Automatic Trust Calculation for QoS-based Web Service System

Bo Ye*, Maziar Nekovee†, Anjum Pervez* and Mohammad Ghavami*

*ESBE, London South Bank University, London SE1 0AA, UK, {yeb, perveza, ghavamim}@lsbu.ac.uk

†Samsung Electronics R&D Institute, South Street, Staines, Middlesex TW18 4QE, UK, m.nekovee@samsung.com

***Abstract*—To choose the best service, how much a service can be trusted is increasingly important for service consumers. In addition, manually assigned feedback costs much time and suffers several drawbacks. Automatic trust calculation is the only feasible method for large-scale service-oriented applications. Hence, we propose an automatic trust calculation using non-trust quality criterion values. To make the calculation accurate, we employ the Kalman Filter to filter out malicious non-trust values instead of directly filtering out malicious trust values. Furthermore, to offer higher detection accuracy we propose an improved algorithm based on the joint probability by taking the relationship between the non-trust criterion value and its variance into account. Although malicious data can be filtered out, dishonest or inaccurate values can still influence trust values. Therefore, similarity between consumers is used to weight the values from others, which is calculated by mostly used Euclidean distance function. However, we modify a collection of distance functions to calculate similarity. Finally, experiments are carried out to access the validation and robustness of our model. The results show that our improved algorithm can offer higher detection accuracy under several malicious situations, and we also discovered another similarity function performed better than the Euclidean function.**

I. Introduction

The rapid growth of service-oriented applications has spurred a considerable amount of research in this field. However, service providers (SP) are usually little known by service consumers (SC). Among various SPs providing identical or similar services with varying Quality of Service (QoS), it is hard for SCs to select appropriate services. To help SCs make decisions, QoS of service-oriented systems has been modelled [1]–[4] and various selection algorithms have been proposed to optimise the results.

Several centralised systems [1]–[4] modelled QoS properties, and then based on consumer requirements and the values of QoS published by SPs, various algorithms were employed to select services. However, the systems [1]–[4] did not focus on how the values of QoS properties were collected. If the values are given by SPs, they may not be fully true, due to dishonesty of certain SPs. Wang *et al.* [5] collected feedback from SCs to evaluate SPs. Similarly SCs may also provide malicious feedback. Hence, among QoS criteria, trust is increasingly important for SCs to select SPs. Although several trust-based systems [6]–[11] have been proposed, they required humans to provide feedback ratings used as trust values. Overall, the existing approaches have the following weaknesses:

1) First of all, the approaches [6]–[11] measure service reputation based on the assumption that the feedback is provided by humans. Manually assigning feedback costs much time and has several disadvantages. For instance, certain humans are not willing to provide feedback and may provide unfair one. It is difficult to ensure the accuracy due to different abilities and knowledge of humans. According to the review of trust techniques in service workflows and relevant contexts [12], Viriyasitavat *et al.* also discovered that lacking of a unified way to formalise trust prevented automation in trust-related processes from realisation. Hence, it is necessary to build an automatic trust measure system without humans providing feedback.
2) Secondly, the authors of [9]–[11] employed only one distance function, the Euclidean function, to calculate the similarity between SCs, and ignored other methods.
3) Except [11], the approaches [1]–[10] treat trust as only one QoS criterion of a service, meaning that one service has only one value representing its overall reputation.
4) Lastly, the systems [1]–[4] did not filter malicious values out. Malicious SCs might provide malicious values to falsely improve the trust in certain SPs, or to degrade the trust in certain providers for commercial benefits.

To address the weaknesses above, an approach to measure the trust both in SPs and SCs has been proposed. This approach groups service quality criteria, and then measures the trust in each quality criterion of a service based on their characteristics. The measure of trust in SPs has been divided into two stages, including Time and Aggregation Domain, because of two reasons. Firstly, a SC may invoke a service many times, so that it can measure the trust in the service based on its own data at Time Domain. Because all data is obtained by itself, it is unnecessary to filter out any information. Secondly, it may compute the trust in a service by using other SCs' data. At this stage, the SC not only needs to aggregate all data, but also has to filter out malicious data. This stage is called Aggregation Domain.

Compared to the existing approaches, our main contributions have been summarised as follows:

1) QoS Criteria have been grouped into several classes based on their characteristics, and trust calculation has

been divided into two domains. An approach has been proposed to compute the trust values automatically.

2) At Aggregation Domain, when a SC aggregates the data from others, the trust in a SC X is employed to weight the data from X. The trust in X is calculated by distance functions, and a poor distance function may lead to bad aggregation results. Hence, a comparative study of different distance functions is carried out to find the most suitable function for trust aggregation.
3) The value of trust in each QoS criterion is calculated.
4) Different from the approaches that directly filters out malicious trust values, our model filter out malicious values of non-trust QoS criteria, because it is harder for SCs to manipulate multiple values than one trust value, and this model can be extended so that each SC can use its own trust-based algorithm to calculate the trust. In addition, the trust value calculated directly from the raw data is more accurate than aggregating trust values from others, because the inaccuracy of trust values from others will accumulate in the trust aggregation.

The rest of this paper is organised as follows. Section II gives a brief review of related work. Section III demonstrates how quality criteria are classified, and how the values of trust in quality criteria and in SCs are calculated automatically. Section IV presents how malicious values are detected and how different values from a variety of consumers are aggregated. Section V describes the processes of automatic trust calculation. The model is evaluated by carrying out different experiments in Section VI. The final Section contains the conclusion and some ideas for further work.

II. Related Work

To help SCs select proper services, QoS of web services has been modelled [1]–[4] and various selection algorithms have been proposed to optimise the results. The systems [1], [2] did not aggregate the values of trust, while in the approaches [3], [4] trust is just one of the QoS criteria considered. Wang *et al.* [5] collected feedback from SCs to evaluate SPs. The approaches of [1]–[4] and [5] used data provided by SPs and SCs respectively. Both SPs and SCs may provide malicious data, and none of them [1]–[5] employed detection algorithm to filter out malicious data. Thereby if malicious SCs exist, they may not select proper services.

Several trust-based systems [6]–[11] have been proposed. Although Conner *et al.* [6] provided high flexibility for SCs to use a variety of scoring functions over the same data for personalised reputation evaluation, this approach was based on the assumption that SCs do not mask their malicious behaviour, meaning that it is hard to detect those malicious SCs that behave well until they gain good trust values and then behave maliciously. X.F. Wang [7] represented trust using two attributes, trust value and trust estimate variance, and employed the Kalman Filter (KF) to filter out malicious trust values and to aggregate feedback. Their experiments showed that the model provided higher robustness to estimate trust values with a lower false detection rate.

TABLE I
Comparison of Trust Calculation Models

Authors	Automatic	Similarity	Direct Trust	Indirect Trust	Detection Algorithm
Conner *et al.* [6]			✓		✓
X.F. Wang [7]			✓	✓	✓
P. Wang *et al.* [8]			✓	✓	
S. Wang *et al.* [9]		✓	✓	✓	✓
Das *et al.* [10]		✓	✓	✓	✓
Yan *et al.* [11]		✓	✓	✓	✓
Our model	✓	✓	✓	✓	✓

S. Wang *et al.* [9] measured the reputation by an approach containing three steps, including feedback checking, feedback adjustment and malicious feedback detection. They collected a user survey form to check the feedback ratings from the users who are lacking in feedback ability, and then adjusted the ratings by calculating the similarity between the feedback. Finally, the method of cumulative sum was adopted to detect malicious feedback. Yan *et al.* [11] presented a user-centric trust model to select services for SCs with regard to SCs' preferences. In the final section, they pointed out that their model can be improved by automating personal service selection. Based on the direct experience and indirect recommendation of services, P. Wang *et al.* [8] presented a trust-based QoS model by employing Dempster-Shafer evidence reasoning theory and belief model. Das *et al.* [10] proposed a dynamic trust calculation model named SecuredTrust. In SecuredTrust, the different factors related to measure the trust were analysed. Euclidean distance function was employed by [9]–[11] to calculate similarity between SCs, and the similarity was used as the weight to aggregate the data from others. However, many more distance functions in [24] are not considered.

