

# Towards Secured P2P Network: Degenerated Node Identification and Evaluation

Liumei Zhang<sup>1,2,\*</sup>, Jianfeng Ma<sup>1</sup>, Yichuan Wang<sup>1</sup> and Di Lu<sup>1</sup>

<sup>1</sup>School of Computer Science and Technology, Xidian University, Xi'an, 710071, China; <sup>2</sup>School of Computer Science, Xi'an Shiyou University, Xi'an, 710065, China

**Abstract:** The security improvement of service provided by peer-to-peer (P2P) network has been widely studied. Security issue of P2P is mainly related to the users of P2P services and threat of malicious software. Especially, some malicious softwares could threaten other P2P users and even threaten the security of the entire P2P network. In this paper, based on the analytical hierarchy process, a degenerated node evaluation model has been proposed to evaluate the node resource usage of the P2P network. Moreover, attribute clustering based collaborative filtering algorithm is depicted for calculating the similarities between the identified degenerated node and all the other nodes within a hybrid autonomous cluster. The algorithm utilizes similarity characteristics of node task resource attributes, especially CPU related attributes, to filter redundant data by feature selection. Experiments show that the proposed model and method are feasible and have vital significance for providing decision support and recommendation to the search node. This proposed model is evaluated as being practical for secure computing of P2P network.

**Keywords:** Analytical hierarchy process, Collaborative filtering, P2P Network, Network security, Degenerated Node, Pair-Wise Comparison.

## 1. INTRODUCTION

In recent years, P2P network has brought great convenience and has improved resource sharing over the Internet. Hybrid model of P2P network united the pure decentralized and centralized P2P to provide advanced search ability. According to the capabilities (computing power, memory size, and residence time and network connection bandwidth), P2P nodes fall into two categories; normal node and search node. An autonomous cluster is created when a search node unifies the number of nearby normal nodes. Inside the cluster, there exists a centralized directory based P2P model. Then the entire P2P network is formed by the connection of various clusters *via* pure P2P model. It is extremely effective in eliminating the adverse effects of network congestion and search delays caused by flooding algorithm of pure P2P model. At the same time, as the search node in each cluster is responsible for monitoring all of the normal node behavior, it is able to control the local malicious behavior in the network. Thus, to some extent, it improves the load balance of the whole network.

P2P systems pose unique challenges from a computer security perspective. Like any other form of software, P2P applications can contain vulnerabilities. What makes this particularly dangerous for P2P software, however, is that peer-to-peer applications act as servers as well as clients, which means that they can be more vulnerable to remote exploits [1]. Attack towards P2P network varies in kind including poisoning attacks, distributed denial of service attacks,

betrayal (vampire) attack, etc. Security issue of P2P network has increased and drawn research attention worldwide [2-4].

As nodes play a dormant role in a hybrid P2P network, capture of any node in a cluster by any form of attack surely endangers the entire network. Thus, a secure P2P should first secure all the nodes within such a network. A secure P2P then surely contains nodes that act as "normal". Degenerated nodes can thus be considered as malfunctioned and it is reasonable to assume that the node has such behavior. Such issue can be identified as a mining problem that falls into outlier detection. If the node behavior can be measured by such technique, security of P2P can further be ensured.

In a hybrid P2P cluster, nodes can be considered as physical machines or virtual machines scattered and connected by means of communication medium. As degenerated nodes behave abnormal when compared with all the other normal nodes and to itself before the capture, the system resource it uses surely differs. If such difference can be identified, such degenerated node is able to be determined and can report to search node (super node) for further broadcasting such warnings to rest of the nodes. Thus, network security can be ensured. For deciding the degenerated node, all nodes need to be analyzed to determine the node behavior. Active node can be obtained from analyzing the characteristics of its behavior and usage on a resource usage perspective. Resource usage can be depicted with the resource used by a node when certain computing task is executed. However, resource usage information can be utilized to evaluate if a node is feasible and to decide whether a degenerated node is still a problem. Therefore, this situation should be defined as a decision-making problem.

\*Address correspondence to this author at the 2 South Taibai Road, Xi'an, China. Postcard: 710071; Tel: +8618092330186; E-mail: rikrun@gmail.com

The analytic hierarchy process (AHP) [5] is a decision-making method designed to help solve the multiple criteria complex problem in many application fields. Such method is effective and practical for complex and unstructured decision problem [6]. Thus to solve decision-making problems, the AHP method has been widely accepted in many application fields [7, 8]. Finding out the degenerated node with certain resource usage through the AHP method will provide great convenience for further study.

