POSTER: Scanning-free Personalized Malware Warning System by Learning Implicit Feedback from Detection Logs

Jyun-Yu Jiang[†], Chun-Liang Li[†], Chun-Pai Yang[†], Chung-Tsai Su[‡]
[†]Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
[‡]Trend Micro Inc., Taiwan
jyunyu.jiang@gmail.com, {r01922001, b99902109}@csie.ntu.edu.tw,
chungtsai_su@trend.com.tw

ABSTRACT

Nowadays, World Wide Web connects people to each other in many ways ubiquitously. Followed along with the convenience and usability, millions of malware infect various devices of numerous users through the web every day. In contrast, traditional anti-malware systems detect such malware by scanning file systems and provide secure environments for users. However, some malware might not be detected by traditional scanning-based detection systems due to hackers' obfuscation techniques. Also, scanning-based approaches cannot caution users for uninfected malware with high risks. In this paper, we aim to build a personalized malware warning system. Different from traditional scanning-based approaches, we focus on discovering the potential malware which has not been detected for each user. If users and the system know the potentially infected malware in advance, they can be alert against the corresponding risks. We propose a novel approach to learn the implicit feedback from detection logs and give a personalized risk ranking of malware for each user. Finally, the experiments on real-world detection datasets demonstrate the proposed algorithm outperforms traditional popularity-based algorithms.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering; K.6.5 [**Management of computing and information systems**]: Security and Protection

Keywords

Computer Security; Malware Detection; Malware Warning System; Personalized Collaborative Filtering.

1. INTRODUCTION

The concept of malware has been studied for a long time [1]. The malware, abbreviated from the malicious software, is the software which intentionally injures or harms users' systems or devices such as computer viruses and worms. A usual way to detect and prevent the malware is using anti-malware software or systems. An anti-malware system monitors users' devices in many ways, such as analyzing network packets and scanning users' file system [8]. Although the anti-malware software can recognize malware patterns well from contents of packets or binaries, there are still some drawbacks. First, a scan may be time-consuming when there are myriad and large files in the file system [2]. Second, some computer malware may still be undetected due to undiscovered variations with distinct binary codes and incorrect usages of users. For example, a user might accidentally turn off the firewall so that the anti-malware software cannot work sufficiently. Furthermore, all scanning-based systems cannot detect any "nonexistent" malware because the devices have not been infected. Therefore, a warning system can detect uninfected malware with high risk of future infection without scanning is desired.

In this work, we focus on building a personalized malware warning system without any traditional scanning technique. We present a novel approach based on detection logs and collaborative filtering (CF) algorithms to tackle the malware detection problem. Although CF algorithms have been widely used in recommender systems [3, 7], applying CF algorithms to detect malware has not been well-studied. Furthermore, CF algorithms can detect malware without scanning, so it is possible to discover the "future" malware. That is, we can help the user to identify high-risk existing malware before the infection. As the evidences, experiments on large-scale real-world detection logs show that CF is useful for malware detection and malware warning system.

2. PROBLEM STATEMENT

Given a set of users U and a set of known malware W, each entry (u, w) in the malware detection logs $S \subseteq U \times W$ represents that the user u was infected by the malware w. For each user u, the personalized malware warning systems aims to estimate the infection risks $r(u, w_i)$ and $r(u, w_j)$ such that $r(u, w_i) > r(u, w_j)$ for all w_i with the higher risk of user u than w_j. Then the system ranks the malware for each user by the estimated risks.

Consider that risks $r(u, w)$ can fill in a $|U|$ by $|W|$ matrix R. Each entry in the detection logs is treated as the explicit feedback, which includes the fact of infection, malware's characteristics and user's security risks [3]. Usually, R is sparse because the detection logs could only provide limited explicit feedback. In our detection logs, each user has only 6.28 detection entries averagely. Therefore, we need to infer the implicit feedback which indirectly reflects the "preferences" of malware and users from provided explicit feedback. The task can be treated as a one-class collaborative filtering problem [4]. By applying CF algorithms,

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.
ACM 978-1-4503-2957-6/14/11.
http://dx.doi.org/10.1145/2660267.2662359.

we could estimate the other entries in matrix R and give a satisfactory ranking of malware for each user.

Related Work. In addition to matching the binaries with lists of malicious patterns [8], there are many previous work applying machine learning techniques into the malware detection [9, 10] and classification [6] in recent years. Ye *et al.* analyzed the Windows API execution sequences and developed a malware detection system with some rule-based classifiers [10]. Tahan *et al.* proposed a detection algorithm with several features extracted from the segments of executable files and malware's meta-data [9]. Rieck *et al.* cluster malware into several classes with behaviors and assign unknown malware to discovered classes [6]. However, all of them analyzed and detected malware in binary-level or behavior-level. Also, their methods are in need of scanning users' file systems or launching malware in sandbox system. To the best of our knowledge, we are the first to apply collaborative filtering techniques into malware detection without any scan.