Thereby, trust is increasingly crucial, and the comparison of the methods in [6]–[11] is made in Table I. It is obvious that none of them covered all the parameters. Hence, based on trust, the QoS properties are modelled to address the existing approaches' drawbacks and to cover a wider range of parameters, which the existing models have not considered.

Much work has been done in other fields and several detection algorithms have been proposed including Bayesian-based [20]–[22], KF-based algorithms [7], etc. In [20], [21], a Bayesian reputation approach was proposed to calculate the trust value based on the beta probability density functions. In [21], intuitive parameters needed to be tuned manually without guarantee of any quantitative confidence. In [22], the trust was modelled as a three-dimension belief (b, d, u), which represented the positive, negative and uncertain probabilities. Although the trust in [20]–[22] was modelled as predicted probability values, prediction variance was ignored by them, which was considered in [7]. X.F. Wang [7] proposed a general trust model for a more robust reputation evaluation by employing KF [25], and their experimental results showed that KF-based algorithm outperformed others. Hence, KF-based algorithm is adopted in this paper. In addition, the value of some QoS property, such as response time, cannot be measured exactly, because there is network delay and a service-oriented system is complex and dynamic. KF-based algorithm is also employed to produce estimates of the response time, because KF can use a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produce estimates of unknown variables that tend to be more precise.

III. Quality Criterion

In this section, some concepts related to trust are introduced first. Based on these concepts, this section presents how the value of the trust in each individual criterion and other SCs are calculated.

Definition 1. ***Quality Criterion:*** *This encompasses a number of QoS properties used to evaluate a web service, including Price (PR), Response Time (RT), Availability (AY), Success Rate (SR) and Trust (T).*

There are two ways to categorise Quality Criteria. Firstly, based on the way how it affects the overall QoS of a service, Quality Criterion can be classified as either **Positive Criterion**, whose increase benefits the overall QoS, or **Negative Criterion** whose decrease benefits the overall QoS. Secondly, on the basis of the nature of a criterion, criteria then fall into three major classes:

1) **Ratio Criterion:** The value of a criterion can be presented as a ratio, which can be directly used as the trust of the criterion, such as availability, success rate etc. Please note that Ratio Criteria are not the criteria whose values obtained from SPs are rate. For example, compensation rate, whose values gotten from SPs are rate, is not a ratio criterion.
2) **Non-ratio Criterion with Accurate Observation:** A criterion's value cannot be presented as a ratio, and its exact values can be obtained by consumers, eg. price.
3) **Non-ratio Criterion without Accurate Observation:** A criterion's value cannot be presented as a ratio, and its values cannot be gotten accurately, because of observation and systematic error, eg. response time.

For a criterion of a service, a SC can get its value in two different ways. The SC can obtain the value of a criterion published by the SP, or it can get the value by invoking the service. Hence there are two major classes of criterion values:

1) **Published Value of a Criterion (PV):** A criterion's value is published by a SP when a service is published. This can be updated by SP at any time.
2) **Actual Value of a Criterion (AV):** The value of a criterion is collected by SCs after invoking a service, which may be different from PV. For instance, a SP may publish $40ms$ as a service's response time, but the actual response time may be $43ms$ when a consumer invokes the service.

In this paper, trust is denoted by T. One consumer A has the value of the trust in another one B, meaning that A knows how much he can trust B. Trust can be classified as either criterion or reference trust, on the basis of trust purpose. For an illustrative purpose, a criterion C of a service S provided by a provider P is denoted by $P.S.C$.

1) **Criterion Trust:** A's trust in $P.S.C$, denoted by $T(A \to P.S.C)$. It identifies how much the service's criterion $P.S.C$ can be trusted.
2) **Reference Trust:** A's trust in another consumer B's capacity of referring to SPs' ability to do something, defined by $T(A \to B)$. Please note that a SP can also have a reference trust, because a SP can also be a SC, recommending another SP.

Based on the ways how it affects the overall trust, a trust can be divided into **Positive Part**, which increases the trust, and **Negative Part**, which decreases the trust.

Similarly, the actual value of a criterion can also be classified as either **Positive Actual Value**, which increases the trust of the criterion, or **Negative Actual Value**, which decreases the criterion's trust.

A. Criterion Trust Calculation

After defining relevant concepts, how the trust value is derived from values of non-trust quality criteria is presented. To begin with, a few notations are introduced.

- $T(A \to P.S.C)_j$: The value of A's trust in $P.S.C$ after j^{th} time A invokes a service S;
- c represents $P.S.C$'s PV;
- c_j is the actual value obtained by A after j^{th} time invoking service S.

Assume that C is a negative criterion, and then $T(A \to P.S.C)_j$ is calculated by the following equations.

Number of positive C values,

$$\mathrm{num}_j^{po} = \begin{cases} \mathrm{num}_{j-1}^{po} + 1 & c_j \leq c \\ \mathrm{num}_{j-1}^{po} & c_j > c \end{cases} \tag{1}$$

Number of negative C values,

$$\mathrm{num}_j^{ne} = \begin{cases} \mathrm{num}_{j-1}^{ne} & c_j \leq c \\ \mathrm{num}_{j-1}^{ne} + 1 & c_j > c \end{cases} \tag{2}$$

The value of positive part of $T(A \to P.S.C)_j$ is calculated by,

$$T_j^{po} = \begin{cases} \sqrt{\frac{\mathrm{num}_{j-1}^{po}(T_{j-1}^{po})^2 + (1-\frac{c_j}{c})^2}{\mathrm{num}_j^{po}}} & c_j \leq c \\ T_{j-1}^{po} & c_j > c \end{cases} \tag{3}$$

The value of negative part of $T(A \to P.S.C)_j$,

$$T_j^{ne} = \begin{cases} T_{j-1}^{ne} & c_j \leq c \\ \sqrt{\frac{\mathrm{num}_{j-1}^{ne}(T_{j-1}^{ne})^2 + (1-\frac{c_j}{c})^2}{\mathrm{num}_j^{ne}}} & c_j > c \end{cases} \tag{4}$$

At last, the value of A's trust in $P.S.C$ is computed as follows:

$$T(A \to P.S.C)_j = 1 + T_{j-1}^{po} - \frac{\mathrm{num}_j^{ne}}{\mathrm{num}_j^{ne} + \mathrm{num}_j^{po}} \cdot T_j^{ne} \tag{5}$$

Please note that the values which are equal to PV are always classified as positive ones.

B. Reference Trust Calculation

To aggregate data from others, a SC needs to know how much he can trust them. In this thesis similarity between two SCs is used as a consumer A's reference trust in another one B, because A can trust B more, if values of the trust maintained by A are more similar to B's. Using the value of trust in B, A can know how much he can trust the services referred by B. Before introducing the method of calculating reference trust, several notations are explained as follows.

