ICIC Express Letters
Volume **11**, Number **3**, March **2017**
 ISSN 1881-803X
pp. **543–549**

AN INCREMENTAL APPROACH FOR DETECTING RECOMMENDATION ATTACKS BASED ON ENSEMBLE LEARNING

Quanqiang Zhou

Computer Engineering Institute
Qingdao University of Technology
No. 777, Jialingjiang Rd., Huangdao Dist., Qingdao 266520, P. R. China
zhouqiang128@126.com

Received July 2016; accepted October 2016

Abstract. *Existing supervised approaches for detecting recommendation attacks learn the training sets in batch mode. With this problem in mind, in this paper we propose an incremental detection approach for recommendation attacks based on ensemble learning. Firstly, we use the active example selection (AES) method with the new added training sets to create the base training sets. Then, we use the Naive Bayes learner to learn the created base training sets to generate the base classifiers. The weights of these base classifiers are calculated using the training error rate. Finally, we use the generated base classifiers to detect the recommendation attacks. The weighted voting policy is used to output the detection results. The experimental results on MovieLens dataset show that the proposed approach can detect the recommendation attacks with high recall, precision, and AUC.*
Keywords: Collaborative filtering, Recommendation attacks, Incremental detection, Ensemble learning

1. **Introduction.** Collaborative filtering recommender systems have been shown to have significant vulnerabilities to "shilling attacks" or called "recommendation attacks" [1, 2]. To detect such attacks, a variety of unsupervised and supervised approaches have been proposed. Unsupervised approaches require certain prior knowledge rather than labeled training samples [3, 4, 5]. One common problem faced by unsupervised approaches is that some prior knowledge used in these approaches is difficult to get. Supervised approaches [6, 7, 8] usually have good detection performance. They learn the training sets in batch mode. That is when new training set is produced, these approaches have to re-learn all the old and new added training sets.

The Naive Bayes learner [9] is a probabilistic learner based on the theorem of Bayes. Ensemble learning techniques [10] try to combine a group of classifiers (called base classifiers), each of which solves the same classification problem, in order to obtain a better classification accuracy. EAES (ensemble learning based on active example selection) algorithm [11] is an ensemble learning algorithm. In this algorithm, the AES (active example selection) method is used to choose the informative examples for creating the base training sets.

In this paper, to learn the training sets incrementally we propose an approach for detecting recommendation attacks based on ensemble learning. The key part of our proposed approach for achieving incremental learning is that it generates a group of base classifiers for each new added training set and uses the weighted voting policy to integrate all the base classifiers. The main contributions of this paper are summarized as follows. (1) We propose an algorithm to generate a group of classifiers and weights based on the new added training sets. In this algorithm, we use the AES method to generate the base training sets. We use the Naive Bayes learner as the base learning algorithm. We use the training error rate to calculate the weights of base classifiers. (2) We propose

an incremental detection algorithm for detecting recommendation attacks. In particular, we use the generated base classifiers to detect the recommendation attacks. We use the weighted voting policy to integrate all the base classifiers. (3) We conduct experiments on MovieLens dataset to verify the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section 2 describes the proposed incremental detection approach. Section 3 presents the experimental results and evaluations. The conclusions and future work are discussed in Section 4.

2. **Incremental Detection Approach.** As shown in Figure 1, the proposed incremental detection approach consists of two stages: stage of training and stage of detecting. At the stage of training, the incremental training sets D_1, D_2, ..., D_I are used as the inputs of the base classifiers and weights generation algorithm. Then, this algorithm will generate $I \times K$ classifiers and weights $\{C_{1,1}, C_{1,2}, \ldots, C_{1,K}, \alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{1,K}\}$, $\{C_{2,1}, C_{2,2}, \ldots, C_{2,K}, \alpha_{2,1}, \alpha_{2,2}, \ldots, \alpha_{2,K}\}, \ldots, \{C_{I,1}, C_{I,2}, \ldots, C_{I,K}, \alpha_{I,1}, \alpha_{I,2}, \ldots, \alpha_{I,K}\}$. At the stage of detecting, the generated base classifiers are used to detect the test set. After that, the weighted voting policy is used to output the final detection results.

