Dynamic Reputation Based Trust Management Using Neural Network Approach

Reza Azmi[1] Mahdieh Hakimi[2], and Zahra Bahmani[3]

[1] College of Engineering, Alzahra University
Tehran, Iran

[2] College of Engineering, Alzahra University
Tehran, Iran

[3] College of Engineering, Alzahra University
Tehran, Iran

Abstract

Multi-agent systems like Peer-to-Peer (P2P) Networks employ scalable mechanisms that allow anyone to offer content and services to other system users. The open accessibility of these networks makes them vulnerable to malicious users wishing to poison the system. This paper proposed a novel trust and reputation system, using RBF artificial neural network to determine trust level and mitigate the number of unreliable downloads.

Keywords: *Peer-to-Peer Network, Trust, Reputation, RBF Artificial Neural Network.*

1. Introduction

Agents in P2P Networks are anonymous and heterogeneous also there is no central authentication system. In these distributed systems flexibility and low participation cost encourages a much larger number of participants. Accessing to data and shared services in dynamic networks like P2P environments are related to trust and reputation of peers. Recently these systems are usually applied to file sharing and social networks. P2P networks tend to be more scalable, robust and adaptive than other forms of distributed systems. In this paper we proposed a novel dynamic trust model as one kind of decision support systems, using RBF artificial neural network to determine trust level and mitigate the number of unreliable downloads. Recommended trust models are applied broadly that help peers to download from reliable providers. They differ in selections of recommenders and in aggregations of recommendations.

2. Related Work

Various techniques have been proposed to secure P2P networks over the last decade. Abdul-Rahman[1] captured the most important characteristics of trust and reputation and proposed the general structure for developing trust and reputation in a distributed system. Most of the later works in this area followed their ideas, but in different application domain, such as [2, 3, 4, 5]. EigenTrust model of Kamvar[4] is built on the notion of transitive trust. A major issue of applying this model is to find pre-trusted peers that guarantee convergence of the algorithm and avoid malicious collectives. Wang[5] applied Naive Bayesian network to recommendation trust. The model can be used to solve the problem of different estimation process of the same online service. In [6, 7, 8, 9] proposed the trust and reputation system based on artificial neural network which are built on back-propagation algorithm to train the MLPs neural network. In proposed paper we used RBF neural network since these networks tend to learn much faster and require fewer training samples than MLPs.

3. Determination of Trust and Reputation Level by Neural Network

In this paper we focus on pure P2P networks for file exchange and, more precisely, on the Gnutella architecture because it is closest to the ideal structure of the P2P networks, where all participants have a uniform role.

3.1 Basic idea of proposed system

In a P2P overlay network like Gnutella, exchanging a file is containing two phases such as 1.Searching a file, 2.Downloading it. But some other researches like [2, 3, 4, 5] proposed this protocol by adding two other phases such as 3.Pooling and 4.Evaluating the votes. In this way before a requester decides about downloading the file from a provider, first it asks other peers about reputation of him/her.

In addition this paper studied about influence of layering concept on evaluating trust, for the first time and we use artificial neural network to achieve higher reliable trust and reputation about provider. In other word, as it's showed by literature review about trust and reputation systems, all the collected votes about reputation of a file provider from other peers, are in the same level. Whereas being more hops between responses to the requester causes more malicious responses and spoofing. In this paper we assume that peers in the first layer get to the requester through one hope, for the second layer there are two hopes and so on.
Since peers are heterogeneous, they may have different preferences and judge issues by different criteria. On the other hand in this paper same as [1] we consider different level of trust like Very Trustworthy, Trustworthy, Untrustworthy, Very Untrustworthy (see Table 1). We use these levels of trust to determine the output of neural network.

Table 1: Different level of Trust

Meaning	***Trust Level***
Very Trustworthy	VT
Trustworthy	T
Untrustworthy	U
Very Untrustworthy	VU

It is a difficult problem to predict the character of a client. Furthermore, because the open environments are dynamic, it is much more complicated to predict the distribution of clients with different characters on specific time. Therefore we use RBF neural network to evaluate other peer's recommendation. Artificial neural networks are robust to noise data and support incremental training. So our proposed trust model tries to overcome some disadvantages of previous trust systems by training the RBF neural network and tuning its weights by similarity matching to find the index of best center.
Before describing more details about application of neural network in our model, we need to explain about Gnutella phases as they presented in [3].

3.2 Phase 1: Resource Searching

At first peer **A** who is a peer that looking for a resource, broadcasts a Query indicating the resource it is looking for. Every other peers which receiving the query and willing to offer the requested resource for download, sends back a QueryHit message stating how it satisfies the query and providing its ID and its pair(IP, port), which peer **A** can use for downloading.

