A Trust-based Model for Quality of Web Service

Bo Ye, Anjum Pervez, Mohammad Ghavami
the Faculty of Engineering, Science and Built Environment
London South Bank University
London, UK
{yeb,perveza,ghavamim}@lsbu.ac.uk

Maziar Nekovee
BT Research,
Polaris 134 Adastral Park,
Martlesham, Suffolk, UK
maziar.nekovee@bt.com

***Abstract*—In order to choose the best services among various services across the world, how much a service can be trusted is increasingly important for service consumers. In addition, if services are invoked by machines, it is increasingly crucial that the trust in services can be calculated automatically. However, most existing approaches are based on the assumption that the trust value can be provided by consumers, but 'how' is not solved. In this paper, criteria of Quality of Service (QoS) are classified into different groups, and an automatic trust calculation is introduced. After that, an approach based on the Kalman Filter is presented to filter out malicious values and to estimate real values. Through aggregating the values provided by other consumers, the value of the trust in different QoS criteria can be obtained. Finally, experiments are carried out to access the validation and robustness of our model.**

Keywords-Web service; Quality of Service; Trust; Kalman Filter

I. INTRODUCTION

While more and more systems are developed based on Service and Cloud Computing, more and more different kinds of services are going to emerge on the Internet. Because of various users' requirements and many service attributes, selection of a proper service is not easy for a user in Service Oriented Architecture (SOA) systems. Currently, most approaches are based on Quality of Service (QoS) to select services.

Among these QoS criteria, reputation is a really important one because service consumers can access a large number of services providing identical or similar functions. Most providers publish values of their services' QoS. However, how much these values can be trusted is really crucial. Electronic commerce has a similar issue, and various electronic markets and online electronic commerce companies have built reputation systems, e.g., Amazon, eBay, Yahoo!, Slashdot. Many researchers have considered trust-based system, an effective way to identify malicious consumers to minimize their threat and protect the system from possible misuses and abuses.

Therefore, it is crucial to measure web service trust and many researchers have proposed various approaches using different techniques. Although many efficient and robust measure solutions [1]–[6] has been proposed in previous research, these approaches still mostly have following weaknesses.

First of all, most approaches measure service reputation based on the feedback provided by human, and therefore it is difficult to ensure the accuracy. This kind of systems cannot be built without humans participating. Due to different abilities and knowledge of human, it is impossible for them to provide the same feedback, even if they use the identical service. Hence, it is necessary to build an automatic trust measure system.

Secondly, existing trust measure solutions mostly collect feedbacks only between service consumers and providers, but rarely use information among service consumers. Most of such systems are centralized, and not all systems have a center server. Hence, how a new service consumer can obtain the trust level of a service in distributed systems needs to be considered.

In addition, due to existing malicious service consumers, how those malicious ones can be filtered out becomes increasingly crucial. Malicious consumers might provide malicious values to falsely improve the trust, or to degrade the trust in certain service providers for commercial benefits.

To address the weaknesses above, an approach to measure the trust both in service providers and consumers has been proposed. This approach first groups QoS criteria, and then measures the trust in each QoS criterion of a service based on the characteristics of QoS criteria. The measure of trust in services has also been divided into two main stages, including Time Domain and Aggregation Domain, because a service consumer can measure the trust in service providers at different domains. First, it may invoke a service many times, so that it can measure the trust based on its own observation at Time Domain. All data obtained itself at Time Domain, and therefore it is unnecessary to filter out any information. Second, it may also obtain the trust by using the data from others, called as Aggregation Domain. At this stage, the consumer not only needs to aggregate all information, but also has to filter out malicious data.

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

1) Quality of Web Service Criteria have been grouped into several classes based on their characteristics, and the stages of trust measure have been classified into two main domains. A trust measure approach has been proposed to overcome the weaknesses mentioned above.
2) Both negative and positive malicious values can be detected by our approach, and the trust in both service providers and consumers has been measured, which makes the measure more reliable. Trust in consumers reflects its trust level from another one's perspective.
3) At Aggregation Domain, when a service consumer ag-

ISBN: 978-1-61208-270-7

gregates the information from numerous other service consumers, it uses the trust in other consumer X to weight the trust in a service provider from X.

The rest of this paper is organized as follows. The major related research has been introduced in Section I. Section III presents how quality criteria are grouped and how the values of trust in quality criteria and reference trust are calculated automatically. Section IV describes how malicious values are detected and how different values from a variety of consumers are aggregated to calculate the value of trust. The model is evaluated by carrying out different experiments in Section V. Finally, the conclusion is given in the Section VI.

II. Related Work

The characterization of reputation systems and threats facing them from a computer science perspective can be found in [4]. Wang and Lin [5] reviewed the reputation-based trust evaluation mechanisms in literature and outline some trust issues in e-commerce environments. An overview of existing and proposed systems that can be used to derive measures of trust and reputation for Internet transactions was given in [6].

Ardagna and Pernici [7] proposed a modelling approach to service selection problem, but trust is just one of those considered quality criteria. Although this approach is effective to select a service for a consumer, it did not focus on the detection of malicious consumers. Then if malicious consumers exist, it may not select appropriate services for consumers.

In [8], the reputation was modelled as a three-dimension belief (b, d, u), which represent the positive, negative and uncertain probabilities. In [9]–[11], a Bayesian reputation approach was proposed to calculate the reputation value based on the beta probability density functions (PDF). In [10], intuitive parameters needs to be tuned manually without guarantee of any quantitative confidence. Wang, Liu and Su [12] proposed a general trust model for a more robust reputation evaluation. The trust in [8]–[10] was modelled as predicted probability values, but prediction variance was ignored by them, which was considered in [12]. However, all these models only use feedbacks between service providers and consumers, but did not use feedbacks among service consumers.

