Applied Soft Computing 34 (2015) 693–704

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

GenTrust: A genetic trust management model for peer-to-peer systems[☆]

Ugur Eray Tahta[a,b], Sevil Sen[a,*], Ahmet Burak Can[a]

[a] *Department of Computer Engineering, Hacettepe University, 06800 Ankara, Turkey*
[b] *ASELSAN, 06370 Ankara, Turkey*

ARTICLE INFO

Article history:
Received 12 July 2014
Received in revised form 9 March 2015
Accepted 28 April 2015
Available online 4 June 2015

Keywords:
Evolutionary computation
Genetic programming
Trust models
Reputation
Peer-to-peer systems

ABSTRACT

In recent years, peer-to-peer systems have attracted significant interest by offering diverse and easily accessible sharing environments to users. However, this flexibility of P2P systems introduces security vulnerabilities. Peers often interact with unknown or unfamiliar peers and become vulnerable to a wide variety of attacks. Therefore, having a robust trust management model is critical for such open environments in order to exclude unreliable peers from the system. In this study, a new trust model for peer-to-peer networks called GenTrust is proposed. GenTrust has evolved by using genetic programming. In this model, a peer calculates the trustworthiness of another peer based on the features extracted from past interactions and the recommendations. Since the proposed model does not rely on any central authority or global trust values, it suits the decentralized nature of P2P networks. Moreover, the experimental results show that the model is very effective against various attackers, namely individual, collaborative, and pseudospoofing attackers. An analysis on features is also carried out in order to explore their effects on the results. This is the first study which investigates the use of genetic programming on trust management.

1. Introduction

Open nature of peer-to-peer (P2P) systems facilitates join or leave of users to the network without worrying about any obligations. However this freedom creates potential threats for good behaving peers. Since there is no central authority to manage inter-peer interactions, malicious peers can easily perform attacks or take advantage of system resources without contributing to the system. A way to mitigate such threats is to create artificial trust relationships among users based on peer interactions. Trust models can help in such open environments to quantify trustworthiness numerically and create trust relationships among peers. However, it is hard to measure and formulate trust with numeric values. Furthermore, measuring trust without a priori knowledge is a challenging problem in P2P systems since peers mostly interact with unknown peers. Therefore, trust management in P2P environments is a difficult research problem.

When there is a central authority, trust management is relatively easy problem. In some e-commerce applications, a central authority collects user inputs about completed interactions and this information is used to make trust decisions about future interactions. Although fake users and interactions can pollute the collected information, this model mostly works on e-commerce applications. However, P2P systems need more complex trust management models due to the lack of a central authority. Peers need to store and manage trust information about each other [1–3]. On the other hand, uncertain information collected from neighboring peers might be deceptive. Malicious peers might deliberately provide wrong information to the system and this might not be detected since there is no central authority. Therefore, trust models in P2P systems should be able to recognize various attacks and help benign peers to find trustworthy peers. While doing this task, ambiguous information collected from other peers should be processed carefully to make correct decisions.

The trust decision problem can be considered as a classification problem, and machine learning techniques could be employed to distinguish malicious peers from benign peers. This paper proposes a genetic programming (GP) based trust management model (GenTrust), extending our previous work [4] with greater experimental

[☆] This paper is an extended, improved version of the paper "Evolving a Trust Model for Peer-To-Peer Networks Using Genetic Programming" presented at EvoComNet2014 and published in: Applications of Evolutionary Computing, Proceedings of 17th European Conference, EvoApplications 2014, Granada, Spain, April 23–25, 2014, LNCS 8602, Springer, 2014.

[*] Corresponding author. Tel.: +90 3122977500.
E-mail addresses: uetahta@aselsan.com.tr (U.E. Tahta), ssen@cs.hacettepe.edu.tr (S. Sen), abc@cs.hacettepe.edu.tr (A.B. Can).

http://dx.doi.org/10.1016/j.asoc.2015.04.053
1568-4946/© 2015 Elsevier B.V. All rights reserved.

verification and analysis on features. The proposed model helps to identify malicious peers and find trustworthy peers using the features derived from peer interactions and recommendations. The model has evolved with these features by using genetic programming, which provides a mathematical function to measure trust values of peers. A peer ranks its neighbors according to trust values, and makes trusting decisions using these values. Each peer stores trust relationships for the peers they have interacted in the past. As peers gain more neighbors with time, malicious peers are excluded from the system using trust relationships. The evolved model is evaluated against various attackers, namely individual attackers, collaborators, and pseudospoofers. The results show that the model decreases the number of attacks considerably. Features of the model are also analyzed and their effects on the performance is assessed. Satisfaction related features are found more influential in trust decisions. Cross training and testing are performed among various attacker types to understand the model's adaptability on different attacker behaviors. These experiments show that the models trained on complex attack behaviors are also successful in simple attack behaviors.

Organization of the paper is as follows. Section 2 gives a summary of the state of the art research. Sections 3 and 4 explain the proposed trust model and the simulation environment respectively. Section 5 presents the experiments and discusses their results extensively. Section 6 outlines the conclusions of the study.