3.3 Phase 2: Polling

Upon reception of the QueryHit messages, peer **A** selects a top list of favorite peers T and polls its peers about the reputations of them. In the poll request, peer **A** includes IDs of peers in T about which it is enquiring and a public key PKpoll generated on the fly for the poll request, with which responses to the poll will need to be encrypted. The poll request is sent through the P2P network and therefore peer **A** does not need to disclose its ID or its IP to be able to receive back the response. Peers receiving the poll request and wishing to express an opinion on any of the peers in the list, send back a PollReply expressing their votes and declaring their (IP, port) pair. Peer A hash of the votes and pair (IP, port) is also added in order to allow peer **A** to check the integrity of the message. The PollReply is then encrypted with PKpoll to ensure its confidentiality (of both the vote and the voters) when in transit.

3.4 Phase 3: Vote Evaluation

As a result of the previous phase, peer **A** receives a set of votes, where, for each peer in T, some votes can express a good opinion while some others can express a bad opinion. To base its decision on the votes received, peer **A** needs to trust the reliability of the votes. Thus, peer **A** first uses the hash to detect tampered-with votes and discard them. Second, peer **A** detects votes that appear suspicious, for example since they are coming from IPs suspected of representing a clique. Third, peer **A** selects a set of voters that it directly contacts (by using the (IP, port) pair they provided) to check whether they actually expressed that vote. For each selected voter, peer **A** directly sends a TrueVote request reporting the votes it has received, and expects back a confirmation message TrueVoteReply. This forces potential malicious peers to pay the cost of using real IPs as false witnesses.

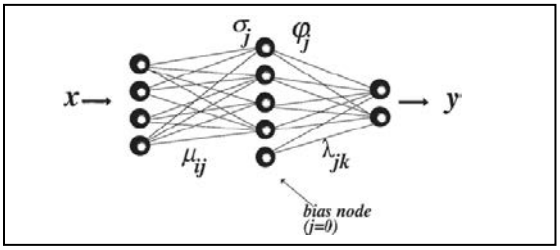
3.5 Application of RBF neural network in proposed trust model

When a peer's looking for a resource, broadcasts a query. Every other peers which willing to offer the requested resource, sends back a message. Then the requester polls about the reputations of the provider. After evaluating received recommendations or votes, finally the requester has to select the provider who seems to be the best on the list, in different aspects like download speed, file type and file quality. In this point we use Radial-Basis Function (RBF) artificial neural network to solve the problem (see Figure 1). RBF is a single-hidden-layer feed forward network with linear output transfer functions and nonlinear transfer functions, on the hidden layer nodes. RBF networks provide a powerful alternative to Multi-Layer Perception (MLPs) for function approximation or classification. They train faster and require fewer training samples than MLPs.

There are several techniques for training these networks. RBF networks are nonlinear hybrid networks typically containing a single hidden layer of processing elements. This layer uses Gaussian transfer functions, rather than the standard sigmoid functions employed by MLPs. The primary adjustable parameters in Figure 1 are the final layer weights, $\{\lambda_{jk}\}$, connecting the jth hidden node to the kth output node. There are also weights $\{\mu_{ij}\}$ connecting the ith input node with the jth hidden node [10].

The mathematical embodiment of the RBF takes the following form. The kth component of the output vector y_p corresponding to the pth input pattern x_p is expressed as:

$$[y(x_p)]_k = \sum_{j=0}^{h} \lambda_{jk}\, \phi_j\left(\|x_p - \mu_j\|; \Sigma_j\right) \quad (1)$$



Where $\phi_j(\ldots)$ denotes the nonlinear transfer function of hidden node j. In the RBF neural network there are three basic parameters like 1- centers, 2- spreads, 3- weights.

Fig. 1 The basic radial basis function structure[10].

We use K-means clustering algorithm for determining centers:

1. Initialization: random $\mu_j(t=0)$

2. Sampling: draw $x_p(t)$ from input space

3. Similarity matching: find index of best center
$k = \arg min_j \|x_p(t) - \mu_j(t)\|$ (2)

4. Updating: adjust centers
$\mu_k(t+1) = \mu_k(t) + \eta * [x_p(t) - \mu_k(t)]$ (3)

5. Continuation: increment t by 1, go to 2 and continue until no noticeable changes of centers occurred.

Next we use normalizing method to find spreads:
$$\sigma = \frac{Maximum\ \ distance\ \ between\ \ any\ \ 2\ centers}{\sqrt{number\ \ of\ \ centers}} = \frac{d_{max}}{\sqrt{m_1}} \quad (4)$$

$$\varphi_i(\|x - t_i\|^2) = exp\left(-\frac{m_1}{d_{max}^2}\|x - t_i\|^2\right)$$

$$i \in [1, m_1] \quad (5)$$

At last using LMS method for tuning weights:
they are part of a sentence, as in

$$\lambda_{jk} = \lambda_{jk} + \eta * [y(k) - y'(k)]$$

$$0 < \eta < 1, \quad (6)$$

y′ is predicted output of network and y is actual output of network.