3. PROPOSED APPROACH

The CF algorithms for recommender systems assume that users with similar interests may like similar items, and vice versa. This assumption is also applicable to the malware detection problem. Users with similar habits may have same risks of infection of analogous malware. On the other hand, the malware with same infection channels may infect similar users who are careless about such risks. The similarity suggests that CF algorithms may be valid to the malware detection problem. Therefore, we proposed a novel CF based approach which consists of three parts: (1) most popular prediction, (2) matrix factorization and (3) hybrid prediction.

3.1 Most Popular Prediction

We first propose the popularity-based method called "Most Popular Prediction" (MPP). MPP predicts values with the occurrence of malware in training detection logs as follows:

$$\mathrm{MPP}(u, w) = \sum_{u \in U} \mathit{infected}(u, w),$$

where $\mathit{infected}(u, w)$ shows how many times the user u is infected by malware w in training detection logs. This algorithm can be treated as an Maximum Likelihood Estimator. It assumes there is a probability distribution of infected malware. In an ideal world, the system could have known such distribution and obtain excellent prediction performance. Besides, MPP is not dependent on user's individual logs, and thus can perform well even if there are only few logs for a user without the representativeness. It is also the baseline method in performance comparison.

3.2 Matrix Factorization

When the provided logs are more plentiful, the latent factor CF algorithms could work well. Many state-of-the-art latent factor CF algorithms are based on matrix factorization (MF) [3]. An important assumption of MF is that is the low-rank assumption of the utility matrix. Then the problem can be treated as the task approximating the predicted matrix R with the product of a matrix $P : |U| \times k$ and a matrix $Q : |W| \times k$ as follows:

$$\hat{R} = P \times Q^T,$$

where k is the dimension of the latent factors. Each row p_u in P can be treated as the latent factor or feature factor of the user u. Similarly, each row q_w in Q is the latent factor of the malware w. Then we can predict the value $\hat{r}(u, w)$ for a user-malware pair (u, w) with the inner product $p_u \times q_w^T$ in the latent factor space.

As mentioned in Section 2, the problem is a one-class CF problem so that we have no any numerical rating. We have only the instances of positive class, i.e., the infected malware in logs. Actually, our goal is actually to rank malware with higher risks in higher positions. In this work, we adopt the framework of Bayesian personalized ranking (BPR) [5] with MF optimization. Based on the assumption of BPR, we assume the infected malware has higher risks than other malware without infection for a user, thereby creating training data $D_S : U \times W \times W$ as follows:

$$D_S = \left\{(u, w_i, w_j) \mid w_i \in W_u^+, w_j \in W \setminus W_u^+\right\}.$$

Here W_u^+ represents the set of infected malware for the user u. For each user, BPR optimizes the pairwise error between infected and uninfected malware for personalized ranking.

To adopt the scenario of malware ranking, we model the probability that a malware w_i really has higher risk than w_j for a user u according to the BPR optimization criterion [5] as follows:

$$P\left(r(u, w_i) > r(u, w_j)\right) = \frac{1}{1 + e^{-\left(\hat{r}(u, w_i) - \hat{r}(u, w_j)\right)}}.$$

Here $\hat{r}(u, w_i)$ is the predicted risk of w_i for the user u. Hence, we can learn the model by maximizing the log-likelihood over the training data D_S and estimating parameters. The maximizing procedure is converted to solve the objective as follows:

$$\min \sum_{(u, w_i, w_j) \in D_S} \ln\left(1 + e^{-\left(\hat{r}(u, w_i) - \hat{r}(u, w_j)\right)}\right) + \frac{1}{2}||U||^2 + \frac{1}{2}||W||^2.$$

The latter two terms in the equation are L2 regularization for reducing overfitting. Because the number of possible uninfected malware w_j is large, the sampling techniques are utilized in the training procedure. Therefore, we can optimize the malware ranking for each user. It can also be considered as involving the prior probabilities for model parameters in a Bayesian view. The results of BPR with MF are denoted as BPR-MF in this paper.

3.3 Hybrid Prediction

Both MF and MPP have their advantages in different cases. Hybrid prediction aims to predict values by aggregating predictions of two proposed methods. As two kinds of predictions are in different scales, they need to be standardized before aggregation. We estimate the mean value and standard deviation to standardize predictions. For example, the standardized value of MF predictions can be calculated as follows:

$$\mathit{stdMF}(u, w) = \frac{\hat{r}(u, w) - \hat{\mu}_u}{\hat{\sigma}_u},$$

where $\hat{\mu}_u$ and $\hat{\sigma}_u$ are mean and standard deviation estimated by predicted values for the user u. The standardized MPP predictions can be calculated similarly. Then we can aggregate predictions of two methods as:

$$\mathit{Hybrid}(u, w) = \alpha \cdot \mathit{stdMF}(u, w) + (1 - \alpha) \cdot \mathit{stdMPP}(u, w),$$

where $0 \leq \alpha \leq 1$ is a parameter determining the weights of two methods. Note that when $\alpha = 1$ the hybrid prediction is identical to the MF method, and when $\alpha = 0$ the hybrid prediction is identical to the MPP method. The results of hybrid prediction are denoted as Hybrid.P in this paper.

