ISSN 2007-9737

$Ward_pHMM$: A Shilling Attack Detection Technique Using $Ward_p$ Method and Hidden Markov Model

Keya Chowdhury[1], Abhishek Majumder[1], Joy Lal Sarkar[1],
Sukanta Chakraborty[1], Sudipta Roy[2]

[1] Tripura University,
Department of Computer Science & Engineering,
India

[2] Assam University,
Department of Computer Science and Engineering,
India

{ckeya28,abhi2012, joylalsarkar, visitsukanta, sudipta.it}@gmail.com

Abstract. Collaborative Filtering Recommender Systems (CFRSs) are widely employed in several applications because of its satisfying performance in the customized recommendation. Recent studies show that CFRSs are at risk of shilling attacks where attackers inject shilling profiles into the system. Malicious user injected ratings not only severely impact genuineness of recommendations but also user's trustworthiness within recommendation systems. Existing unsupervised clustering technique uses Ward method, which is an iterative method of low scalability. For addressing this issue, in this work an unsupervised SA detection technique named $Ward_pHMM$ has been proposed. It uses $Ward_p$ and Hidden Markov Model (HMM). In this proposed method HMM is used to measure difference of user's rating behavior. It generates User Suspicious Degree (USD) of each user by analyzing user's Suspicious Degree Range of Items (SDRI) and User's Matching Degree (UMD). Then $Ward_p$ method is applied to merge users based on USD and to acquire group of Attack Users (AUs). For performance analysis of the proposed technique, Amazon-ratings sample dataset was used. The performance comparison shows that proposed $Ward_pHMM$ technique outperforms baseline technique with respect to precision, recall and F1-score.

Keywords. Profile injection attack, Hidden Markov model, user matching degree, user suspicious degree, $Ward_p$ method.

1 Introduction

The Collaborative Filtering (CF) based recommendation system is developed for filtering out irrelevant resources [1]. In the recommender systems, collaborative Filtering Recommender System (CFRS) is considered as a popular and productive technique. CFRS work on the principle that identical users have identical tastes.

For recommender frameworks CF has been the source of vulnerability, due to its open and interactive nature. Usually a user-based CF algorithm makes recommendation by searching out similar user patterns, which are obtained from the preferences of numerous totally non-identical people [1]. If profiles contain biased information, they may be thought as real users and eventually lead to biased recommendations. Therefore, relevant data is buried under a good deal of irrelevant information. The filtering procedure of CF depends on the profiles of different clients, so the usage of Collaborative Filtering in recommender framework is vulnerable to Shilling Attacks (SAs). For their own benefits attackers use biased rating profiles [2]. SA can be classified as, Push Attack (PA) and Nuke Attack (NA). The PA has been used for promoting someone's own items by giving maximum ratings and the NA has been used for demoting product of someone's rivals by giving minimum ratings [3].

Item	i_1^S	...	i_n^S	i_1^F	...	i_k^F	$i_1^\emptyset$	...	$i_m^\emptyset$	i^T
Rating	$\varphi(i_1^S)$	...	$\varphi(i_n^S)$	$\varphi(i_1^F)$	...	$\varphi(i_k^F)$	Null	Null	Null	$\gamma(i^T)$

Fig. 1. Attack profile's generalized form [3]

The increasingly prevalent shilling attackers apply biased rating profiles to the frameworks to control items recommendation. It not only brings down the recommending accuracy but also harms the reliability of intermediated exchange stages and members. For example, in any online site if any retailer and customer both give high positive ratings for the product then that customer can be recognized as genuine purchaser.

Otherwise, if some purchaser only gives low negative ratings to every item then he or she can be recognized as fake user or an attacker who is performing SAs. But in some cases, the genuine purchaser also gives moderate ratings, then it become difficult to recognize without proper investigation. Thus, in RS detection of SAs is a major challenge. SA detection has drawn the attention of lot of researchers. Many techniques have been proposed by different researchers for SA detection. A brief study on different existing SA detection techniques has been presented in section 2. Out of these, UD-HMM [4] is a promising SA detection technique. In detecting most types of SAs UD-HMM performed very well. But in case of UD-HMM, the detection performance is not well when obfuscated attack takes place. The obfuscated attack uses Standard Average Attack (SAA). Every filter item is selected from top x% of most popular items with equal probability. For addressing this problem, an unsupervised SA detection method named Ward$_p$HMM has been proposed in this paper. It uses Ward$_p$ [5] method and Hidden Markov Model (HMM).

Proposed method concentrates on detecting the genuine users and the attack users.

Contributions of the paper are:

I. A shilling attack detection technique, namely Ward$_p$HMM, has been proposed. Here, HMM is used to calculate the User's Preference Sequence (UPS).

II. The proposed technique uses Ward$_p$ clustering method for obtaining the group of attack users. This is because Ward$_p$ method uses feature weights, which produces results that are superior to those produced by Ward method.