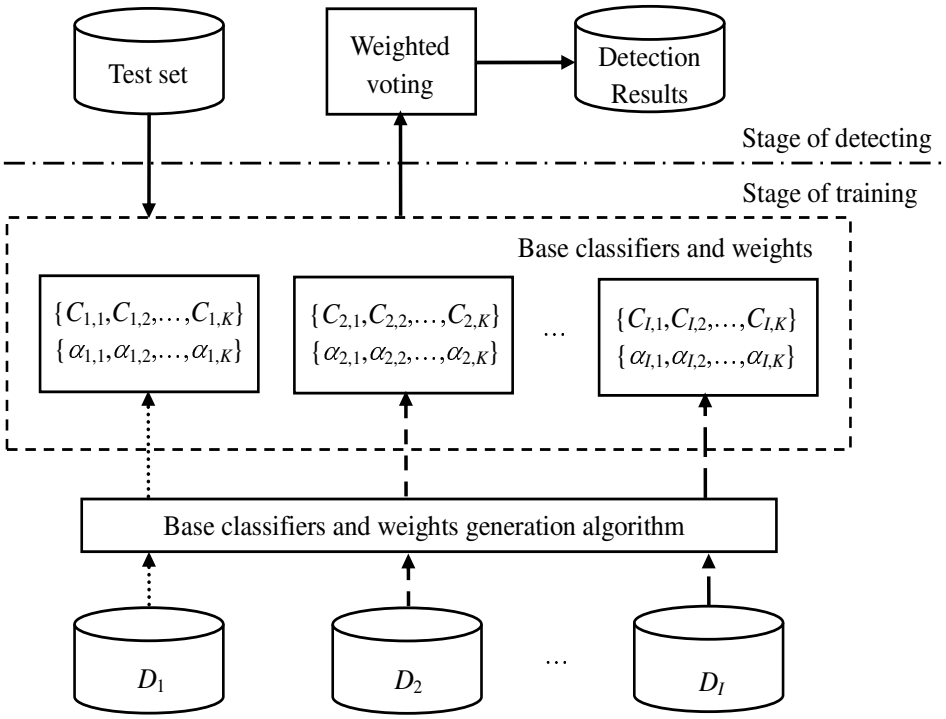

FIGURE 1. Framework of the proposed incremental detection approach based on ensemble learning

The details of the proposed incremental detection approach will be discussed in the following subsections. We first introduce the proposed base classifiers and weights generation algorithm. Then, we show the proposed incremental detection algorithm.

2.1. **Base classifiers and weights generation algorithm.** To learn the first or the new added training sets, we use the following proposed algorithm to generate the base classifiers and weights which are used to detect attach profiles.

Let D denote a training set, $\{C_1, C_2, \ldots, C_K\}$ denote the base classifiers, and $\{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ denote the corresponding weights of the base classifiers. The proposed base classifiers and weights generation algorithm is shown as follows.

Algorithm 1 Base classifiers and weights generation algorithm

Input: D

Output: $\{C_1, C_2, \ldots, C_K\} \cup \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$

(1) Choose λ_0 genuine profiles and λ_0 attack profiles from D, randomly, to create the balanced seed training set D_0, where λ_0 denotes a small integer. Let $V_0 = D - D_0$ denote the validation set.

(2) Use the Naive Bayes learner to learn D_0 to generate a classifier C. Based on this classifier, measure the usefulness of examples in set V_0 using the following formula [11]:

$$e_c(x_p) = \frac{1}{2}\sum_{m=0}^{1}\left(y_{pm} - f_m(x_p, C)\right)^2, \tag{1}$$

where, $m \in \{0,1\}$, 0 denotes genuine profile, 1 denotes attack profile, y_{pm} denotes the label of instance x_p, and $f_m(x_p, C)$ denotes the probability of C classifying x_p as class m. Note that, the most useful instance is the one which causes the largest error on the current classifier.

(3) Choose λ most useful instances $\{x_1, \ldots, x_\lambda\}$ from V_0. Let $D_0 = D_0 + \{x_1, \ldots, x_\lambda\}$ and $V_0 = V_0 - \{x_1, \ldots, x_\lambda\}$ where λ denotes a small integer.

