entropy

MDPI

Article

A Cloud Theory-Based Trust Computing Model in Social Networks

Fengming Liu [1,*], Xiaoqian Zhu [1], Yuxi Hu [2], Lehua Ren [1] and Henric Johnson [3]

1 School of Management Science and Engineering, Shandong Normal University, Ji'nan 250014, China; zhuxq2015@126.com (X.Z.); renlehua@163.com (L.R.)
2 Computer Science Faculty, The University of California at Davis, One Shields Ave., Davis, CA 95616, USA; huyuxi@163.com
3 Faculty of Computer Sciences, Blekinge Institute of Technology, 37141 Karlskrona, Sweden; henric.johnson@bth.se
* Correspondence: fmliucn@gmail.com; Tel./Fax.: +86-531-8618-0508

Academic Editor: Raúl Alcaraz Martínez
Received: 4 November 2016; Accepted: 22 December 2016; Published: 28 December 2016

Abstract: How to develop a trust management model and then to efficiently control and manage nodes is an important issue in the scope of social network security. In this paper, a trust management model based on a cloud model is proposed. The cloud model uses a specific computation operator to achieve the transformation from qualitative concepts to quantitative computation. Additionally, this can also be used to effectively express the fuzziness, randomness and the relationship between them of the subjective trust. The node trust is divided into reputation trust and transaction trust. In addition, evaluation methods are designed, respectively. Firstly, the two-dimension trust cloud evaluation model is designed based on node's comprehensive and trading experience to determine the reputation trust. The expected value reflects the average trust status of nodes. Then, entropy and hyper-entropy are used to describe the uncertainty of trust. Secondly, the calculation methods of the proposed direct transaction trust and the recommendation transaction trust involve comprehensively computation of the transaction trust of each node. Then, the choosing strategies were designed for node to trade based on trust cloud. Finally, the results of a simulation experiment in P2P network file sharing on an experimental platform directly reflect the objectivity, accuracy and robustness of the proposed model, and could also effectively identify the malicious or unreliable service nodes in the system. In addition, this can be used to promote the service reliability of the nodes with high credibility, by which the stability of the whole network is improved.

Keywords: social network; cloud model; trust evaluation; reputation trust; transaction trust

1. Introduction

With the rapid development of social networks, they have been widely used in a variety of different fields, such as file sharing, information retrieval, and collaborative computing [1,2]. In a social network, each node is independent and contributes two roles: server and client. Different nodes have different service ability and reliability, and the participation of each node is voluntary and random. These characteristics determine that the traditional network security management and control methods cannot be implemented effectively. Therefore, there can be many frauds and unreliable services in a social network, such as document forgery in file sharing systems, and also random events that can suspend the services in social systems [3,4]. All these drawbacks seriously affect the network performance, therefore, building a reliable trust management system which can significantly improve the service performance and fundamentally promote the development of peer-to-peer (P2P) networks is of great importance [5].

Trust management has been extensively studied in the past decades, and various kinds of trust models have been proposed [6–17]. Although these models have provided the corresponding evaluation methods or mechanisms to partially solve trust problems in different application scenarios, the description and solution of the randomness, fuzziness and unpredictability of trust in complex network environments still needs to be improved. The randomness, which also called the contingency, is an uncertainty feature which is manifested by whether the result of an event happens or not. The fuzziness, also called unclarity, comes about because of the fuzziness of concepts, so it is hard to confirm whether an object fits a concept or not, so there is no clear definition in quality, and there is also no clear boundary in quantity. The randomness and fuzziness are often interconnected, and it is difficult to distinguish their independent existence. Human cognition is essentially an image of the objective world. The uncertainty of the objective world determines the uncertainty of the human subjective cognitive process. Therefore, the unpredictability of trust is an uncertainty of human subjective cognition, which manifests as randomness and fuzziness. To address this issue, corresponding solutions that leveraged probability theory [18] and D-S theory [19,20] have been proposed, where the randomness of trust is emphasized but the fuzziness of trust is not objectively reflected. On the other hand, based on fuzzy logic, Esposito et al. [21] proposed the concept of smart cloud storage service selection. The service selection was resolved with the distributed application of fuzzy inference or Dempster-Shafer theory of evidence. Ullah et al. [22] and Hao et al. [23] applied fuzzy theory to study trust, and described the fuzziness of trust in details, but the randomness of trust has not been well considered. Overall, the problem with the above solutions is that they fail to comprehensively consider the relationship between randomness and fuzziness and draw a sharp line of equality between them. Instead, Li [24] proposed the cloud model theory that organically integrates fuzziness and randomness. This model better reflects human understanding of the essence of the objective world. Therefore, our trust model is mainly based on the cloud model theory to construct trust, which reflects the nature of trust and makes the result of trust evaluation more objective.

In this paper, we propose a trust management model based on a cloud model that uses a specific tectonic operator to achieve the transformation from qualitative concepts to quantitative computation. Our model can effectively express fuzziness, randomness and the relationship between them for the subjective trust. In the node trust we distinguish between reputation trust and transaction trust, and design their respective evaluation methods. To get the reputation trust, we design a two-dimension trust cloud evaluation model based on nodes' comprehensive and trading experience. The expected value reflects the average trust status of the nodes. We use entropy and hyper-entropy to describe the uncertainty of trust. Then, we propose calculation methods to comprehensively compute the transaction trust of each node. Thus, based on trust cloud, the node can choose a trade strategy. We performed simulation experiments on the Ecological Network Computing Environment (ENCE) platform, and the results demonstrate the objectivity, accuracy and robustness of our cloud trust model. It can effectively identify malicious or unreliable nodes to promote the service reliability of the nodes with high credibility and stability, so the service stability of the whole network is improved. The rest of the paper is organized as follows: Section 2 introduces the related work in the areas of cloud models and trust models. Section 3 describes the design and implementation of the proposed trust computing model based on cloud theory in details. Section 4 discusses the experiments carried on a standard dataset and demonstrates the efficiency of our trust model. Finally, Section 5 summarizes this paper and points out possible future directions of this research.

2. Related Works

2.1. Cloud Model

The cloud model is a qualitative and quantitative conversion model proposed by Li in [24] which can realize the uncertainty conversion between a qualitative concept and its quantitative counterpart. It mainly reflects two kinds of uncertainty: fuzziness and randomness.

It is common that there are uncertain phenomena and things in Nature and human society. How to express and process these uncertain phenomena and things in a better way has always been an important research topic in the field of natural science. The main tool to study random phenomena is probability theory. Probability theory quantitatively describes the likelihood of uncertain events, which lays a solid theoretical foundation to mathematically understand the two phenomena of uncertainty and randomness. Since Zadeh created fuzzy set theory and proposed the concept of fuzzy information processing methods in [15], fuzzy set theory has gradually become the main tool to deal with vagueness and uncertainty with a considerable number of achievements in both theory and application. As a consequence, the cloud model theory is developed on the basis of cross-penetration of these two theories, which forms a qualitative concept and its quantitative conversion model through a specific construction algorithm, effectively putting fuzziness and randomness together.

The cloud model is primarily reflected by three digital characteristics of the cloud: Expectation (*Ex*), Entropy (*En*) and Hyper-entropy (*He*), which integrate the fuzziness and randomness to reflect the quantitative characteristic of the qualitative concept on the whole [25,26]. *Ex* is the Expected value, which is defined as the computed mean value of cloud drops. *En* is the Entropy value, which is definied as the degree of uncertainty for qualitative concepts. *He* is the Hyper-entropy, which is definied as the degree of uncertainty of entropy.

A cloud model can better express the randomness and fuzziness of Nature without the need to determine the membership function through a significant number of experiments and accumulated experience. The basis of a fuzzy set is a membership function, and the problem of fuzzy math application is how to accurately determine the correct membership function in a fuzzy concept. Because there is no need to determine membership functions in cloud model theory, it is applicable in many areas to measure specific uncertain phenomena. In the field of intelligent control, Li et al. [27] discussed the objectivity, ordinary and meaning of uncertain existence in human knowledge and intelligent computing. In the area of network security, Zhao et al. [28] proposed the incorporation of cloud model theory into network intrusion detection methods. Di et al. [29] introduced the cloud model theory into spatial data mining and knowledge development for qualitative and quantitative conversion, concept decomposition and knowledge expression which provides new methods for generating concept from data and concept hierarchy structures. He et al. [30] proposed a trust model based on a cloud model, introducing the cloud model theory into the trust model and describing the trust relationship through a trust cloud.