Collaborative filtering is designed to generate recommendations for the corresponding application domain user. For example, a study [9] proposed a collaborative filtering algorithm based on spatial clustering, then divided the recommended process into two stages of offline and online. Cheinshung Hwang [10] proposed the fuzzy set theory based clustering collaborative filtering algorithm for the prediction of Web pages. Therefore, by uniting collaborative filtering method with the results of the AHP method, similarity between the degenerated node and other nodes is able to obtain. Then, high similarity nodes recommendation can be provided from the degenerated to the search node.

In this paper, an AHP model for processing influencing factors between degenerated node identification decisions and resource usage has been proposed. P2P system administrators can determine the importance of criterion for the resource usage of pair-wised comparison matrix. Then alternative nodes can be prioritized or weight calculated. To determine the degenerated node in a cluster, AHP model is depicted based on the CPU usage as it is the key factor associated with the resource usage. By referencing item rating collaborative filtering, an attribute clustering based collaborative filtering is proposed to provide degenerated nodes recommendation by using the results of AHP model for the search nodes. Finally, search nodes are able to inform all the other nodes in the whole P2P network to be aware of such degeneration. Thus the security of entire network can be ensured.

**2. DEGENERATED NODE EVALUATION MODEL**

**2.1. Establishment of Degenerated Node Evaluation Index System**

Three basic steps of AHP are applied to solve the decision problem: First, index system is established for analyzing the relationship between the factors of the system. Pair-wised comparison matrix is created for evaluating the importance of each element on the same layer towards the above layer; second, judgment matrix is formed to calculate relative weights of elements being compared. Then, consistency of judgment matrix is checked; third, global weights of all levels are calculated. Finally, total ranking of each scheme towards the overall goal is obtained.

Establishing hierarchy of index system is based on the selection of impact factors which have certain influence on the evaluation results. And the process of establishing the index system is an evaluation process by a series of indicators. Index system refers to a scale set for comprehensive measurement of an evaluation object. Such scale set is composed of a series of indexes. Comprehensive evaluation index system is a widely accepted method to analyze systems in the social, economic and management science fields. Evaluation system often has a hierarchical structure including goal and criterion layers to form a multi-level evaluation system.

Building hierarchy for degenerated node evaluation can be considered as an evaluation model for degenerated node identification. Therefore, from the perspective of CPU usage, CPU usability and cyclicity are the two main performance criteria. Therefore, sub criterion of CPU usability can be obtained such as the average CPU rate, maximum CPU rate, cycle per instruction and task duration. Fig. (1) illustrates the index system of such evaluation.

From Fig. (1), degenerated node can be obtained. However, for one node, single task resource usage information cannot represent a definite behavior as task resource usage of a node consists of arbitrary attribute vector. Therefore, there is a need to analyze the overall task resource usage of a node to select the most representative task resource usage information. The second hierarchy then is depicted to select the most representative task resource usage information of one node. Fig. (2) represents the index system of the overall task resource usage evaluation of one node. Compared with Fig. (1), Fig. (2) introduces the task validity as evaluation index to depict the time interval between task execution and current.

Building hierarchy for degenerated node evaluation can be considered as an evaluation model for degenerated node identification. Therefore, from the perspective of CPU usage, CPU usability and cyclicity are the two main performance criteria. Therefore, sub criterion of CPU usability can be obtained such as the average CPU rate, maximum CPU rate, cycle per instruction and task duration. Fig. (1) illustrates the index system of such evaluation.

**2.2. Pair-Wise Comparison Matrix**

When completing the establishment of index system, all factors that influence the identification of a degenerated node should be analyzed by constructing pair-wise comparison matrix to judge criterion weights and check the consistency of the judgment matrix.

Before the comparison, all factors should first be analyzed. For node running in one P2P autonomous cluster, it is assumed that if the average CPU rate is high, then the running node consumes high CPU resources and is a degenerated node. The same rule can also work when the maximum CPU rate is high.