III. To analyze performance of Ward$_p$HMM technique, extensive experiments have been carried out and compared with UD-HMM technique.

The paper is arranged as follows. Section 2 discusses background and related work. Proposed Shilling Attack (SA) detection technique is presented in section 3. Section 4 discusses performance analysis and comparison of both UD-HMM and Ward$_p$HMM. Finally, in section 5 conclusions and future work has been discussed.

2 Background and Related Work

The section discussed profile injection attack in brief. It also presents the existing profile injection attack detection techniques.

2.1 Profile Injection Attack

An attack against a CFRS requires a group of Attack Profiles (APs) which attacker injects. Attackers may inject shilling profiles with highest or lowest rating to the target items to be promoted or demoted respectively [6]. An AP contains lot of biased ratings [7].

Figure 1 illustrates an attack profile's generalized form [3]. i^S represents the selected item which is mainly used for characterizing the attack and $\varphi(i^S)$ represents the selected item's rating. i^F represents the filler item which normally rate items randomly to be looked like normal profile.

It is difficult to detect. $\varphi(i^F)$ represents the ratings of filler item. i^{φ} represents the unrated items and $\varphi(i^{\emptyset})$ represents the ratings of unrated item. i^T represents the target items, which gets highest ratings for promotion or lowest ratings for demotion by the attackers. $\gamma(i^T)$ represents the ratings of target item.

2.2 Shilling or Profile Injection Attack Detection techniques

The detection of profile injection attacks in CFRSs has attracted huge attention from research community [6, 7]. Over past few decades, many shilling attack detection techniques were proposed. There are three types of detection techniques: (i) unsupervised detection method, (ii) supervised detection method and (iii) semi-supervised detection method.

The supervised classification techniques need labeled training classifiers and sample information. They can appropriately detect attacks of known kind. Example of supervised classification technique is RAdaBoost [8]. Yang et al. proposed this detection technique. It detects different types of attacks based on 18 statistical features of malicious users.

However, due to huge amount of features this method is computationally intensive. Burke et al. [9] proposed a length variance (lengthVar), which is a generic attribute. For a user the number of ratings is represented by length. This attribute measures the variation in length for some provided user from average length. For finding fake profiles that are correlated with items' subset, a variance-adjusted H_V value was proposed by Bryan et al. [10], whose objective is that fake profiles will have a maximum H_v value.

Previous knowledge of attacks is normally not required in unsupervised clustering techniques. This include candidate attack users to be labeled and range of attack profiles injected. Utilizing multidimensional scaling a hybrid two phase detection method was proposed by Lee et al. [11]. It is an unsupervised clustering technique. The detection performance of this method is very good with high filler size while detecting the Average Attack (AA). With small filler size, the detection performance is not well when detecting Random Attack (RA).

Zhang et al. proposed a HMM and Hierarchical Clustering (HC) based technique named UD-HMM [4] to identify the profile injection attacks in CFRSs. This method first calculates the User's Suspicious Degree (USD) by utilizing the HMM and then uses the HC to detect the group of attack users.

The method outperforms the baseline methods in detecting different kinds of profile injection attack. However, when detecting the obfuscated attack based on standard average attack the detection performance is not good. To detect attackers based on beta distribution Yang et al. [12] used a novel Beta-Protection (βP) method. This method does not require previous information about the rating distribution of each product. Beta-Protection (βP) is used to immune the missing values.

In most of the semi-supervised detection techniques, there is little quantity of users who are labeled but massive quantity of users are unlabeled. So some existing works emphasize on modeling of both unlabeled and labeled consumer profiles. Zhang et al. [13] developed one Semi-Supervised SA detection method. From the product reviews, the method detects the malicious users. Performance of this method is good when known types of attacks are detected.

However, it requires some labeled profiles to create the training classifiers. Wu et al. [14] proposed hPSD (semi-supervised hybrid learning) model for detecting SA. This model uses both user-item relations and user features to gain maximum rate of SA detection. Cao et al. [15] and Wu et al. [16] proposed a Semi-SAD (Semi-supervised learning based SA Detection) method. This method takes advantage from both unlabeled and labeled user profiles for detecting SA.

3 Proposed Technique

In this section, the proposed technique named $Ward_pHMM$ has been presented. $Ward_pHMM$ is an unsupervised SA detection technique, which uses HMM and $Ward_p$ method. Here the $Ward_p$ [5] method creates feature weights by utilizing L_p norm, which can be seen as feature rescaling factor. The clusters formed by $Ward_p$ depend on p. The proposed scheme has two parts.

The first part is measurement of difference in rating behaviors of user and second part is the detection of attack profiles.

3.1 Prerequisite

In this section, prerequisites for working of the $Ward_p$HMM method have been presented.