2.2. Trust Model

The foundation of a trust model is based on how to measure and compute the degree of trust. Trust is a common phenomenon in society and exists in many fields such as technology, politics and psychology. Marsh [31] first proposed trust models in sociology and psychology. Although the model applies a trust model in the computer related research field, it is too complex with the introduction of a large number of variables and over-emphasizes the importance of the agent (an intelligent node which can make decisions by itself). In addition, it also ignores the evaluation of the other agents in the network. All these features make it difficult to realize Marsh's trust model in the current P2P network structure.

Beth et al. [18] proposed the trust quantification concept by dividing the trust into recommendation trust (that comes from the third-party node without direct connection in a network) and direct trust (that comes from the direct connection node in a network) and defining the methods of trust computing. However, the shortcomings of this model are two-fold: first, the scale used to define the success of trust is not compliant with the understanding of trust in reality; secondly, using the simple arithmetic average method to calculate trust degree provides opportunities for malicious recommendations on the recommended path. Moreover, Blaze et al. [6] first introduced trust management jargon and defined trust management. Based on trust management, various fields have developed different automatic management systems, for instance see Schilke et al. [7], Rose et al. [9] and Jairak et al. [32].

Abdul-Rahman et al. [33] further introduced an approach that can achieve trust in a P2P network based on Marsh's model with a small amount of calculation, but it is time-consuming to maintain and update the complex large database of each node and the model has no specific description of how to recommend other nodes.

Das and Islam [34] proposed a trust computation model called SecuredTrust for evaluating agents in multi-agent environments. By analyzing ten different parameters related to evaluating the trust of an agent, they proposed a comprehensive quantitative model for measuring trust. This model achieved a good transformation from qualitative concept to quantitative computation.

Chatzopoulos et al. [35] also integrated an incentive scheme and a reputation mechanism, and proposed a framework which addressed any mechanism that considers selfish users. The peer-to-peer reputation exchange scheme showed that the degrees of trust of nodes could be calculated using the feedback of the other nodes. Chatzopoulos et al. also proposed a reputation middleware named OPENRP, which provided a unified interface for crowd computing and opportunistic networking applications [36]. The middleware evaluated and updated the reputation of participating peers based on their mutual opportunistic interactions. The influence of reputation is considered when establishing the node trust. However, the recommendation trust of a node couldn't be ignored.

Wang et al. [37] proposed a trust reputation model based on a Bayesian network. The model adopted trust and reputation mechanisms, respectively, to deal with direct trust and recommendation trust. It focused on solving the recommendation trust, whose theoretical basis is the Bayes formula. However, its reasoning is purely built on a probability model without considering the fuzziness of trust itself and kept away from the trust management practice. Yuan et al. [38] and Tang et al. [39] considered the fuzziness of the subjective trust by constructing a subjective trust management model on the basis of fuzzy set theory, and introduced linguistic variables and fuzzy logic into the research on subjective trust reasoning. However, the predetermined membership function in fuzzy mathematics makes the study lack flexibility.

Wang et al. [40] proposed a subjective trust quantification evaluation method based on a cloud model. The method introduced three digital characteristics of subjective trust to quantify on object's trust and the changes of object trust were described by a mutative trust cloud. This model also combines the fuzzy and uncertainty together, and provides a theoretical basis for trust decisions.

Although trust models have been studied extensively, they all have their own limitations. These limitations are mainly reflected in two aspects: the uncertainty of trust and fuzziness of subjective trust are not well balanced, and only taking the nodes' trust and distrust degrees into account makes any accurate evaluation of the service of nodes unrealistic. In the meantime, the study on cloud theory trust models in recent years has exposed some problems, such as a lack of trust parameter acquisition methods, over-complicated trust evaluation methodology and a lack of ways to prevent unreasonable recommendation algorithms [41]. Other major problems of the cloud theory trust model are the storage of trust parameters and how to manage and express the entity trust effectively in a simple and accurate way.

3. Trust Computing Based on a Cloud Model

There is an uncertain trust relationship before transactions in social network nodes, which needs to select a target for judgment. We need to construct a model of trust to distinguish the uncertainty of trust in a way that human minds can accept and evaluate the trust condition of nodes. How to derive trust information effectively and how to accurately express trust information, including the trust degree calculation methods as well as trust reasoning mechanism is the key issue when building a trust evaluation model. In this paper, a cloud model theory is applied in the trust evaluation process of P2P networks, describing the trust information in the form of a trust cloud. The advantage of a qualitative and a quantitative conversion model has been brought into full play, which allows us to conduct the trust evaluation process more realistically and accurately.

3.1. Definitions

3.1.1. Definition 1: Node

The node is the provider or consumer who provides or consumes services in a social network. The services in a social network need to pay or not to pay, therefore, it is called a transaction.

3.1.2. Definition 2: Reputation Trust Degree

The reputation trust degree is a comprehensive assessment of a node's reputation obtained from other nodes by evaluating all transaction information. The reputation trust degree is expressed by *STrust* in this paper, where $STrust \in [0, 1]$. The larger the value of *STrust* is, the better the credibility of nodes and the higher the level of trust.

3.1.3. Definition 3: Transaction Trust Degree

The transaction trust degree is a node's trust degree obtained by evaluating the historical transaction records. It provides a basis to predict its future trust status. The trading trust degree is expressed by *TTrust* in this paper, where $TTrust \in [0, 1]$. The larger the value of *TTrust* is, the more the previous trading behavior is trusted. The transaction trust degree is divided in two parts: direct transaction trust degree and recommended transaction trust degree.

3.1.4. Definition 4: Direct Transaction Trust Degree

The direct transaction trust degree is a node's transaction trust degree from the direct transacting nodes. The direct transaction trust degree is expressed by *DTrust*, $DTrust \in [0, 1]$.

3.1.5. Definition 5: Recommended Transaction Trust Degree

The recommended transaction trust degree is a node's transaction trust degree obtained from a third-party's evaluation, for which there exists no historical transaction records in the past. The recommended transaction trust degree is expressed by *RTrust*, $RTrust \in [0, 1]$.

3.1.6. Definition 6: Cloud Model

The cloud model is denoted as the set $T(Ex,\ En,\ He)$. The cloud $T' = (kEx,\ kEn,\ kHe)$ is called the product of cloud T with constant k, denoted as $k \times T$.

3.1.7. Definition 7: Cloud Synthesis

Given two trust clouds $T_1(Ex_1,\ En_1,\ He_1)$ and $T_2(Ex_2,\ En_2,\ He_2)$, the synthesis of the cloud union is written as $T = T_1 \oplus T_2$, in which $\oplus$ represents a logical add.

3.1.8. Definition 8: Trust Clouds

A trust cloud is also called a trust evaluation cloud, which is denoted as the set $TC(Ex,\ En,\ He)$.

3.2. Model Design

3.2.1. The Basic Idea

The internal attributes and external manifestations of things are closely related because the internal real properties are reflected by the external information. The novel trust cloud evaluation model designed in this paper is based on a two-dimensional metric: reputation trust and transaction trust. We leverage the main qualitative and quantitative conversion tool cloud model to effectively quantify the comprehensive status and historical transaction behavior of the nodes in order to make trust evaluations for the nodes in the network. Figure 1 shows the node comprehensive trust degree calculation model.

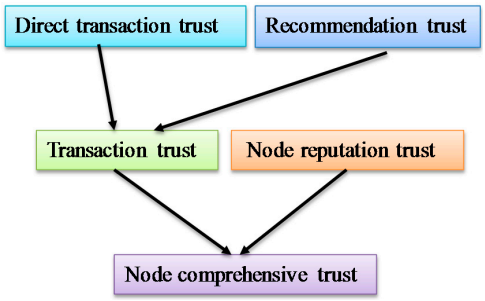

Figure 1. Comprehensive trust model of nodes.

The comprehensive trust degree evaluation method is designed accordingly; when a node p wants to transact with node q, p first obtains the reputation trust degree of q, and then derives the transaction trust of node q according to the direct transaction trust it had with node q and the recommendation trust from the recommended node. The formula of calculating the comprehensive trust degree of p to q is as follows:

$$T_{pq} = \omega_i \times STrust_q + \omega_j \times TTrust_{pq} \quad (1)$$

Among them, $STrust_q$ is the reputation trust degree of node q, $TTrust_{pq}$ is the transaction trust degree of p to q, ω_i and ω_j are the weights in the calculation process.

3.2.2. The Index of Trust Evaluation

Selecting evaluation indexes that are easy to measure and have strong representatives from a large amount of evaluation information or original data is another major component in the proposed model. We follow the principle of dynamics and diversity on the basis of scientific, hierarchy, computability, maneuverability and competences. We define different trust evaluation indexes for different evaluated objects. The evaluation index means different attributes of node for evaluating trustworthiness, as presented in Figure 2.

