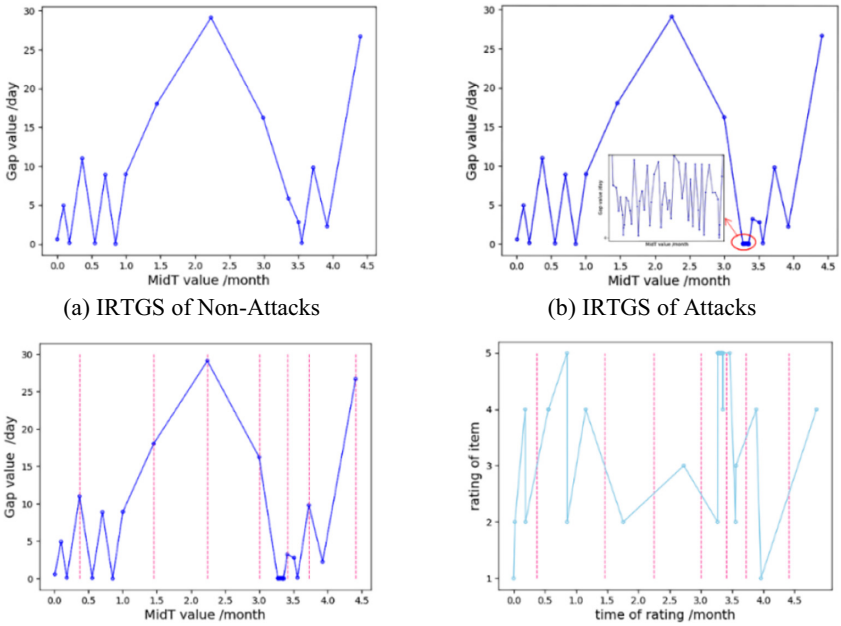
**Output:** the list of mark points: $mark$

1: Initialize $mark$ to an empty list
2: $max \leftarrow$ GetMax($gap$), $min \leftarrow$ GetMin($gap$)
3: $m \leftarrow$ obtaining the subscript of $max$
4: $n \leftarrow$ obtaining the length of $gap$
5: **if** $max - min > \alpha$ **then**
6: $\quad sl \leftarrow$ Extract($gap, 0, m$), $sr \leftarrow$ Extract($gap, m+1, n$)
7: $\quad tl \leftarrow$ Extract($midT, 0, m$), $tr \leftarrow$ Extract($midT, m+1, n$)
8: $\quad$ Add $midT[m]$ in $mark$
9: $\quad S \leftarrow sl, T \leftarrow tl$
10: $\quad$ **PRH**($S, T, \alpha$)
11: $\quad S \leftarrow sr, T \leftarrow tr$
12: $\quad$ **PRH**(S, $T, \alpha$)
13: **end**
14: **return** $mark$

### 4.2 Identifying Abnormal Rating Actions Based on T-Distribution

After the abnormal rating actions are detected by the following steps:

1) Calculate the T value between time windows to determine whether each window is similar with the properties of others.
2) Compare the relationship between each T value and its corresponding boundary value, which is shown in Table 2. If the T value beyond the boundary value is




(a) IRTGS of Non-Attacks (b) IRTGS of Attacks

(c) Recording mark points (d) Cutting rating history

**Fig. 1.** DTI partitioning the rating history (Color figure online)

denoted as 1 (dissimilar), and if it is denoted as 0 within the boundary value range (similar). Sequentially, we obtain time window-time window Symmetric Matrix, where its values only have 0 or 1, so in this paper, we call it 0-1 Matrix.

3) Analyze the 0-1 Matrix, and we select the windows whose 1 value quantities are more than the mean of all windows 1 value quantities in rating history to get suspicious window set. Based on the shilling attack features, abnormal window must be minority, so the case that the windows get 1 value is more than the normal windows. Thus, we can easily get suspicious windows by comparing the number of 1 value in each window.
4) Compare each suspicious window obtained in step 3. If time gap of the window is more than the mean time gaps of all windows in the rating history and the rating actions number of the window are less than the mean rating actions quantities of all windows in the rating history, the window is identified as abnormal window.

