·CTCIS 2016 推荐论文· DOI:10.15961/j.jsuese.2017.01.023

基于项目流行度和新颖度分类特征的托攻击检测算法

于洪涛[1,2],周倩楠[1,2],张付志[1,2]
(1.燕山大学 信息科学与工程学院,河北 秦皇岛 066004;2.河北省计算机虚拟技术与系统集成重点实验室,河北 秦皇岛 066004)

摘 要:针对有监督检测方法在检测托攻击时准确率不高的问题,提出一种基于项目流行度和新颖度分类特征的托攻击检测算法。首先,根据真实概貌和攻击概貌在选择评分项目方式上不同,从流行度和新颖度角度,提出有效区分正常用户和攻击用户的特征;然后,基于这些特征提出一种集成检测框架,通过Boosting提升技术产生多个差异较大的基分类器,并且通过融合带有权重的基分类器的预测值得到最终的检测结果。实验结果表明,基于项目流行度和新颖度分类特征的托攻击检测算法能够提高攻击检测的准确率和召回率。

关键词:托攻击;项目流行度;项目新颖度;Boosting技术;集成检测
中图分类号:TP393 **文献标志码**:A **文章编号**:2096-3246(2017)01-0176-08

An Shilling Attack Detection Algorithm Based on Item Popularity and Novelty Degree Features

YU Hongtao[1,2], *ZHOU Qiannan*[1,2], *ZHANG Fuzhi*[1,2]
(1. School of Info. Sci. and Eng., Yanshan Univ., Qinhuangdao 066004, China;
2. Key Lab. for Computer Virtual Technol. and System Integration of Hebei Province, Qinhuangdao 066004, China)

Abstract: The existing supervised detection algorithms have low precision when they detect shilling attacks. To address this problem, a shilling attack detection algorithm based on the features of popularity and novelty degree was proposed. Firstly, according to the difference between genuine and attack profiles in choosing items to rate, several features were extracted, which can effectively distinguish normal and attack users in perspective of popularity and novelty. Secondly, an ensemble detection framework based on these features was proposed. The Boosting technology was used to generate different base classifiers and the detection results were obtained by combining the predicted results of the base classifiers with weight. The experimental results showed that the shilling attack detection algorithm based on the features of popularity and novelty degree can improve the precision and recall of attack detection.

Key words: shilling attacks; item popularity; item novelty; Boosting technology; ensemble detection

协同过滤推荐系统为解决“信息过载”问题提供了一条有效途径,在电子商务站点得到越来越多的应用。然而由于推荐系统自身的开放性,一些恶意用户通过向推荐系统中注入大量的虚假评分,使其推荐结果产生偏差,进而影响用户对推荐系统的信任[1]。这种虚假评分的行为被称为“托攻击(shilling attacks)[2]”或“概貌注入攻击(profile injection attacks)”。为了降低攻击者对推荐系统的影响,需要寻找一些特征来区分正常用户和攻击用户。传统的攻击检测方法大多都是根据真实概貌和攻击概貌的评分值存在差异来提取特征,但是这种基于用户评分值差异提取的特征存在两个问题:1)攻击用户通过选择流行项目并且评最高分(例如,流行攻击)来减小与正常用户的评分值差异,那么利用此类特征容易造成正常用户的误判;2)此类特征对于低填充规模攻击有较低的检测准确率和召回率。

为了解决以上问题,本文在提取特征时,一方面,考虑将用户对项目的评分值用一种统计指标值代替;另一方面,不考虑用户对项目的评分值大小,只考虑用户对某个项目是否有评分。另外,对于低

收稿日期:2016-09-17
基金项目:国家自然科学基金资助项目(61379116);河北省自然科学基金资助项目(F2015203046);河北省高等学校科学技术研究重点资助项目(ZH2012028)
作者简介:于洪涛(1964—),副教授,博士. 研究方向:推荐系统. E-mail:yu5771@163.com

填充规模的攻击,通过改进相似度计算方法,增加相似度权重,进而改进 DegSim 这一特征指标。

本文提出一种基于项目流行度和新颖度分类特征的托攻击检测算法。本文的主要贡献如下:

1)基于项目流行度和项目新颖度,提出了 7 个不依赖于项目具体评分值大小的托攻击检测特征。

2)根据提出的检测特征,提出了一种基于 Boosting 技术的托攻击集成检测框架。

3)在 MovieLens 数据集上进行了对比实验,验证了本文所提方法的有效性。

1　相关工作

托攻击检测算法大致可以分为两种:有监督检测算法和无监督检测算法。现有的有监督检测方法中,Chirita 等[3]提出了几个简单的统计指标,在此基础上提出两种简单的攻击检测算法来达到检测攻击概貌的目的。Williams 等[4]提出一些通用特征和专用特征,并利用经典的机器学习算法对攻击概貌进行检测。基于文献[4]中提出的特征,He 等[5]提出了一种基于粗糙集的检测方法。基于 Williams 等[4]提出的特征,伍之昂等[6]针对不同的攻击模型提出了一种特征选择算法。李文涛等[7]提出了一种基于流行度分类特征的托攻击检测算法,一定程度上提高了检测精度。Zhou[8]通过利用文本分类技术提取特征,提出了一种有监督检测方法。Zhang 等[9]提出了一种基于元学习的集成检测算法,利用集成学习技术有效提升了检测精度。Zhang 等[10]通过构建有序项目序列,提出了一种基于决策树的集成检测算法,一定程度上提高了检测性能。