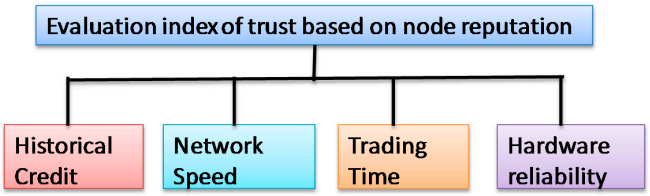

Figure 2. Trust evaluation index.

3.3. Computation of Reputation Trustworthiness Based on the Cloud Model

3.3.1. Trust Classification and Quantitative Description

It is extremely difficult to make accurate numerical evaluations because of the inherent fuzziness of trust. Therefore, it is not necessary to accurately classify the trust level in order to maintain the semantic information the trust itself contains. In this paper, we use discrete values to describe the trust degree levels. The trust relationship between nodes is expressed by the level of trust. The higher the level is, the more reliable and the higher the credibility is. We define five trust levels of nodes as follows:

Level 1: the node's credibility is quite poor, so it cannot be trusted completely.
Level 2: the node's credibility is poor and the service quality needs to be improved in the transaction process.
Level 3: the node's credibility is at a middle level, with basic credibility.

Level 4: the node has a high credibility and trust comprehensive conditions are high.
Level 5: the node is completely credible, and one can be at ease with it.

In this paper, we denote trust levels with trust space TS, a set of trust levels. Each element in this set represents a qualitative notion of trust in natural language. The trust space is defined as below:

$$TS = \{“full - confidence”, “high - confidence”, “basic - confidence”, “low - confidence”, “no - confidence”\}$$

3.3.2. Trust Cloud Parameter Acquisition

We use a discrete interval scale $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5)$ to represent the value of five trust levels in trust space TS and the credentials of the expressed qualitative concept. The credibility value of a node and its uncertainty degree are both in the range of [0, 1], as shown in Table 1.

Table 1. Description and scale of credibility level.

Trust Level	Trusted State Language Description	Trust Scale	Times of Evaluation
1	No-confidence	$\beta_5 = 0$	α_5
2	Low-confidence	$\beta_4 = 0.25$	α_4
3	Basic-confidence	$\beta_3 = 0.5$	α_3
4	High-confidence	$\beta_2 = 0.75$	α_2
5	Full-confidence	$\beta_1 = 1$	α_1

According to the trust space TS and discrete interval scale $(\beta_1,\ \beta_2,\ \beta_3,\ \beta_4,\ \beta_5)$, we use a five-dimensional vector $V_{AB}(\alpha_1,\ \alpha_2,\ \alpha_3,\ \alpha_4,\ \alpha_5)$ to express the evaluation information of expert A to node B in the trust assessment process, which is called the trust vector between A and B. The expert node is a kind of special node to evaluate the services of other nodes, which is an integral part of the P2P network. We ask that its trusted state language description be full-confidence. α_1 is the time of full-confidence from node A for evaluating the services of node B, and so on. The evaluation information of experts is collected repeatedly on the nodes of given evaluation indexes that correspond to the levels of trust in the trust space. For example, expert A evaluates an index of evaluation object B for 100 times rating, including 60 times as the basic credible, 20 times as the higher credible, 20 times for completely credible. For this index, the trust vector between A and B can be expressed as $V_{AB}(20,\ 20,\ 60,\ 0,\ 0)$.

Through the reverse cloud generator algorithm [23], we realize the conversion from quantitative trust vector to the three cloud parameters in the trust evaluation cloud $TC(Ex,\ En,\ He)$.

The uncertainty in cloud mode is described by the other two characteristic parameters entropy and hyper-entropy and entropy and hyper-entropy are defined by the standard deviation. Therefore, we use the two cloud parameters En and He in the trust evaluation cloud to describe the uncertainty of trust. The definitions of Ex, En, He are as follows:

$$Ex = \frac{1}{n}\sum_{i=1}^{n}\beta_i\alpha_i \tag{2}$$

$$En = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\beta_i\alpha_i - Ex)^2} \tag{3}$$

$$He = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\beta_i\alpha_i - Ex)^2 - En^2} \tag{4}$$

These three digital characteristics are the basis of the cloud model, and we finally get the cloud parameter values Ex = 0.65, En = 0.2037, He = 0.02. Figure 3 shows a trust relationship nephogram of $V_{AB}(20,\ 20,\ 60,\ 0,\ 0)$.

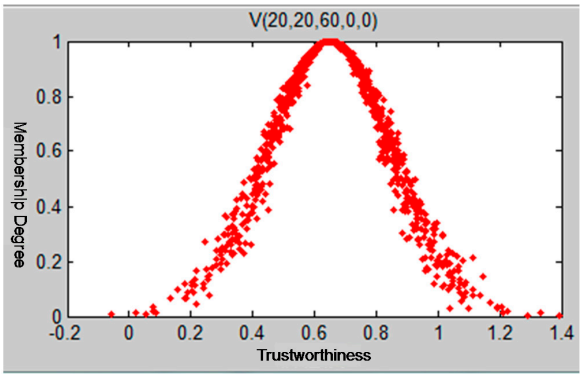

Figure 3. Nephogram of $V_{AB}(20,\ 20,\ 60,\ 0,\ 0)$.

The membership degree is the degree of a node that belongs to the confidence interval, in which it corresponds to the trustworthiness.

3.3.3. Computation of the Trust Evaluation Cloud

According to the trust index assessment system, there are two steps to calculate a node's trust evaluation cloud. First we calculate the trust evaluation cloud of the evaluated node for each index. Then based on the index weight factors, we combine the calculation of the evaluation value on the same hierarchy by following the steps to a target derivation. Finally, we can get the trust cloud parameter values of the evaluated node through computing. The method to calculate the trust evaluation cloud of an index is as follows:

Let set $Z = \{Z_1, Z_2, \cdots, Z_n\}$ represent the n expert nodes and $X = (x_{ij})_{n\times m}$ be the evaluation matrix of m indicators for n nodes to the evaluated nodes. Then x_{ij} is the evaluation value of the jth indicator of the evaluated node B from node $x_i (i = 1, \cdots, n; j = 1, \cdots, m)$. So, the Algorithm 1 is shown as:

Algorithm 1:

Input: the evaluation value of n experts' nodes to the jth evaluation index of evaluated B, $\Omega_j = (x_{1j}, x_{2j}, \cdots, x_{nj})$;

Output: the cloud of trust evaluation $TC_j(Ex_j,\ En_j,\ He_j)$ evaluated node B on jth index.

(1) According to the set Ω_j, calculate the sample average value $\overline{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$, the standard variance $std_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2}$ and the sample variance $s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2$.

(2) Calculate $\hat{Ex}_j = \overline{x_j}$.

(3) Calculate $\hat{En}_j = std_j$.

(4) Calculate $\hat{He}_j = \sqrt{s_j^2 - \hat{En}^2}$.

We also develop the method of calculating combined multi-indicators. In the trust evaluation index system, the process to combine each evaluation index of the same level into a comprehensive evaluation cloud is called cloud synthesis, which essentially generates the node's trust evaluation cloud. Because of the weight of each index, in the cloud synthesis process, the weighting factors of all indexes for the evaluation of cloud should be taken into account, denoted as $\omega = \left\{\omega_j > 0, \sum_{j=1}^{m}\omega_j = 1\right\}$.

According to Definitions 6 and 7, the formula of m index evaluation cloud weighted synthesis is:

$$T(Ex, En, He) = (\omega_1 \times T_1) \oplus (\omega_2 \times T_2) \oplus \cdots \oplus (\omega_m \times T_m) \tag{5}$$

where:

$$Ex = (\omega_1 Ex_1, \omega_2 Ex_2, \cdots, \omega_m Ex_m) = \sum_{i=1}^{m} \omega_i Ex_i \tag{6}$$

$$En = (En_1, En_2, \cdots, En_m) = \frac{1}{n} \sum_{i=1}^{m} En_i \tag{7}$$

$$He = (He_1, He_2, \cdots, He_m) \tag{8}$$

3.3.4. Computation of Reputation Trust

According to the node's comprehensive conditions, we use the weighted synthesis formula to get the node reputation trust evaluation cloud. Under the premise of getting the node evaluation cloud, we compare the trust basic cloud with the node trust evaluation cloud.