(4) Repeat step (2) and step (3) until classifier C achieves specified performance level or V_0 is empty.

(5) Use the Naive Bayes learner to learn D_0 to generate the kth base classifier C_k. Use C_k to classify instances in D to compute the training error ϵ_k of C_k.

(6) Repeat step (1) and step (5) K times to generate K base classifiers $\{C_1, C_2, \ldots, C_K\}$ and the corresponding training errors $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_K\}$ where K denotes an integer.

(7) Use the training errors $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_K\}$ to compute the weights of the base classifiers. The weight ϵ_k of the kth base classifier can be calculated as follows [11]:

$$\alpha_k = \frac{\exp(-\epsilon_k)}{\sum_{k=1}^{K}\exp(-\epsilon_k)}. \tag{2}$$

(8) Return $\{C_1, C_2, \ldots, C_K\} \cup \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$.

2.2. **Incremental detection algorithm.** Let $\{D_1, D_2, \ldots, D_I\}$ denote the incremental training sets, D_{test} denote the test set, R_{test} denote the detection result set. The proposed incremental detection algorithm based on ensemble learning is shown as follows.

Algorithm 2 Incremental detection algorithm based on ensemble learning

Input: $\{D_1, D_2, \ldots, D_I\}$, D_{test}

Output: R_{test}

/*stage of training*/

(1) For each training set $D \in \{D_1, D_2, \ldots, D_I\}$, call Algorithm 1 to generate $I \times K$ base classifiers $\{C_1, C_2, \ldots, C_{I\times K}\}$ and the corresponding weights $\{\alpha_1, \alpha_2, \ldots, \alpha_{I\times K}\}$.

/*stage of detecting*/

(2) For each user $u \in D_{test}$, let $P_k(0|u, C_k)$ denote the probability of C_k classifying user u as genuine profile, $P_k(1|u, C_k)$ denote the probability of C_k classifying user u as attack profile, 0 denote genuine profile, and 1 denote attack profile. Based on the weighted voting policy, the final detection results for user u can be computed as follows [11]:

$$f(u) = \begin{cases} 0, & \sum_{k=1}^{I\times K}\alpha_k P_k(0|u, C_k) > \sum_{k=1}^{I\times K}\alpha_k P_k(1|u, C_k), \\ 1, & \sum_{k=1}^{I\times K}\alpha_k P_k(0|u, C_k) \le \sum_{k=1}^{I\times K}\alpha_k P_k(1|u, C_k). \end{cases} \tag{3}$$

(3) For each user $u \in D_{test}$, put $f(u)$ into set R_{test}.

(4) Return R_{test}.

In Algorithm 2, the characteristic of incremental detection is shown at the stage of training. In this stage, the algorithm does not need to learn the whole training sets but only the new added trainings sets. The new generated base classifiers can contain the

knowledge of the new attack profiles in the training set. In the stage of detecting, the weighted voting policy is used to combine all the base classifiers.

3. Experiments and Evaluations.

3.1. **Experimental data and settings.** The MovieLens dataset [12] is used in this paper. It consists of 1,000,209 ratings on 3,952 movies by 6,040 users. The ratings are integer values between one and five. The user profiles in this dataset are labeled as genuine profiles.

Four common attack models are used to generate the attack profiles. These attack models are random, average, bandwagon, and 20% AoP attack [13, 14].

Table 1 shows the final experimental data. It consists of four incremental training sets D_1, D_2, D_3, D_4, and one test set.

Table 1. Experimental data

Type of data	Training set				Test set
	D_1	D_2	D_3	D_4	
Genuine	500	500	500	500	500
Random	50	0	0	0	25
Average	0	50	0	0	25
Bandwagon	0	0	50	0	25
20% AoP	0	0	0	50	25

As shown in Table 1, to create the training sets we randomly select 500 genuine profiles from the MovieLens dataset, in turn. Attack profiles are generated by the four attack models with filler sizes [2] $\{1\%, 3\%, 5\%, 10\%, 15\%\}$, respectively. Ten attack profiles are constructed for each filler size.