$$y'(k) = \begin{cases} 1 & if\ confidence\ condition\ is\ true \\ 0 & otherwise \end{cases} \quad (7)$$

, y(k) = output of function φ from [0,1] (8)

Generally inputs of this RBF neural network are other peer's recommendations about a file provider based on their assigned layer. As mentioned before these recommendations are in VT, T, U, VU format. Thus total number of input neurons of this neural network will be equals to product of network TTL in these 6 trust levels. In order to normalization of inputs of trust levels, we divide number of received recommendations from each trust level into the all number of them. In RBF network hidden layer's neurons do the similarity matching and find index of best center. As previous terms express weights of hidden layer's neurons calculate by K-means clustering algorithm. Finally neuron's weights of output layer are difference between predicted output of network and actual output. The requester can finally download the resource and, depending on its satisfaction for the download, update its reputation information for the provider. Every peer keeps the last trained neural network in its memory. After each interaction, the neural network could be trained by requester's experiment and tuned network's weights. If the neural network was built beyond a certain period of time or some recommenders have changed their trust models, or the requester changes his trust estimation accuracy requirement, the requester collects up-to-date trust data and retrains the neural network. After a while the network is well trained and it would help peers to find reliable file providers.

3.6 Phase 4: Resource downloading

Peer A can finally download the resource and, depending on its satisfaction for the download, update its reputation information for peer **B**.

4. Experiments

We evaluate our system in a simulation of a peer-to-peer network with implementation of the trust computation model in RBF neural network developed with Matrics programming language on the Matlab.

4.1 Simulation setup

Our simulation involves 20 peers with 2 very trusted peers, 10 trusted peers, 6 untrusted peers and 2 very untrusted peers (malicious peers), and with a random topology of these nodes. For implementation of RBF neural network we assumed 20 nodes, each node has 4 states of pooling like VT, T, U and VU, so we had 4^{20} number of sampling spaces. First we made 1,000 samples randomly and applied K-means clustering algorithm for determining centers. Next the RBF neural network has been made up of 100 centers and it has been trained for 20,000 iterations.

4.2 Simulation results

The goal of this simulation is to see whether the trust and reputation system based on RBF neural network will help peers with different characters make accurate decisions and decrease the number of unreliable downloads. Thus we compare the percent of accurate suggestions for each training steps. In figure 2 we can see that the curve of *accurate suggestions* is increasing during the training steps. It means that output of the neural network finds the trust level of the file provider. It's been showed in figure 2 that training of the RBF neural network starts with 63 % of accurate suggestions in first step and then it's received to 89.9 % of accurate suggestions.

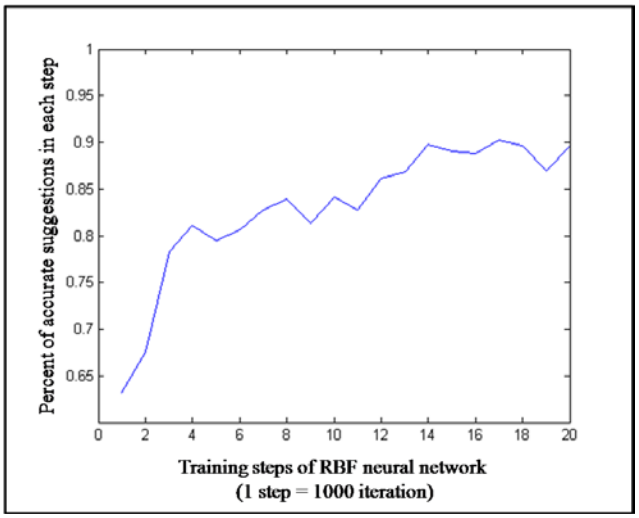


Fig. 2 Simulation result with 2 malicious in the network

4.3 Behavior analysis

In order to see the impact of increasing malicious peers in P2P network on output of reputation system that is based on RBF neural network, we have continued our simulation experiments by a P2P network with 2, 4 and 8 malicious peers in each time. Figure 2 is about the network with 2 malicious peers and in Figure 3 we can see the output of RBF trust model, while number of malicious peers has been duplicated. As shown in figure 3, training of the RBF neural network starts with 61% of accurate suggestions in first step and finally it reaches to 88%. Similarly, figure 4 with 8 malicious peers in P2P network indicates that rate of accurate suggestions from 66% reaches to 92% .

By comparing the result of simulation experiments which mentioned above, we can conclude that despite of increasing malicious peers in P2P network, there is no significant impact on training of the RBF neural network.