Let set $TB_j(Ex_j,\ En_j,\ He_j)(j = 1, 2, \cdots, 5)$ be a trust basic cloud and TC(Ex, En, He) be the trust evaluation cloud. Through the normal cloud generator, we generate the cloud droplets (x_j, u_j) of cloud TC, $i = 1, 2, \cdots, N$ Let set η_i be the degree of the membership for each x_i in basic cloud TB_j and $\delta_j = \frac{1}{N}\sum_{i=1}^{N} \eta_i$ be the similarity between cloud TC and the basic cloud.

The algorithm to calculate similarity between two clouds is given as follows:

(1) Generate a normal random member $En' = N(En, He)$ in the cloud TC, in which En is the expected value and He is the standard deviation.

(2) Generate a normal random member $x_i = N(Ex, En')$, in which Ex is the expected value in the cloud TC and En' is the standard deviation.

(3) Generate a normal random member $\hat{En_j} = N(En_j, He_j)$ in the cloud TB_j in which En_j is the expected value and He_j is the standard deviation.

(4) Calculate $\eta_{ij} = exp\frac{-\left(x_i - Ex_j\right)^2}{2\left(\hat{En_j}\right)^2}$.

(5) Repeat the above four steps sequentially until the required N of η_{ij} is generated.

(6) Calculate $\delta_j = \frac{1}{N}\sum_{i=1}^{N} \eta_{ij}$.

The larger value of δ_j indicates the greater similarity between the cloud TC and basic cloud TB_j, thus the node being evaluated is closer to the established standard. The higher trust degree means that the node is more credible as the trust degree equals their similarity.

3.3.5. Decision Algorithm of Trust Cloud

Let set $TS(Exs,\ Ens,\ Hes)$ to be the trust basic cloud, namely the standard trust cloud and $TC(Ex,\ En,\ He)$ to be the node's trust evaluation cloud. The trust cloud decision Algorithm 2 is then as follows:

Algorithm 2:

Input: trust basic cloud $TS(Exs,\ Ens,\ Hes)$, the node's evaluation trust cloud $TC(Ex,\ En,\ He)$

Output: the value of trust degree

(1) If $En/He < \sigma$ return 0, which means that the trust evaluation cloud is too discrete. Where, σ is called the uncertainty factor, whose value is generally derived through statistical analysis.

(2) In the case that $Ex \geq Exs$: if $En \leq Ens$ return 1, which means that the trust evaluation cloud is higher than the trust reference; if $Ex - 3En \geq Exs - 3Ens$ return 1, which means that trust evaluation cloud has high uncertainty, but its total trust value is higher than the reference value of the cloud; in other situations, we use the method of computing similarity between clouds introduced in the last section to make a judgment.

(3) In the case that $Ex < Exs$: if $En \leq Ens$ return 0, which means that trust evaluation cloud is smaller than the trust reference value, the larger uncertainty; if $Ex < Exs$, $En \leq Ens$ and $Ex - 3En \geq Exs - 3Ens$, we use the method of computing similarity between clouds. In other situations, return 0. This means that the total trust value of the trust evaluated cloud is less than the value of the reference cloud.

If $\sigma < 5$, the trust cloud is in the criteria of the evaluation node's trust. If it is greater than this value, the data has a high degree of dispersion, because hyper entropy is a measure of the degree of the entropy's uncertainty change. As long as the uncertainty factor is not greatly beyond the range, it generally will not affect the trust judgments. We can only take the expected value of the two parameters and entropy for analysis. The larger the expected value is, the smaller the range of random distribution of the trust cloud is. It is an ideal trust cloud that can reflect the trust information more effectively and more stably through our cloud model.

3.4. Computation of Transaction Trust Based on Cloud Model

In the process of P2P network trust evaluation, it is also necessary to take the node's historical trading experience into consideration after calculating the node's reputation trust degree, which reflects the complete credible degree of a node. Obviously, the trading trust degree is a continuous accumulation process. Recording the feedback evaluation data of the trading nodes after each transaction and the node's experience will become richer and richer as the number of transactions increases.

We then conduct a comprehensive analysis calculation by balancing weights between the direct transaction trust and recommendation trust. In the initial stage, when the node in the network wants to trade with strange nodes, the information of the strange nodes' trust condition is largely collected from the recommendation of other entity that has traded with this node. After having several transactions with the node, when calculating the trading trust, both the direct trading historical experience with itself and other node's recommendation are considered. When two nodes have a certain number of transactions, they establish mutual trust with each other. In this case, we can rely more on direct trading information when calculating the interact trust. The calculation formula for the trading trust degree is given as follows:

$$Trust_{pq} = ep_{pq} \times \lambda \times DTrust_{pq} + \left(1 - ep_{pq}\right) \times RTrust_p \tag{9}$$

where $DTrust_{pq}$ is the direct trading trust of node p to node q and $RTrust_p$ is the recommendation trust about node q. The parameter ep_{pq} is called the balance weight, which reflects the experience value of node p to q. The greater its value is, the more experience node p has, and thus the more sureness for the judgment of the trust condition of the node q is. λ is the time attenuation factor.

3.4.1. Computation of Direct Trading Trust

By querying the evaluation records of node p to q in the time window $win = [t_{start}, t_{end}]$, the formula of calculating direct trading trust is as follows:

$$DTrust_{pq} = S_{pq} = \sum_{k=1}^{I(p,q)} \frac{S_{pq}(k)}{I(p,q)} \tag{10}$$

where, $S_{pq} = \left(S_{pq}(1), S_{pq}(2), \cdots, S_{pq}(\mathrm{n})\right)$ is called the direct trading trust degree, which is the feedback score of node p to trade with node q for several transactions in the time window $win = [t_{start}, t_{end}]$. $I(p,q)$ is the trading times of node p to q. In order to obtain more accurate results, we introduce the following three factors into Equation (10).

The first factor is the time attenuation factor λ. The trading feedback rating can better reflect the node's recent behavior and trust condition if the time is closer. On the contrary, the trading feedback rating has a smaller impact on calculating the direct trading trust degree if the time is longer.

The second factor is the volume factor $C(i)$. When the node's turnover is larger, the feedback score can affect the direct trading trust degree more. The volume factor can effectively prevent some nodes from using the trust value accumulated by some small amounts in large transactions. In addition, the serious attitude of trading nodes to the large amount of transactions makes the evaluation results reflect the node's behavior more accurately. The following formula is used to calculate the volume factor:

$$C(i) = \frac{amount(i)}{amount(i) + M} \tag{11}$$

where $amount(i)$ is the amount of the *ith* transaction and M is the control coefficient of the turnover factor, whose value is a positive number.

The third factor is the acceleration factor $A(i)\frac{1}{1+e^{-n}}$, where n is the number of failed transactions. The acceleration factor urges the trust value to drop rapidly when the transaction fails. In order to avoid the situation that nodes are punished because of one or two inadvertent errors, its value will instead increase rapidly when the number of failures increases.

At the moment t, the assessment of node p to node q is $S_{pq}(i)$, $S_{pq}(i) \in [0, 1]$, whose value can be (1, 0.75, 0.5, 0.25 or 0). t' is the time of a transaction in time window $win = [t_{start}, t_{end}]$. The attenuation factor is $\lambda^{t-t'}$ and $I(p, q)$ is the trading times of node p with node q. Then Equation (10) can be rewritten as Equation (12) after incorporating the above three factors, which provides a more comprehensive method to calculate the direct trading trust:

$$DTrust_{pq} = \sum_{i=1}^{I(p,q)} \frac{S_{pq}(i) * \lambda^{t-t'} * C(i) + A(i) * \frac{1}{1+e^{-n}}}{I(p, q)} \tag{12}$$

3.4.2. Computation of Recommendation Trust

In most existing social network trust evaluation models, the calculation of recommendation trust mainly takes the credibility of the recommended node into account. In this paper, we introduce the experience factor and time attenuation factor as well as the turnover factor into the calculation process. Assumes that node q has direct transactions with n other nodes in the time window $win = [t_{start}, t_{end}]$; node q will save n corresponding historical trading feedback evaluation records. The following is the formula of calculating recommendation degree in the traditional model:

$$RTrust_q = \frac{\sum_{k=1}^{n} {\lambda_k}^{t-t'} \times ep_{kq} \times Cr_k \times DTrust_{kq}}{n} \tag{13}$$

Among them, the time attenuation factor is $\lambda \in [0, 1]$. The parameter t is the current time and t' is the refresh time of the trust value after direct transactions among nodes. Cr_k is the recommendation degree of node k. ep_{kq} is the empirical value of node k relative to q, which can be computed as follows:

$$ep_{kq} = \frac{1}{\pi} \times [\arctan\left(\frac{count_{kq}}{M}\right) + \arctan(\frac{amount_{kq}}{N})] \tag{14}$$

where M and N are positive numbers set by the actual conditions. The parameters $count_{kq}$ and $amount_{kq}$ are the trading times and total turnover, respectively, between node k and node q in a certain time period.