无监督检测方法中,Zhang 等[11]利用奇异值分解技术得到一个低维线性模型,依据信任分歧计算概貌之间的信任程度来检测攻击概貌。Mehta 等[12]基于主成分分析提出了一种无监督检测算法,在检测标准攻击时具有较好的效果。Bryan 等[13]基于攻击概貌之间的高相似性利用聚类技术提出了一种无监督检测算法。李聪等[14]提出了一种缺失评分潜在因素分析模型。Chung 等[15]提出了一种基于 β 分布的检测方法。文献[16]利用信息熵提出了一种基于多维风险因子的推荐攻击检测方法。周全强等[17]依据项目流行度划分窗口,提出了一种基于仿生模式识别的未知托攻击检测方法。

2　基于流行度和新颖度分类的托攻击集成检测框架

为了提高攻击检测的性能,本文提出一种基于流行度和新颖度分类的特征提取算法和一个集成检测框架来检测攻击用户。其中,集成检测框架如图 1 所示。在特征提取阶段,通过引入流行度和新颖度,运用特征提取算法将训练集和测试集分别映射到特征空间。在基分类器生成阶段,采用 Boosting[18]技术,将训练集中每个样本分别赋以不同权重,从而保证每轮迭代过程中基训练集的差异,运用 C4.5 算法,生成 k 个不同的基分类器。在攻击检测阶段,通过融合多个带有权重的基分类器的预测结果,得到最终的检测结果。

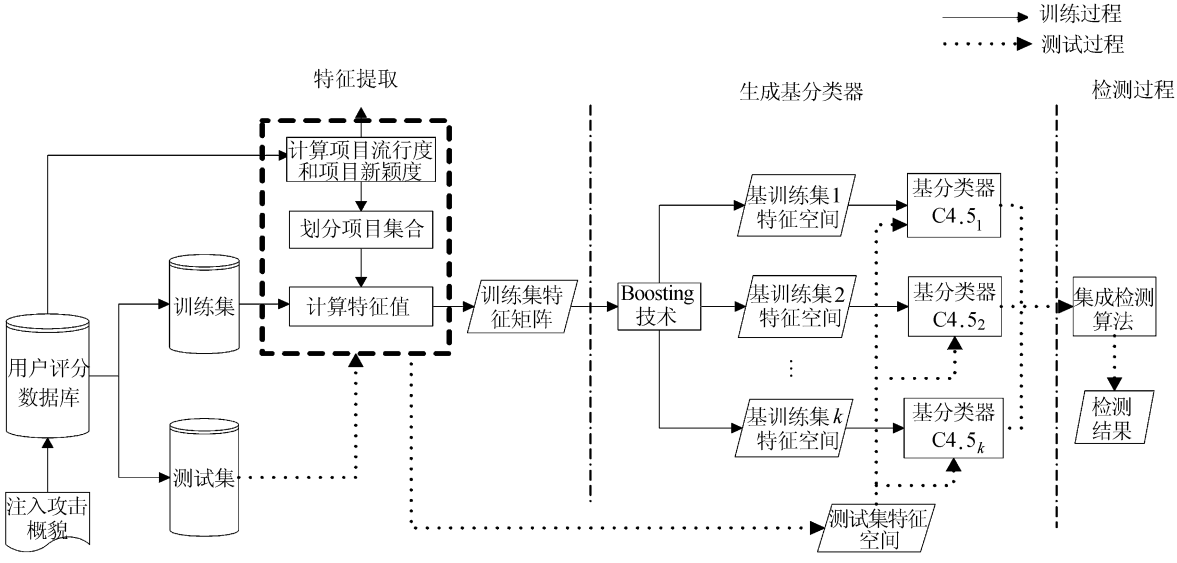


图 1　基于流行度和新颖度分类特征的托攻击集成检测框架

Fig. 1　Ensemble framework for shilling attack detection based on item popularity and novelty degree features

2.1 特征提取

为了得到托攻击检测特征,首先给出项目流行度和项目新颖度的定义。

定义1 项目流行度(item popularity, *IPop*):是指评分数据库中所有真实用户对项目 i 的评分次数,即

$$IPop_i = \sum_{u \in D_g} \psi(r_{u,i}) \tag{1}$$

式中:D_g 表示用户评分数据库中所有真实概貌的集合;$r_{u,i}$ 表示用户 u 对项目 i 的评分值;若 $r_{u,i} = \varnothing$,则 $\psi(r_{u,i}) = 0$;若 $r_{u,i} \neq \varnothing$,则 $\psi(r_{u,i}) = 1$。

定义2 项目新颖度[19](item novelty, *INov*):是指用户 u 购买项目 i 与购买记录中其他项目的不相似程度,即

$$INov_i = \frac{\sum_{u \in D_g, r_{u,i} \neq \varnothing} Nov_{u,i}}{|D_g|} \tag{2}$$

式中:$|D_g|$ 表示 D_g 集合中的数目;$Nov_{u,i}$ 表示用户 u 对某一项目 i 的新颖度,其计算方法如下:

$$Nov_{u,i} = \frac{\sum_{r_{u,j} \neq \varnothing} (1 - w(i,j))}{N_u} \tag{3}$$

式中:N_u 表示用户 u 的评分项目数;$w(i,j)$ 表示项目 i 和项目 j 的相似度,这里为余弦相似度。

根据定义1,将所有项目按照流行度降序排序,采用10折交叉验证将项目分为流行项目集合 I_{pop} 和非流行项目集合 I_{unpop}。同理,根据定义2,将所有项目按照项目新颖度分为新颖项目集合 I_{nov} 和非新颖项目集合 I_{unnov}。