**Table 2.** The degree of freedom corresponds to boundary value in the 95% confidence level

| Degree of freedom | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Boundary | 12.71 | 4.303 | 3.182 | 2.776 | 2.571 | 2.447 | 2.365 | 2.306 |

5) Calculate the ratings mean of abnormal windows, exclude the rating actions less than the mean (push) or the rating actions more than the mean (nuke), and the rest are the abnormal rating actions.

The T value is calculated by modified two-sample T-distribution hypothesis testing method [21]. The formula applied to the procedures is as follows:

$$\bar{x}_i^* = \frac{1}{m}\sum\nolimits_{k=1}^{m} x_{ik} (1 \le m \le 5) \tag{1}$$

$$\bar{x}_j^* = \frac{1}{n}\sum\nolimits_{k=1}^{n} x_{jk} \ (1 \le n \le 5) \tag{2}$$

The formula (1) and (2) are functions of rating action means in a time window. $\bar{x}_i^*$ is the modified mean of the $i$th time window and $\bar{x}_j^*$ represents modified mean of the $j$th time window. Since the score range is from 1 to 5 in this context, the max rating difference is 5, leading to that the difference between the rating means of time windows can only be within the range of 5 and the calculation of T value is related to the difference, which is no obvious effect. In order to increase the difference, we improve the function through the $m$ in formula (1) and $n$ in formula (2), refer to the number of rating kinds rather than the number of ratings. $x_{i_k}$ in formula (1) refers to the $k$th rating in the $i$ window and $x_{j_k}$ in formula (2) is the $k$ th rating in the $j$ window.

$$\bar{x}_i = \frac{m}{g}\bar{x}_i^* \tag{3}$$

$$\bar{x}_j = \frac{n}{h}\bar{x}_j^* \tag{4}$$

The formula (3) is the conversion function of $\bar{x}_i^*$ and $\bar{x}_i$, and formula (4) shows relationship between $\bar{x}_j^*$ and $\bar{x}_j$, where $g$ and $h$ refers to the number of rating in the $i$th and $j$th time windows.

$$s_i^2 = \frac{1}{g}\sum\nolimits_{k=1}^{g} \left(x_{i_k} - \bar{x}_i\right)^2 \tag{5}$$

$$s_j^2 = \frac{1}{h}\sum\nolimits_{k=1}^{h} \left(x_{j_k} - \bar{x}_j\right)^2 \tag{6}$$

The formula (5) and (6) are variances of a time window.

$$T_{ij} = \frac{\bar{x}_i^* - \bar{x}_j^* - (a_0 - a_i)}{\sqrt{gs_i^2 + hs_j^2}} \sqrt{\frac{mn(m+n-2)}{m+n}} \sim t(m+n-2) \tag{7}$$

T value can be expressed as formula (7), where $a_0$ refers to the rating mean in entire rating history, and $a_i$ refers to the rating mean of rating history excluding the $i$ th window. $m + n - 2$ is the degree of freedom about T value, which corresponds to the boundary value, shown in Table 2. If the degree of freedom is 0, it means that the rating type of both windows is 1, in this case, we default to similar attributes of the two windows.

Subsequently, we take the item selected in the Sect. 4.1 as an example. Table 3 shows the value matrix $T_{ij}$ through T-distribution procedure obtained. And then, compare each T value with the corresponding boundary value to obtain the matrix 0-1 in Table 4. The third window far exceeds the boundary value and third window is suspicious window.