To create the test set, we randomly select 500 genuine profiles from the remaining MovieLens dataset. Attack profiles are generated by the four attack models with filler sizes $\{1\%, 3\%, 5\%, 10\%, 15\%\}$, respectively. Five attack profiles are constructed for each filler size. We repeat this process to generate ten test sets. The average detection results of these test sets are reported in the experiments.

Thirteen attributes [6] are used to extract features of user profiles. These features include WDMA, DegSim, DegSim', Length Variance, RDMA, WDA, FAC (random attack, push), FAC (bandwagon attack, push), FMD (average attack, push), PV (average attack, push), FMD (bandwagon attack, push), FMV (average attack, push), and FMD (random attack, push), respectively.

We follow paper [11] to set the parameters of the proposed approach. In particular, we set $\lambda_0 = 1$, $\lambda = 2$ in Algorithm 1. In Algorithm 2, we set $K = 15$. Specified performance level in Algorithm 2 means that the AUC of classifier C is larger than 0.9 on the training set.

3.2. **Evaluation metrics.** Three standard metrics of recall, precision, and AUC (area under the ROC curve) [7, 15] are used to evaluate the detection performance in our experiments. These metrics are defined as follows [7, 15]:

$$Recall = \frac{TP}{TP + FN}, \tag{4}$$

$$Precision = TP\frac{TP}{TP + FP}, \tag{5}$$

$$AUC = \frac{\sum_{i=1}^{N} rank_i - N \times (N+1)/2}{N \times P}, \tag{6}$$

where, TP is the number of attack profiles which are correctly detected, FN is the number of attack profiles misclassified as genuine profiles, FP is the number of genuine profiles misclassified as attack profiles, N is the total number of genuine profiles, P is the total number of attack profiles. Rank the user profiles in the test set according to their posterior probability of detection results in reverse order. $rank_i$ is the order number of the ith genuine user.

3.3. **Experimental results and analysis.** To verify the effectiveness of the proposed approach, we set two groups of comparative experiments as follows.

(1) We compare the proposed approach IncDA with Batch-EAES. The only difference between these two methods is that Batch-EAES requires relearning four times and the training sets for each time are D_1, $D_1 \cup D_2$, $D_1 \cup D_2 \cup D_3$, and $D_1 \cup D_2 \cup D_3 \cup D_4$, respectively. IncDA requires incremental learning four times and the training sets for each time are D_1, D_2, D_3, and D_4, respectively.

(2) We compare IncDA with Single-SVM [6] and Meta-SVM [7]. Single-SVM is a classical supervised approach. Meta-SVM can represent the latest research results in supervised detection approach. $D_1 \cup D_2 \cup D_3 \cup D_4$ are used as their training sets. Note that, Single-SVM and Meta-SVM operate in batch mode.

3.3.1. *Comparison of IncDA and Batch-EAES.* Recall, precision, and AUC of Batch-EAES and IncDA are shown in Figure 2, Figure 3 and Figure 4.

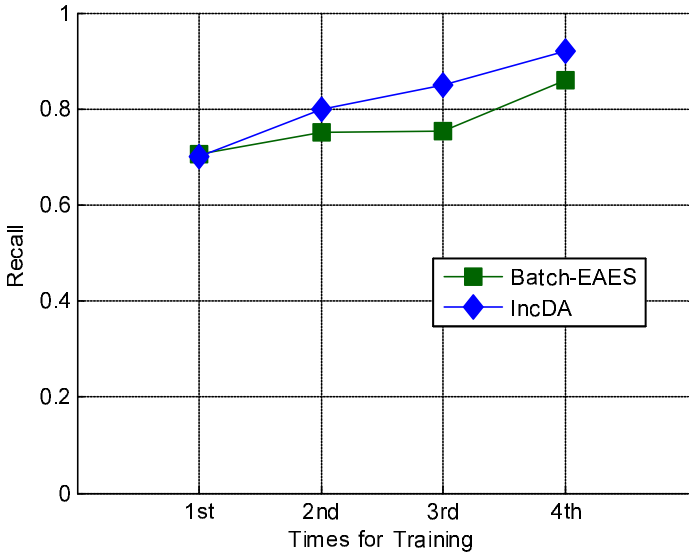

FIGURE 2. Recall of Batch-EAES and IncDA on the test set

As shown in Figure 2, Batch-EAES and IncDA have the same recall at the beginning since they use the same training set at the first point. As more training sets are used in these two methods, their recall also increases. The reason for this phenomenon is that both methods can learn more instances of attack profiles from the new added training sets.