3.1.1 User Preference Sequences (UPS)

In CFRS, authentic clients ordinarily rate objects according to their real preference. Whereas, Attack Clients (ACs) rate objects to bias framework's output according to their specific requirements. The rankings based on the rating given by ACs do not reflect the genuine preferences. In this way, a large difference exists between attack and genuine clients with respect to rating patterns. Such difference can be analyzed based on User Rating Item Sequence (URIS). URIS is given as:

$$\mathrm{URIS_u} = \{\mathrm{i}^{\mathrm{v}}_{\mathrm{m_1 s_1}}, \mathrm{i}^{\mathrm{v}}_{\mathrm{m_2 s_2}}, \dots, \mathrm{i}^{\mathrm{v}}_{\mathrm{m_n s_n}}\}, \tag{1}$$

where, $\mathrm{i_{m_1}}$, $\mathrm{i_{m_2}}$, …, $\mathrm{i_{m_n}}$represents the items, v represents the user and s represents the rating time.

The process of obtaining User Preference Sequences (UPS) is:

a. Based on rating information of user, observation sequence is constructed. Observation sequence represents item's rating sequence of every consumer.
b. For getting the Hidden Markov Model λ_0, firstly parameters of HMM λ ={ M, A, B } are set to small arbitrary values. Here, M represents initial probability distribution, A is matrix of transition probability, and B is matrix of emission probability. The observation sequence $OB = \{OB_1, OB_2, \dots, OB_S\}$ is used to train the HMM. Here, s denotes the length of the number of items rated by user v (i.e., observation sequence). Then for re-estimating HMM parameters Baum-Welch algorithm [17] is used.
c. Finally from the re-estimated HMM parameters, UPS and State Transition Sequence of each user is obtained using Viterbi algorithm [17].

3.1.2 User Matching Degree

For a user $v \in V$, let the observation sequence be $OB_v = \{OB_1, OB_2, \dots, OB_S\}$ and preference sequence (i.e., the equivalent hidden state sequence of user v) be $Q_v = \{q_1, q_2, \dots, q_S\}$. Then the User v's Matching Degree (UMD_v) can be calculated as:

$$\mathrm{UMD_v} = \sqrt[s]{\widehat{\mathrm{M}}(\mathrm{q_1})\widehat{\mathrm{B}}(\mathrm{q_1}, \mathrm{OB_1}) \prod_{\mathrm{i=2}}^{S} \widehat{\mathrm{A}}(\mathrm{q_{j-1}}, \mathrm{q_j})\widehat{\mathrm{B}}(\mathrm{q_{j-1}}, \mathrm{OB_j})}, \tag{2}$$

where, the initial State Probability (SP) matrix is represented by $\widehat{\mathrm{M}}$, the Observation Probability (OP) matrix is represented by $\widehat{\mathrm{B}}$ and the State Transition Probability (STP) matrix is represented by $\widehat{\mathrm{A}}$.

3.1.3 User Suspicious Degree (USD)

The ACs injects some specific number of APs into the CFRS to create the supported attack impact. Since the ACs must rate the similar target object, so ratings for the target object is very important.

3.1.3.1 Item Rating Sequence

Let i be an item and J be the set of all items in the dataset. Rating sequence of item $i \in J$ is known as Item Rating Sequence (IRS). It refers to series of item i's ratings $\mathrm{rt}^{s_1}_{v_1,i}, \mathrm{rt}^{s_2}_{v_2,i}, \dots, \mathrm{rt}^{s_n}_{v_n,i}$ provided by users $v_1, v_2, \dots, v_n$ at time $s_1, s_2, \dots, s_m$. IRS of item i (IRS_i) can be written as:

$$\mathrm{IRS_i} = \left\{\mathrm{rt}^{s_1}_{v_1,i}, \mathrm{rt}^{s_2}_{v_2,i}, \dots, \mathrm{rt}^{s_n}_{v_n,i}\right\}. \tag{3}$$

3.1.3.2 Item Entropy

Uncertainty of arbitrary variables is often measured by entropy. Let in the CFRS the set of different user provided ratings be G. $P_{i,e}$ represents the probability for item $i \in J$, that the users have given e points. Then, the Item Entropy of i (IE_i) can be calculated as:

$$\mathrm{IE_i} = -\sum_{\mathrm{e \in G}} \mathrm{P_{i,e}} \log_2 \mathrm{P_{i,e}}, \tag{4}$$

$$P_{i,e} = \frac{\sum_{rt^{s_k}_{v_{k,}i} i \in IRS_i} \Gamma(rt^{s_k}_{v_{k,}i}, e)}{\sum_{e \in G} \sum_{rt^{s_k}_{v_{k,}i} i \in IRS_i} \Gamma(rt^{s_k}_{v_{k,}i}, e)}, \quad (5)$$

where, $\Gamma(rt^{s_k}_{v_{k,}i}, e)$ denotes a discriminator function. If $rt^{s_k}_{v_{k,}i} = e$, then $\Gamma\left(rt^{s_k}_{v_{k,}i}, e\right) = 1$; otherwise $\Gamma(rt^{s_k}_{v_{k,}i}, e) = 0$, for dataset, G = {1, 2, 3, 4, 5}.