In the above recommendation trust computing model, if the recommendation information has not been updated for a long time, its influence on the node's trust degree calculation will diminish. This is why we introduce the time attenuation factor. Considering the total turnover, this prevents the nodes in the network from accumulating experience values too quickly by using a small amount with high faith transactions. The experience factor is also introduced in the model, because the experience of a node is richer, the recommended information it gives is closer to the reality, which is more trustworthy.

The reliability of the recommended node equals the recommendation degree, which is calculated as follows:

$$T(q, w) = \sum_{i=1}^{I(q)} s(q, i) \times \frac{sim(E(q, i), w)}{\sum_{j=1}^{I(p)} sim(E(q, j), w)} \tag{15}$$

$$sim(v,w) = 1 - \sqrt{\frac{\sum_{x \in IS(v,w)} \left(\frac{\sum_{i=1}^{I(x,v)} s(x,i)}{I(x,v)} - \frac{\sum_{i=1}^{I(x,w)} s(x,i)}{I(x,w)} \right)^2}{|IS(v,w)|}} \tag{16}$$

Among them, $sim(v,w)$ is the recommend similarity of node v and w, which is the similarity of their feedbacks to other nodes' trust conditions. $T(q,w)$ represents the trust value of node q to node w. The total amount of transactions between node q and node v in a time window is denoted as $I(q,v)$. The parameter $IS(v,w)$ is a set of nodes that have performed transactions with both node v and w. The nodes with high recommended credibility can be given a high weight for their recommendation information. By filtering the nodes whose feedback information has a large deviation from the trust evaluating standard, our model can also resist conspiracy attacks of malicious nodes.

The traditional models of calculating recommendation degree only use the trust degree value to calculate the evaluated node's trust conditions without taking the uncertainty of trust into consideration. From what has been introduced above, we know that the cloud model can reflect the trust condition of the evaluated nodes more comprehensively by using three digital characters. Integrating Ex, which expresses the node's average trust value with the entropy En and the hyper entropy He, reflects the uncertainty of trust effectively.

We then introduce how to use the trust cloud to calculate the trust conditions of evaluated node q. Let set $TC_q(Ex,\ En,\ He)$ to be the trust cloud of the evaluated node q. We collect feedback records about qıs trading conditions from other nodes in the time window $win = [t_{start}, t_{end}]$. The node that has traded with node q for i times is then denoted as $E(q,i)$. The parameter $Cr(w)$ represents the recommending reliability of node w and $S(q,i)$ is the feedback rating of node $E(q,i)$ to node p for the ith transaction. Taking the records' ratings as the sample values, the node's recommending reliability is represented as the weights of sample numbers. The total number of samples is quite large. We then use the samples as an input and calculate the node trust cloud based on the following Algorithm 3.

Algorithm 3:

Input: service node p, evaluated node q, total number of samples N
Output: the evaluated node's trust cloud $TC_q(Ex,\ En,\ He)$.

(1) $0 \rightarrow SumCr(q)$, which sets parameters as zero.

(2) Retrieve $Feedback(q, win) \rightarrow Feedback$, which collects the rating record of node q in the time window win.

(3) For $1 \rightarrow I$ to $Length(Feedback)$, we perform the following steps: Feedback source of $Feedback(i) \rightarrow E(q,i)$ which collects the nodes that give the rating records; Retrieve Feedback by $(E(q,i), win)$, which collects the rating records of node $E(q,i)$ to other trading nodes; Compute similarity $(q, E(q,i))$ by calculating the recommending reliability of node $E(q,i)$; Calculate $SumCr(q) + Cr(E(q,i)) \rightarrow SumCr(q)$.

(4) For $i \rightarrow 1$ to $Length(Feedback)$, perform $\frac{Cr(E(q,i)) \times N}{SumCr(q)} \rightarrow num(i)$, which takes the $num(i)$ rating records value of $Sum[q,i]$ as the sample input of the one-dimensional reverse cloud generator.

(5) Output $TC_q(Ex,\ En,\ He)$.

In the calculation of the above node's trust cloud, the value of total number of samples N should be as large as possible. It first ensures that the value $\frac{Cr(E(q,i)) \times N}{SumCr(q)}$ is positive in the algorithm, and then it can also reduce the errors that are generated by the reverse cloud generator algorithms. To get the trust cloud's expectations by this algorithm, we use the following formula:

$$Ex = \overline{X} = \frac{\sum_{i=1}^{Length(Feedback)} S(q,i) \times \left| \frac{Cr(E(q,i)) \times N}{SumCr(q)} \right|}{N} \tag{17}$$

The characteristic parameter Ex of the trust cloud basically covers the trust information of the evaluated nodes. The uncertainty of trust is described by the other two characteristic parameters, entropy and hyper entropy. It is essential to build a more robust, accurate trust model to explore the node's real trust behavior by fully mining the trust information contained in the three characteristic parameters of trust cloud.

When the total number of samples N is certain, $En^2 + He^2$ decides the discrete degree of trust. Therefore, we introduce the discrete factor $\lambda = \sqrt{En^2 + He^2}$ to measure and reflect the level of uncertainty of trust. We can use the three characteristic trust cloud parameters to distinguish a malicious recommended node from a goodwill recommended node. The analysis process is then as follows:

(1) The goodwill recommended node using characteristic parameter expectation value to measure the trust value of evaluated node is relatively accurate and the volatility of node's behavior is low, so the discrete factor is small.
(2) The behavior of nodes that sometimes provide malicious information, and sometimes provide goodwill information is unstable with high volatility, so the discrete factor is large.
(3) The nodes that frequently provide malicious information, whose behavior and uncertainty factor is stable, will certainly generate a cloud expect value that becomes pretty low.

3.5. Trading Node Selection Strategy Based on Trust Cloud

Through a large amount of factual analysis, we know that the discrete factor of goodwill nodes is relatively stable. This factor is usually not greater than a normal number, which is called the threshold of goodwill node discrete factor, written as Δ. The value of Δ can be derived from a large amount of experimental statistical analysis by using the discrete factors of goodwill and malicious nodes. Its value should not be too big as the upper bound of the goodwill node's uncertain factor. We generally get the value according to the actual situation.

To select one as the trading object, we should comprehensively consider the node's recent behavior fluctuation and average trust condition in the P2P network. How to determine the node's selection sequence when there are multiple trading nodes with the same expectation value is difficult. From the period theatricals analysis of the cloud model, we know that when the discrete factor of the trust cloud $En + He$ is smaller, the expectation value is higher. The following gives the algorithm for selecting the trading nodes.

Assume that the destination node is u, then there are two conditions for u to be selected as the trading nodes: $T(w) \geq \theta$ and $\lambda \leq \Delta$. The value of θ is determined by the judgment of the service request node and the trading risk size. So, the Algorithm 4 is shown as:

Algorithm 4:

(1) Through the filter entity set p, get a subject $P' = \{T(u) \geq \theta \; and \; \lambda \leq \Delta\}$.
(2) If the set P' is non-empty, choose the node according to the reputation value from high to low and turn to the next step. If the set P' is empty, turn to step (4).
(3) In the process of choosing nodes in accordance with the order, if there are two or more nodes that have same reputation value, compare their $\lambda's$ values, preferring the nodes with a smaller λ, and turn to step (2).
(4) Choose the node whose reputation value is the highest as the trading object in set p.

The above node selection strategy is different from the traditional ones, which generally consider the magnitude of the trust value based on a reputation mechanism by selecting the trading nodes with a high trust degree according to the reputation value from high to low in the trading nodes set. Instead, we design the trading node selection strategy by considering the node's trust expected value as well as

introducing another decision condition, namely the discrete factor, which can reflect the stability of trust behavior more accurately.

Because the node's behavior in set P' has a high stability selecting the nodes with high trust expectation values from this set greatly increases the probability of successful trading. The discrete factor also solves the problem of how to choose the nodes when they have the same trust degrees.

4. Experiment Analysis

4.1. Experiment Set

To evaluate the proposed trust-aware propagation mechanisms, a simulator is developed on the Ecological Network Computing Environment (ENCE) platform [42], which has been applied to simulations of innovative network applications in P2P, grid and Web service environments [43,44]. On the ENCE platform the software interfaces, the common structural module, and the emulator are designed using Java language, and support "plug and play" and flexible API operations. All the data was collected from a file sharing system in the P2P network.