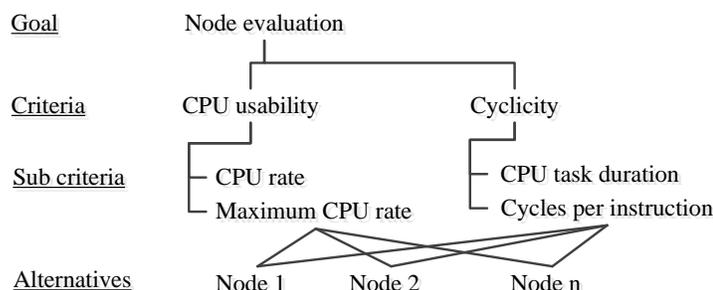


Fig. (1). Degenerated node evaluation index system.

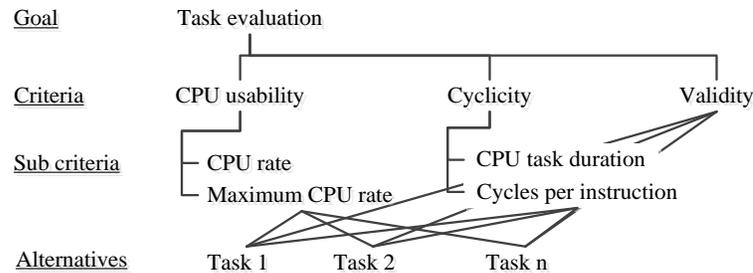


Fig. (2). Task evaluation index system.

Table 1. Degenerated node evaluation criteria and sub-criteria.

Goal: Node Evaluation				
criteria	CPU usability (C1)		Cyclicality (C2)	
Sub-criteria	CPU usage (C11)		CPU task duration (C21)	
	Maximum CPU usage (C12)		Cycles per instruction (C22)	
Goal: Task Usage Evaluation				
criteria	CPU usability (B1)		Cyclicality (B2)	Task validity (B3)
Sub-criteria	CPU usage (B11)		CPU task duration (B21)	
	Maximum CPU usage (B12)		Cycles per instruction (B22)	

Table 2. Global weights of all criteria.

Node Evaluation							
Criteria	C1	C2	C11	C12	C21	C22	
Weight	0.83	0.17	0.73	0.1	0.14	0.03	
Task Evaluation							
Criteria	B1	B2	B3	B11	B12	B21	B22
Weight	0.36	0.07	0.57	0.31	0.05	0.06	0.01

On the other hand, if the CPU task duration is short, it indicates a node CPU scheduling frequent tasks. Otherwise, the long CPU task duration indicates infrequent tasks scheduling and such node needs to be identified as a degenerated node. Cycles per instruction describe the instruction type that CPU executes. If the cycles per instruction are short, this may indicate that the instruction is a conventional instruction including operational instruction, short instruction and short data operation instruction. Typically, those instructions are used only for calling the register. Therefore, this node can be identified as a degenerated node. A long cycle per instruction indicates that the co-processor is required for big data operation or conducting abnormal control, etc. More precisely, the operation duration is not long and soon the node can be terminated. So, it is unnecessary to identify it.

From Figs. (1) and (2), we can summarize the criteria and sub-criteria that we have identified as being important in the degenerated node identification decisions. Table 1 shows the result.

Then, the pairwise comparison matrices can be developed to determine the criteria and sub-criteria weights. The weights for all the pairwise comparison matrices can then be

computed. Table 2 shows the resulting global weights of all criteria.

For all nodes in alternative layers, the metrics value of nodes toward sub criterion is obtained by summing up the multiplication result between data and the weight value of such sub criterion. Furthermore, degenerated node evaluation results can be obtained by summing up the multiplication result between such metrics value and weight value of the criterion. Finally, degenerated node is determined once all nodes are calculated and prioritized.

For search node, the degenerated node is now identified according to the defined weight value distribution. As the nodes in a P2P network are abundant in numbers, deciding only one degenerated node to report is insufficient for achieving the entire network security. It is necessary to find out other degenerated nodes. Such degenerated nodes can be obtained by calculation of similarities between the identified degenerated node and all the other nodes. Then all nodes that have high similarity value are recommended to search node and collaboration filtering is required.

Table 3. Node attributes matrix.

	$a_{11}$	$a_{12}$		$a_{1j}$		$a_{1k}$
$I_1$	1	0	...	1	...	0
$I_2$	0	1	...	1	...	1
...	...	...	...	...	...	...
$I_i$	1	0	...	1	...	0
...	...	...	...	...	...	...
$I_n$	0	1	...	0	...	1

### 3. NODES SIMILARITY CALCULATION

#### 3.1. Collaborative Filtering for Node Clustering

According to the above statements, if a task occupies high average CPU rate, maximum CPU rate, short cycles per instruction and long task duration, the node is then a degenerated node. So, the alternative cluster of the degenerated node is provided.