- P_A: The set of SPs whose certain services have been invoked by A before, meaning A have data on some services provided by P_A.
- $p_i.S_A$ $(p_i \in P_A)$: The set of p_i's services which have been invoked by A, and it is a subset of all p_i's services.
- $P_{A\cap B}$=$P_A \cap P_B$: The set of SPs whose certain services have been invoked by both A and B.
- $p_x.S_{A\cap B}$=$p_x.S_A \cap p_x.S_B$ $(p_x \in P_{A\cap B})$: The set of p_x's services which have been invoked by both A and B, and it is a subset of all p_x's services.
- $\overrightarrow{T_A}$ and $\overrightarrow{T_B}$ represent the trust matrix maintained by A and B respectively.

To simplify equations, T_A and T_B are short for $T(A \rightarrow p.s.c)$ and $T(B \rightarrow p.s.c)$ respectively in this section. Distance functions in [24] are modified to calculate the similarity between A and B, and they are explained as follows. In following equations, $p \in P_{A\cap B}$, $s \in p.S_A \cap p.S_B$ and $c \in s.C$.

$$S_1 = 1 - \left(\frac{\sum_p \sum_s \sum_c (T_A - T_B)^p}{|T_A|} \right)^{\frac{1}{p}} \tag{6}$$

Eq. (6) is the most widely used. When $p = 2$, this equation is Euclidean distance function used in [9], [11].

$$S_2 = 1 - \sqrt{(E(\overrightarrow{T_A}) - E(\overrightarrow{T_B}))^T G^{-1} (E(\overrightarrow{T_A}) - E(\overrightarrow{T_B}))/10} \tag{7}$$

where G represents the pooled covariance matrix calculated with $\overrightarrow{T_A}$ and $\overrightarrow{T_B}$.

$$S_3 = 1 - \max(|\overrightarrow{T_A} - \overrightarrow{T_B}|) \tag{8}$$

$$S_4 = 1 - \sum_p \sum_s \sum_c \sqrt{\frac{(T_A - T_B)^2}{\mathrm{var}(\overrightarrow{T_A})}}/100 \tag{9}$$

where $var(\overrightarrow{T_A})$ means the variance of $\overrightarrow{T_A}$.

$$S_5 = 1 - \sum_p \sum_s \sum_c \frac{(T_A - T_B)^p}{1 + \min\{|T_A|, |T_B|\}} \tag{10}$$

$$S_6 = 1 - \sum_p \sum_s \sum_c \frac{(T_A - T_B)^p}{\max\{|T_A|, |T_B|\}} \tag{11}$$

$$S_7 = 1 - \sum_p \sum_s \sum_c \frac{|T_A - T_B|}{\max\{|T_A|, |T_B|\}} \tag{12}$$

$$S_8 = 1 - \sum_p \sum_s \sum_c \frac{|T_A - T_B|}{1 + \min\{|T_A|, |T_B|\}} \tag{13}$$

$$S_9 = 1 - \sum_p \sum_s \sum_c \frac{|T_A - T_B|}{1 + \max\{|T_A|, |T_B|\}} \tag{14}$$

$$S_{10} = 1 - \sum_p \sum_s \sum_c \frac{|T_A - T_B|}{1 + |T_A| + |T_B|} \tag{15}$$

$$S_{11} = 1 - \sum_p \sum_s \sum_c \left| \frac{T_A}{1 + |T_A|} - \frac{T_B}{1 + |T_B|} \right| \tag{16}$$

It is obvious that the more similar their values are, the more A can trust B.

C. Trust Transitivity

Definition 2. ***Transitive Trust:*** *Assume that there are two SCs A and B, A has a reference trust in B, and B has a criterion trust in $P.S.C$. A needs to know how much he can trust in $P.S.C$, and it has no information about it. A's trust in $P.S.C$ can be derived by aggregating B's criterion trust in $P.S.C$ and A's trust in B as follows:*

$$T(A \rightarrow P.S.C) = T(A \rightarrow B) \cdot T(B \rightarrow P.S.C)$$

It is common to collect reference trusts from several SCs to make better decisions, which can be called consensus trust.

Definition 3. ***Consensus Trust:*** *The consensus trust of two SCs' trust in criterion C of service S provided by a provider P is a trust that reflects trust in a fair and equal way. A consumer A needs to obtain the value of the trust in $P.S.C$, and it knows little about $P.S.C$. However, it has trust in other SCs X and Y, and both of them have a trust in $P.S.C$. The derived consensus trust in $P.S.C$ can be defined by:*

$$T(A \rightarrow S.C) = \frac{|T(A \rightarrow X) \cdot T(X \rightarrow P.S.C)| + |T(A \rightarrow Y) \cdot T(Y \rightarrow P.S.C)|}{|T(A \rightarrow X)| + |T(A \rightarrow Y)|}$$

IV. Criterion Value Estimate and Improved Detection Algorithm

Malicious SC can be classified as either adulating SC, which tries to falsely improve the trust in certain SPs, or defaming SC, trying to degrade the trust in certain SPs. The authors in [7] used this algorithm to filter out malicious values of the trust, however we use it to filter out malicious values of non-trust quality criteria to retain the accuracy of the trust. In addition, based on the algorithm we not only estimate the value of non-trust quality criteria, but also predict its variance. We further improve this algorithm by taking the relationship between the value of non-trust quality criteria and its variance into account.

A. Criterion Value Estimate

Because the value of a criterion C of a service S provided by a provider P obtained by a consumer A each time is independent, it is reasonable to model the distribution of the values of $P.S.C$ as Normal distribution. For each criterion, its values follow normal distribution with $\{\mu^r, \sigma^r\}$, where μ^r is the real value of $P.S.C$'s μ, and σ^r is the actual $P.S.C$'s variance.

Assume that A is going to use estimated values from other consumers to predict $P.S.C$'s $\{\mu^r, \sigma^r\}$. SC i's estimated values of $P.S.C$'s $\{\mu^r, \sigma^r\}$ are denoted as $\{\mu_i^e, \sigma_i^e\}$. After aggregating i's estimated values, A's estimated values are denoted as $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$. Because of incomplete knowledge of $P.S.C$, i's estimated values usually have a deviation from A's estimated values $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$. Because the estimated values are from independent consumers, the relation between i's estimate and A's estimate is modelled as follows:

$$\begin{aligned}\mu_i^e &= \mu_{A,i}^e + \lambda_\mu \text{ and } p(\lambda_\mu) \sim \text{Normal}(0, \Lambda_\mu)\\ \sigma_i^e &= \sigma_{A,i}^e + \lambda_\sigma \text{ and } p(\lambda_\sigma) \sim \text{Normal}(0, \Lambda_\sigma)\end{aligned}$$

Note that λ_μ is different from σ_i^e. λ_μ is an estimate noise covariance when A estimates the real value μ^r, while σ_i^e is estimated covariance from SC i, which may be malicious. Similarly, λ_σ is an estimate noise covariance when A estimates the real value σ^r.

