PeerMate: A Malicious Peer Detection Algorithm for P2P Systems based on MSPCA

Xianglin Wei
Department of Computer Science and Engineering, PLA University of Science and Technology
Nanjing, China
wei_xianglin@ieee.org

Tarem Ahmed
Department of Computer Science, International Islamic University Malaysia
Kuala Lumpur, Malaysia
tarem@ieee.org

Ming Chen
Department of Computer Science and Engineering, PLA University of Science and Technology
Nanjing, China
cm@plaust.edu.cn

Al-Sakib Khan Pathan
Department of Computer Science, International Islamic University Malaysia
Kuala Lumpur, Malaysia
sakib@iium.edu.my

***Abstract*—Many reputation management schemes have been introduced to assist peers to choose the most trustworthy collaborators in P2P environment where honest peers coexist with malicious ones. These schemes indeed provide some useful information about the reliability of peers, but still suffer from various attacks including slandering, collusion and so on. Consequently, how to detect malicious peers plays a critical role in successful work of these mechanisms, and it will also be our focus in this paper. Firstly, we divide the malicious peers into six categories; secondly, we bring forward PeerMate, a malicious peers detection algorithm based on Multiscale Principal Component Analysis (MSPCA) and Quality of Reconstruction (QR), to detect malicious peers in reputation based P2P systems; finally, we show through simulations that PeerMate can detect malicious peers efficiently and accurately.**

Keywords-component; P2P, MSPCA

I. INTRODUCTION

In order to stimulate peers to contribute resources and to assist peers to select the most trustworthy collaborators, several reputation management schemes have been proposed [1][2]. These schemes try to evaluate the transactions performed by peers and assign reputation values to them to reflect their past behavior features. And these reputation values will be the basis for identifying trustworthy peers to reduce the blindness of peer selection. Although these schemes have been proved to be theoretically attractive, they still have a long way to go before practical deployment. Because they are still faced with various attacks including self-promoting, whitewashing, slandering, collusion [3] and Sybil attack [4]. To simplify the description, these P2P systems with reputation management schemes will be referred to as reputation based P2P (RP2P) systems for short, and those peers who initiate attacks will be referred to as malicious peers, other peers besides malicious ones will be called honest peers.

As a burgeoning field, malicious peer detection has attracted a few researchers' attention in recent years which are detailed in Section III. These methods either concentrate on malicious peers of some particular categories or are based on global assumption, in this work however, we focus on developing a general detection algorithm PeerMate. The main differences between PeerMate and existing methods are: on one hand, PeerMate aims at detecting malicious peers of multi-categories rather than some particular categories; on the other hand, PeerMate is based only on reputation information, which can be collected in many current RP2P systems, rather than global information.

Our contributions in this work are threefold: Firstly, to our best knowledge, this is the first paper that aims at detecting malicious peers from the aspect of signal processing in RP2P systems; secondly, we develop PeerMate to detect malicious peers based on MSPCA; finally, the efficiency of PeerMate is evaluated through simulations.

The rest of the paper is organized as follows. Related work is summarized in Section II, Section III introduces PeerMate, and in Section IV many simulations are conducted to evaluate the performance of PeerMate. Finally, we conclude our main works and further research in Section V.

II. RELATED WORK

Mekouar et al. proposed a Malicious Detector Algorithm in [5] to detect liar peers that send wrong feedback to subvert reputation system. That is, after each transaction between a pair of peers, both peers are required to generate feedback to describe the transaction. If there is an obvious gap between the two pieces of feedback, both are regarded being suspicious.

Ji et al. proposed a group based metric for protecting P2P network against Sybil Attack and Collusion by dividing the whole network into some trust groups based on global structure information which is hard to obtain [6].

In [3], Lian et al. recommended various trace based collusion detection approaches including pair-wise detector and traffic concentration detector with data of Maze file sharing application based on trace analysis.

Recently, an upload entropy scheme is developed by Liu et al. to prevent collusions and further enhance robustness of private trackers' sites [1]. But the threshold of this scheme needs to be settled manually.

Lee et al. put forward a simplified clique detection method to detect the colluders [7], but their method is restricted to colluders who form a clique.