**Definition:** A cluster C is defined as the Alternative Cluster of Degenerated Node (ACDN), if C is composed with nodes that have task usage data with high CPU rate, maximum CPU rate, high cycles per instruction and long task duration.

Traditional collaborative filtering algorithm generates recommendations by the calculation of nearest neighbor information of users. Item-based collaborative filtering first calculates the correlation between the items. Then, it predicts the users score towards unrated items by referencing rating of correlated items [11]. Therefore, the user interest is able to be predicted by analyzing the cluster commonalities. Moreover, clustering is able to partition users with similar interest into a same cluster [12] and predict the users interest trend based on analysis of sub cluster. This paper assumes that user is the search node with a P2P autonomous cluster. The collaborative filtering algorithm proposed here is dedicated to recommend ACDN to search node according to the result of the degenerated node obtained from AHP method. Node activity is measured by the data obtained from the computing tasks of each node. At this time, activity of the node is able to be measured by using task resource usage data of each node. Then, the rating table is required to map the raw data for applying collaborative filtering. Finally, the ACDN is able to be obtained by utilizing clustering method and rating table.

#### 3.2. Similarity Identification for Node Attributes

For clustering convenience, a node attribute space is introduced to summarize the node rating score into a number of feature clusters, which is denoted as  $\Omega = A_1, A_2, \dots, A_k$ , where,  $k$  is the number of attributes of a node. For some nodes, however, a specific attribute may have multiple attribute values. Therefore, this paper adopted a single attribute to obtain similarity of a node on a particular attribute. Then summing all the similarities of attributes, and calculating the average similarity, similarity between the nodes can be attained. For example, attributes space of the

node  $N_1$  and  $N_2$  is denoted as  $A_1 = \{a_{11}, a_{12}, \dots, a_{1k}\}$ . Therefore, we can get the node attribute matrix  $n \times k$  from attribute  $A_1$  as shown in Table 3.

In Table 3, column  $k$  indicates that attribute  $A_1$  has  $k$  values, while row  $n$  indicates the number of nodes. 1 and 0 represent whether  $A_1$  attribute of the node conforms to the attribute values of all the other nodes. 1 means it conforms and 0 means that it does not conform.

After the node attribute matrix is constructed, we are able to calculate similarity for further measurement of similarity degree between node  $N_1$  and  $N_2$  toward attribute  $A_1$ . A set of feature vector is then formed with respect to the conformity value of a node towards certain attribute. For example, the feature vector of nodes  $N_1$  and  $N_2$  toward attribute  $A_1$  is depicted as  $\overrightarrow{N_1A_1} = \{N_1a_{11}, N_1a_{12}, \dots, N_1a_{1k}\}$  and  $\overrightarrow{N_2A_1} = \{N_2a_{11}, N_2a_{12}, \dots, N_2a_{1k}\}$ .

Then, the attribute similarity of  $N_1$  and  $N_2$  toward attribute  $A_1$  can be represented as:

$$\begin{aligned}
 S_1 &= sim(N_1A_1, N_2A_1) \\
 &= 1 - \frac{\overrightarrow{N_1A_1} \oplus \overrightarrow{N_2A_1}}{K} \\
 &= 1 - \frac{\sum_{i=1}^k \overrightarrow{N_1a_{1i}} \oplus \overrightarrow{N_2a_{1i}}}{K}
 \end{aligned} \tag{1}$$

Where,  $sim(N_1A_1, N_2A_1)$  is the similarity,  $\overrightarrow{N_1A_1} \oplus \overrightarrow{N_2A_1}$  is the attribute value with no commonality between  $N_1$  and  $N_2$  toward  $A_1$ . Such attribute value is capable for the generation of probability value of certain attribute of the node after XOR operation. Then the attribute value, with no commonality, is summed and divided by  $K$ . The result depicts the incoherence degree towards  $A_1$ . It is to be noted that  $K$  refers to the total value numbers of  $A_1$ .