**Table 3.** $T_{ij}$ value matrix

| $T_{ij}$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 2.534 | **53.013** | 0.157 |
| 2 | 2.683 | 0 | **50.833** | 2.43 |
| 3 | **52.195** | **49.962** | 0 | **52.132** |
| 4 | 0.307 | 2.348 | **53.008** | 0 |

**Table 4.** The matrix 0-1

| $T_{ij}$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 0 | **1** | 0 |
| 2 | 0 | 0 | **1** | 0 |
| 3 | **1** | **1** | 0 | **1** |
| 4 | 0 | 0 | **1** | 0 |

# 5 Experimental Evaluation

## 5.1 Datasets and Evaluation Metrics

In the experiments, we use public available dataset MovieLens 100 k, which is available on the GroupLens web site and GroupLens Research has collected. The dataset contains 1682 items, 100,000 ratings that were evaluated by 943 users. Each user makes at least 20 ratings, and rating has five ranks from 1 to 5. Statistically analyzing rating time intervals of 1682 items, the mean is 20,029 s and the top 100 is 176.8 s. To facilitate our experiment, we omitted the items with less than 10 rating quantities, because the number of rating was too small to reflect the authenticity of the experiment, and the dataset after filtering is shown in Table 5.

**Table 5.** Mainly information of the dataset in the experiment

| Dataset | Item | User | Rating |
|---|---|---|---|
| MovieLens 100K | 1152 | 943 | 97,953 |

We used two indicators [8–10] to evaluate the experimental results: The detection rate in formula (8) is defined as the number of detected attack events (abnormal rating actions) divided by the number of the total attack events. The false alarm rate in formula (9) is defined as the number of normal events (rating actions) that are recognized as attack events divided by the number of all normal events. Here, an event means that a rating action.

$$DetectionRate = \frac{Detected\ Attack\ Events\ Quantity}{Total\ Attack\ Events\ Quantity} \tag{8}$$

$$FalseAlarmRate = \frac{False\ Quantity}{Normal\ Quantity} \tag{9}$$

## 5.2 Comparative Approaches

At present, there are many methods [11–20] for attacks detection, but we put forward methods based on time and item view, so we compare it with similar methods [8–10] in

this paper. The comparison of relevant methods has Gao et al. [9, 10] and Zhang et al. [8] in the simulation experiment on the MovieLens 100k dataset. The following is brief introduction to three methods of them:

- **TS** [8]: The method constructs a time series of rating for an item to compute the features of sample average and sample in each window, and based on duration of attacks, observing the time series of two features can expose attack events, in which sample average is represented by **TS-Ave** and sample entropy is represented by **TS-Ent**.
- **TIC** [9]: The distributions of ratings in diverse time intervals are compared to detect anomaly intervals through the calculation of chi square distribution ($\chi^2$).
- **DP** [10]: According to two features of shilling attacks (the rating of item is always maximum and minimum as well as it takes a very short time to inject large attacks), firstly, this method dynamically divides item-rating time series via significant points, and then, identify abnormal intervals through chi square distribution.

### 5.3 Experiment Performance

To estimate the performance of our method in different attack quantities, filler sizes, item types and attack models, we use MovieLens dataset to take four item types: fad, fashion, style, and scallop; three attack models: average attack, random attack, and bandwagon attack. Assume that original users are normal users, and attack profiles are those generated from attack models. Based on the features of shilling attacks in Sect. 4, the time gap between attacks in the same period within 1000 s, and the time point of each abnormal rating action is a random point within the attack period. 50 items are randomly selected from 1152 items. The number of attacks is 10, 20, 30, 40, 50 and filler size is set to 1%, 3%, 5%, 7%, 10%, which divided into two categories experience: push attack and nuke attack. We repeat the experiments twenty times, and calculate the detection rate and false alarm according to the mean of the experiments.

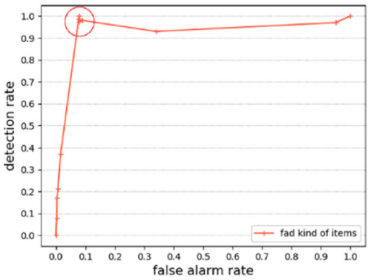
**Threshold Setting in Experiments.** $\alpha$ is the threshold that controls the end circulation of DTI. The impact of $\alpha$ value is related to the item types. Based on the rating features of items in [9], 1,152 items are divided into four types as follows:

- **Fad:** There are 75 items in total. Feature is relatively dense rating time and fewer rating quantities.
- **Fashion**: There are 41 items in total. Feature is relatively dense rating time and larger rating quantities.
- **Style**: There are 250 items in total. Feature is scattered rating time and fewer rating quantities.
- **Scallop**: There are 786 items in total. Feature is continuous topic and larger rating quantities.