Figure 2 shows the structure of node's reputation trust evaluation index. Based on the data from fifteen experts, we get the index weight for each indicator, as shown in Table 2. The fifteen experts are the expert nodes which we selected in advance from the network, and they meet the requirements of full trust. Through the comprehensive calculation, we get the original data in Table 2.

Table 2. Indicator weight.

Indicator	Weight
Historical reputation U1	0.36
Network speed U2	0.30
Trading hours U3	0.23
Hardware reliability U4	0.11

4.2. The Calculation of Reputation Trust Degree

Assuming that relevant experts participate in a node's research evaluation, we calculate the three characteristic parameters through the reverse cloud generator by using the evaluation vector for each indicator 15 times, as shown in Table 3. According to the complexity of our experiment, 15 iterations could obtain optimum results.

Table 3. Experts' evaluation data on each indicator.

Evaluation Index	Evaluation Vector	$T(Ex, En, He)$
U1	$V1(7, 3, 5, 0, 0)$	$T1(0.7893, 0.2028, 0.0683)$
U2	$V2(8, 3, 3, 1, 0)$	$T1(0.8201, 0.3149, 0.0920)$
U3	$V3(7, 2, 3, 2, 1)$	$T1(0.7025, 0.3121, 0.1034)$
U4	$V4(5, 4, 3, 2, 1)$	$T1(0.6762, 0.3049, 0.0930)$

We then synthesize the trust evaluation cloud of the four indexes generated from Table 3 using the Algorithm 1, and finally get the evaluated node's trust evaluation cloud $T(0.9969, 0.2503, 0.0809)$. We generate the comparison chart of the trust basic cloud and the trust evaluation cloud after getting the trust evaluation cloud, and then calculate the node's trust degree represented by the cloud evaluation cloud by combing the Algorithm 2.

Assume that the trust evaluation cloud of node A, B, C and D in the file sharing system are $T_A(0.8, 0.1, 0.01)$, $T_B(0.8, 0.4, 0.01)$, $T_C(0.4, 0.1, 0.01)$, $T_D(0.4, 0.4, 0.01)$, respectively and the base cloud of trust is $T_S(0.6, 0.2, 0.01)$. What follows in Figures 4–7 is a comparison of the trust evaluation

clouds and the base clouds of trust. The abscissa represents the trust cloud's expectations, and the ordinate represents the entropy in Figures 4–7.

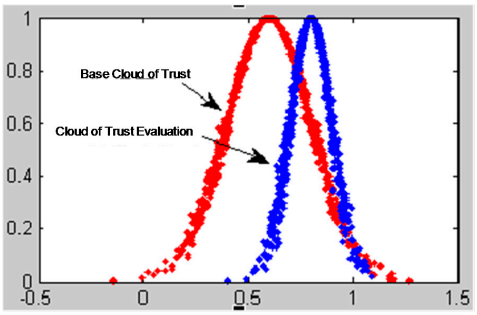

Figure 4. Comparison between T_A and T_S. (The abscissa represents the trust cloud's expectations, and the ordinate represents the entropy.)

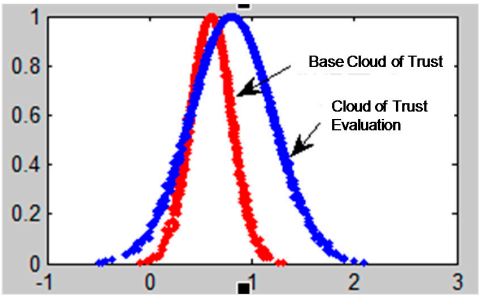

Figure 5. Comparison between T_B and T_S.

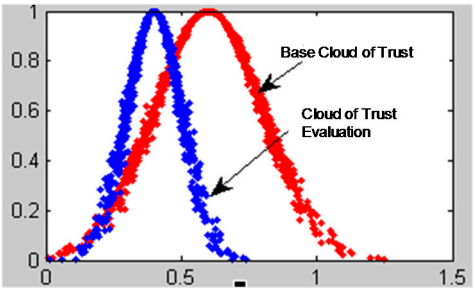

Figure 6. Comparison between T_C and T_S.

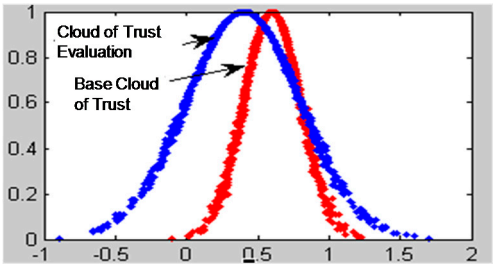

Figure 7. Comparison between T_D and T_S.

The question is which one of the above four trust evaluation clouds is more reliable? In Figure 4, the expectation of trust cloud T_A is greater than the base cloud of trust and the entropy of the base cloud of trust is greater than $T_A's$, which indicates that the trust evaluation cloud meets the requirements, belonging to ideal state. From Figure 7, we can see that the trust value of trust cloud T_D is smaller than

the expectation value of the base cloud of trust and the entropy of the base cloud of trust is smaller than $T_D's$, which indicates that the trust cloud does not meet the trust requirements. In Figure 5, the expectation value of trust cloud T_B is greater than the base cloud of trusts, and its entropy is also greater than the base cloud of trusts. In this case, we should consider comparing the value of $Ex - 3En$. Consequently, we obtain a value of $Ex - 3En$ and the base cloud of trust that is greater than the evaluation cloud $T_B's$ through calculation. Confronted with such a complicated situation, the conclusion is not obvious from the comparison of parameters. According to the description in Section 3, we use the two-cloud similarity algorithm to judge, in which we obtain a similarity value of $\delta_j = 0.7239$. In Figure 6, the expectation of the trust cloud T_C is smaller than the base cloud of trust and so is its entropy. We should also consider the value of $Ex - 3En$ with a determined similarity value of $\delta_j = 0.5865$.

4.3. Calculation of Direct Trading Trust Degree

Three factors should be considered while calculating the direct trading trust degree: attenuation factor, acceleration factor and volume factor. Table 4 is the successful trading records between node p and node q in the P2P network. In our experiment, we set $\lambda = 0.5$, $A(i) = 0.5$, and $M = 100$. These values are the result of several experiments.

Table 4. Direct trading records.

$I(p,q)$ between p and q	$S_{pq}(i)$ of p to q	Trading Time $t - t'$	Turnover Amount	$c(i) = amount(i)/(amount(i) + M)$	True or False
1	0.75	1	2400	0.9600	T
2	1	2	1800	0.9474	T
3	0.75	2	1200	0.9230	T
4	0.75	1	900	0.9000	T
5	0.5	0.5	1500	0.9375	T
6	0.5	0.25	700	0.8750	T
7	1	2	1100	0.9167	T
8	0.25	1	1500	0.9375	T
9	0.25	1.5	1000	0.9090	T
10	0.75	1	2000	0.9524	T

We can calculate the value of the volume factor $C(i)$ using the turnover amount and the trading time $t - t'$ in Table 4. Let the number of failed transactions be $n = 2$, we can then calculate the direct trust degree of node p to q using Equation (12) and we obtain a value of $DTrust_{pq} = 0.8512$.

4.4. The Calculation of Recommendation Degree

To calculate the trust cloud of node q, we first collect the rating records of the other nodes that have transactions with node q in a time window $win = [t_{start}, t_{end}]$, as shown in Table 5.

Table 5. Score records of recommended nodes.

Recommended Nodes $E(q,i)$	The Score to Node q $s(q,i)$	Credibility $Cr(E(q,i))$ of Recommended Node $E(q,i)$	$Ex = \overline{X}$
a	0.75	0.9	0.8726
a	1		
a	0.75		
b	0.75	0.8	
b	0.75		
b	0.5		
c	0.5	0.5	
c	0.5		
c	0.5		
c	0.25		

We first collect the feedback score records of the nodes that have transactions with node q in a month and use Equations (11)–(16) to calculate the recommendation reliability of the node $E(q,i)$. Then we calculate the node q's trust cloud by using the Algorithm 3. Taking the value of trust rating $s(q,i)$ as the sample value and the weights of sample numbers as the node's recommendation reliability $Cr(E(q,i))$, assuming $N = 100$, we generate the trust cloud $TC_q(Ex, En, He)$ of node q and obtain $RTrust = 0.8726$ after inputting the data to the one-dimension reverse cloud generator.