III. PEERMATE

Firstly, we present the detection context and divide malicious peers into six categories, and then introduce PeerMate.

A. Detection Scenario

GRep. Before presenting the detection algorithm, we first describe the detection context GRep, which is derived from current popular RP2P systems, such as TVTorrents (www.tvtorrents.com), EigenTrust [2] and Maze (http://maze.tianwang.com). In GRep, the objects exchange process is divided into several time slots (rounds). Each peer initiates requests during each round, and the request process follows the typical P2P request model in literature [8]. Besides, each peer is assigned an initial reputation value. And a peer's reputation value will increase by R_u when the peer uploads a valid object and will decrease by R_d when the peer downloads a valid object, and $R_u \geq R_d$.

Malicious peers. According to their different behavior features, the malicious peers can be roughly divided into six categories: **MP1**: peers that utilize P2P's resources without providing appropriate amount of resources (i.e., free-riders), such as BitTyrant and BitThief clients; **MP2**: peers that upload inauthentic objects to persecute the community, such as the clients controlled by the music industry which inject fake files to KaZaA; **MP3**: peers that collaborate with each other to promote their reputation values through uploading object to each other preferentially, such as the colluders in Maze, and these reputation values will be used to download their desired objects; **MP4**: Sybil peers that only request peers who create them for objects; **MP5**: peers that create Sybil peers for uploading to promote their own reputation values; **MP6**: peers that exploit P2P's resources for their malicious purposes like worm dispatching, denial of service and so on [9]. Moreover, there also exist some other malicious peers with more complex behaviors. For example, some peers may belong to multi categories at the same time and their behavior is a combination of the behaviors of multi categories. As another example, a peer is being nice until its reputation is high, and from then on exploits the system. But all in all, these malicious peers have different behaviors from honest peers. For the sake of simplicity, these malicious peers are called strategic malicious peers, and we will discuss them further in Section IV.

Other peers besides malicious ones will be called honest peers. Furthermore, we assume honest peers count for the majority of all the peers.

Reputation matrix. Let N be the total number of peers and R_p^T be the reputation value of peer p at the end of the T^{th} round, $1 \leq p \leq N$. Consequently, the reputation value of all the peers can form a reputation vector $RV^T = (R_1^T, R_2^T, \ldots, R_N^T)$ at the end of the T^{th} round. Besides, from the aspect of one single peer p, R_p^t, $1 \leq t \leq T$, can form a reputation time-series $RS_p = (R_p^1, R_p^2, \ldots, R_p^T)$. Then we can obtain a reputation matrix $R^{T \times N}$ as (1) at the end of the T^{th} round.

$$R^{T\times N} = \begin{bmatrix} R_1^1 & R_2^1 & \cdots & R_N^1 \\ R_1^2 & R_2^2 & \cdots & R_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ R_1^T & R_2^T & \cdots & R_N^T \end{bmatrix} \quad (1)$$

The i^{th} column of $R^{T \times N}$ is the reputation time-series of peer i. And the t^{th} row of $R^{T \times N}$ is the reputation vector at the end of round t. While each entry R_p^t is the reputation value of peer p at the end of the t^{th} round.

In RP2P systems, $R^{T \times N}$ may be iteratively calculated by each peer as in EigenTrust, or be collected and calculated by a centralized facility, such as the tracker servers in private tracker site or the central server in Maze. Besides, Distributed Hash Table (DHT) based method provides a feasible way to collect reputation information in distributed manner. Hence, at least in some way, we can always obtain $R^{T \times N}$. For the sake of simplicity, we will use $\boldsymbol{R}$ to represent $R^{T \times N}$ in the following analysis. Notice that due to network congestion and churn in overlay network, a few elements in $\boldsymbol{R}$ may be inaccurate or missing, and we will discuss this in Section IV-C.

B. Basic idea of PeerMate

As illustrated in Section III-A, all the malicious peers are with various objectives when joining the system. Despite of this, they possess an identical feature, which also differentiates them from honest ones, i.e. they behave differently from honest peers. Since the reputation value of a peer reflects its behavior feature, different behaviors will lead to different reputation values, which will afterward lead to their different reputation time-series in $\boldsymbol{R}$. Therefore, we can distinguish malicious peers from honest ones if we can extract their different behavior features, which are embedded in the different deterministic features of their reputation time-series in $\boldsymbol{R}$.