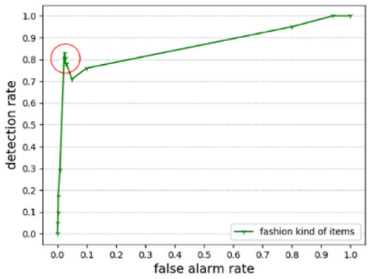
We evaluate the impact $\alpha$ on the performance IDTI in four kinds of items. Here, 50 push attacks (filler size is 0) are injected into the system and testing.

Figure 2 shows the effects of α on four item types. We can see that one $\alpha$ value corresponds to a detection rate and a false detection rate, where $\alpha$ value increases from

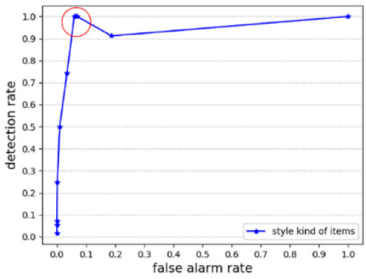
0, and unit of $\alpha$ is hour (h). In the experiment, we expect the detection rate to be as large as possible and the false alarm rate as small as possible. The most ideal state is that $\alpha$ exists one value, which makes the detection rate reaches 1, and the false alarm rate is 0. We mark ranges of $\alpha$ optimal values in red and structure Table 6. By integrating the intersecting parts of the four item types, we test the best experiment results for four kinds of items when the α value is equal to 0.389 h, as shown in Table 6. In the following experiments, we set α value for DTI as 0.389 h.



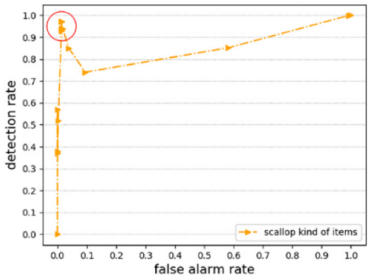

(a) Impact of α on fad items




(b) Impact of α on fashion items




(c) Impact of α on style items




(d) Impact of α on scallop items

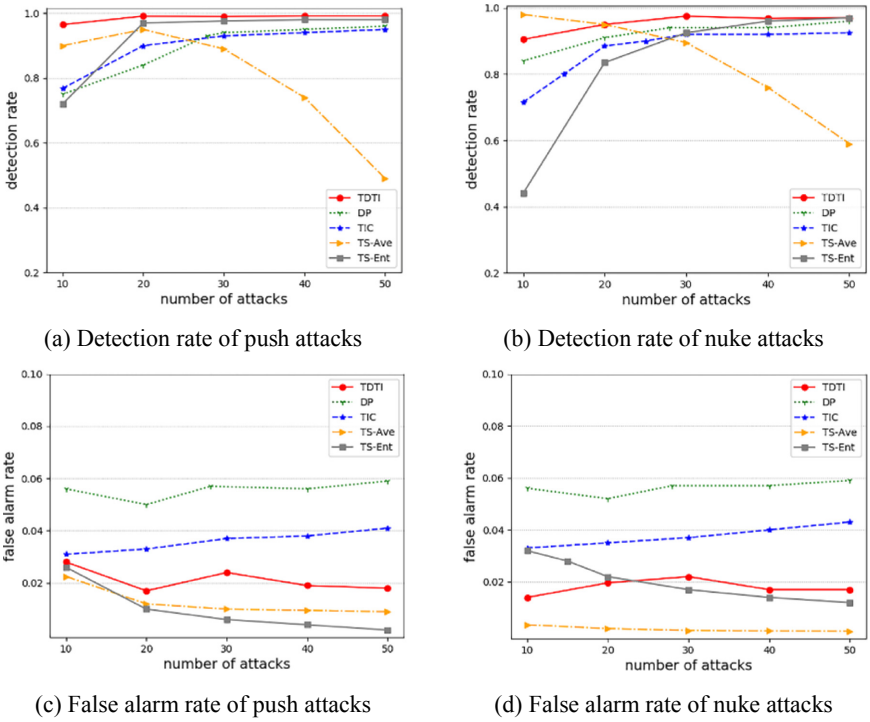
**Fig. 2.** Impact of $\alpha$ on the detection rate and false alarm rate in four kinds of items