The more standardized item's rating distribution is, the lesser item's entropy will be. If the probability of that item is greater, it suggests that the ACs have rated on that item and also that item is a target object.

3.1.3.3 Item Suspicious Degree (ISD)

Let, ζ_i given in eq. 6 represents the normalized value of item i's reciprocal entropy, φ_v given in eq. 7 represents the normalized value of user v's reciprocal matching degree. For item i ∈ J the Item Suspicious Degree (ISD_i) is given in eq. 8:

$$\zeta_i = \frac{1/IE_i - 1/IE_{max}}{1/IE_{min} - 1/IE_{max}}, \quad (6)$$

$$\varphi_v = \frac{1/UMD_v - 1/UMD_{max}}{1/UMD_{min} - 1/UMD_{max}}, \quad (7)$$

$$ISD_i = \frac{\sum_{v \in V_G} \varphi_v}{|V_G|} \times \zeta_i, \quad (8)$$

where, the group of users who gave item i high ratings has been represented by V_G. If the value of rating is more than 3, it is referred as high rating for the dataset. The lowest value of User Matching Degree (UMD) is UMD_{min}. Highest value of UMD is UMD_{max}. The user v's matching degree is UMD_v. IE_{min} and IE_{max} be the lowest and highest value of item entropy respectively. Item i's entropy is IE_i.

Here, φ_v indicates that the user is more suspicious if the variation of UMD is higher. ζ_i indicates that the user is more suspicious if the item ratings distribution is more dispersed.

3.1.3.4 Suspicious Degree Range of Items

Suspicious Degree Range of Items (SDRI) rated by user v ∈ V, $SDRI_v$, can be calculated as:

$$SDRI_V = ISD^v_{max} - ISD^v_{min}, \quad (9)$$

where, ISD^v_{min} and ISD^v_{max} represent minimum and maximum value of ISD rated by user v respectively.

3.1.3.5 User Suspicious Degree

For user v ∈ V, let USD_v is user v's User Suspicious Degree (USD). φ_v is given in eq. 7. It represents the normalized value of user v's reciprocal matching degree. ϕ_v given in eq. 10 represents the normalize value of ISD rated by v. So using the linear weighted combination of ϕ_v and φ_v, in eq. 11 USD_v can be calculated as:

$$\phi_v = \frac{SDRI_V - SDRI_{min}}{SDRI_{max} - SDRI_{min}}, \quad (10)$$

$$USD_v = f \times \varphi_v + (1 - f) \times \phi_v, \quad (11)$$

where, Weight Factor is represented by f, $SDRI_v$ represents the Suspicious Degree Range of Items (SDRI) rated by user v, $SDRI_{max}$ and $SDRI_{min}$ represent highest value and lowest value of SDRI rated by users, respectively.

3.1.4 Ward$_p$ Method

$Ward_p$ [5] is an agglomerative hierarchical clustering technique. This method creates feature weights by utilizing L_p norm [18, 19]. For transforming the weights into feature rescaling factor L_p norm is used. The clusters created by $Ward_p$ are reliant on p. The $Ward_p$ technique given in eq. 12 can be calculated as:

$$Ward_p(Z_i, Z_j) = \frac{M_{Z_i} M_{Z_j}}{M_{Z_i} + M_{Z_j}} \sum_{r \in R} wt^p_{lr} |c_{Z_{i,r}} - c_{Z_{j,r}}|^p, \quad (12)$$

$$wt_{lr} = \frac{1}{\sum_{o \epsilon R} [D_{lrp}/D_{lop}]^{\frac{1}{(p-1)}}}, \quad (13)$$