More concretely, the ideas lying behind malicious peer detection are as follows. Firstly, in order to extract the deterministic feature of reputation time-series in $\boldsymbol{R}$ and cope with inaccurate or missing data, we apply MSPCA to $\boldsymbol{R}$ to obtain the reconstructed reputation matrix $\hat{R}$ since we have found that the variability in $\boldsymbol{R}$ can be captured in a space of lower dimension; secondly, we define a metric QR in Section III-D, and then distinguish malicious peers from honest ones through a threshold γ.

C. MSPCA-based feature extraction

MSPCA. MSPCA combines the ability of PCA to de-correlate the variables by extracting a linear relationship, with that of wavelet analysis to extract deterministic features and approximately de-correlate auto-correlated measurements [10]. Furthermore, we add wavelet coefficients de-noising process to the MSPCA in [10], so the MSPCA here contains four steps:

Step 1: Wavelet decomposition of $\boldsymbol{R}$: apply wavelet decomposition $\boldsymbol{W}$ to each column of $\boldsymbol{R}$ to get wavelet coefficient matrix Z_L, Y_m (m = 1,…, L) at each scale; then filter the wavelet coefficients according to MAD method [11] and arrive at $\bar{Z}_L$, $\bar{Y}_m$ (m=1,…, L).

Step 2: Principal component analysis of wavelet coefficient matrix: firstly, apply PCA to wavelet coefficient matrix $\bar{Z}_L$, $\bar{Y}_m$ (m = 1,…, L) at each scale; secondly, select the number of principal components reserved according to scree plot method [12]; finally, reconstruct the wavelet coefficients matrix $\hat{Z}_L$, $\hat{Y}_m$.

Step 3: Wavelet reconstruction of the reputation matrix: reconstruct the matrix based on $\hat{Z}_L$, $\hat{Y}_m$ (m=1,…, L) through inverse wavelet transformation $\boldsymbol{W}^T$.

Step 4: Principal component analysis of reconstructed matrix: apply PCA to reputation matrix obtained in Step 3 to reduce the dimensionality and then obtain reconstructed matrix $\hat{R}$.

For the sake of clarity, this process is illustrated in Fig. 1.

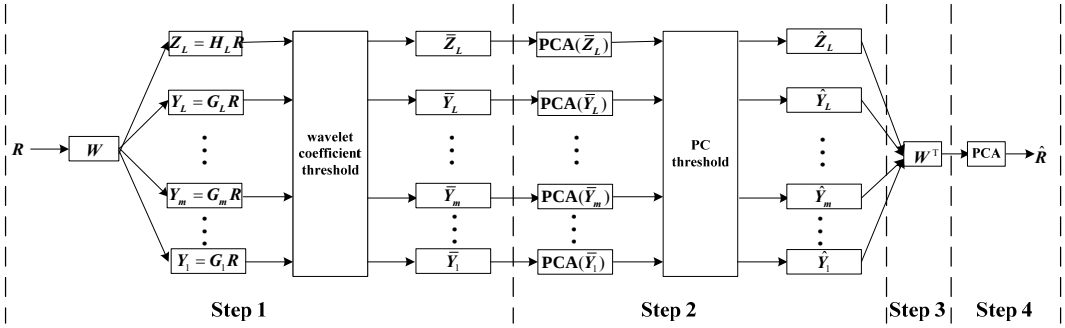

Fig. 1. Four steps of MSPCA

During Step 1 and 2, the inaccurate or missing elements will be treated as noise and will be eliminated, while the reputation values of malicious peers will be treated as noise with higher possibility than that of honest ones since the deterministic feature of $\boldsymbol{R}$ is distorted to the feature of honest peers.

During Step 4, after PCA, the first r principal components will be reserved to construct a new space called normal sub-space, while the other N-r principal components will be considered as abnormal sub-space. Then, $\hat{R}$ will be reconstructed with the normal sub-space. In general, the reconstruction process will tend to result in a large change to the reputation time series of malicious peers. Consequently, the reconstruction error of malicious peers will be higher than those of honest ones, and we can find them out through investigating the reconstruction error between $\boldsymbol{R}$ and $\hat{R}$.