Below we use the expectations and discrete factors of the trust cloud to distinguish the goodwill recommended nodes from malicious ones. For the evaluation, we choose three representative nodes: first one is a goodwill recommended node *good-Entity*, whose service is reliable and whose expectation discrete factors are respectively recorded as Ex_1, λ_1; the second one is the unstable recommended node with 0.1 small probability *mal-Entity1*, whose expectation and discrete factor are respectively recorded as Ex_2, λ_2; The third one is a malicious recommended node *mal-Entity2* which always provides false information and its expectation and discrete factor are recorded as Ex_3, λ_3. We then collect the score records of the nodes, which have transactions with the three nodes in two months and use Equation (16) to calculate the recommend reliability of the three nodes. We also calculate the three nodes' trust evaluation cloud by using the trust cloud algorithm and the one-dimensional normal reverse cloud generator, denoted as $TC_1(Ex_1,\ En_1,\ He_1)$, $TC_2(Ex_2,\ En_2,\ He_2)$, $TC_3(Ex_3,\ En_3,\ He_3)$, where $\lambda_1 = \sqrt{(Ex_1)^2 + (En_1)^2}$, $\lambda_2 = \sqrt{(Ex_2)^2 + (En_2)^2}$, $\lambda_3 = \sqrt{(Ex_3)^2 + (En_3)^2}$.

The result of the calculation is presented in Table 6. We obviously notice that the trust cloud expectation of goodwill node *good-Entity* is high and its discrete factor is small, which indicates that it has been providing good service information. The unstable node *mal-Entity1* trust cloud expectation is high as well, because the service it provided is alternatively good and bad, while the value of its discrete factor is large. The malicious recommend node *mal-Entity2* always provides false information, so its discrete factor is small and the trust cloud's expectation is the lowest. It is also quite easy to distinguish various types of nodes through the trust cloud, especially those malicious nodes that intend to achieve high reputations by using unfair tactics.

Table 6. Parameters of the node trust cloud.

Node	**Good-Entity**		**Mal-Entity1**		**Mal-Entity2**	
Trading Circles	Ex_1	λ_1	Ex_2	λ_2	Ex_3	λ_3
20	0.8875	0.1244	0.7250	0.3025	0.1125	0.1474
40	0.8250	0.1146	0.6000	0.3825	0.0500	0.1000
60	0.9250	0.1146	0.7375	0.3305	0.1375	0.1474
80	0.8500	0.1225	0.7250	0.3527	0.1250	0.1250
100	0.8375	0.1193	0.8125	0.2607	0.1500	0.1225
120	0.8500	0.1225	0.6250	0.3212	0.1250	0.1250
140	0.9000	0.1225	0.7250	0.3527	0.1000	0.1225

4.5. Calculation of Trading Node Selection Strategy Based on Trust Cloud

In this section, we compare the Algorithm 4 based on the trust cloud with the one solely based on degree of trust. We allow nodes to sometimes provide good service and sometimes provide malicious service varying from 30 times to 70 times, which can effectively verify the advantage of the trading node selection strategy based on cloud theory. Figure 8 shows the influence curve of the trading node selection strategy proposed in this paper and the traditional strategy to the node's successful trading rate.

From Figure 8, we know that the traditional algorithm has a greater influence on a node's successful trading rate when the recommended node shows unstable behavior. Our proposed node selection strategy is based on trust cloud and has a smaller influence on the trading success rate. This indicates that the proposed and novel algorithm can effectively identify the nodes with unstable behavior and simultaneously improve the node's trading success rate.

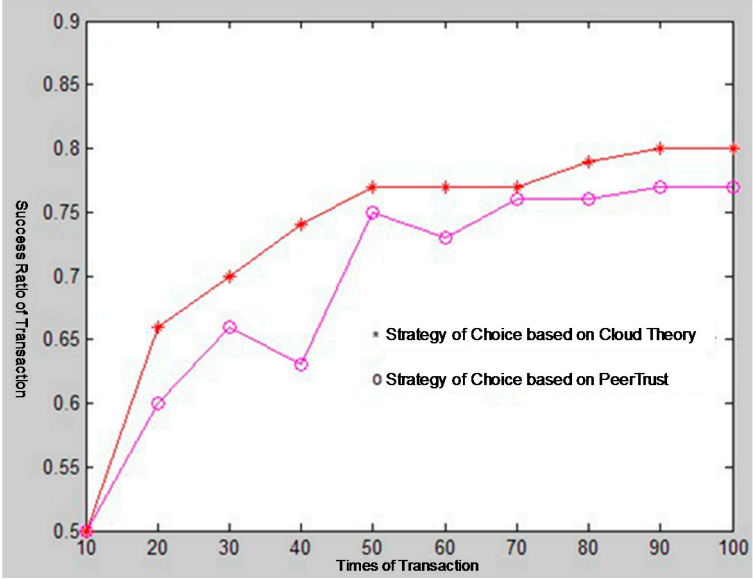

Figure 8. Comparison of two different selection strategies for a node's trading success ratio.

5. Summary and Prospects

5.1. Summary

Social network security is of high relevance and interest in the research community, and has been one of the most important factors impacting the application of social technology. A variety of different trust models have been developed and thus greatly promoted the development and application of social networks. However, trust in social networks is strongly subjective, which also means that it is difficult to measure accurately. Most of the current trust evaluation models in social networks fail to take full account of trust's inherent attributes such as the subjectivity and uncertainty. This disadvantage prevents us from making comprehensive and accurate judgments on a node's trust conditions. In response to this problem, in this paper we introduce a cloud model for trust evaluation in P2P network systems.

We have designed a two-dimensional trust cloud evaluation model. Our model is based on a cloud model and integrates the structure and characteristics of a social network. In addition, our model considers both nodes' reputation trust and historical transaction trust in order to ensure that the assessments of the nodes' trust conditions are more comprehensive and accurate. Our experiments demonstrate that our model effectively solves the problem of trust's uncertainty by leveraging three characteristic parameters of cloud models that better reflect the overall trust conditions of nodes.

We also presented a method for calculating a reputation trust value based on a trust cloud using an index and the corresponding index weight. In the trust decision mechanism, the node's average trust condition is described by an expected value Ex, and the uncertainty of trust is expressed by two other parameters, En and He. This method offers more advantages in the retention of uncertainty inherent attributes of trust compared to the method of using a single datapoint to represent a degree of trust.

When computing a node's trading degree of trust, we propose a calculation method based on a trust cloud, which can identify those nodes with unstable trading behaviors by using a discrete factor. Our approach effectively prevents and suppresses dishonest or malicious nodes.

By combining the trust expectation value and the discrete factor in the trust cloud model as the deciding conditions, we discuss our trading node selection strategy, which takes into full consideration a node's stability and trust conditions and solves the puzzle of how to choose between nodes when they have the same trust values.

We applied the trust evaluation model constructed above in a typical P2P network file sharing system and conducted a sample analysis. The result reflects the objectivity and accuracy of our model and proves the basis for selecting reliable trading nodes in the file sharing system.

5.2. Research Prospects

Not much research has been performed on trust evaluation in the area of cloud models. Considering the factors impacting trust evaluation, we applied the cloud model to social networks for the evaluation of trust degree. However, our model is not very mature. We are interested in further studying and exploring the following few aspects of this research domain:

In social networks, recommending nodes usually need to get trust evaluation information from other nodes by using layers of recommendation trust generated from a trust chain. The attenuation of trust information and synthesis calculation in the process of dissemination in the trust chain needs to be further investigated with the use of a cloud model.

Furthermore, a punishment mechanism and an incentive mechanism can be introduced into the proposed trust evaluation model proposed in this paper, which would generally punish the malicious nodes and reward the ones with high credibility.

Finally, we believe that the comprehensive application of combining a cloud model and other security technology is also of interest in future studies. For example, we can use a Bayesian feedback trust cloud model for updating the trust in social networks.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (No. 61170038, 61472231), the National Social Science Foundation of China (No. 14BTQ049), and a project of International Cooperation in Training of Excellent Backbone Teachers for Advanced University in Shandong Province.