1)流行项目流行度分布(popular item popularity distribution, *PIPD*):是指某用户概貌中,已评分流行项目的流行度占用户流行度的比例,其计算公式为:

$$PIPD_u = \frac{\sum_{i \in I_{\text{pop}}} \delta(r_{u,i})}{\sum_{i \in I} \delta(r_{u,i})} \tag{4}$$

式中,$\delta(r_{u,i})$ 的计算公式如下:

$$\delta(r_{u,i}) = \begin{cases} IPop_i, & r_{u,i} \neq 0; \\ 0, & r_{u,i} = 0 \end{cases} \tag{5}$$

2)非流行项目流行度分布(unpopular item popularity distribution, *UNPIPD*):是指某用户概貌中,已评分非流行项目的流行度占用户流行度的比例,其计算公式为:

$$UNPIPD_u = \frac{\sum_{i \in I_{\text{unpop}}} \delta(r_{u,i})}{\sum_{i \in I} \delta(r_{u,i})} \tag{6}$$

式中,$\delta(r_{u,i})$ 的计算同式(5)。

3)新颖项目新颖度分布(novel item novelty distribution, *NIND*):是指某用户概貌中,已评分新颖项目的新颖度占用户新颖度的比例,其计算公式为:

$$NIND_u = \frac{\sum_{i \in I_{\text{nov}}} \Gamma(r_{u,i})}{\sum_{i \in I} \Gamma(r_{u,i})} \tag{7}$$

式中,$\Gamma(r_{u,i})$ 的计算方式如下:

$$\Gamma(r_{u,i}) = \begin{cases} INov_i, & r_{u,i} \neq 0; \\ 0, & r_{u,i} = 0 \end{cases} \tag{8}$$

4)非新颖项目新颖度分布(unnovel item novelty distribution, *UNIND*):是指某用户概貌中,已评分非新颖项目的新颖度占用户新颖度的比例,其计算公式为:

$$UNIND_u = \frac{\sum_{i \in I_{\text{unnov}}} \Gamma(r_{u,i})}{\sum_{i \in I} \Gamma(r_{u,i})} \tag{9}$$

式中,$\Gamma(r_{u,i})$ 的计算同式(8)。

由于用户在流行项目集合里评分次数较多,在新颖项目集合里评分次数较少,因此可结合文本分类中卡方检验知识,提取检测特征。

χ^2 统计(Chi-square)[20]表征2个统计量之间相互关联的程度。关联程度越大,χ^2 值越大;反之,关联程度越小,χ^2 越小。在文本分类领域,χ^2 统计用来表示特征 t 和类别 c 之间的依赖关系,其近似值为:

$$\chi^2 \approx \frac{N \times (A \times D - C \times B)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{10}$$

式中,A 为特征 t 与 c 类文档同时出现的次数,B 为特征 t 出现而 c 类文档不出现的次数,C 为 c 类文档出现而特征 t 不出现的次数,D 为特征 t 与 c 类文档均不出现的次数,N 为文档总数。若特征 t 与 c 类文档相互独立,那么特征 t 的 χ^2 值为0。

根据式(10),提出以下2个检测特征:

5)流行项目的卡方估计值(Chi-square of popular item, *CHIP*):表示用户 u 的评分项目与流行项目之间的关联程度,计算公式为:

$$CHIP_u = |I| \times (A_{u,I_{\text{pop}}} \times D_{u,I_{\text{pop}}} - C_{u,I_{\text{pop}}} \times B_{u,I_{\text{pop}}})^2 \times \frac{1}{(A_{uI_{\text{pop}}} + C_{uI_{\text{pop}}}) \times (B_{uI_{\text{pop}}} + D_{uI_{\text{pop}}}) \times (A_{uI_{\text{pop}}} + B_{uI_{\text{pop}}}) \times (C_{uI_{\text{pop}}} + D_{uI_{\text{pop}}})} \tag{11}$$

式中,$A_{u,I_{\text{pop}}}$ 表示被用户 u 评过分且属于流行项目 I_{pop} 集合的项目数,$B_{u,I_{\text{pop}}}$ 表示被用户 u 评过分且不

属于流行项目 I_{pop} 集合的项目数,$C_{u,I_{pop}}$ 表示没有被用户 u 评过分但属于流行项目 I_{pop} 集合的项目数,$D_{u,I_{pop}}$ 表示没有被用户 u 评过分且不属于流行项目 I_{pop} 集合的项目数。

6) 新颖项目的卡方估计值(Chi-square of novel item,*CHIN*):表示用户 u 的评分项目与流行项目之间的关联程度,计算公式为:

$$CHIN_u = |I| \times (A_{u,I_{nov}} \times D_{u,I_{nov}} - C_{u,I_{nov}} \times B_{u,I_{nov}})^2 \times \frac{1}{(A_{uI_{nov}} + C_{uI_{nov}}) \times (B_{uI_{nov}} + D_{uI_{nov}}) \times (A_{uI_{nov}} + B_{uI_{nov}}) \times (C_{uI_{nov}} + D_{uI_{nov}})} \tag{12}$$