D. QR based malicious peer detection

Based on the analysis in Section III-C, in order to capture the reconstruction error of each column, we define a metric Quality of Reconstruction (QR):

$$QR_i = 1 - sum((\hat{R}_i - \boldsymbol{R_i})^2) / sum((\boldsymbol{R_i})^2)) \quad (2)$$

where $\hat{R}_i$ is the i^{th} column of reconstructed matrix $\hat{R}$ and $\boldsymbol{R_i}$ is the i^{th} column of the original reputation matrix, $1 \le i \le N$.

According to (2), we can detect malicious peers through their QR based on some threshold γ. To be more concrete, if the QR of a column is lower than a threshold γ, it will be treated as a suspicious malicious peer, otherwise, it will be considered as an honest peer.

Algorithm 1 PeerMate

Input: $\boldsymbol{R}=[\boldsymbol{R_1}, \boldsymbol{R_2}, \ldots, \boldsymbol{R_N}]$ △ the reputation matrix, $\boldsymbol{R_i}$ is the i^{th} column of $\boldsymbol{R}$

Output: SMPS △suspicious malicious peers set

1: obtain reconstructed reputation values matrix $\hat{R} = [\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_N]$ of $\boldsymbol{R}$ through MSPCA
2: **for** each column i
3: calculate QR_i according to (2)
4: **end for**
5: obtain QR vector $QRV=[QR_1, QR_2, \ldots, QR_N]$
6: **for** each QR_i in QRV
7: **if** $QR_i < \gamma$
8: i is considered as a malicious peer, add i to **SMPS**
9: **end if**
10: **end for**

Our detection algorithm PeerMate is illustrated in Algorithm 1. Firstly, in line 1, PeerMate applies MSPCA to $\boldsymbol{R}$ and obtain reconstructed matrix $\hat{R}$. Secondly, from line 2 to 5, PeerMate calculates QR_i for each column i of $\boldsymbol{R}$ and obtain $QRV=[QR_1, QR_2, \ldots, QR_N]$. Finally, from line 6 to 10, for each element QR_i in QRV, peer i will be regarded as suspected malicious peer and will be added to the suspicious malicious peer set (**SMPS**) if its QR_i is lower than threshold γ.

Time complexity. The time complexity of PeerMate mainly lies on MSPCA, whose time complexity is $O(LTN^2)$, where L is the decomposition level of Wavelet, and is usually low. Therefore, the complexity of PeerMate is $O(LTN^2)$. Moreover, the storage cost of PeerMate is $O(TN)$.

IV. EXPERIMENTAL RESULTS

In this section, we conduct some simulations to evaluate the performance of PeerMate. We first demonstrate our simulation context, and then present the simulation results as well as the impaction of parameters. Discussions on the simulation results as well as possible usability of PeerMate are given at last.

A. Simulation context

In our simulation, the basic workload model follows the typical workload model in literature [8]. Concretely, the objects arrive at constant rate $\lambda_O > 0$, and the popularity of them follows the *Zipf* distribution. When an object arrives, its popularity rank is determined by selecting randomly from the *Zipf*(1) distribution. On average, a client requests a constant number of objects per round, from which it chooses objects to comply with a *Zipf* probability distribution with a parameter of 1.0. Moreover, we assume that all objects in the system are of equal size. The malicious peers are selected randomly from all the peers. Table I describes the parameters setup in the simulation experiment.

Table I Simulation Parameters

Symbol	Meaning	Base value
N	# of peers	200
O	# of objects	4000
λ_R	per-user request rate	2 objects /round
λ_O	object arrival rate	varies
P_M	the ratio of # of malicious peers to # of peers	varies
P_h	the possibility that strategic malicious peer act as honest peer	varies

B. Comparison benchmark and metrics

Comparison benchmark. We choose two recently proposed representative existing schemes as comparison benchmark: Upload Entropy (UEntropy) and Interaction

Entropy (IEntropy) schemes [1]. These two schemes are both aimed at incentive peers for sharing content in Private BT society. And those peers with lowest entropy will be considered as the least trustworthy collaborators, in other words, they are the suspicious malicious peers. Therefore, in order to guarantee the fairness of comparison, in UEntropy and IEntropy schemes, those peers that have the lowest entropy will be treated as suspicious malicious peers.