2. Related work

Trust is a social concept and hard to measure and formalize with theoretical foundations. Although some approaches have formulated trust as a result of direct experiences [5], most trust models use recommendations of others to build trust relationships [6–8]. However, it might be hard to correctly evaluate trustworthiness using recommendations, since recommendations may contain deceptive or subjective opinions [9]. To better address different aspects of the trust, some approaches use trust and distrust concepts [10,11], and some approaches formulate trust in different contexts [12,13].

Although it is hard to quantify trust numerically, reputation systems provide a means to address trust concept. As in some e-commerce applications, the users with higher reputation might be considered as more trustworthy. However, ensuring honesty of feedbacks in reputation systems is still a problem [14,15]. Dissemination of bogus feedbacks with multiple fake users (sybil attacks [16]) should also be prevented. Otherwise, malicious users can create many fake users and pollute reputation of others according to their intention [17,18]. Furthermore, users should also have long-lived identities to build higher reputations [14]. Otherwise, it is hard to maintain reputation when users join and leave the system frequently [19]. If reputation and trust concepts are modeled correctly, economic activity can be increased in e-commerce applications since some activities may not happen without trust [20].

Trust models have found many applications in P2P systems due to their open and malicious nature [21,22]. Since interactions mostly happen among unfamiliar peers, a trust model can improve the success rate of interactions and prevent some attacks [1–3]. Trust models on P2P systems are affected by the structure of network. In the unstructured overlay networks like Gnutella [23], trust information about other peers are obtained by flooding trust queries to the network [2,24,25]. Generally, each peer stores trust information about its neighbors. Trust queries enable to collect recommendations about unfamiliar peers and make decisions about them. Some trust models are designed based on structured P2P networks [1,3,26]. A distributed hash table (DHT), such as Chord [27], is used to manage trust information. The DHT algorithm determines which peer(s) will be in charge of storing the trust information about a peer. This provides efficient access to the global trust information without flooding queries to the whole network. However anonymity of the trust holders are revealed in this approach. Some models proposed cryptographic protocols to provide anonymity for trust holders [28].

Most trust models on P2P systems are generally based on probabilistic and statistical methods. Aberer and Despotovic [1] assume that number of complaints can be a measure of trustworthiness. Some approaches apply basic majority voting and averaging principles on the information collected from past experiences and recommendations, such as in XRep [29] or P2PRep [30]. EigenTrust model[3] uses transitivity of trust to calculate indirect trust relations. PeerTrust [26] uses transaction and community context parameters in trust calculation so application dependent factors and whole system related issues can be addressed better. Wang and Vassileva [31] use a Bayesian network model to evaluate different aspects of interactions on a P2P file sharing application. Selcuk et al. [24] use a vector-based trust metric and limited flooding approach to evaluate trustworthiness. PowerTrust [32] utilizes a random-walk strategy and power nodes in an overlay network to improve global reputation accuracy. GossipTrust [25] defines a randomized gossiping [33] protocol for efficient aggregation of trust values. Nguyen et al. [34] propose a Bayesian model of trust and use different context of trust but do not provide any experimental results. Conner et al. [35] proposes a method for customized trust evaluations by using different scoring functions over the same feedback data. Josang et al. [36] uses subjective logic to analyze trust networks. M-Trust [37] utilizes confidence in reputation for a better trust evaluation. Wu [38] proposes a trust model for predicting availability of wireless links on mobile P2P networks. A stable group model is proposed to address mobility issues and increase trust query success rate. Selvaraj and Anand [39] propose credential trees for evaluating trust among peers. In this model, peers use policies and credentials to make trust decisions. Anand and Bhaskar [40] integrates a trust model with a security model to solve some security issues of P2P systems. The model increases controlled scalability and availability of content in P2P systems. SORT [41] uses service and recommendation contexts of trust to measure trustworthiness better in providing services and recommendations.

Although most trust models use probabilistic and statistical methods, there are some approaches using machine learning techniques to classify good and malicious peers. Weihua Song et al. [42] use neural networks to derive trust values in multi agent systems. Neural network approach helps to classify recommendations as qualified or unqualified when choosing service providers. Beverly and Afergan [43] use a support vector machine based approach to select neighbors efficiently and then reduce the communication cost. Linear discriminant analysis and decision trees are used by Liu et al. [44] to help peers to build a knowledge base using past interaction history, which helps to identify successful transactions. Some approaches use Hidden Markov Models (HMM) in trust models [45–47] since HMM can be an effective method to model behaviors of entities in a system efficiently.

The evolutionary computation techniques are mainly applied to intrusion detection in the security domain [48,49]. Most of these studies focus on developing effective and efficient detection methods. Moreover they usually apply either genetic programming (GP) or genetic algorithms (GA). The first GP application for intrusion detection was from by Crosbie and Stafford [50]. Since then, there are many useful applications in the field. In [51], Abraham and Grosan compare the genetic programming technique with other machine learning methods for intrusion detection [51], and show that genetic programming techniques both outperform other techniques and are lightweight. Another advantage of evolutionary computation is to generate readable, easy-to-understand