式中,$A_{u,I_{nov}}$ 表示被用户 u 评过分且属于流行项目 I_{nov} 集合的项目数,$B_{u,I_{nov}}$ 表示被用户 u 评过分且不属于流行项目 I_{nov} 集合的项目数,$C_{u,I_{nov}}$ 表示没有被用户 u 评过分但属于流行项目 I_{nov} 集合的项目数,$D_{u,I_{nov}}$ 表示没有被用户 u 评过分且不属于流行项目 I_{nov} 集合的项目数。

从用户评分角度发现:①*DegSim* 相似度计算中,用户 u 与 v 之间的相似度不能等同于用户 v 与 u 之间的相似度;② 计算中没有考虑评分为空的值。因此,通过改进 *DegSim* 这一特征,提出针对低填充规模攻击的检测特征。

7) 相似度权重均值(weighted degree of average similarity,*WDSA*):指用户与 k 近邻用户之间的平均相似度,其计算公式为:

$$WDSA_u = \frac{1}{k} \sum_{u=1}^{k} sim(u,v) \times \frac{co\_rateitem(u,v)}{N_u} \tag{13}$$

式中:$sim(u,v)$ 为皮尔逊相似度,这里需要将评分为空的值当成 0 参与到计算中;$co\_rateitem(u,v)$ 表示用户 u 和用户 v 共同评分的项目个数;N_u 的含义同定义 2;k 为用户的最近邻数,本文选取 $k = 25$。

基于以上分析,给出特征提取算法的描述如下:

算法 1　特征提取算法

输入:D_g,U,I;

输出:特征矩阵 $\boldsymbol{V}$。

BEGIN

(1) for $i \in I$ do
(2)　for $u \in D_g$ do
(3)　　$IPop_i \leftarrow$ 计算每个项目的流行度
(4)　　$INov_i \leftarrow$ 计算每个项目的新颖度
(5)　end for
(6) end for
(7) $I \leftarrow \{I_{pop}, I_{unpop}\}$ /* 按流行度划分项目集合 */
(8) $I \leftarrow \{I_{nov}, I_{unnov}\}$ /* 按新颖度划分项目集合 */
(9) for $u \in U$ do
　　$V_u \leftarrow \{PIPD_u, UNPIPD_u, NIND_u, UNIND_u, CHIP_u, CHIN_u, WDSA_u\}$ /* 根据式(4) ~ (13) 计算特征 */
(10)　$V \leftarrow V \cup V_u$
(11) end for
(12) return V

END

算法 1 中第 1 ~ 6 行是计算每个项目的流行度和新颖度,然后按照项目流行度降序排序,把所有项目分为流行项目集合和非流行项目集合(第 7 行),同理,按照项目新颖度降序,把所有项目分为新颖项目集合和非新颖项目集合(第 8 行)。第 9 ~ 11 行利用式(4) ~ (13) 对每个用户 u 计算 *PIPD*、*UNPIPD*、*NIND*、*UNIND*、*CHIP*、*CHIN*、*WDSA*,最后返回所有用户特征向量组成的特征矩阵 $\boldsymbol{V}$(第 12 行)。

2.2　基于流行度和新颖度的集成检测算法

2.2.1　基分类器的生成

采用 Boosting[18] 技术,首先将映射到特征空间的训练集中每个样本赋一个初始权重,然后将带有权重的训练集数据用 C4.5 进行训练,得到一个基分类器。通过检测结果得到每个样本的分类错误率 ε_i,进而得到基分类器的分类权重 α_i。根据基分类器的分类权重得到下一次迭代中每个样本的权值,然后再进行训练得到另一个基分类器。基分类器生成算法描述如下:

算法 2　基分类器生成算法

输入:训练集 *train*,迭代次数 T,基分类器 C4.5,类标签集 $y_i \in \{0,1\}$;

输出:基分类器集 *C4.5set*,基分类器权重集 α。

BEGIN

(1) 利用算法 1 的特征提取算法计算训练集的特征矩阵 $\boldsymbol{V}_{train}$;
(2) 将基分类器集 *C4.5set* 初始化为空集;
(3) 将每个样本的初始权重 w_j^1 设为 $1/m$,$j = 1,2,\cdots,m$;
(4) for $i = 1$ to T do
(5)　对带有权值 w_j^i 的训练集特征矩阵 $\boldsymbol{V}_{train}$ 用 C4.5 算法进行分类,得到第 i 个基分类器 $C4.5set^i$;
(6) 计算第 i 个分类器的分类错误率:

$$\varepsilon_i = \frac{1}{m}\left[\sum_j w_j^i \zeta(C_i(\boldsymbol{V}_{\text{train}j}) \neq y_j)\right];$$