Evaluation metrics. Let **MPS** (Malicious Peers Set) be the malicious peers set, **HPS** (Honest Peers Set) be the set of honest peers, and **SMPS** (Suspected Malicious Peers Set). Then we define two metrics TPR (True Positive Ratio) and FNR (False Negative Ratio) as follows.

TPR=|**SMPS**∩**MPS**|/|**MPS**|;

FNR=|**SMPS**∩**HPS**|/|**HPS**|.

where | | represents the rank of a set, and ∩ stands for the intersection of two sets. Consequently, both TPR and FNR range from 0 to 1.

Simulation scenarios. We consider two typical simulation scenarios here. One is simple and the other is more complex. Under the simple scenario, there are no strategic malicious peers in the system. In contrast, there exist strategic malicious peers in the system under complex scenario.

C. Simulation results under simple scenario

Comparison results of the three schemes. We first compare detection results of PeerMate, UEntropy and IEntropy schemes with $\lambda_O = 2$, $P_h = 0$, $P_M = 0.2$ and $\gamma = 0.9$. Other parameters are as those listed in Table I.

After 200 rounds, we can obtain a reputation matrix $\boldsymbol{R}^{200\times200}$, and then we apply the three schemes to $\boldsymbol{R}^{200\times200}$ respectively, the results are shown in Table II. As Table II demonstrates, the TPR of PeerMate is 97.5% which is the highest among the three, while the TPRs of UEntropy and IEntropy are 42.5% and 0 respectively. Moreover, the FNR of PeerMate is 1.87% which is the lowest among the three, while the FNRs of UEntropy and IEntropy are 14.37% and 25% respectively.

All in all, PeerMate achieves the best performance among the three, and PeerMate can detect malicious peers accurately with very small FNRs.

After manual inspection, we find that those malicious peers that cannot be distinguished by PeerMate are those peers that belong to MP5 since they have similar behaviors to that of honest ones. To detect these malicious peers, we can investigate the behavior of malicious peers that belong to MP4, since they always download content from malicious peers belonging to MP5. Moreover, about 1.87% honest peers are judged as malicious ones because their objects may be at a lower rank among all the objects during the request process.

We also notice that almost none of the malicious peers can be detected by IEntropy, this tells us that interaction information cannot help us find malicious peers effectively.

Table II Comparison results of the three schemes

Scheme	FNR	TPR
UEntropy	14.37%	42.5%
IEntropy	25%	0
PeerMate	1.87%	97.5%

Detection results with missing data. As we have mentioned before, the reputation values in $\boldsymbol{R}$ may be inaccurate or missing. Consequently, we investigate how PeerMate could adapt to missing data context with different ratios of missing data from 0 to 60%, while the missing data are selected randomly from $\boldsymbol{R}$. Other parameters are: $\lambda_O = 2$, $P_h = 0$, $P_M = 0.2$ and $\gamma = 0.9$. For the sake of simplicity, we fix $\boldsymbol{R}$ as follows: if R_i^t is missing, then we set $R_i^t = (R_i^{t-1} + R_i^{t+1})/2$, if $1 < t < T$; $R_i^t = R_i^{t+1}$, if $t = 1$; $R_i^t = R_i^{t-1}$, if $t = T$.

After obtaining the reputation matrix at the end of the 200th round, we remove some elements of the matrix randomly with missing ratio from 0 to up to 60%. And then PeerMate is applied to the matrices, and the results are shown in Fig. 2.

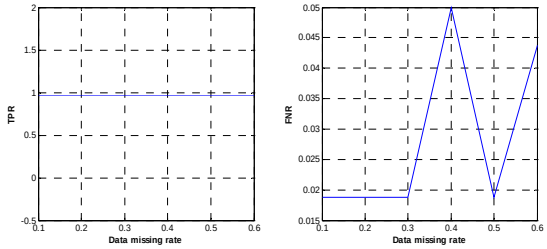

Fig. 2 The detecting results of PeerMate with different data missing rates

From Fig. 2, we can see that the FNR of PeerMate tends to fluctuate between 1.87% and 5% as the missing rate increases; in contrast, the TPR of PeerMate keeps at 97.5% as the missing rate increases. This means PeerMate is robust to missing data context since data missing cannot change the deterministic features of honest and malicious peers.