Based on the Kalman Filter [25], the estimation of $\{\mu^r, \sigma^r\}$ is governed by the linear stochastic difference equations:

$$\begin{aligned}\mu_{A,i}^e &= F_\mu \mu_{A,i-1}^e + Bu_{i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu)\\ \sigma_{A,i}^e &= F_\sigma \sigma_{A,i-1}^e + Bu_{i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma)\end{aligned}$$

where, F is the factor for relationship between the previous estimate based on the estimate of SC $i-1$ and the current estimate based on i's estimate, and u is the optional control input to the estimate $\{\mu_A^e, \sigma_A^e\}$. Because in our model there is no control input, u is 0. Hence, our estimates are governed by the following linear difference equations:

$$\begin{aligned}\mu_{A,i}^e &= F_\mu \mu_{A,i-1}^e + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu)\\ \sigma_{A,i}^e &= F_\sigma \sigma_{A,i-1}^e + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma)\end{aligned}$$

In the Kalman Filter, there are two steps: $Predict$ step and $Update$ step. P_μ and P_σ represent predict error covariance of $\mu_{A,i}^e$ and $\sigma_{A,i}^e$ respectively. The $Predict$ step is responsible for obtaining the priori estimate, denoted by$\{\bar{\mu}_{A,i}^e, \bar{\sigma}_{A,i}^e\}$, for the next step based on the previous estimate $\{\mu_{A,i-1}^e, \sigma_{A,i-1}^e\}$. Similarly, priori predict error covariances are denoted by $\bar{P}_\mu$ and $\bar{P}_\sigma$. The $Update$ step is responsible for incorporating a new SC's estimate $\{\mu_i^e, \sigma_i^e\}$ to obtain an improved posteriori estimate $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$.

$Predict$ step:

$$\begin{aligned}\bar{\mu}_{A,i}^e = F_{\mu,i}\mu_{A,i-1}^e, \quad \bar{\sigma}_{A,i}^e = F_{\sigma,i}\sigma_{A,i-1}^e\\ \bar{P}_{\mu,i} = F_{\mu,i}^2 P_{\mu,i-1} + W_{\mu,i}, \quad \bar{P}_{\sigma,i} = F_{\sigma,i}^2 P_{\sigma,i-1} + W_{\sigma,i}\end{aligned} \tag{17}$$

$Update$ step:

$$K_{\mu,i} = \frac{\bar{P}_{\mu,i}}{\bar{P}_{\mu,i} + \Lambda_{\mu,i}}, \quad K_{\sigma,i} = \frac{\bar{P}_{\sigma,i}}{\bar{P}_{\sigma,i} + \Lambda_{\sigma,i}} \tag{18}$$

$$\begin{aligned}\mu_{A,i}^e &= \bar{P}_{\mu,i} + K_{\mu,i}(\mu_i^e - \bar{\mu}_{A,i}^e),\\ \sigma_{A,i}^e &= \bar{P}_{\sigma,i} + K_{\sigma,i}(\sigma_i^e - \bar{\sigma}_{A,i}^e)\end{aligned} \tag{19}$$

$$P_{\mu,i} = (1 - K_{\mu,i})\bar{P}_{\mu,i}, \quad P_{\sigma,i} = (1 - K_{\sigma,i})\bar{P}_{\sigma,i} \tag{20}$$

To compute the parameters $F_{\mu,i}$, $\Lambda_{\mu,i}$, $W_{\mu,i}$, $F_{\sigma,i}$, $\Lambda_{\sigma,i}$, $W_{\sigma,i}$, the following equations are used:

$$F_{\mu,i} = \frac{\sum_{m=1}^{i-1} \mu_{A,m}^e \mu_{A,m-1}^e}{\sum_{m=1}^{i-1} (\mu_{A,m}^e)^2}, \quad F_{\sigma,i} = \frac{\sum_{m=1}^{i-1} \sigma_{A,m}^e \sigma_{A,m-1}^e}{\sum_{m=1}^{i-1} (\sigma_{A,m}^e)^2} \tag{21}$$

$$\Lambda_{\mu,i} = \frac{1}{i}\sum_{m=1}^{i-1}(\mu_m^e - \mu_{A,m}^e)^2, \quad \Lambda_{\sigma,i} = \frac{1}{i}\sum_{m=1}^{i-1}(\sigma_m^e - \sigma_{A,m}^e)^2 \tag{22}$$

$$\begin{aligned}W_{\mu,i} &= \frac{1}{i}\sum_{m=1}^{i-1}(\mu_{A,m}^e - F_i \mu_{A,m-1}^e)^2,\\ W_{\sigma,i} &= \frac{1}{i}\sum_{m=1}^{i-1}(\sigma_{A,m}^e - F_i \sigma_{A,m-1}^e)^2\end{aligned} \tag{23}$$

B. Improved Detection Algorithm

Given significance probability levels δ_μ and δ_σ, the problem of determining if the SC i is not malicious is to find the threshold values $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ so that:

$$P(|\mu_i^e - \mu_{A,i}^e| \le \Delta_{\mu,i}) = \delta_\mu, P(|\sigma_i^e - \sigma_{A,i}^e| \le \Delta_{\sigma,i}) = \delta_\sigma \tag{24}$$

In addition, $\mu_i^e - \mu_{A,i}^e$ and $\sigma_i^e - \sigma_{A,i}^e$ follow zero mean normal distribution with variance $P_{\mu,i} + \Lambda_{\mu,i}$ and $P_{\sigma,i} + \Lambda_{\sigma,i}$ respectively. Hence, there are also equations:

$$\begin{aligned}P(|\mu_i^e - \mu_{A,i}^e| \le \Delta_{\mu,i}) = 1 - 2\Phi(\frac{-\Delta_{\mu,i}}{\sqrt{P_{\mu,i} + \Lambda_{\mu,i}}}),\\ P(|\sigma_i^e - \sigma_{A,i}^e| \le \Delta_{\sigma,i}) = 1 - 2\Phi(\frac{-\Delta_{\sigma,i}}{\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}}})\end{aligned} \tag{25}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Hence, after solving Eq. (24) and (25), $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ can be obtained:

$$\begin{aligned}\Delta_{\mu,i} = -\Phi^{-1}(\frac{1-\delta_\mu}{2})\sqrt{P_{\mu,i} + \Lambda_{\mu,i}},\\ \Delta_{\sigma,i} = -\Phi^{-1}(\frac{1-\delta_\sigma}{2})\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}}\end{aligned} \tag{26}$$

Using the threshold values $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$, malicious values of μ^e and σ^e can be detected respectively. However, a malicious consumer can still manipulate the model by setting σ^e or μ^e to be the lower or upper limit. Although a malicious consumer i can set its feedback $\{\mu_i^e, \sigma_i^e\}$ to be upper or lower limit, the probability of such feedback may be very low or even zero. Hence, to increase the detection accuracy, joint probability of σ^e and μ^e is used to filter out this type of malicious values. Differences between $\{\mu_i^e, \sigma_i^e\}$ and $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$ are denoted by $d_{\mu,i}$ and $d_{\sigma,i}$, which follow zero mean normal distribution with variance $P_{\mu,i} + \Lambda_{\mu,i}$ and $P_{\sigma,i} + \Lambda_{\sigma,i}$ respectively. The joint probability of $d_{\mu,i}$ and $d_{\sigma,i}$ can be calculated by:

$$\begin{aligned}P_{d_\mu,d_\sigma}(d_{\mu,i}, d_{\sigma,i}) &= P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i})P_{d_\mu}(d_{\mu,i})\\ &= P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i})P_{d_\sigma}(d_{\sigma,i})\end{aligned} \tag{27}$$

where $P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i})$ and $P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i})$ give the conditional distributions of d_σ given $d_\mu = d_{\mu,i}$ and of d_μ given

$d_\sigma = d_{\sigma,i}$ respectively, and $P_{d_\mu}(d_{\mu,i})$ and $P_{d_\sigma}(d_{\sigma,i})$ give the distributions for d_μ and d_σ respectively.