As shown in Figure 3, IncDA obviously outperforms Batch-EAES in terms of precision. These results illustrate that only a small number of genuine profiles are misclassified. The proposed approach IncDA can effectively identify the genuine profiles.

As shown in Figure 4, the AUC of Batch-EAES decreases as the increasing of training sets. This is due to the fact that the increasing of training sets increases the imbalance of the final training set in batch mode. However, the proposed IncDA holds a high AUC since it operates incrementally. Above results show the success of the proposed approach.

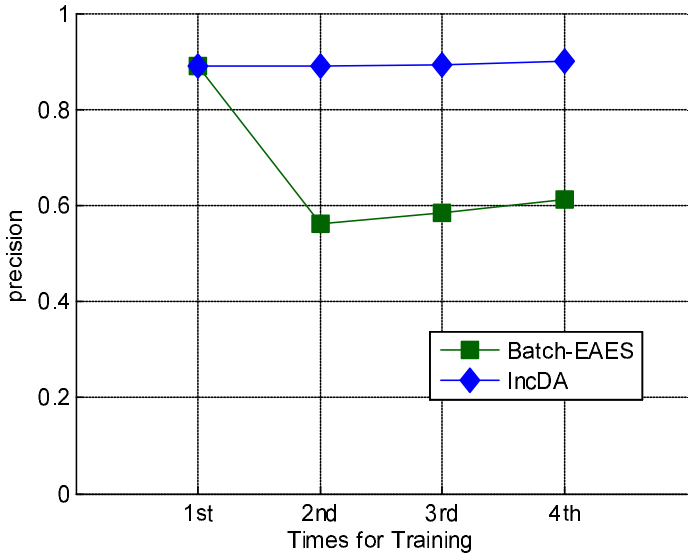

FIGURE 3. Precision of Batch-EAES and IncDA on the test set

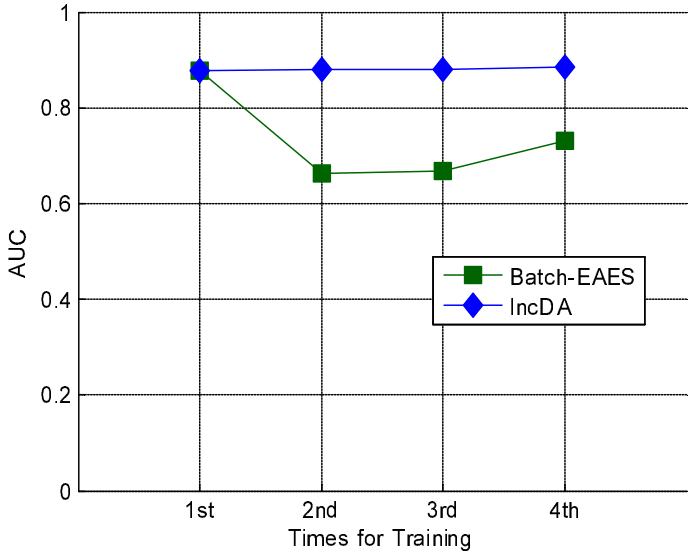

FIGURE 4. AUC of Batch-EAES and IncDA on the test set

TABLE 2. Detection results of Single-SVM, Meta-SVM, and IncDA

Methods	Evaluation metrics		
	Recall	Precision	AUC
Single-SVM	0.90	0.50	0.67
Meta-SVM	0.91	0.80	0.83
IncDA	0.90	0.90	0.85

3.3.2. *Comparison of Single-SVM, Meta-SVM, and IncDA.* Recall, precision, and AUC of Single-SVM, Meta-SVM, and IncDA are shown in Table 2.