Next, the average trust degree of similarity,  $EA(sim)$ , between  $N_1$  and  $N_2$  is calculated as:

$$\bar{S} = EA[sim(N_1, I_2)] = \frac{\sum_{i=1}^m sim(N_1A_i, N_2A_i)}{m} \tag{2}$$

Then, the expectation value of trust degree among all attributes is obtained to describe the mean value of similarity

between the nodes.  $m$  is the number of attributes used for such description.

Therefore, similarity between nodes can be obtained as:

$$S = sim(N_1, N_2) = \frac{\sum_{i=1}^k (S_1 - \bar{S})}{\sqrt{\sum_{i=1}^k (S_1 - \bar{S})^2}} \quad (3)$$

This describes the similarity between node  $N_1$  and  $N_2$ . Where,  $k$  is the total value numbers of  $A_1$ .

### 3.3. Node Similarity Clustering

In this paper, we simply used the k-means algorithm for the node group analysis. Assuming that the dataset and cluster centroid is  $n$  dimensional vector, following two steps are repeated until the convergence:

**Step 1:** For each  $x^i$ , the nearest centroid  $j$  is obtained and marked into different categories. We need to assign  $x^i$  to cluster  $j$  to assign all points into its nearest centroid.

$$set\ c(i) = arg\ min\ ||\ x^i - u^j || \quad (4)$$

**Step 2:** The cluster centroid is updated to the average value of all points to determine the new centroid.  
set  $distance = d$

$$set\ cluster\_center = EA(distance) \quad (5)$$

It describes the similarity between node  $N_1$  and  $N_2$ . Where,  $k$  represents the total value numbers of  $A_1$ .

Assuming that there are  $n$  nodes, thus the recommended collection is  $N = \{N_1, N_2, \dots, N_n\}$ . After treating with K-means algorithm, the clusters can be described as  $C = \{c_1, c_2, \dots, c_j\}$ . Where,  $j$  is the total cluster number,  $c_i$  contains the nodes with high preference and interest similarity. The realization is shown as follows:

**input:** ClusterNum  $j$  and Matrix( $n \times k$ );

**output:** The number of clusters about the matrix is  $j$ ;

Steps:

**Step 1:** Search  $n$  nodes within the node attribute matrix, depicted with collection  $N = \{N_1, N_2, \dots, N_n\}$ ;

**Step 2:** Randomly choose  $j$  nodes. Set their attribute data as the initial cluster centroid, depicted with collection  $C' = \{c'_1, c'_2, \dots, c'_j\}$ ;

**Step 3:** Empty  $j$  clusters, depicted with collection  $C = \{c_1, c_2, \dots, c_j\}$ ;

**Step 4:** Perform the following actions on the rest of the nodes:

Algorithm 1: Node clustering algorithm	
1:	for all $n_i \in N$ do
2:	for all $c' \in C'$ do
3:	$sim(c_i, c'_i)$ ;
4:	end for
5:	$sim(n_i, c'_m) = max\{sim(n_i, c'_m), \dots, sim(n_i, c'_m)\}$ ;
6:	$c'_m = c'_m \cup n_i$ ;
7:	end for

Where in the Node clustering algorithm,  $sim(n_i, c'_i)$  is the similarity between  $n_i$  and centroid  $c'_i$ ,  $sim(n_i, c'_m) = max\{sim(n_i, c'_m), \dots, sim(n_i, c'_m)\}$  and  $c'_m = c'_m \cup n_i$  is the clustering process.

**Step 5:** Calculate the mean value of all nodes in the new cluster and update the centroid;

**Step 6:** Repeat Step 4 and Step 5 until the centroid is stable, with an output of  $s$  clusters.

With nodes clustering algorithm, we can find the nodes that have the highest similarity. If one node has the highest similarity with the degenerated node, it should also be identified as a degenerated node when considering task factors such as CPU usage and CPU cycle. Thus we get lemma 1.

**lemma 1:** for the node with highest similarity with the AHP defined degenerated node, it should be included in ACDN.

## 4. EXPERIMENT ANALYSIS

### 4.1. Node Evaluation Simulation

The experiment data was sampled from task usage table part-00000-of-00500 of Google clusterdata-2011-1 [13].

Before using AHP model, Liberatore's [14] five-point rating scale was adopted for rating each sub-factor of alternative nodes to reduce the time and effort in making pair-wise comparisons. Table 4 shows the modified pair-wise comparison matrix of the rating scale. The matrix was normalized to obtain relative weight value of each metric for measurement of experimental dataset. As can be seen from both the tables, for CPU rate, Maximum CPU rate and Cycles per instruction attributes, the terms we used were: very-high, high, moderate, low and very-low to normalize the raw data according to

Table 4. Five-point rating scale.