**Table 6.** $\alpha$ optimal value in different kinds of item ($\alpha$ unit: h)

| Item types | α | Detection rate | False alarm rate |
|---|---|---|---|
| Fad | 0.278–0.389 | 1 | 0.0708–0.072 |
| Fashion | 0.389–0.444 | 0.82–0.84 | 0.023–0.025 |
| Style | 0.278–2.78 | 1 | 0.062–0.063 |
| Scallop | 0.278–0.389 | 0.96–0.97 | 0.015 |
| Synthesize | 0.389 | 0.979 | 0.028 |

**Comparison with Other Approaches.** In order to demonstrate the performance improvement of TDTI, we compare TDTI with three methods [8–10] based on the

simulation experiments on the MovieLens 100 k dataset. Since the detection methods [8–10] are immune to diverse attack model, the impact of the filler size and attack model isn't considered here. The experiment results are shown in Fig. 3.




(a) Detection rate of push attacks

(b) Detection rate of nuke attacks

(c) False alarm rate of push attacks

(d) False alarm rate of nuke attacks

**Fig. 3.** Detection rates and false alarm rate comparison with DP, TIC and TS

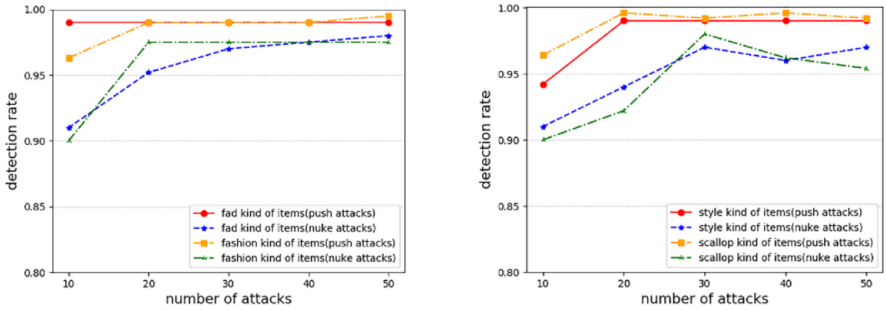
- **Detection rate of push attacks:** TDTI has the best experiment performance. We take the Fig. 3(a) as an example, compared with TS-Ent, DP, TIC and TS-Ave, the performance improvements by TDTI are 0.245, 0.215, 0.097 and 0.065 when attack quantity is 10. And when more than 20, TDTI maintains steady state, ranging from 0.99 to 0.992. It is obvious that TDTI has more accurate detection results, because DTI algorithm more accurately divide the abnormal rating actions into a time window, and we employ T-distribution to identify abnormal window. The feature of T-distribution is that small sample has obvious detection effect.
- **Detection rate of nuke attacks:** The experiment performance of TDTI is stable and higher. As shown in Fig. 3(b), when Attack quantity at 10, TDTI is 0.905, and TS-Ave is 0.98, which is 0.075 higher than ours. However, when the attack quantity is 50, TS-Ave is 0.59, and TDTI is 0.97, where we improve 0.38. Because TS-Ave is static window, as the number of attacks increases, it is easy for the attack events to overflow window.