The impact of γ. γ plays a critical role in Algorithm 1, since it determines the TPR and FNR of PeerMate to some extent. Therefore, we investigate PeerMate with $\lambda_O = 2$, $P_M = 0.2$ and $\gamma = 0.84$, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9, 0.91, 0.92 and 0.93 respectively, and then show the results in Fig. 3. From Fig. 3, we can see that FNR of PeerMate increases from 0.63% to 3.13% as γ increases from 0.84 to 0.93. And the TPR of PeerMate increases from 55% to 97.5% as γ increases from 0.84 to 0.93. As the primary purpose of PeerMate is to detect as many malicious peers as possible with low FNR, therefore, $\gamma = 0.9$ is preferred here.

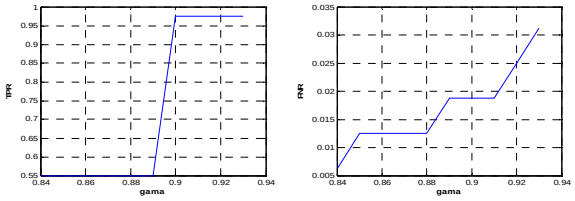

Fig. 3 The detecting results of PeerMate with different γ

The impact of P_M. We also investigate how PeerMate can adapt to different P_M. Therefore, we examine PeerMate with $\lambda_O = 2$, $\gamma = 0.9$ and $P_M = 0.1$, 0.2, 0.3, 0.4, 0.5, 0.6 and 0.7 respectively, and then show the results in Fig. 4.

From Fig. 4 we can see that the TPR of PeerMate is very high even when up to 60% of the peers are malicious, but when 70% of the peers are malicious, the TPR of PeerMate decreases to 50%. Since when there are too many malicious peers in the system, they can distort the first r principal components obtained in Step 4 of MSPCA, and then the QR of each column. The FNR of PeerMate, in contrast, increases from 0.5% to 3.5% as P_M increases from 10% to 30%, and then decreases from 3.5% to 0 as P_M increases from 30% to 70%. All in all, this indicates

that PeerMate can adapt to the context well even when up to 60% of the peers are malicious.

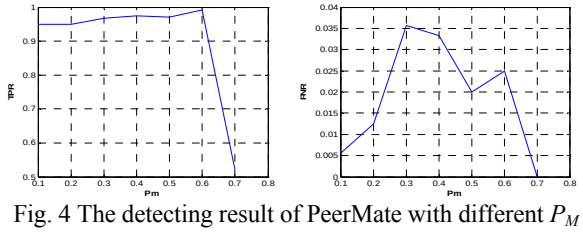

Fig. 4 The detecting result of PeerMate with different P_M

D. Simulation results under complex scenario

Besides the simple scenario discussed before, we further consider two complex scenarios here to evaluate the performance of PeerMate.

Possibility model. In order to avoid being detected, during each round, many strategic malicious peers will act as honest ones with certain possibility P_h. Therefore, we investigate PeerMate with $\lambda_O = 2$, $P_M = 0.2$, $\gamma = 0.9$ and P_h =0.1, 0.2, 0.3, 0.4 0.5, 0.6, 0.7 and 0.8 respectively and show the results in Fig. 5.

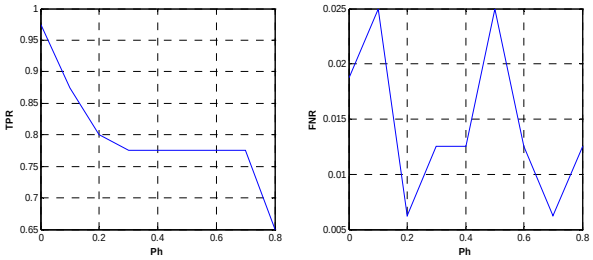

Fig. 5 The detecting result of PeerMate with different P_h

As Fig. 5 demonstrates, The TPR of PeerMate decreases from 97.5% to 65% as P_h increases from 0 to 0.8. The TPR of PeerMate is higher than 80% when P_h is lower than 0.3. Moreover, the FNR of PeerMate fluctuates between 0.65% and 2.5% and keeps at a low level. This means the performance of PeerMate is acceptable when P_h is lower than 0.3.