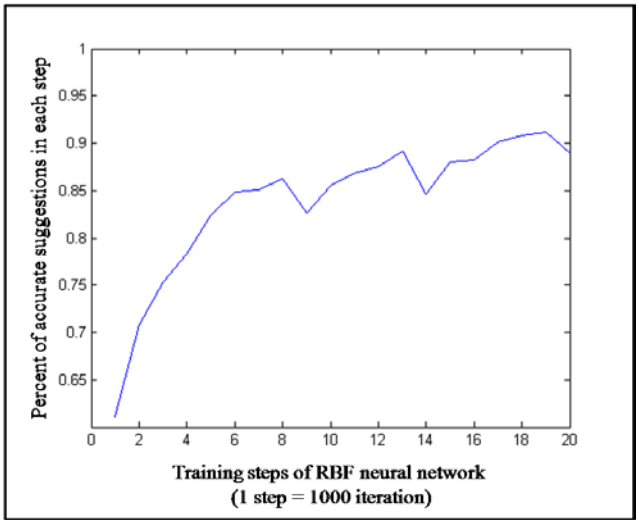


Fig. 3 Simulation result with 4 malicious in the network

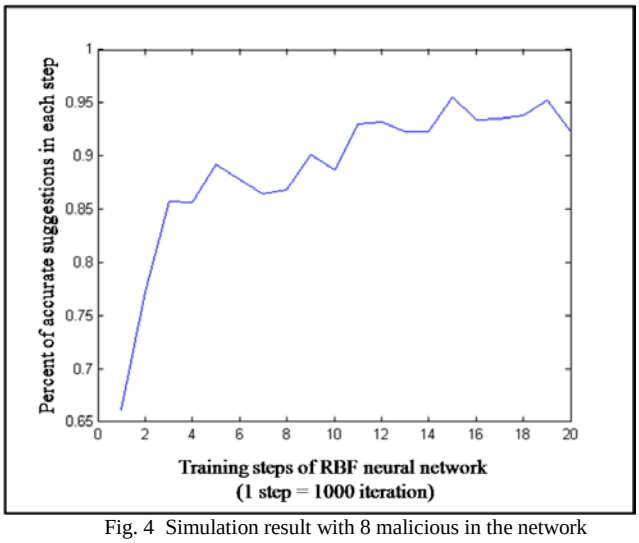


Fig. 4 Simulation result with 8 malicious in the network

5. Conclusions

In this paper we decide to use the RBF neural-network based recommendation trust model since experiments in this area have

discovered that hidden variables capture by hidden layers of the model, and these networks are robust to noise in the training data, also they have fast speed with high accuracy. In addition adaptability and non-linear aggregation of heterogeneous agent's recommendations are other properties of RBF based trust system.

References

[1] A. Abdul-Rahman, S. Hailes, "Supporting trust in virtual communities", In Proceedings of the Hawai'i International Conference on System Sciences, Maui, Hawaii, Jan 4-7 2000.

[2] F. Cornelli, E. Damiani, "Implementing a Reputation-Aware Gnutella Servent". In Proc. International Workshop on Peer-to-Peer Computing, Pisa, Italy, May 24, 2002.

[3] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, "Managing and sharing servents' reputations in p2p systems", IEEE Transactions on Knowledge and Data Engineering, 15(4), 840-854, 2003.

[4] S. Kamvar, M. T. Schlosser, and H. Garcia-Molina." The EigenTrust Algorithm for Reputation Management in P2P Networks". In Proc.12th International World Wide Web Conference (WWW'03), May, 2003.

[5] Y. Wang, J. Vassileva, " Trust and Reputation Model in Peer-to-Peer Networks", Proc. IEEE Conference on P2P Computing, Linkoeping, Sweden, 2003.

[6] W. Song, V. Phoha, X. Xu, "An Adaptive Recommendation Trust Model in Multiagent System", IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04).

[7] B. Zong, F. Xu, J. Jiao, J. Lv , "A Broker-Assisting Trust and Reputation System Based on Artificial Neural Network", In IEEE 978-1-4244-2794-9/09/ 2009.

[8] Baohua H., Heping H., Zhengding L.,"Identifying Local Trust Value with Neural Network in P2P Environment". IEEE 0-7803-9179-9/05, 2005.

[9] Fuke Sh., Pan Ch., Xiaoli R.,"Research of P2P Traffic Identification Based on BP Neural Network". in Proc. 3rd Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing(IIHMSP), Taiwan, Vol.2, pp. 75-78, 2007.

[10] D. Lowe, "Radial basis function networks and statistics, in Statistics and Neural Networks: Advances at the Interface", New York: Oxford University Press, pp. 65–95,1999.