$$D_{lrp} = \sum_{n \in Z_l} |y_{nr} - c_{lr}|^p, \quad (14)$$

$$c_{lr} = \frac{1}{|Z_l|} \sum_{y_n \in Z_l} y_{nr}, \quad (15)$$

Algorithm 1. Algorithm of the Ward$_p$HMM technique

Input: User-item rating dataset
Output: Group of AUs Set$_{AUs}$

1. Start.
2. URIS ← 0, UPS ← 0, USD ← 0
3. For each user v ∈ V
4. URIS$_v$ ← 0
5. For each item i ∈ J do
6. If $r_{v,i} > 0$ then
7. URIS$_v$ ← URIS$_v$ ∪ i
8. End if
9. End for
10. Sort items of URIS$_v$ based on its rating timestamps
11. URIS ← URIS ∪ URIS$_v$
12. End for
13. Initialize λ ={ M, A, B } to small arbitrary values
14. Repeat
15. Train the HMM parameter using λ_0 and URIS based on Baum-Welch algorithm
16. Until M, A and B converge
17. For every user v ∈ V
18. Find the best hidden sequence $Q_v = \{q_1, q_2, \ldots, q_S\}$ by Viterbi algorithm
19. UPS ← UPS ∪ Q_v
20. End for
21. For every item i ∈ J do
22. Compute item i's entropy IE$_i$ by using eqs. 4 and 5
23. End for
24. For every user v ∈ V do
25. Compute user v's UMD$_v$ by using the set of UPS applying eq. 2.
26. End for
27. For every item i ∈ J do
28. Compute item i's ISD$_j$ by using eqs. 6-8
29. End for
30. For every user v ∈ V do
31. Compute user v's SDRI$_v$ by using eq. 9
32. End for
33. For each user v ∈ V do
34. Compute user v's USD$_v$ by using eqs. 10 and 11
35. USD ← USD ∪ USD$_v$
36. End for
37. Set L ← |USD| and wt_{lr} =1/R.
38. Each USD be a cluster, represented as $Z = \{Z_1, Z_2, \ldots, Z_L\}$
39. Repeat
40. L ← L – 1
41. Join the clusters Z_i and Z_j to create the new cluster $Z_{Z_i \cup Z_j}$ which are the nearest according to Ward$_p$ given in eq. 12.
42. Update each wt_{lr} for l={1,2, ..., L} and r ={1,2, ..., R} by using eq.13.
43. Until L = 2
44. A1 ← Compute mean of USD for cluster Z_1.
45. A2 ← Compute mean of USD for cluster Z_2.
46. If A1 > A2 then
47. Set$_{AUs}$ ← get$_{AUs}$ (Z_1)
48. Else
49. Set$_{AUs}$ ← get$_{AUs}$(Z_2)
50. End if
51. Return Set$_{AUs}$
52. End

$$y_{nr} = \frac{x_{nr} - \overline{x_r}}{\text{range}(x_r)}, \tag{16}$$

where, M_{Z_i} and $c_{Z_{i,r}}$ represent the cardinality of item i at cluster Z_i and centroid of item i at cluster Z_i with respect to its feature r, respectively.

M_{Z_j} and $c_{Z_{j,r}}$ represent the same for item j at cluster Z_j. L = |USD|, $l \in L$, where L represents the length of User Suspicious Degree. wt_{lr} is the weight of feature $r \in R$ in the cluster having centroid at c_l. D_{lrp} is the distance of the feature r in cluster Z_l having centroid at c_l with respect to p. p represents the optimal exponent ($p \geq 1$, $p \leq 5$). c_{lr} is the centroid of feature r in cluster Z_l. y_{nr} represents cluster feature value. $\overline{x_r}$ is the average of feature r over the whole user's suspicious degree and x_{nr} denotes an entity in the user's suspicious degree.

3.3 Proposed Ward$_p$HMM Algorithm

For detecting shilling attackers, the proposed algorithm mainly needs three tasks, (i) calculation of User Preference Sequences (UPS), (ii) generation of User Suspicious Degree (USD) and (iii) shilling attacker detection based on Ward$_p$ method. Algorithm 1 presents the proposed Ward$_p$HMM technique.

For calculating UPS firstly, this model generates test set from rating database and attack profiles (Line 2). Secondly, for each user URIS is constructed using eq. 1 (Lines 4-13). Thirdly, the Hidden Markov Model parameters are initialized (Line 14) and the HMM is trained (Lines 15-17). Finally, to generate UPS the trained HMM is used (Lines 18-21).

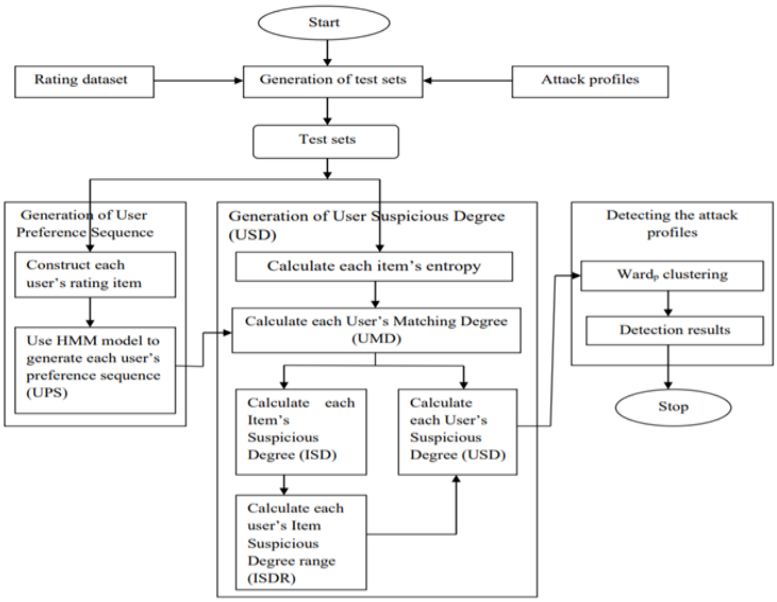

Fig. 2. Workflow of the Ward$_p$HMM method

For generating USD at first, each item's entropy is calculated using eq. 4 (Lines 22-24). Then, using eq. 2 each user's UMD is calculated from UPS (Lines 25-27). Secondly, each item's ISD (Lines 28-30) and SDRI (Lines 31-33) is calculated using eq. 8 and eq. 9 respectively.