Mixture model. Here we evaluate PeerMate with strategic malicious peers that belong to multi categories at the same time and their behaviors are a combination of the behaviors of multi categories. Generally speaking, the mixture of malicious behavior cannot change the essential difference between the behaviors of malicious peers and honest ones. Concretely, we add a few malicious peers whose behaviors are as follows. They act as the behaviors of MP1, MP2, …, MP6 with certain possibility. And we find that the TPR of PeerMate is 95% with FNR equals to 2.5%. This means PeerMate is also good at finding malicious peers with mixture behaviors out since mixture behaviors cannot change the deterministic features of honest and malicious peers.

E. Discussion

Summary. From these simulations, we can draw the following conclusions:

First, PeerMate can achieve high accuracy even under strategic or data missing context. Second, PeerMate cannot distinguish malicious peers of MP5 from honest ones, but we can find them out through further checking the interaction collaborators of malicious peers belong to MP4. Third, each reputation management scheme needs to implement some algorithms to detect malicious peers; this can be proved through a game theory model and will be presented in our future papers. Fourth, towards at accurate detection, each reputation management scheme needs to implement some algorithms to collect reputation information of all the peers.

PeerMate applications. PeerMate can be applied to Maze-like and EigenTrust-like systems directly since they have common context. Besides, it can also be applied to other RP2P systems if some schemes are implemented to collect and compute the reputation values of all the peers.

V. Conclusions and Future Works

In this paper, we present PeerMate, a novel malicious peer detection algorithm in RP2P systems, which distinguishes malicious peers from honest ones by MSPCA and QR. Simulation results indicate that PeerMate achieves high detection accuracy and flexibility.

As a future task, we are planning to extend PeerMate to make it adaptable to real-time online detection in RP2P systems.

VI. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61070173, Jiangsu Province Natural Science Foundation of China under Grant No. BK2010133 and Jiangsu Province Natural Science Foundation of China under Grant No. BK2009058.

References

[1] Z. Liu, P. Dhungel, D. Wu, C. Zhang and Keith W. Ross, "Understanding and Improving Incentives in Private P2P Communities," In ICDCS 2010, Italy, Jun. 2010.

[2] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. "The EigenTrust Algorithm for Reputation Management in P2P Networks," In Proceedings of the Twelfth International World Wide Web Conference, Budapest, May 2003.

[3] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li, "An empirical study of collusion behavior in the maze p2p file-sharing system," in IEEE ICDCS, June 2007.

[4] J. R. Douceur. "The Sybil attack," In First International Workshop on Peer-to-Peer Systems (IPTPS '02), Mar. 2002.

[5] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-Peer's Most Wanted: Malicious Peers, " Comp. Net., vol. 50, no. 4, Mar. 2006, pp. 545–62.

[6] W. Ji, S. Yang and B. Chen, "A Group-Based Trust Metric for P2P Networks: Protection against Sybil Attack and Collusion," International Conference on Computer Science and Software Engineering, 2008 Volume: 3 , Page(s): 90 - 93.

[7] H. Lee; J. Kim; K. Shin, "Simplified clique detection for collusion-resistant reputation management scheme in P2P networks," 2010 International Symposium on Communications and Information Technologies (ISCIT), 2010 , Page(s): 273 - 278.

[8] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," In 19-th ACM Symposium on Operating Systems Principles, Bolton Landing, NY, USA, October 2003.

[9] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," Technical Report of Purdue University (CSD TR No.07-013), 2007.

[10] B. R. Bakshi. "Multiscale PCA with Application to Multivariate Statistical Process Monitoring," AIChE journal, 1998, 44(3): 1596-1610.

[11] Donoho, I. M. Johnstone, G. Kerkyacharian, et al, "Wavelet Shrinkage: Asymptopia?," J. R. Stat. Soc. B, 57, 2797-2814, 1995.

[12] A. Lakhina, M. Crovella, C. Diot, "Diagnosing Network-Wide Traffic Anomalies," SIGCOMM, Portland, Oregon, USA, 2004.