In Eq. (27), $P_{d_\mu}(d_{\mu,i})$ and $P_{d_\sigma}(d_{\sigma,i})$ can be calculated by the probability density function of normal distribution. Therefore, the problem is to determine $P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i})$ or $P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i})$. In this paper, they are calculated by using a set of clean data $\{\overrightarrow{d_\mu}, \overrightarrow{d_\sigma}\}$ which can be collected from those highly trusted consumers. Before introducing the method of calculating $P_{d_\mu}(d_{\mu,i})$ and $P_{d_\sigma}(d_{\sigma,i})$, basic notations are provided as follows:

- $\{d_{\mu,min}, d_{\mu,max}\}$, $\{d_{\sigma,min}, d_{\sigma,max}\}$: denote the minimum and maximum values of $\overrightarrow{d_\mu}$ and $\overrightarrow{d_\sigma}$ respectively;
- $N_{inv,\mu}$, $N_{inv,\sigma}$: The intervals between minimum and maximum values of $\overrightarrow{d_\mu}$ and $\overrightarrow{d_\sigma}$ are divided into $N_{inv,\mu}$ and $N_{inv,\sigma}$ smaller intervals respectively;
- N_d: The number of the elements of the set $\{\overrightarrow{d_\mu}, \overrightarrow{d_\sigma}\}$;
- $N_{[d_{\mu,k-1}, d_{\mu,k}]}$: The number of elements in $\{\overrightarrow{d_\mu}, \overrightarrow{d_\sigma}\}$ whose d_μ is in the k^{th} interval $[d_{\mu,k-1}, d_{\mu,k}]$;
- $N_{[d_{\sigma,k-1}, d_{\sigma,k}]}$: The number of elements whose d_σ is in $[d_{\sigma,k-1}, d_{\sigma,k}]$;
- $N_{[d_{\mu,k-1}, d_{\mu,k}],[d_{\sigma,k-1}, d_{\sigma,k}]}$: The number of elements whose d_μ is in the k^{th} interval $[d_{\mu,k-1}, d_{\mu,k}]$ and d_σ is in $[d_{\sigma,k-1}, d_{\sigma,k}]$.

The first step of this method is to divide the intervals $[d_{\mu,min}, d_{\mu,max}]$ and $[d_{\sigma,min}, d_{\sigma,max}]$ into $N_{inv,\mu}$ and $N_{inv,\sigma}$ smaller intervals respectively. In the following step, assume $d_{\mu,i}$ and $d_{\sigma,i}$ are in the a^{th} interval $[d_{\mu,a-1}, d_{\mu,a}]$ and b^{th} interval $[d_{\sigma,b-1}, d_{\sigma,b}]$ respectively, and then $P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i})$ and $P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i})$ are calculated by the following equations:

$$P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i}) = \frac{N_{[d_{\mu,a-1}, d_{\mu,a}],[d_{\sigma,b-1}, d_{\sigma,b}]}}{N_{[d_{\sigma,b-1}, d_{\sigma,b}]}} \tag{28}$$

$$P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i}) = \frac{N_{[d_{\mu,a-1}, d_{\mu,a}],[d_{\sigma,b-1}, d_{\sigma,b}]}}{N_{[d_{\mu,a-1}, d_{\mu,a}]}} \tag{29}$$

Due to calculation error, the results of $P_{d_\sigma|d_\mu}(d_{\sigma,i}|d_{\mu,i})P_{d_\mu}(d_{\mu,i})$ and $P_{d_\mu|d_\sigma}(d_{\mu,i}|d_{\sigma,i})P_{d_\sigma}(d_{\sigma,i})$ in Eq. (27) may be different. Hence, two significant probability level δ_μ^{joint} and δ_σ^{joint} are given. At last, for each pair $\{\mu_i^e, \sigma_i^e\}$, if the joint probabilities of its diversion from estimation values $\{\mu_{A,i}^e, \sigma_{A,i}^e\}$ calculated by Eq. (27) are greater than δ_μ^{joint} and δ_σ^{joint} respectively, they are classified as non-malicious values.

V. Processes of Trust Calculation

A SC can calculate the trust value in a SP in two different ways. Firstly, it can invoke a service many times, and obtain the direct experience, called at Time Domain. Secondly, it can collect trust values in a service from others, and get the indirect experience, called at Aggregation Domain.

A. Time Domain

Each time a SC invokes a service, it can obtain the values of all criteria. The consumer can use these values to calculate the trustworthiness of the service. Assume that it is i^{th} time that a consumer A invokes a service S provided by a provider P. The non-trust criteria of S include Price (PR), Response Time (RT), Availability (AY) and Success Rate (SR). In the four criteria, AY and SR are ratio criteria; P is a non-ratio criterion with exact observation; RT is a non-ratio criterion without exact observation. This section describes how estimate values $\{\mu_i^e, \sigma_i^e\}$ of each criterion can be obtained by using A's collected criterion values.

1) Estimate values of ratio criteria: Because at time domain all values are collected by the SC itself, it is unnecessary to filter out malicious values. It is easy to calculate ratio criteria's value. For instance, the success rate c_{sr} of S can be calculated by the following equations:

If the i^{th} invocation of S is successful:

$$\mathrm{num}_{\mathrm{success},i} = \mathrm{num}_{\mathrm{success},i-1} + 1$$

If the i^{th} invocation of S fails:

$$\mathrm{num}_{\mathrm{success},i} = \mathrm{num}_{\mathrm{success},i-1}$$

Finally:

$$c_{sr} = \frac{\mathrm{num}_{\mathrm{success},i}}{i}$$

Because the success rate is accurate, $\mu_i^{e,SR}$ is directly assigned with c_{sr} and there is no variance value $\sigma_i^{e,SR}$ for success rate.

2) Estimate values of non-ratio criteria with exact observation: Assume that pr_i is the value obtained after the i^{th} invocation of S by A, and then PR's values $\{\mu_i^{e,PR}, \sigma_i^{e,PR}\}$ can also be calculated by the equations as follows:

$$\mu_i^{e,PR} = \frac{(i-1)\mu_{i-1}^{e,PR} + \mathrm{pr}_i}{i}$$
$$\sigma_i^{e,PR} = \frac{(i-1)\sigma_{i-1}^{e,PR} + (\mathrm{pr}_i - \mu_i^{e,PR})^2}{i}$$

3) Estimate values of non-ratio criteria without exact observation: Exact values of RT cannot be obtained, not only because computer is a complex dynamic system, but also because of network delay. It is impossible to get exact response time of a service. Hence, the method of criterion value estimation in Section IV is used to estimate the value $\{\mu_i^e, \sigma_i^e\}$. Assume C^{RT} is the vector of RT values obtained by A, and c_i^{RT} is the RT value gotten by A after j^{th} time's invocation of the service S. $\{U^{e,RT}, V^{e,RT}\}$ is the vector of RT values estimated by A, and $\{\mu_i^{e,RT}, \sigma_i^{e,RT}\}$ are the estimated values after getting the value c_i^{RT}.