(7) 依据分类器的分类错误率，计算第 i 个分类器的分类权重 α_i；

(8) 依据第 i 个分类器的分类权重更新下一轮样本的权重 w_j^{i+1}；

(9) 将训练好的基分类器放到基分类集中，即

$$C4.5set \leftarrow C4.5set \cup \{C4.5set^i\};$$

(10) 将每个基分类器的权重放到权重集合中，即

$$\alpha \leftarrow \alpha \cup \{\alpha_i\};$$

(11) end for

(12) return $C4.5set, \alpha$

END

2.2.2 托攻击检测算法描述

融合多个基分类器的预测结果，得到最终的检测结果。检测算法如下：

算法 3 基于流行度和新颖度分类的集成检测算法

输入：测试集 $test$，$C4.5set$，α，T；

输出：用户类标签集 T_{result}。

BEGIN

(1) 利用算法 1 的特征提取算法计算测试集的特征矩阵 $\boldsymbol{V}_{\text{test}}$；

(2) for each $u \in Vtest$ do

(3) for each $C4.5set^i \in C4.5set$ do

(4) 将特征矩阵 $\boldsymbol{V}_{\text{test}}$ 用已训练好的基分类器 $C4.5set^i$ 检测，得到预测结果 $pre_i \leftarrow pre_{C4.5set^i}(u)$；

(5) 将用户 u 的预测结果和分类器权重融合，即 $p(u) \leftarrow \alpha_i pre_i$；

(6) end for

(7) 组合基分类器对每个样本的预测结果，即 $result(u) \leftarrow sum(p(u))$；

(8) 与阈值 $1/2sum(\alpha)$ 进行比较，确定用户 u 的类别 $T(u)$；

(9) end for

(10) $T_{\text{result}} \leftarrow T_{\text{result}} \cup \{T(u)\}$；

(11) return T_{result}

END

3 实验数据与评价

3.1 实验数据与设置

实验选用 MovieLens 100K 数据集，该数据集由 943 个用户对 1 682 部电影的 100 000 条评分数据构成，且每个用户至少对 20 部电影进行了评分。

实验中，将数据集中的用户平均分成 2 组，分别作为训练集和测试集中的真实用户概貌。训练集和测试集中的攻击概貌分别由 3 种攻击模型（随机攻击、均值攻击和流行攻击）按照一定的填充规模和攻击规模生成，所产生的攻击概貌均为推攻击。填充规模分别为 1%、3% 和 5%；攻击规模分别为 3%、5%、10% 和 12%。

3.2 评价标准

本文采用托攻击检测中常用的准确率（*precision*）、召回率（*recall*）作为检测方法的评价指标。其定义为：

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$recall = \frac{TP}{TP + FN} \tag{16}$$

式中，TP 表示被正确检测的攻击概貌的数目，FP 表示被误判的真实概貌的数目，FN 表示未被检测出来的攻击概貌的数目。

3.3 实验结果与分析

3.3.1 对比算法

为了评价算法的性能，将本文提出的检测算法（简称 EMDegreeJ48）与 2 个集成检测算法进行实验对比。

1）EMJ48-6：选取 *RDMA*、*WDMA*、*WDA*、*DegSim*、*MeanVar*[4] 和 *Hv*[13] 作为检测特征，利用决策树 C4.5 算法作为基分类器的 Adaboost 算法。

2）EMSVM-6：选取 *RDMA*、*WDMA*、*WDA*、*DegSim*、*MeanVar*[4] 和 *Hv*[13] 作为检测特征，利用支持向量机（SVM）作为基分类器的 Adaboost 算法。

3.3.2 信息增益

信息增益（IG）[21] 通常用来评价特征对分类系统的重要程度，信息增益越大，说明该特征对分类系统的贡献越大，也就是该特征越重要。本文所提特征在测试集上的平均信息增益如表 1 所示。

表 1 特征的平均信息增益

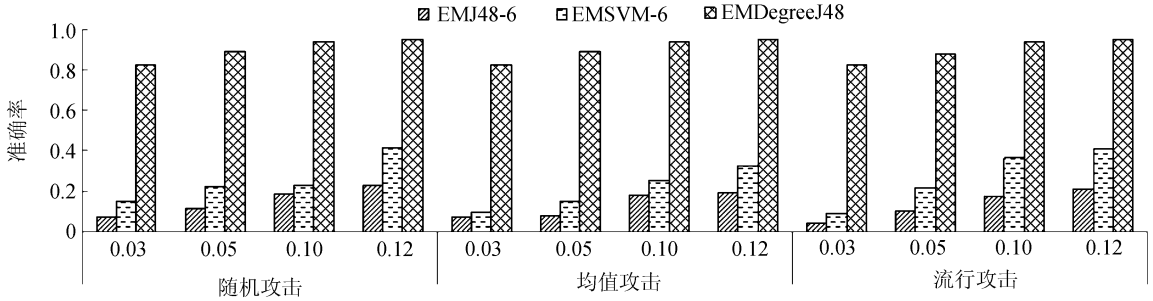
Tab. 1 Features of average information gain

Feature	Random 攻击		Average 攻击		Bandwagon 攻击	
	IG	rank	IG	rank	IG	rank
PIPD	0.298 1	4	0.296 7	4	0.238 8	5
UNPIPD	0.298 1	5	0.296 7	5	0.238 8	6
NIND	0.314 3	2	0.313 2	2	0.314 7	2
UNIND	0.314 3	3	0.313 2	3	0.313 8	3
CHIP	0.329 0	1	0.321 6	1	0.320 8	1
CHIN	0.251 3	6	0.252 7	6	0.250 7	4
WDSA	0.205 4	7	0.190 1	7	0.169 7	7