- **False alarm rate:** TDTI is better than TIC and DP. As shown in Fig. 3(c) and Fig. 3(d), TDTI is higher than TS. However, TS has lower detection rate as result of boundedness of static window. TDTI is lower than TIC, DP and relatively stable in the below 0.028. We can see TDTI is more satisfactory for attack detection, because the abnormal windows by TDTI have less normal rating actions.

**Comparison of Between Rating Features and Between Attack Models.** We detect the effect of TDTI in the four item types: fad, fashion, style and scallop, and three kinds of attack models: random attack, average attack, and bandwagon attack.

We test TDTI in the four kinds of items. Experiment conditions: average attack is injected into the system, and filler size is set to 5%.

Figure 4 shows the detection rate of IDTI on the different item types. IDTI has satisfactory performance on four item types. The detection rate of push attacks is generally higher than nuke attacks. When the attack quantity is 10, the difference between push and nuke is the largest, where the detection rate of fad item injecting push attacks is 0.99 and nuke is 0.91. The best experiment effect of four types is fad, whose push attack gets to 0.99 and nuke attack is up to 0.98. Because the fad rating features are dense rating time and less quantity, which makes it easier for our method to divide abnormal rating actions into a time window, and T-distribution has obvious detection effect on less quantity windows.

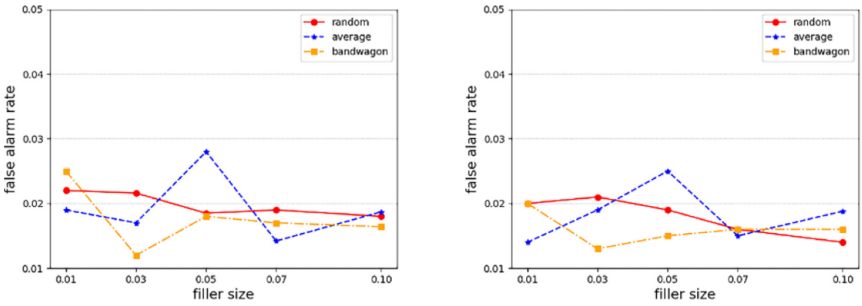



(a) Fad and fashion kinds of items (b) Style and scallop kinds of items

**Fig. 4.** Impact of attack size on four kinds of items

We test the impact of the filling size on diverse attack models. Experiment conditions: the attack quantity is set to 10, and the selected item in the Bandwagon attack is 50th because the item always has the topic degree and its rating quantity is largest in the 1152 items.

Figure 5 shows the false alarm rate of TDTI on the diverse attack models. TDTI is unaffected by the attack models and has stable results. The false alarm rate of three models is below 0.028, and the lowest is 0.012. Because TDTI is based on the view of item and consider time factor, our method is immune to different attack models.




(a) Push attacks (b) Nuke attacks

**Fig. 5.** Impact of filler size on attack models

## 6 Conclusions

Based on the characteristics of the shilling attacks, this paper proposes the DTI algorithm to divide the rating history into multiple time windows, and then the T-distribution is employed to calculate the similarity between windows. Finally, the proposed method identifies shilling attacks by analyzing T value, the time difference and rating actions quantity of each window.

Compared to other methods, the detection rate of our method achieves the desired value, but our false alarm rate still needs to be improved. Therefore, we will focus on how to further decline the false alarm rate in our future work.

**Acknowledgment.** This work is supported by the National Nature Science Foundation of China (91646117, 61702368) and Natural Science Foundation of Tianjin (17JCYBJC15200, 18JCQNJC00700).