Because variance of RT values cannot be gotten from SPs, it cannot be estimated directly. Under this situation, the real RT value $\mu_i^{e,RT}$ is estimated using the Kalman Filter, and $\sigma_i^{e,RT}$ is computed indirectly in the process of estimating $\mu_i^{e,RT}$. First, Eq. (21), (22) and (23) are used to compute the parameters F, Λ, W. They are calculated as follows.

$$F_{\mu,i} = \sum_{m=1}^{i-1} \mu_m^{e,RT} \mu_{m-1}^{e,RT} / \sum_{m=1}^{i-1} (\mu_m^{e,RT})^2;$$

$$\Lambda_{\mu,i} = \frac{1}{i} \sum_{m=1}^{i-1} (c_m - \mu_m^{e,RT})^2;$$

$$W_{\mu,i} = \frac{1}{i} \sum_{m=1}^{i-1} (\mu_m^{e,RT} - F_i \mu_{m-1}^{e,RT})^2$$

Similarly, Eq. (17), (18), (19) and (20) are employed, and $\mu_i^{e,RT}$ are predicted as follows sequentially.

$$\bar{\mu}_i^{e,RT} = F_{\mu,i} \mu_{i-1}^{e,RT};\ \bar{P}_{\mu,i} = F_{\mu,i}^2 P_{\mu,i-1} + W_{\mu,i};$$
$$K_{\mu,i} = \bar{P}_{\mu,i} / (\bar{P}_{\mu,i} + \Lambda_{\mu,i});\ P_{\mu,i} = (1 - K_{\mu,i}) \bar{P}_{\mu,i};$$
$$\mu_i^{e,RT} = \bar{P}_{\mu,i} + K_{\mu,i} (c_i^{e,RT} - \bar{\mu}_i^{e,RT})$$

Finally, $\sigma_i^{e,RT}$ is assigned with $P_{\mu,i} + \Lambda_{\mu,i}$.

B. Aggregation Domain

Because of incomplete knowledge on SPs, each SC can collect data from other consumers, and aggregate these values to obtain more accurate values, although certain values may be malicious.

Assume a consumer A is going to calculate the trust in $P_x.S_x$, and has no data on this service. Hence, it collects data from its neighbour consumers who have used $P_x.S_x$. The set of A's neighbours is denoted by CS, which is obviously a subset of its all neighbours. The values of the four criteria SR, AY, PR and RT from a SC $cs_i \in CS$ are denoted by $\{\mu_i^{e,SR}\}$, $\{\mu_i^{e,AY}\}$, $\{\mu_i^{e,PR}, \sigma_i^{e,PR}\}$ and $\{\mu_i^{e,RT}, \sigma_i^{e,RT}\}$ respectively. The process of calculating trust is described as follows.

1) The data is filtered first by using the method introduced in Section IV, malicious values are filtered out.
2) A trust matrix T maintained by a consumer is a three dimensional one, $P \times S \times C$. Eq. (6) is used as an example for illustrating how A's trust in a consumer $cs_i \in CS$ can be computed. The following equation is used to calculate the similarity between A and cs_i, which is A's trust in cs_i.

$$1 - \left(\frac{\sum_p \sum_s \sum_c (T(A \to p.s.c) - T(B \to p.s.c)^p}{|T(A \to p.s.c)|} \right)^{\frac{1}{p}}$$

where $p \in P_A \cap P_{cs_i}$, $s \in p.S_A \cap p.S_{cs_i}$ and $c \in s.C$.
3) Based on the transitive property of trust introduced in Section III-C, A's trust in $P_x.S_x$'s criteria is calculated similar to the approach presented III-A. The following equations explain how A's trust in $P_x.S_x.PR$ is calculated. PR_{pb} denotes the published value of $P_x.S_x.PR$. The set of A's neighbours whose $P_x.S_x.PR$ actual values are positive and the set of neighbours whose values are negative are denoted by CS^{po} and CS^{ne} respectively.

$$\mathrm{Tsum}^{po} = \sum_{\mathrm{cs}_i \in \mathrm{CS}^{po}} T(A \to \mathrm{cs}_i)$$

$$\mathrm{Tsum}^{ne} = \sum_{\mathrm{cs}_i \in \mathrm{CS}^{ne}} T(A \to \mathrm{cs}_i)$$

$$T^{po} = \sqrt{\sum_{\mathrm{cs}_i \in \mathrm{CS}^{po}} \frac{T(A \to \mathrm{cs}_i)}{\mathrm{Tsum}^{po}} (1 - \frac{\mu_{\mathrm{cs}_i}^{e,PR}}{PR_{\mathrm{pb}}})^2}$$

$$T^{ne} = \sqrt{\sum_{\mathrm{cs}_i \in \mathrm{CS}^{ne}} \frac{T(A \to \mathrm{cs}_i)}{\mathrm{Tsum}^{ne}} (1 - \frac{\mu_{\mathrm{cs}_i}^{e,PR}}{PR_{\mathrm{pb}}})^2}$$

$$T = 1 + T^{po} - \frac{\mathrm{Tsum}^{ne}}{\mathrm{Tsum}^{po} + \mathrm{Tsum}^{ne}} T^{ne}$$

VI. Performance Evaluation

In this section, several experiments are carried out to evaluate the robustness of the proposed model, compared to the algorithm (RLM) in [7]. This trust model presented in this paper, Improved RLM (IRLM), is evaluated in a simulated environment. Because the experimental results in [7] have shown that KF-based algorithm outperforms Bayesian-based algorithms, and so on, IRLM is not compared with those approaches again. At last, all similarity functions are evaluated and compared.

To evaluate the robustness of IRLM, an experiment is carried out in an environment with malicious values. The probability of malicious values is set up to 40%, because it is almost impossible that more than a half of SCs behave maliciously and it is impossible to perform well in an environment with more than 50% malicious SCs.

Definition 4. ***True Malicious Rate:*** *The percentage of correctly detected malicious values.*

The number of all malicious values and correctly detected malicious ones are denoted by num_m and num_c respectively, and true malicious rate is calculated by $\frac{\mathrm{num}_c}{\mathrm{num}_m}$.

Definition 5. ***False Malicious Rate:*** *The percentage of wrongly detected non-malicious values.*

The number of all non-malicious and wrongly detected malicious values are denoted by num_{non} and num_w respectively, and false malicious rate is calculated by $\frac{\mathrm{num}_w}{\mathrm{num}_{\mathrm{non}}}$.

This experiment is similar to the experiment in [7], and the variance σ^e of malicious values is set to be an extreme low value. All results are shown in Fig. 1 and 2, and IRLM performs almost the same as the original RLM algorithm in this situation, as the increase of the malicious value probability. Because IRLM is based on RLM, it will not perform worse than RLM. IRLM aims at another situation which is not considered by RLM. Hence, in this experiment, IRLM does not outperform RLM, either. Results in Fig. 1 and 2 further proved these two points.