Finally, each user's USD is generated using eq. 10 to obtain the set of USD (Lines 34-37).

For detecting shilling attackers based on Ward$_p$ method firstly, the L and wt_{lr} are set (Line 38). Then the Ward$_p$ method is used for grouping set of USD in two clusters using eq. 12 (Lines 39-44). Finally, the group of Attack Users (AUs) (Lines 45-52) is generated. The cluster with higher mean value of USD is denoted as the group of AUs. The workflow of the proposed Ward$_p$HMM technique is shown in figure 2.

4 Performance Analysis and Comparison

In this section, performance of the proposed Ward$_p$HMM technique has been analyzed.

It also presents a comparison between the performance of Ward$_p$HMM and UD-HMM. For this experiment Amazon-ratings dataset [20] is used. This Amazon-ratings dataset contains 1210271 User-Ids, 249274 Product-Ids, 2023070 Ratings and 4231 Timestamps.

Ratings vary from 1 to 5. Where 5 indicate most liked and 1 indicates disliked. Here, Amazon-ratings dataset is sampled randomly containing 5000 User-Ids, 757 Product-Ids, 5255 Ratings and 1613 Timestamps. Shilling Profile (SP) has been constructed based on the obfuscated attack model [21] with different filler size and attack size, which is injected in dataset. Here, UD-HMM parameters α and N are set to 0.7 and 5 respectively.

4.1 Performance metrics used

To analyze performance of proposed Ward$_p$HMM technique, here it has been compared with UDHMM with respect to precision, recall and F1-score. The precision, recall, and F1-score are defined as:

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}, \quad (17)$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative(FN)}}, \quad (18)$$

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \qquad (19)$$

where, number of APs correctly detected is defined by True Positive (TP). Number of authentic users correctly detected is defined by True Negative (TN). Number of authentic users misclassified as attack ones is defined as False Positive (FP). Number of APs misclassified as authentic users is defined as False Negative (FN).

4.2 Performance Comparison

To compare the precision, recall and f1-score values of Ward$_p$HMM and UD-HMM methods, experiments have been performed on the Amazon-ratings sampled dataset with different attack size and filler size.

Figure 3 and figure 4 present effect of attack size on precision when the filer size is 3% and 5% respectively. The precision values are captured for different Filler Size (FS) and Attack Size (AS) under obfuscated attack based on standard average attack. When filler size is set to 3%, precision value of Ward$_p$HMM ranges from 0.8 to 0.9. In case of UD-HMM it ranges from 0.34 to 0.41. On the other hand, when filler size is set to 5% precision value of Ward$_p$HMM is between 0.85 to 0.91. In case of UD-HMM, it ranges between 0.19 and 0.29. So, this indicates that the proposed method detects the attack users more correctly compared to the UDHMM method.

Figure 5 and figure 6 present the effect of attack size on recall value when the filler size is 3% and 5% respectively. The recall values are recorded for various filler size and attack size under obfuscated attack on the Amazon-ratings sampled dataset. When filter size is set to 3% recall value of Ward$_p$HMM ranges from 0.88 to 0.93. In case of UD-HMM it ranges from 0.37 to 0.95. However, the highest value of UD-HMM is slightly more than that of Ward$_p$HMM but in most of the cases Ward$_p$HMM performs better than UD-HMM. On the other hand, when filter size is set to 5% recall value of Ward$_p$HMM is between 0.86 and 0.94.

In case of UD-HMM, it ranges between 0.55 and 0.94. Overall, recall value of Ward$_p$HMM method is higher than the existing UD-HMM method.