Author Contributions: Fengming Liu conceived and designed the experiments; Lehua Ren performed the experiments; Yuxi Hu analyzed the data and contributed reagents/materials/analysis tools; Fengming Liu and Xiaoqian Zhu wrote the paper and Henric Johnson revised the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, B.; Jajodia, S.; Kankanhalli, M.S. Building trust in peer-to-peer systems: A review. *Int. J. Secur. Netw.* **2006**, *1*, 103–112. [CrossRef]
2. Hughes, D.; Coulson, G.; Walkerdine, J. Free riding on Gnutella revisited: The bell tolls. *IEEE Comput. Soc.* **2005**, *6*, 258–265. [CrossRef]
3. Tian, C.; Yang, B.; Zhong, J.; Liu, X. Trust-based incentive mechanism to motivate cooperation in hybrid P2P networks. *Comput. Netw.* **2014**, *73*, 244–255. [CrossRef]
4. Sánchez-Artigas, M.; Herrera, B. Understanding the effects of P2P dynamics on trust bootstrapping. *Inf. Sci.* **2013**, *236*, 33–55. [CrossRef]
5. Zhao, Y.; Jiang, C. Research of trust model in P2P file-sharing system. *Procedia Environ. Sci.* **2012**, *12 Pt B*, 1208–1212.
6. Blaze, M.; Feigenbaum, J.; Lacy, J. Decentralized trust management. In Proceedings of the Symposium on Security and Privacy, Oakland, CA, USA, 6–8 May 1996; pp. 164–173.
7. Schilke, O.; Cook, K.S. Sources of alliance partner trustworthiness: Integrating calculative and relational perspectives. *Strateg. Manag. J.* **2015**, *36*, 276–297. [CrossRef]
8. Johnson, H.; Lavesson, N.; Zhao, H.; Wu, S.F. On the Concept of Trust in Online Social Networks. In *Trustworthy Internet, Part 3*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 143–157.
9. Rose, A.M.; Rose, J.M.; Norman, C.S. Material Control Weakness Corrections: The Enduring Effects of Trust in Management. *Behav. Res. Account.* **2016**, *28*, 41–53. [CrossRef]
10. Nejdl, W.; Olmedilla, D.; Windlett, M. PeerTrust: Automated trust negotiation for peers on the semantic web. In Proceedings of the Workshop on Secure Data management in a Connected World, Toronto, ON, Canada, 30 August 2004; pp. 118–132.

11. Erlandsson, F.; Johnson, H.; Boldt, M. Privacy Threats Related to User Profiling in Online Social Networks. In Proceedings of the Third International Workshop on Security and Privacy in Social Networks 2012 (SPSN-2012), Canterburry, UK, 3–5 September 2012.
12. Chen, Z.; Ge, L.; Wang, H.; Huang, X.; Lin, J. A trust-based service evaluation and selection model in pervasive computing environment. *J. Softw.* **2006**, *17*, 200–210.
13. Zhang, X.; Nie, R. A self-government model peer-to-peer trust management. *J. Beijing Univ. Technol.* **2008**, *34*, 211–215.
14. Tian, L.; Lin, C. Evaluation mechanism for user behavior trust based on DSW. *J. Tsinghua Univ. (Sci. Technol.)* **2010**, *50*, 763–767.
15. Wang, L.; Guo, Y.; Zhan, Y. Fuzzy trust model for wireless sensor networks with intrusion tolerance. *J. Commun.* **2010**, *31*, 37–44.
16. Liu, L.; Zhou, D.; Xie, X.; Li, J. Services trust evaluation model based on cloud computing. *Softw. Guide* **2011**, *10*, 75–77.
17. Fan, L.; Wang, S.; Liu, W. Evaluation method based on human trust mechanism for mobile e-commerce trust. *Comput. Sci.* **2012**, *39*, 190–193.
18. Beth, T.; Borcherding, M.; Klein, B. Valuation of trust in open networks. In Proceedings of the European Symposium on Research in Security, Brighton, UK, 7–9 November 1994; Springer: Berlin, Germany, 1994; pp. 3–18.
19. Jøsang, A.; Knapskog, S.J. *A Metric for Trusted Systems*; Global IT Security, Wien, Austrian Computer Society: Wien, Austria, 1998; pp. 541–549.
20. Jøsang, A. Trust-based decision making for electronic transactions. In Proceedings of the 4th Nordic Workshop on Secure Computer Systems, Kista, Sweden, 1–2 November 1999; pp. 99–105.
21. Esposito, C.; Ficco, M.; Palmieri, F.; Castiglione, A. Smart Cloud Storage Service Selection Based on Fuzzy Logic, Theory of Evidence and Game Theory. *IEEE Trans. Comput.* **2016**, *65*, 2348–2362. [CrossRef]
22. Ullah, Z.; Khan, M.S.; Ahmed, I.; Javaid, N.; Khan, M.I. Fuzzy-Based Trust Model for Detection of Selfish Nodes in MANETs. *Adv. Inf. Netw. Appl.* **2016**, 965–972. [CrossRef]
23. Hao, F.; Min, G.; Lin, M.; Luo, C.; Yang, L.T. MobiFuzzyTrust: An Efficient Fuzzy Trust Inference Mechanism in Mobile Social Networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2944–2955. [CrossRef]
24. Li, D. Uncertainty in knowledge representation. *Eng. Sci.* **2000**, *2*, 73–79.
25. Liu, C.; Li, D.; Du, Y.; Han, X. Some statistical analysis of the normal cloud model. *Inf. Control* **2005**, *34*, 235–248.
26. Li, D.; Liu, C. Study on universality of the normal cloud model. *Eng. Sci.* **2004**, *6*, 28–34.
27. Liu, C.; Li, D.; Du, Y.; Han, X. Artificial intelligence with uncertainty. *J. Softw.* **2004**, *15*, 1583–1594.
28. Zhao, W.; Li, D. Intrusion detection using cloud model. *Comput. Eng. Appl.* **2003**, *29*, 158–164.
29. Di, K.; Li, D.; Li, D. Cloud theory and its applications in spatial data mining and knowledge discovery. *J. Image Graph.* **1999**, *4*, 930–935.
30. He, R.; Niu, J.; Zhang, G. *CBTM: A Trust Model with Uncertainty Quantification and Reasoning for Pervasive Computing*; LNCS 3758; Springer: Berlin, Germany, 2005; pp. 541–552.
31. Marsh, S. *Formalising Trust as a Computational Concept*; University of Stirling: Stirling, UK, 1994; pp. 22–45.
32. Jairak, R.; Praneetpolgrang, P.; Chirawichitchai, N. A Roadmap for Establishing Trust Management Strategy in E-Commerce Services Using Quality Based Assessment. *Int. J. Inf. Eng. Electron. Bus.* **2016**, *6*, 1–9. [CrossRef]
33. Abdul-Rahman, A.; Hails, S. Supporting trust in virtual communities. In Proceedings of the Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; pp. 6007–6015.
34. Das, A.; Islam, M.M. SecuredTrust: A Dynamic Trust Computation Model for Secured Communication in Multiagent Systems. *IEEE Trans. Depend. Secur. Comput.* **2012**, *9*, 261–274. [CrossRef]
35. Chatzopoulos, D.; Ahmadi, M.; Kosta, S.; Hui, P. Have you asked your neighbors? A Hidden Market approach for device-to-device offloading. In Proceedings of the 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, 21–24 June 2016; pp. 1–9.
36. Chatzopoulos, D.; Ahmadi, M.; Kosta, S.; Hui, P. OPENRP: A reputation middleware for opportunistic crowd computing. *IEEE Commun. Mag.* **2016**, *45*, 115–121. [CrossRef]
37. Wang, Y.; Vassileva, J. Bayesian network-based trust model. In Proceedings of the IEEE International Conference on Web Intelligence, Halifax, NS, Canada, 13–17 October 2003; pp. 372–378.

38. Yuan, W.; Li, J.; Hong, P. Distributed peer-to-peer trust model and computer simulation. *J. Syst. Simul.* **2006**, *18*, 938–942.
39. Tang, W.; Hu, J.; Chen, Z. Research on a fuzzy logic-based subjective trust management model. *Comput. Res. Dev.* **2005**, *42*, 1654–1659. [CrossRef]
40. Wang, S.; Zhang, L.; Li, H. Evaluation approach of subjective trust based on cloud model. *J. Softw.* **2010**, *21*, 1341–1352. [CrossRef]
41. He, R.; Niu, J.; Yuan, M.; Hu, J. A novel cloud-based trust model for pervasive computing. In Proceedings of the Fourth International Conference on Computer and Information Technology, Wuhan, China, 14–16 September 2004; pp. 693–700.
42. Gao, L.; Ding, Y.-S.; Ren, L.-H. A novel ecological network-based computation platform as grid middleware system. *Int. J. Intell. Syst.* **2004**, *19*, 859–884. [CrossRef]
43. Ding, Y.-S.; Gao, L.; Ruan, D. Communication mechanisms in ecological network-based grid middleware for service emergence. *Inf. Sci.* **2007**, *177*, 722–733. [CrossRef]
44. Gao, L.; Ding, Y.-S.; Ying, H. An adaptive social network-inspired approach to resource discovery for the complex grid systems. *Int. J. Gen. Syst.* **2006**, *35*, 347–360. [CrossRef]