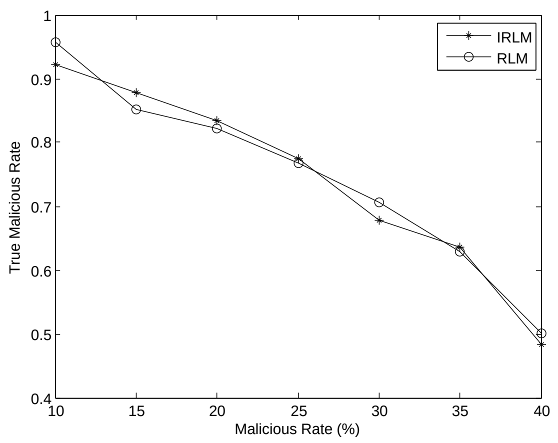

Fig. 1. Average true malicious rate with the variance being set to be an extreme low value

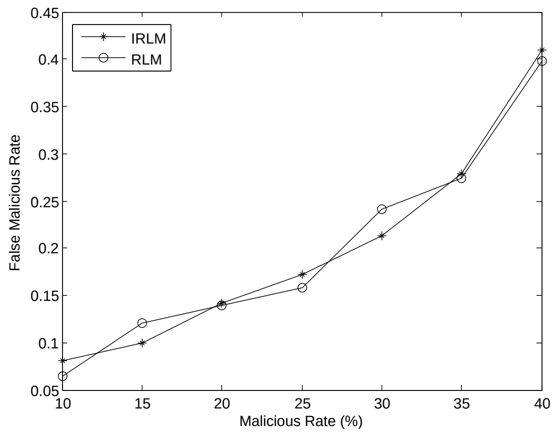

Fig. 2. Average false malicious rate with the variance being set to be an extreme low value

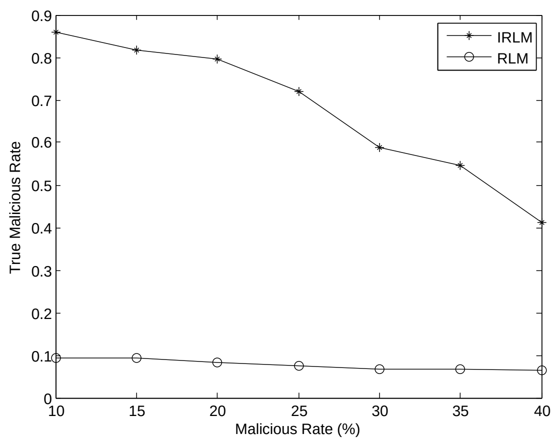

Fig. 3. Average true malicious rate with three types of malicious values

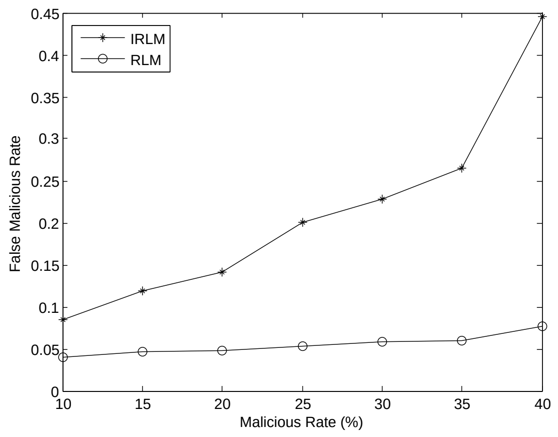

Fig. 4. Average false malicious rate with three types of malicious values

Furthermore, in another experiment with malicious values, three different types of malicious values are listed as follows:

- σ^e is set to be the real value and μ^e is set to be the lower or upper limit;
- σ^e is set to be the lower or upper limit and μ^e is set to be real value;
- Both σ^e and μ^e are set to be the lower or upper limit.

The values of σ^e in IRLM correspond to P in two dimension tuple, $\{\langle R\rangle, P\}$, of RLM. RLM used manual feedback and the values of P were also set manually, therefore, there is no relationship between R and P in RLM. In IRLM both σ^e and μ^e are derived from the collected values, and σ^e and μ^e are related to each other. However, consumers can still manipulate the values and provide inaccurate values to other consumers. In this situation, malicious consumers may try to mislead the results gradually. Hence, seen from Fig. 3, IRLM still has high detection efficiency in this strategic malicious environment while RLM does not. However, as shown in Fig. 4, IRLM suffers higher false detection rate than RLM.

Before evaluating how distance functions affect the aggregation of trust value, the parameters in Eq. (6), (10) and (11) needs to be fixed. Therefore an experiment is carried out, and Fig. 5 shows the value of the trust by the use of Eq. (6) with the parameter p in the range $[1, 100]$. Seen from Fig. 5, this parameter affects the result of the aggregation a little. Hence, in the following experiment p is assigned with 50. The results of Eq. (10) and (11) are similar, and both of their parameters are set to be 50.

To evaluate how similarity functions affect the aggregation result, all functions are run on the same data. In the experiment whose results are shown in Table. II, dishonest SCs try to falsely improve the value of the trust of a bad service. Hence, the smaller the deviation is from the ideal value, the better the similarity function is. The first row represents the ideal trust values, and other rows show the results of corresponding similarity function. Seen from the results of the Table. II, S_2, Eq. (7) works the best.

Table. III shows the results with dishonest consumers trying to degrade the value of the trust. We can see that the aggregation results of S_2 have the smallest deviation from the ideal values. Hence, Eq. (7) performs the best.

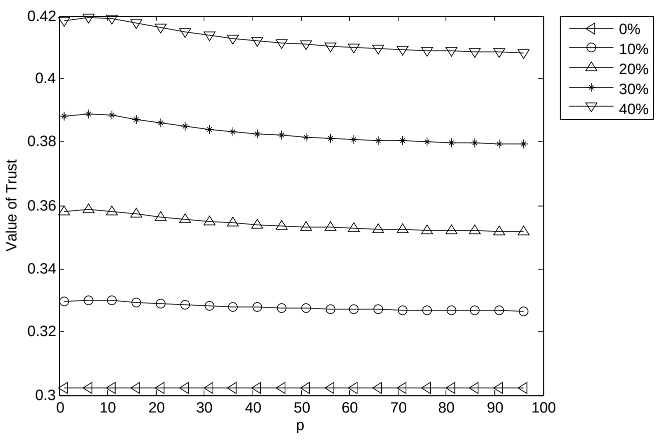

Fig. 5. p of Similarity Function S_1 versus Value of Trust

TABLE II
VALUE OF TRUST AGGREGATED WITH DIFFERENT PERCENTAGE OF DISHONEST CONSUMERS TRYING TO FALSELY IMPROVE THE TRUST

	0%	**10%**	**20%**	**30%**	**40%**
Ideal	0.3	0.3	0.3	0.3	0.3
S_1	0.3017	0.3296	0.3583	0.3858	0.4161
S_2	0.3016	0.3088	0.3175	0.3224	0.3302
S_3	0.3017	0.3243	0.3488	0.3718	0.3986
S_4	0.2991	0.6438	0.6718	0.6877	0.6937
S_5	0.3022	1.0722	0.8082	0.7639	0.7416
S_6	0.2986	0.6667	0.6830	0.6949	0.6985
S_7	0.3008	0.4926	0.5692	0.6140	0.6407
S_8	0.3011	0.5109	0.5840	0.6259	0.6497
S_9	0.3011	0.489792	0.5660	0.6116	0.6387
S_{10}	0.3011	0.4972	0.5726	0.6169	0.6428
S_{11}	0.3011	0.4944	0.5702	0.6149	0.6413