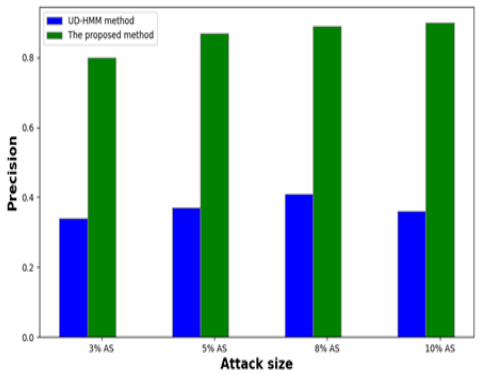

Fig. 3. Effect of attack size on precision when filter size is 3%

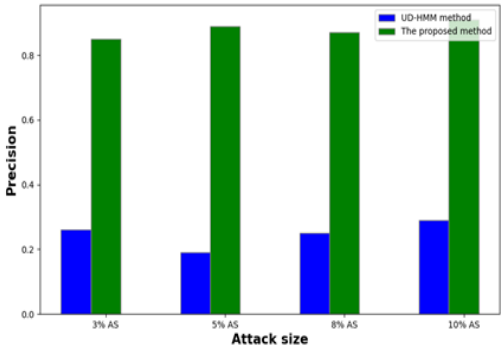

Fig. 4. Effect of attack size on precision when filter size is 5%

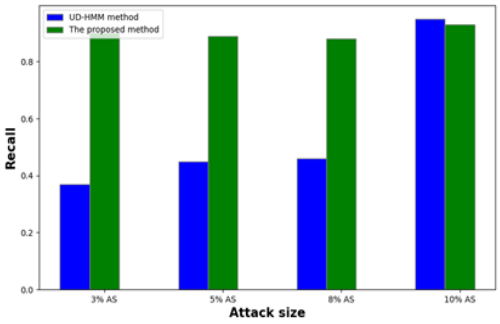

Fig. 5. Effect of attack size on recall when filter size is 3%

This signifies that proposed method's detection performance is better than the UDHMM method.

Figure 7 and 8 presents the effect of attack size on F1-score when the filter size is 3% and 5% respectively. When the filter size is 3%, F1-score

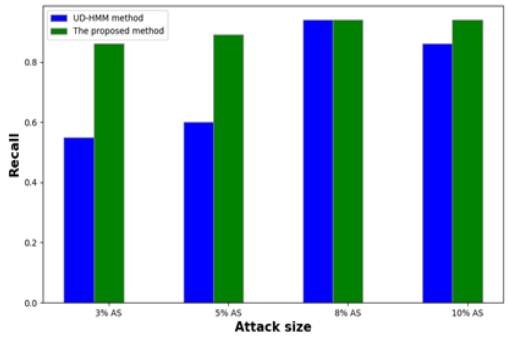

Fig. 6. Effect of attack size on recall when filter size is 5%

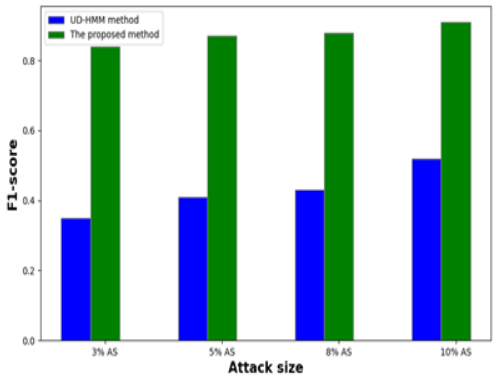

Fig. 7. Effect of attack size on F1-score when filter size is 3%

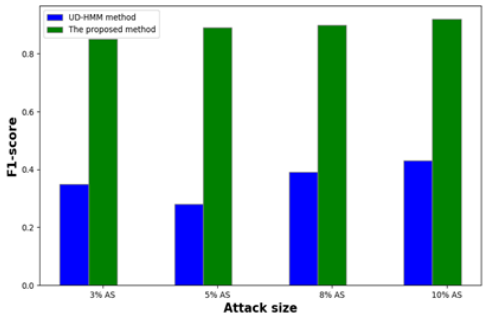

Fig. 8. Effect of attack size on F1-score when filter size is 5%

of the proposed Ward$_p$HMM method is in the range of 0.84 and 0.91.

In case of UD-HMM it is from 0.35 to 0.52. On the other hand, when the filter size is 5% F1-score of Ward$_p$HMM method varies between 0.85 and 0.92. In case of UD-HMM it varies from 0.35 to 0.43. The Ward$_p$HMM method has higher F1-score value than the existing UD-HMM method. So, Ward$_p$HMM method detects the genuine users and attack users more accurately than the existing UD-HMM method. This signifies that with respect to detection performance proposed method outperforms UD-HMM method

5 Conclusions and Future Work

CFRS is a very efficient way for handling the problem of information overloading. However, CFRSs are very much vulnerable to numerous shilling attacks due to insertion of variety of malicious user profiles in the system.

These malicious user profiles affect the user recommendations. For addressing this issue in this paper, a shilling attack detection technique named Ward$_p$HMM has been proposed. For overcoming the problem of Ward method during clustering, the proposed scheme uses Ward$_p$ method. Performance of the Ward$_p$HMM method has been analyzed using the Amazon-ratings sample dataset.

It has been observed that Ward$_p$HMM method outperforms UD-HMM method with respect to precision, recall and F1-score. Ward$_p$ method has still some scope for improvement. The Ward$_p$ method requires the calculation of centroids which make the proposed technique considerably computation intensive. Therefore, development of light weight shilling attack detection technique remains as future work.

Acknowledgement

The authors are thankful to Mobile Computing Laboratory, Department of Computer Science & Engineering, Tripura University for providing the necessary infrastructure.

References

1. **Si, M., Li, Q. (2020).** Shilling attacks against collaborative recommender systems: A review. Artificial Intelligence Review, Vol. 53, No. 1,

pp. 291–319. DOI: 10.1007/s10462-018-9655-x.