As shown in Table 2, Single-SVM has high recall but low precision and AUC. This is due to the fact that this method can detect most of the attack profiles at the same time; however, it misclassifies many genuine profiles as attack profiles.

Both Meta-SVM and IncDA have high recall, precision, and AUC. The proposed incremental detection method IncDA is as good as the batch-based method Meta-SVM in terms of detection performance. The reasons for this phenomenon can be described as

follows: (1) IncDA employs the ensemble learning, in particular the generated base classifiers, to improve its capability of detecting attack profiles. (2) IncDA uses the weighted voting policy to combine the existing and the new added base classifiers, effectively.

4. **Conclusions and Future Work.** In this paper, we propose an incremental approach for detecting recommendation attacks based on ensemble learning. To generate a group of classifiers and weights for each new added training set, we propose a base classifiers and weights generation algorithm based on the AES method, Naive Bayes learner, and the training error rate. We propose an incremental detection algorithm for recommendation attacks based on the generated classifiers, weights, and weighted voting policy. The experiments on MovieLens dataset demonstrate the effectiveness of the proposed approach. In our future work, we plan to study a method for reducing the ineffective base classifiers which have already been integrated in the detection approach.

Acknowledgment. This work is supported by the Shandong Provincial Natural Science Foundation of China (No. ZR2014FP014).

REFERENCES

[1] S. Zhang, Y. Ouyang, J. Ford and F. Makedon, Analysis of a low-dimensional linear model under recommendation attacks, *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, pp.517-524, 2006.

[2] Q. Zhou, Feature extraction approach for recommendation attacks based on mutual information, *ICIC Express Letters*, vol.10, no.1, pp.59-66, 2016.

[3] B. Mehta, H. Thomas and F. Peter, Lies and propaganda: Detecting spam users in collaborative filtering, *Proc. of the 12th International Conference on Intelligent User Interfaces*, Honolulu, Hawaii, pp.14-21, 2007.

[4] B. Mehta and W. Nejdl, Unsupervised strategies for shilling detection and robust collaborative filtering, *User Modeling and User-Adapted Interaction*, vol.19, nos.1-2, pp.65-79, 2009.

[5] J. S. Lee and D. Zhu, Shilling attack detection – A new approach for a trustworthy recommender system, *INFORMS Journal on Computing*, vol.24, no.1, pp.117-131, 2012.

[6] C. A. Williams, B. Mobasher and R. Burke, Defending recommender systems: Detection of profile injection attacks, *Service Oriented Computing and Applications*, vol.1, no.3, pp.157-170, 2007.

[7] F. Z. Zhang and Q. Q. Zhou, A meta-learning-based approach for detecting profile injection attacks in collaborative recommender systems, *Journal of Computers*, vol.7, no.1, pp.226-234, 2012.

[8] F. Z. Zhang and Q. Q. Zhou, HHT-SVM: An online method for detecting profile injection attacks in collaborative recommender systems, *Knowledge-Based Systems*, vol.65, pp.96-105, 2014.

[9] G. H. John and P. Langley, Estimating continuous distributions in Bayesian classifiers, *Proc. of the 11th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, pp.338-345, 1995.

[10] L. Rokach, A. Schclar and E. Itach, Ensemble methods for multi-label classification, *Expert Systems with Applications*, vol.41, no.16, pp.7507-7523, 2014.

[11] S. Oh, M. S. Lee and B. T. Zhang, Ensemble learning with active example selection for imbalanced biomedical data classification, *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol.8, no.2, pp.316-325, 2011.

[12] *http://www.grouplens.org/node/12*.

[13] R. Burke, B. Mobasher and R. Bhaumik, Limited knowledge shilling attacks in collaborative filtering systems, *Proc. of Workshop on Intelligent Techniques for Web Personalization*, Edinburgh, Scotland, 2005.

[14] N. Hurley, Z. Cheng and M. Zhang, Statistical attack detection, *Proc. of the 3rd ACM Conference on Recommender Systems*, New York, pp.149-156, 2009.

[15] D. J. Hand and R. J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, *Machine Learning*, vol.45, no.2, pp.171-186, 2001.