4. EXPERIMENTS

Our experimental data comprises malware detection logs provided by Trend Micro between 6 June, 2013 and 10 June, 2013. For some rare malware and users, we remove some entries such that every user and malware has at least five entries. After filtering, there are 292,113 users and 13,781 malware in 1,880,212 detection entries. We separate detection entries into two datasets with the same size, training data S_{train} and testing data S_{test}. After separating, there are 939,658 entries in S_{train} and 940,554 in S_{test}. For each user u, every method will give a ranking to all malware which did not infect the user u in S_{train}. Our aim is to measure how a method can rank infected or potential malware with a higher position. With the ground truth in S_{test}, we evaluate the quality of rankings with two evaluation measures, including *mean reciprocal rank* (MRR) and *Normalized Discounted Cumulative Gain at position k* (NDCG@k). MRR evaluates the position of the first infected malware in the ranked list. NDCG@k evaluates the overall performance of top-k predictions. While calculating NDCG values, the infected malware will be given score 5, and the uninfected ones gain 0. Instead of using MAP as the evaluation measure, we utilize NDCG because the false alarms (i.e., uninfected predictions) should be penalized.

Table 1 shows the performance of methods. BPR-MF outperforms the MPP baseline in MRR and NDCG at latter positions. MPP is better than BPR-MF in NDCG@1, which represents the accuracy of the first prediction. The reason is that popular malware infects most users. However, less popular malware cannot be predicted well by MPP, thus MPP is worse than BPR-MF in other measures. Aggregating the benefits of two methods, hybrid prediction has the best performance in all measures. The high MRR value shows that the first infected malware is ranked in top-2 positions averagely for each user. The NDCG results also show our approach has good performance in general predictions.

Table 1: Performance Comparison of Methods, K means the dimension of latent factors and α means the parameter in the hybrid prediction.

Method	MRR	NDCG@1	NDCG@5	NDCG@10
MPP (Baseline)	0.5763	**0.4667**	0.2524	0.2603
BPR-MF ($K = 5$)	0.5825	0.4288	0.3013	0.3225
BPR-MF ($K = 10$)	0.5855	0.4381	0.3096	0.3277
BPR-MF ($K = 15$)	**0.5898**	0.4390	**0.3110**	**0.3310**
Hybrid.P ($\alpha = 0.8$)	**0.6245**	**0.5010**	**0.3154**	**0.3380**

Then we analyze the aggregation of two methods in the hybrid prediction. Figure 1 shows the performance of MRR and NDCG@1 with different α in the hybrid prediction. Recall that when $\alpha = 0$ the hybrid prediction is identical to MPP, and when $\alpha = 1$ the hybrid prediction is identical to BPR-MF. The lager α for BPR-MF generally results in better performance. It implies that BPR-MF is more important in aggregation, but MPP still benefits the overall performance in both MRR and NDCG in the hybrid prediction model.

5. CONCLUSIONS AND FUTURE WORK

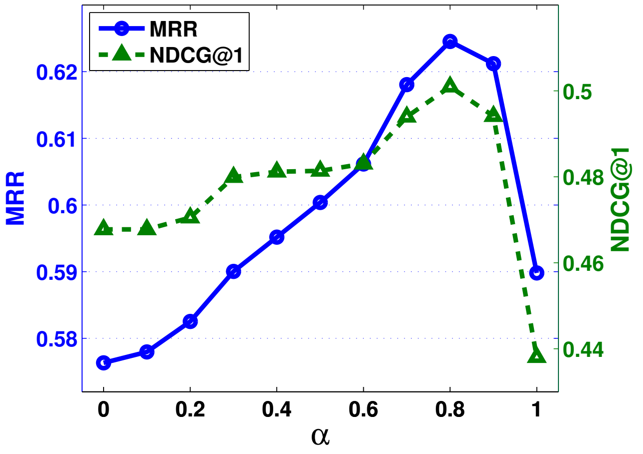

Figure 1: The performance of MRR and NDCG@1 with different α in the hybrid prediction.

In this paper, we first present a framework for personalized malware warning system by matrix factorization techniques of collaborative filtering. Furthermore, we propose the hybrid prediction method aggregating popularity-based approach and matrix factorization approach. The methods are evaluated by real-world malware detection logs and have convincing performance for discovering infecting malware for users individually. Our future work is to combine more meta-data of users and malware into current methods.

6. REFERENCES

[1] F. Cohen. Computer viruses: theory and experiments. *Computers & security*, 6(1):22–35, 1987.

[2] M. Egele, T. Scholte, E. Kirda, and C. Kruegel. A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys (CSUR)*, 44(2):6, 2012.

[3] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[4] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proceedings of the Eighth IEEE International Conference on Data Mining*, 2008.

[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI '09*, pages 452–461, 2009.

[6] K. Rieck, P. Trinius, C. Willems, and T. Holz. Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 2011.

[7] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

[8] P. Szor. *The art of computer virus research and defense*. 2005.

[9] G. Tahan, L. Rokach, and Y. Shahar. Mal-id: Automatic malware detection using common segment analysis and meta-features. *JMLR*, 2012.

[10] Y. Ye, D. Wang, T. Li, and D. Ye. Imds: Intelligent malware detection system. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1043–1047. ACM, 2007.