## References

1. Yuan, J., Jin, Y., Liu, W., Wang, X.: Attention-based neural tag recommendation. In: Li, G., Yang, J., Gama, J., Natwichai, J., Tong, Y. (eds.) DASFAA 2019. LNCS, vol. 11447, pp. 350–365. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-18579-4_21
2. Xu, K., Cai, Y., Min, H., Chen, J.: Top-N trustee recommendation with binary user trust feedback. In: Liu, C., Zou, L., Li, J. (eds.) DASFAA 2018. LNCS, vol. 10829, pp. 269–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91455-8_23
3. Yi, H., Niu, Z., Zhang, F., Li, X., Wang, Y.: Robust recommendation algorithm based on kernel principal component analysis and fuzzy C-means clustering. Wuhan Univ. J. Nat. Sci. **23**(2), 111–119 (2018). https://doi.org/10.1007/s11859-018-1301-6
4. Kaur, P., Goel, S.: Shilling attack models in recommender system. In: 2016 International Conference on Inventive Computation Technologies. IEEE (2016)
5. Oh, H., Kim, S., Park, S., Zhou, M.: Can you trust online ratings? A mutual reinforcement model for trustworthy online rating systems. IEEE Trans. Syst. Man Cybern.: Syst. **45**(12), 1564–1576 (2015)
6. Cheng, Z., Hurley, N.: Effective diverse and obfuscated attacks on model-based recommender systems. In: RecSys, New York, NY, USA, pp. 141–148 (2009)

7. Yu, J., Gao, M., Rong, W., Li, W., Xiong, Q., Wen, J.: Hybrid attacks on model-based social recommender systems. Physica A **483**(2017), 171–181 (2017)
8. Zhang, S., Chakrabarti, A., Ford, J., Makedon, F.: Attack detection in time series for recommender systems. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, Pennsylvania, USA, pp. 809–814, August 2006
9. Gao, M., Yuan, Q., Ling, B., Xiong, Q.: Detection of abnormal item based on time intervals for recommender systems. Sci. World J. **2014**, 845–897 (2014)
10. Gao, M., Tian, R., Wen, J., Xiong, Q., Ling, B., Yang, L.: Item anomaly detection based on dynamic partition for time series in recommender systems. PLoS ONE **10**(8), 135–155 (2015)
11. Chirita, P., Nejdl, W., Zamfir, C.: Preventing shilling attacks in online recommender systems. In: Seventh ACM International Workshop on Web Information and Data Management, Bremen, Germany, pp. 67–74 (2005)
12. Burke, R., Mobasher, B., Williams, C., Bhaumik, R.: Classification features for attack detection in collaborative recommender systems. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, Pennsylvania, USA, pp. 542–547 (2006)
13. Gao, M., Li, X., Rong, W., Wen, J., Xiong, Q.: The performance of location aware shilling attacks in web service recommendation. Int. J. Web Serv. Res. **14**(3), 53–66 (2017)
14. Wang, Y., Qian, L., Li, F., Zhang, L.: A comparative study on Shilling detection methods for trustworthy recommendations. J. Syst. Sci. Syst. Eng. **27**(4), 458–478 (2018). https://doi.org/10.1007/s11518-018-5374-8
15. Li, W., Gao, M., Li, H., Zeng, J., Xiong, Q.: Shilling attack detection in recommender systems via selecting patterns analysis. IEICE Trans. Inf. Syst. **E99.D**(10), 2600–2611 (2016)
16. Fan, Y., Gao, M., Yu, J., Song, Y., Wang, X.: Detection of Shilling attack based on bayesian model and user embedding. In: International Conference on Tools with Artificial Intelligence, pp. 639–646. IEEE (2018)
17. Yang, Z., Cai, Z., Guan, X.: Estimating user behavior toward detecting anomalous ratings in rating systems. Knowl.-Based Syst. **111**(2016), 144–158 (2016)
18. Zhang, F., Zhang, Z., Zhang, P., Wang, S.: UD-HMM: an unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering. Knowl.-Based Syst. **148**(2018), 146–166 (2018)
19. Wu, Z., Wu, J., Cao, J., Tao, D.: Hysad: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In: Proceedings of the 18th International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 985–993 (2012)
20. Zhang, L., Wu, Z., Cao, J.: Detecting spammer groups from product reviews: a partially supervised learning model. IEEE Access **6**(2018), 2559–2567 (2018)
21. Shen, X., Wu, R.: Discussion on t-distribution and its application. Stat. Appl. **4**(4), 319–334 (2015)