2. **Wei, R., Shen, H. (2016).** An improved collaborative filtering recommendation algorithm against shilling attacks. Proceedings of 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), pp. 330–335. DOI: 10.1109/PDCAT.2016.077.
3. **Mobasher, B., Burke, R., Bhaumik, R., Williams, C. (2007)**. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology (TOIT), Vol. 7, No. 4. DOI: 10.1145/1278366.1278372.
4. **Zhang, F., Zhang, Z., Zhang, P., Wang, S. (2018)**. UD-HMM: An unsupervised method for shilling attack detection based on hidden Markov model and hierarchical clustering. Knowledge-Based Systems, Vol. 148, pp. 146–166. DOI: 10.1016/j.knosys.2018.02.032.
5. **Cordeiro de Amorim, R. (2015)**. Feature relevance in ward's hierarchical clustering using the L_p norm. Journal of Classification, Vol. 32, pp. 46–62. DOI: 10.1007/s00357-015-9167-1.
6. **Lam, S.K., Riedl, J. (2004).** Shilling recommender systems for fun and profit. Proceedings of the 13th International Conference on World Wide Web, pp. 393–402. DOI: 10.1145/988672.988726.
7. **Gunes, I., Kaleli, C., Bilge, A., Polat, H. (2014)**. Shilling attacks against recommender systems: A comprehensive survey. Artificial Intelligence Review, Vol. 42, No. 4, pp. 767–799. DOI: 10.1007/s10462-012-9364-9.
8. **Yang, Z., Xu, L., Cai, Z., Xu, Z. (2016).** Re-scale AdaBoost for attack detection in collaborative filtering recommender systems. Knowledge-Based Systems, Vol. 100, pp. 74–88. DOI: 10.1016/j.knosys.2016.02.008.
9. **Burke, R., Mobasher, B., Williams, C., Bhaumik, R. (2006).** Detecting profile injection attacks in collaborative recommender systems. Proceedings of 8th IEEE International Conference on E-Commerce Technology and 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE'06). DOI: 10.1109/CEC-EEE.2006.34.
10. **Bryan, K., O'Mahony, M., Cunningham, P. (2008).** Unsupervised retrieval of attack profiles in collaborative recommender systems. Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 155–162. DOI: 10.1145/1454008.1454034.
11. **Lee, J.S., Zhu, D. (2012).** Shilling attack detection—A new approach for a trustworthy recommender system. INFORMS Journal on Computing, Vol. 24, No. 1, pp. 117–131. DOI: 10.1287/ijoc.1100.0440.
12. **Yang, Z., Cai, Z., Guan, X. (2016).** Estimating user behavior toward detecting anomalous ratings in rating systems. Knowledge-Based Systems, Vol. 111, pp. 144–158. DOI: 10.1016/j.knosys.2016.08.011.
13. **Zhang, L., Wu, Z., Cao, J. (2017).** Detecting spammer groups from product reviews: A partially supervised learning model. IEEE Access, Vol. 6, pp. 2559–2568. DOI: 10.1109/ACCESS.2017.2784370.
14. **Wu, Z., Wang, Y., Wang, Y., Wu, J., Cao, J., Zhang, L. (2015).** Spammers detection from product reviews: a hybrid model. Proceedings of IEEE International Conference on Data Mining. pp. 1039–1044. DOI: 10.1109/ICDM. 2015.73.
15. **Cao, J., Wu, Z., Mao, B., Zhang, Y. (2013).** Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system. World Wide Web, Vol. 16, pp. 729–748. DOI: 10.1007/s11280-012-0164-6.
16. **Wu, Z., Cao, J., Mao, B., Wang, Y. (2011).** Semi-SAD: applying semi-supervised learning to shilling attack detection. In Proceedings of Fifth ACM Conference on Recommender Systems. pp. 289–292. DOI: 10.1145/2043932.2043985.
17. **Rabiner, L.R. (1989).** A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, Vol. 77, No. 2, pp. 257–286. DOI: 10.1109/ 5.18626.
18. **Chan, E.Y., Ching, W.K., Ng, M.K., Huang, J.Z. (2004).** An optimization algorithm for

clustering using weighted dissimilarity measures. Pattern Recognition, Vol. 37, No. 5, pp. 943–952. DOI: 10.1016/j.patcog.2003.11.003.

19. **Huang, J.Z., Ng, M.K., Rong, H., Li, Z. (2005).** Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 5, pp. 657–668. DOI: 10.1109/TPAMI.2005.95.
20. **Kaggle (2021).** https://www.kaggle.com/skillsmuggler/amazon-ratings/data.
21. **Hurley, N., Cheng, Z., Zhang, M. (2009).** Statistical attack detection. Proceedings of Third ACM Conference on Recommender systems, pp. 149–156. DOI: 10.1145/1639714.1639740.

Article received on 07/03/2021; accepted on 07/06/2021. Corresponding author is Abhishek Majumder.