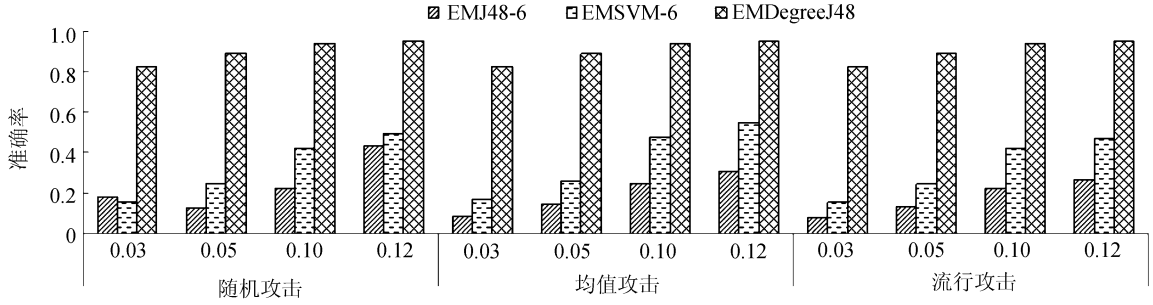
从表 1 中可以看出:在检测 3 种攻击类型时,*CHIP* 的信息增益位居首位,说明在标准攻击下,该指标对分类系统的贡献最大,比其他特征重要。*NIND* 和 *UNIND* 2 个指标均排在前 3 位,说明它对于分类系统的重要性也是不可小觑的。对于 *WDSA* 这一特征指标,虽然它在低填充规模下,起到了相当大的作用,但是随着填充规模的增加,对于分类系统的贡献在减小,因此在测试集上的平均信息增益相比于其他特征表现的值较小,所以排名靠后。

3.3.3　准确率对比

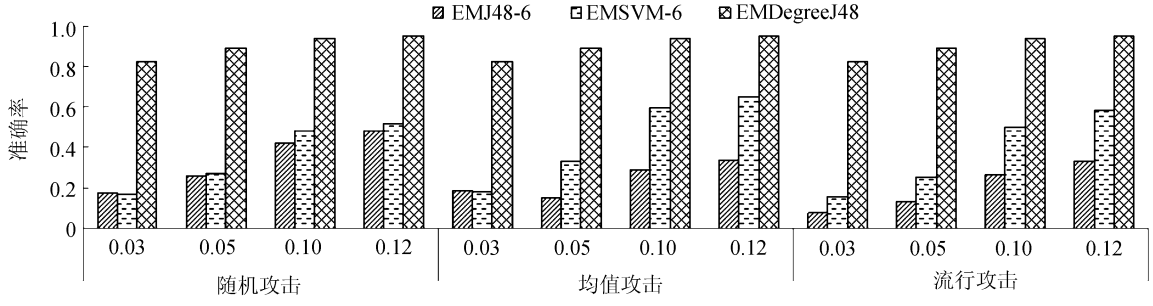
图 2 给出了 EMDegreeJ48、EMJ48-6 和 EMSVM-6 算法在不同攻击类型、不同填充规模和攻击规模下的检测准确率对比。从图 2 可以看出,EMDegreeJ48 算法在 3 种攻击下的准确率均明显高于 EMJ48-6 和 EMSVM-6 算法。以图 2(b)为例,对于填充规模为 3% 的均值攻击,EMDegreeJ48 算法的准确率在 0.824 到 0.95 之间,EMJ48-6 算法的准确率在 0.086 到 0.306 之间,EMSVM-6 算法的准确率在 0.167 到 0.548 之间。这些结果表明本文所提出的检测特征时有效的。从图 2(a)可以看出,对于填充规模为 1% 的随机攻击,EMDegreeJ48 算法的准确率在 0.824 到 0.948 之间,而 EMJ48-6 的准确率在 0.069 到 0.225 之间,EMSVM-6 算法的准确率在 0.149 到 0.416 之间,说明 EMDegreeJ48 算法在低填充规模下仍能有效检测攻击概貌,进一步说明了本文所提出特征的有效性。