VII. CONCLUSION

Trust in QoS of SPs is increasingly important for consumers to select appropriate services. In this paper, QoS criteria have been classified into several groups on the basis of their characteristics, and then a model of automatically calculating the trust in quality criteria has been presented, not only based on the trust in SPs but also on the basis of the trust in SCs, which significantly helped reduce the influence of dishonest SCs. The trust calculation process has been divided into two steps, Time and Aggregation Domain. At time domain, a SC used the values obtained by the consumer itself while at aggregation domain a SC calculated the value of trust using values from others, which may be malicious. Hence at aggregation domain, an improved KF-based algorithm has been presented to filter out malicious values and the trust in other consumer X was used to weight the data provided by X. Finally, our model has been evaluated by several experiments

TABLE III
VALUE OF TRUST AGGREGATED WITH DIFFERENT PERCENTAGE OF DISHONEST CONSUMERS TRYING TO FALSELY DEGRADE THE TRUST

	0%	**10%**	**20%**	**30%**	**40%**
Ideal	0.7	0.7	0.7	0.7	0.7
S_1	0.6950	0.6687	0.6397	0.6072	0.5704
S_2	0.6950	0.6868	0.6754	0.6608	0.6398
S_3	0.6951	0.6729	0.6476	0.6185	0.5848
S_4	0.6947	0.34504	0.3191	0.3065	0.3058
S_5	0.6950	-0.1630	0.1710	0.2273	0.2586
S_6	0.6952	0.2321	0.2697	0.2772	0.2877
S_7	0.6951	-0.0127	0.1992	0.2409	0.2659
S_8	0.6949	0.4652	0.3943	0.3568	0.3397
S_9	0.6949	0.4864	0.4111	0.3693	0.3486
S_{10}	0.6949	0.4538	0.3858	0.3506	0.3353
S_{11}	0.6949	0.4315	0.3701	0.3396	0.3278

and the results have shown that a more accurate value estimate can be made by our model, with higher detection accuracy than the original algorithm under a more strategic malicious environment, although our algorithm suffers a higher false detection rate. Among a collection of similarity functions, Eq. (7) performed the best.

However, our model required a large amount of historic estimation and lots of calculation. Hence, further research will be carried out to reduce the need of storing a large number of historic estimation and calculation. In addition, when aggregating values from other consumers, only first-hand data were used, meaning that a SC only aggregates the data from the SCs that it knows. In future, more data will be considered.

ACKNOWLEDGMENT

This work is partially supported by British Telecom Research. The authors would like to thank Dr. Keith Briggs, Dr. Michael Fitch and Mr. Santosh Kawade.

REFERENCES

[1] Calinescu, R., Grunske, L., Kwiatkowska, M., Mirandola, R., Tamburrelli, G.: “Dynamic QoS management and optimization in service-based systems,” IEEE Trans. Softw. Eng., 2011, 37, (3), pp. 387–409.
[2] Moser, O., Rosenberg, F., Dustdar, S.: “Domain-specific service selection for composite services”, IEEE Trans. Softw. Eng., 2012, 38, (4), pp. 828–843.
[3] Cardellini, V., Casalicchio, E., Grassi, V., Iannucci, S., Presti, F. Lo, Mirandola, R.: “MOSES: A Framework for QoS Driven Runtime Adaptation of Service-Oriented Systems”, IEEE Trans. Softw. Eng., 2012, 38, (5), pp.1138–1159.
[4] Ardagna, D., Pernici, B.: “Adaptive service composition in flexible processes”, IEEE Trans. Softw. Eng., 2007, 33, (6), pp. 369–384.
[5] Wang, S., Sun, Q., Yang, F.: “Quality of service measure approach of web service for service selection”, IET Softw., 2012, 6, (2), pp.148–154.
[6] Conner, W., Rouvellou, I., Iyengar, A., Nahrstedt, K., Mikalsen, T.: “A trust management framework for service-oriented environments”. Proc. Int. Conf. World Wide Web, Madrid, Spain, 2009, pp. 891–900.

[7] Wang, X.F., Liu, L., Su, J.S.: "RLM: A general model for trust representation and aggregation", IEEE Trans. Services Comput., 2012, 5, (1), pp. 131–143.

[8] Wang, P., Chao, K.-M., Lo, C.-C., Farmer, R.: "An evidence-based scheme for web service selection", Inf. Technol. Manag., 2011, 12, (2), pp. 161–172.

[9] Wang, S., Sun, Q., Zou, H., Yang, F.: "Reputation measure approach of web service for service selection", IET Softw., 2011, 5, (5), pp. 466–473.

[10] Das, A., Islam, M.: "Securedtrust: A dynamic trust computation model for secured communication in multiagent systems", IEEE Trans. Depend. Secure, 2012, 9, (2), pp. 261–274.

[11] Yan, S.R., Zheng, X.L., Chen, D.R., Zhang, W.Y.: "User-centric trust and reputation model for personal and trusted service selection", Int. J. Intell. Syst., 2011, 26, (8), pp. 687–717.

[12] Viriyasitavat, W., Martin, A.: "A survey of trust in workflows and relevant contexts", IEEE Commun. Surv. Tutor., 2012, 14, (3), pp. 911 –940.

[13] Jøsang, A., Ismail, R., Boyd, C.: "A survey of trust and reputation systems for online service provision", Decis. Support Syst., 2007, 43, (2), pp. 618–644.

[14] Wang, Y., Lin, K.J.: "Reputation-oriented trustworthy computing in e-commerce environments", IEEE Internet Comput., 2008, 12, (4), pp. 55 –59.

[15] Cho, J.-H., Swami, A., Chen, I.-R.: "A survey on trust management for mobile ad hoc networks", IEEE Commun. Surv. Tutor., 2011, 13, (4), pp. 562 –583.

[16] Govindan, K., Mohapatra, P.: "Trust computations and trust dynamics in mobile adhoc networks: A survey", IEEE Commun. Surv. Tutor., 2012, 14, (2), pp. 279 –298.

[17] Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: "The eigentrust algorithm for reputation management in p2p networks". Proc. Int. Conf. World Wide Web, New York, USA, 2003, pp. 640–651.

[18] Li, X., Zhou, F., Yang, X.: "Scalable Feedback Aggregating (SFA) Overlay for Large-Scale P2P Trust Management", IEEE Trans. Parall. Distr., 2012, 23, (10), pp. 1944–1957.

[19] Xiong, L., Liu, L.: "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities", IEEE Trans. Knowl. Data En., 2004, 16, (7), pp. 843 – 857.

[20] Zhang, Y.C., Fang, Y.G.: "A fine-grained reputation system for reliable service selection in peer-to-peer networks", IEEE Trans. Parall. Distr., 2007, 18, (8), pp. 1134–1145.

[21] Whitby, A., Jøsang, A., Indulska, J.: "Filtering out unfair ratings in bayesian reputation systems". Proc. Int. Conf. Autonomous Agents and Multiagent Systems, New York, USA, 2004, pp. 106–117.

[22] Wang, Y.H., Singh, M.P.: "Trust Representation and Aggregation in a Distributed Agent System". Proc. Conf. Artificial Intelligence, Boston, USA, 2006, pp. 1425–1430.

[23] Nepal, S., Malik, Z., Bouguettaya, A.: "Reputation Management for Composite Services in Service-Oriented Systems", Int. J. Web Serv. Res., 2011, 8, (2), pp. 29–52.

[24] Koloseni, D., Lampinen, J., Luukka, P.: "Differential evolution classifier with optimized distance measures from a pool of distances". Proc. IEEE Congress on Evolutionary Computation, Brisbane, Australia, 2012, pp. 1–7.

[25] Welch, G., Bishop, G.: "An introduction to the kalman filter", 1995.