Score	VL(VS)	L(S)	M	H(L)	VH(VL)	Weight
VL(VS)	1	3	5	7	9	0.513
L(S)	1/3	1	3	5	7	0.261
M	1/5	1/3	1	3	5	0.129
H(L)	1/7	1/5	1/3	1	3	0.063
VH(VL)	1/9	1/7	1/5	1/3	1	0.034

**Table 5. Single node task usage evaluation.**

Criteria	Sub-criteria	Global weights	Task 1		Task 2	
			Score	GW	Score	GW
$B_1$	$B_{11}$	0.31	VL	0.16	VL	0.16
	$B_{12}$	0.05	L	0.01	VL	0.03
$B_2$	$B_{21}$	0.06	VS	0.03	VL	0.002
	$B_{22}$	0.01	VL	0.004	VL	0.003
$B_3$		0.57	VL	0.02	L	0.36

their values. While, for task duration, we used: very-long, long, moderate, short and very-short to normalize the raw data. Then, the weights of very-high, high, moderate, low and very-low were calculated, being equal to 0.513, 0.261, 0.129, 0.063 and 0.034, respectively. The weights of very-long, long, moderate, short and very-short were calculated in the same manner. On the other hand, the attribute task validity was calculated in the same way as the task duration.

The experiment assumed each Machine ID attribute in table part-00000-of-00500 denoting a node. 500 nodes were sampled as the experiment dataset. Each node had 10 items of task resource usage. Next, we mapped the raw data according to the proposed rating scale. For single value, mean, maximum and minimum values were first calculated based on all the 500 nodes raw data. Then, the raw value of such attribute was mapped into partitions according to the five-point rating scale. According to AHP task evaluation model, every node was evaluated and the most representative task usage data was obtained. Table 5 shows the task evaluation of one node with only two task data. Table 6 shows 5 tasks with highest value after application of model. Then, we were able to load task usage data into the degenerated node evaluation model with respect to Table 6.

**Table 6. Task usage evaluation result.**

Node ID	Task ID	Value
2994441279	10	0.5
587080532	10	0.47
17504375	10	0.41
4302816019	10	0.4
2568530361	9	0.37
587080532	10	0.47

**Table 7. Node evaluation result.**

Criteria	Sub-Criteria	Global Weight	Node 1		Node 2	
			Score	GW	Score	GW
$C_1$	$C_{11}$	0.73	VL	0.16	VL	0.16
	$C_{12}$	0.1	L	0.01	VL	0.03
$C_2$	$C_{11}$	0.14	VS	0.03	VL	0.002
	$C_{11}$	0.03	VL	0.005	VL	0.005

Evidently, after all the tested nodes have been applied with the proposed model, the node with the highest value is identified as the degenerated node. Table 7 shows the degenerated node evaluation of 500 nodes with only two nodes. Table 6 represents the result that the node with ID 2994441279 has the highest value. It shall be first put into ACDN. Next, we need to find other nodes which have higher similarity with 2994441279.

**4.2. Node Similarity Calculation**

Before similarity calculation, same five-rating scale was applied to each node for reducing the amount of computation. The mapped data were then transformed according to node attributes matrix as shown in Table 3, that is to say, to decide whether an attribute of a node agrees with the value over 10 task data. As the degenerated node should contain a value that has very high CPU rate, very high maximum CPU rate, very short cycles per instruction and very long task duration, the node attributes matrix is able to be obtained based on such definition. For single task usage data, we assign 1 to the data that has such attribute feature and 0 to the rest of the data.

Experimental dataset was similar to the AHP model application. Again five-point rating scale was applied to 500 nodes raw data to formalize the node attributes matrix. Similarities among each of the 499 nodes towards node 2994441279 were calculated. Table 8 illustrates the top 5 nodes with the highest similarity after result ranking from similarity calculation according to formulas (1), (2) and (3). Next, all the nodes were clustered by casting node clustering algorithm for the similarity. The cluster granularity can be adjusted according to the application scenario. Consequently, the cluster that contained nodes with highest similarities was ACDN. ACDN was a strong reference for search node to report degenerated node over the entire P2P network. The proposed method is conducive and practical to secure P2P network, and has strengthened the concept of P2P security.