(a) 填充规模为 1 % 的准确率



(b) 填充规模为 3 % 的准确率



(c) 填充规模为 5 % 的准确率

图 2　3 种检测算法在不同填充规模和攻击规模的各种攻击下的准确率对比

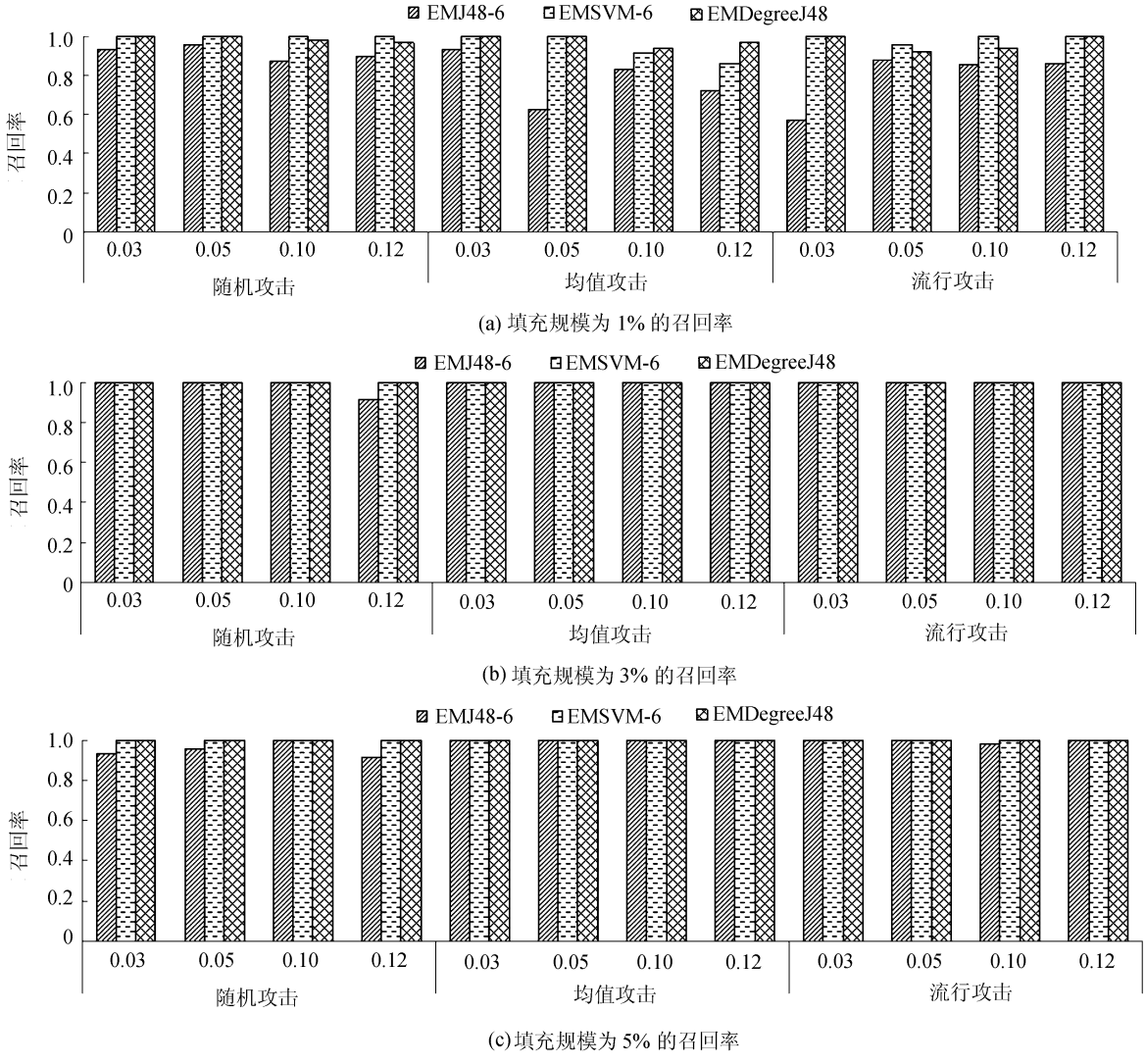
Fig. 2　Precision of three algorithms under various attacks with different filler sizes and attack sizes

从图 2 不难看出,不管是哪一种攻击,在同一填充规模下,随着攻击规模的增大,3 种算法的检测准确率都在上升。以图 2(c)为例,对于填充规模为 5% 的流行攻击,EMDegreeJ48 算法的准确率从 0.824 逐步上升到 0.95,EMJ48-6 算法的准确率从 0.075 上升到 0.331,EMSVM-6 算法的准确率从

0.157上升到0.585 2。因此可以得出结论:随着攻击规模的增大,攻击概貌更容易被检测算法识别。

3.3.4 *召回率对比*

图3给出了EMDegreeJ48、EMJ48-6和EMSVM-6算法在不同攻击类型、不同填充规模和攻击规模下的召回率对比。从图3可以看出,EMDegreeJ48算法在填充规模为3%和5%时,召回率都达到了1。虽然在1%的填充规模下,召回率不全为1,但是随机攻击和均值攻击的召回率最低也达到了0.939,流行攻击的召回率最低也达到了0.917,这说明EMDegreeJ48算法能够检测出大部分攻击概貌。而EMJ48-6算法在检测填充规模为1%的均值攻击时,召回率分别为0.929、0.625、0.83和0.719,召回率最低时为0.625,说明有一部分攻击概貌没被检测出来。对于EMSVM-6算法,召回率几乎都是1,说明该算法和EMDegreeJ48算法一样能有效识别攻击概貌。



(a) 填充规模为1%的召回率

(b) 填充规模为3%的召回率

(c)填充规模为5%的召回率

图3 3种检测算法在不同填充规模和攻击规模的各种攻击下的召回率对比

Fig. 3 Recall of three algorithms under various attacks with different filler sizes and attack sizes

4 结论及进一步工作

从用户评分分布的角度,提出了一种基于项目流行度和项目新颖度分类特征的托攻击检测算法。该算法通过引入项目流行度和新颖度,提出了不依赖于评分值差异的特征,基于这些特征,提出了一种有监督检测算法EMDegreeJ48。在MovieLens 100K数据集上的实验结果表明,本文提出的检测算法能够有效减少对真实概貌的误判,提高了对攻击概貌的检测精度。

下一步工作将针对伪装能力更强的模糊攻击,从信号处理的角度提取能够有效区分真实概貌和攻击概貌的特征指标,实现对模糊攻击的检测。

参考文献:

[1] Yuan W, Guan D, Lee Y K, et al. The small-world trust network[J]. Applied Intelligence, 2011, 35(3): 399 – 410.

[2] Lam S K, Riedl J. Shilling recommender Systems for fun and profit[C]//Proceedings of the 13th International Conference on World Wide Web. New York: ACM, 2004: 393 – 402.

[3] Chirita P A, Nejdl W, Zamfir C. Preventing shilling attacks in online recommender systems[C]//Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management. New York: ACM, 2005: 67 – 74.

[4] Williams C A, Mobasher B, Burke R, et al. Detecting profile injection attacks in collaborative filtering: A classification-based approach[C]//International Workshop on Knowledge Discovery on the Web. Berlin: Springer, 2006: 167 – 186.

[5] He F, Wang X, Liu B. Attack detection by rough set theory in recommendation system[C]//2010 IEEE International Conference on Granular Computing. Los Alamitos: IEEE, 2010: 692 – 695.

[6] Wu Zhiang, Zhuang Yi, Wang Youquan, et al. Shilling attack detection based on feature selection for recommendation systems[J]. Acta Electronica Sinica, 2012, 40(8): 1687 – 1693. [伍之昂,庄毅,王有权,等. 基于特征选择的推荐系统托攻击检测算法[J]. 电子学报, 2012, 40(8): 1687 – 1693.]

[7] Li Wentao, Gao Min, Li Hua, et al. An shilling attack detection algorithm based on popularity degree features[J]. Acta Automatica Sinica, 2015, 41(9): 1563 – 1576. [李文涛,高旻,李华,等. 一种基于流行度分类特征的托攻击检测算法[J]. 自动化学报, 2015, 41(9): 1563 – 1576.]

[8] Zhou Q. Supervised approach for detecting average over popular items attack in collaborative recommender systems[J]. IET Information Security, 2016, 10(3): 134 – 141.

[9] Zhang F, Zhou Q. A meta-learning-based approach fordetecting profile injection attacks in collaborative recommender systems[J]. Journal of Computers, 2012; 7(1): 226 – 234.

[10] Zhang F, Chen H. An ensemble method for detecting shilling attacks based on ordered item sequences[J]. Security and Communication Networks, 2016, 9(7): 680 – 696.

[11] Zhang S, Ouyang Y, Ford J, et al. Analysis of a low-dimensional linear model under recommendation attacks[C]//Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2006: 517 – 524.

[12] Mehta B, Hofmann T. A survey of attack-resistant collaborative filtering algorithms[J]. IEEE Data Engineering Bulletin, 2008, 31(2): 14 – 22.

[13] Bryan K, O'Mahony M, Cunningham P. Unsupervised retrieval of attack profiles in collaborative recommender systems[C]//Proceedings of the 2008 ACM Conference on Recommender Systems. New York: ACM, 2008: 155 – 162.

[14] Li Cong, Luo Zhigang. Detecting shilling attacks in recommender systems based on non-random-missing mechanism[J]. Acta Automatica Sinica, 2013, 39(10): 1681 – 1690. [李聪,骆志刚. 基于数据非随机缺失机制的推荐系统托攻击探测[J]. 自动化学报, 2013, 39(10): 1681 – 1690.]

[15] Chung C Y, Hsu P Y, Huang S H. βP: A novel approach to filter out malicious rating profiles from recommender systems[J]. Decision Support Systems, 2013, 55(1): 314 – 325.

[16] Yu Hongtao, Li Peng, Zhang Fuzhi. Method for detecting recommendation attack based on multiple risk factors[J]. Journal of Chinese Computer Systems, 2015, 36(5): 971 – 975. [于洪涛,李鹏,张付志. 基于多维风险因子的推荐攻击检测方法[J]. 小型微型计算机系统, 2015, 36(5): 971 – 975.]

[17] Zhou Quanqiang, Zhang Fuzhi. Ensemble approach for detecting user profile attacks based on bionic pattern recognition[J]. Journal of Computer Research and Development, 2014, 51(4): 789 – 801. [周全强,张付志. 基于仿生模式识别的用户概貌攻击集成检测方法[J]. 计算机研究与发展, 2014, 51(4): 789 – 801.]

[18] Yu Ling, Wu Tiejun. Assemble learning: A survey of boosting algorithms[J]. Pattern Recognition and Artificial Intelligence, 2004, 17(1): 52 – 59. [于玲,吴铁军. 集成学习: Boosting 算法综述[J]. 模式识别与人工智能, 2004, 17(1): 52 – 59.]

[19] Hurley N, Zhang M. Novelty and diversity in top-*N* recommendation—Analysis and evaluation[J]. ACM Transactions on Internet Technology (TOIT), 2011, 10(4): 14: 1 – 14: 30.

[20] Qu Jun, Lin Xu. Comparison and analysis of feature extraction methods for text categorization[J]. Modern Computer, 2007(4): 10 – 13. [屈军,林旭. 文本分类中特征提取方法的比较与分析[J]. 现代计算机, 2007(4): 10 – 13.]

[21] Ambert K H, Cohen A M. *K*-information gain scaled nearest neighbors: A novel approach to classifying protein-protein interaction-related documents[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2012, 9(1): 305 – 310.

(编辑　杨　蓓)

引用格式:Yu Hongtao, Zhou Qiannan, Zhang Fuzhi, et al. An shilling attack detection algorithm based on item popularity and novelty degree features[J]. Advanced Engineering Sciences, 2017, 49(1): 176 – 183. [于洪涛,周倩楠,张付志,等. 基于项目流行度和新颖度分类特征的托攻击检测算法[J]. 工程科学与技术, 2017, 49(1): 176 – 183.]