**Table 8. Node similarity calculation result.**

Node ID	Similarity
820250647	0.44789675
4820012347	0.44783034
336069364	0.44773205
1429206364	0.44772832
25751970	0.4474831

## CONCLUSION

In this paper, we proposed an AHP-based model in order to evaluate the degenerated node identification decisions. By conducting a usability study with nodes task resource usage data and then demonstrating how the model can be applied in real applications, the paper investigated how easy the proposed AHP model is to work with. Furthermore, an attribute clustering based collaborative filtering method was proposed for identifying the degenerated node. The method first introduced a node attribute space to summarize the node rate score into number of feature clusters. Then, based on the AHP result, it calculated similarities between the nodes. Both the methods were verified with real dataset and thus achieved acceptable results in identifying the degenerated nodes. However, other features of the node have not been included in this paper, and our method is still relatively simple as there are many details which need to be further studied. Hopefully, this work will motivate researchers to investigate solution for the identification of degenerated node and for securing the P2P network.

## CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

## ACKNOWLEDGEMENTS

This paper is supported by Program for Changjiang Scholars and Innovative Research Team in University

(IRT1078), The Key Program of NSFC-Guangdong Union Foundation (U1135002), Major national S&T program (2011ZX03005-002), and the Fundamental Research Funds for the Central Universities (JY0900120301).

## REFERENCES

- [1] Q. H. Vu, M. Lupu, and B. C. Ooi, "Computing, P.t.P.: Principles and applications", 2009.
- [2] A. Mityagin, D. X. Charles, K. E. Lauter, "Security architecture for peer-to-peer storage system", US Patent 8196186, 2012.
- [3] R. Gupta, A. Gavrilescu, J. L. Miller, and G.A. Wheeler, "Peer-to-peer name resolution protocol (pnrp) security infrastructure and method", EP Patent 1361728, 2012.
- [4] C. Selvaraj, "A survey on security issues of reputation management systems for peer-to-peer networks", *Computer Science Review*, vol.6, no. 4, pp.145-160, 2012.
- [5] T. L. Saaty, "Fundamentals of decision making and priority theory: with the analytic hierarchy process", Rws Publications, 1994.
- [6] F. Y. Partovi, "Determining what to benchmark: an analytic hierarchy process approach", *International Journal of Operations and Production Management*, vol. 14, no. 6, pp. 25-39, 1994.
- [7] R. Jain, and B. Rao, "Application of ahp tool for decision making of choice of technology for extraction of anti-cancer bioactive compounds of plant origin. International", *Journal of the Analytic Hierarchy Process*, vol. 5, no. 1, pp. 3-29, 2013.
- [8] S. Uçal, B. İÖztayşi, C. Kahraman, "Fuzzy analytic hierarchy process using type-2 fuzzy sets: An application to warehouse location selection," *Multicriteria Decision Aid and Artificial Intelligence*, pp. 285-308, 2013.
- [9] F. Zhang, H. Sun, and J. Chang, "A spatial clustering-based collaborative filtering algorithm in mobile environment", *Journal of Computational Information Systems*, vol.6, no.7, pp. 2297-2304, 2010.
- [10] C. S. Hwang, "Integrating fuzzy partitional clustering and collaborative filtering for webpage prediction", *WSEAS Transactions on Information Science and Applications*, vol. 3, no. 11, pp. 2094-2099, 2006.
- [11] A. L. Deng, Y. Y. Zhu, and B. Shi, "A collaborative filtering recommendation algorithm based on item rating prediction", *Journal of Software*, vol. 14, no. 7, pp. 1621-1628, 2003.
- [12] X. Liu, and T. Liu, "Attribute clustering recommendation algorithm based on item rating prediction", *Statistics and Decision*, vol. 18, pp. 9-11, 2012.
- [13] C. Reiss, J. Wilkes, and J. Hellerstein, "Google cluster-usage traces: format+ schema", Google Inc., White Paper, 2011.
- [14] M. J. Liberatore, R.L. Nydick, and P. M. Sanchez, "The evaluation of research papers (or how to get an academic committee to agree on something)", *Interfaces*, vol. 22, no. 2, pp. 92-100, 1992.

Received: September 22, 2014

Revised: November 30, 2014

Accepted: December 02, 2014

© Zhang *et al.*; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.