Based on reputation Corner et al. [13] proposed a trust management framework that supports the synthesis of trust-related feedback from multiple services while also allowing each entity to apply various scoring functions over the same feedback data for personalized trust evaluations. However, this approach is based on the assumption that consumers do not mask their malicious behaviour, meaning that it is hard to detect those malicious service consumers that behave well until they gain good trust values and then behave maliciously. Xiong et al. and Srivatsa et al. [14], [15] measured the trust by the use of personalized similarity between itself and other peer X to weight feedback from X.

However, all these models were built based on the assumption that a consumer can provide a trust value in the service. This is the first weakness summarized in Section I. How a trust value can be automatically calculated is ignored in these models.

III. Quality Criterion

In this section, Quality Criterion and relevant concepts are introduced first, and then, how trust values can be calculated automatically is presented.

Definition 1. ***Quality Criterion:*** *This encompasses a number of Quality of Service (QoS) properties used to evaluate a web service. The following is a brief explanation of criteria:*

1) **Price (P):** The price of a web service is a sum of the invoked operations' prices of the web service.
2) **Response Time (RT):** The estimated time between when a request is sent and when a result is received.
3) **Availability (A):** The probability that a web service is available.
4) **Success Rate (SR):** The rate of a web service's ability to response to requests successfully.

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. Based on the nature of a criterion, criteria fall into two major classes:

1) **Ratio Criterion:** The value of a criterion can be expressed as a ratio, and their values can also be directly used as the trust in the criterion, e.g., availability, success rate, etc. Please note that Ratio Criteria are not the criteria whose values obtained from providers are rate. For example, compensation rate's values are rate. However, it is not a ratio criterion.
2) **Non-ratio Criterion:** The value of a criterion can not be expressed by a ratio, e.g., price, response time, etc.

Because there are different ways to obtain the value of a criterion, there are two major classes of criterion value:

1) **Published Value of a Criterion:** The value of a criterion is published by service provider. This can be updated by providers at any time.
2) **Actual Value of a Criterion:** The value of a criterion is obtained by service consumers after invoking a service, which may be different from Published Value. For instance, a provider may publish $40ms$ as a service's response time, but the actual response time may be $43ms$ when the service is invoked.

Definition 2. ***Trust:*** *To simplify description, in this paper, 'Trust' is used as a property of a service, denoted by T.*

For example, a consumer A has a trust in another consumer 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.

1) **Criterion Trust:** A consumer A's trust in a criterion C of a service S provided by a service provider P, denoted by $T(A \rightarrow P.S.C)$. It identifies how much a Published Value of $P.S.C$ can be trusted.

2) **Reference Trust:** Consumer A's trust in consumer B's capacity of referring to other consumers' ability to do something, defined by $T(A \rightarrow B)$. Please note that a service provider can also have a reference trust, because the service provider can also be a service consumer, recommending another service provider.

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

$T(A \rightarrow S.C)_j$ represents A's trust in service S's criterion C after j^{th} time user A invokes web service S. c represents the published value of $S.C$, while c_j is the actual value obtained by the user after j^{th} time invoking service S.

Suppose Criterion C is a negative one, meaning that the decrease of this criterion benefits the trust, then $T(A \rightarrow S.C)_j$ is calculated by the following equations.

The number of positive C values, num_j^{po}, is calculated by:

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

The number of negative C values, num_j^{ne}, is computed by:

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

The following equation is used to calculate the value of positive part of C's trust,

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

Value of negative part of C's trust,

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

At last, $T(A \rightarrow S.C)_j$ is calculated by

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

Please note that the equal values to the Published Value are always classified as positive values.

B. Reference Trust Calculation

A service consumer A's reference trust in another consumer B is used to identify how much B can be trusted by A. Using the value of trust in B, A can know how much he can trust the services or other consumers referred by B.

Because a service provider can provide various services and an identical service can be provided by a number of service providers, $P.S.C$ is used to denote a service S's criterion C provided by a service provider P. $T(A \rightarrow P.S.C)$ represents the value of A's trust value in $P.S.C$, while similarly $T(B \rightarrow P.S.C)$ represents the value of B's trust value.

The set of trust's values obtained by service users can be viewed as a multidimensional space and each user can be a point in the space. Hence, A's trust $T(A \rightarrow B)$ in service consumer B can be calculated by the geometric distance between the points as follows

$$T = 1 - \sqrt{\frac{\sum\limits_{P}\sum\limits_{S}\sum\limits_{C}(T(A \rightarrow P.S.C) - T(B \rightarrow P.S.C))^2}{|T(A \rightarrow P.S.C)|}} \tag{6}$$

Their values are more similar, meaning that their experience is more similar, and then, A can trust B more.

C. Trust Transitivity

If a service consumer A needs to know how much he can trust in criterion C of service provider S, but he has no information about it. However, B has trust in criterion C of service provider S, and A knows how much he can trust B. Then A's trust in $S.C$ can be defined by:

$$A \rightarrow S.C = (A \rightarrow B) \cap (B \rightarrow S.C) \tag{7}$$

In this equation, the symbol $\cap$ does not mean that $A \rightarrow B$ and $B \rightarrow S.C$ intersect. It means that based on the trust's transitive property, A's trust in $S.C$ can be derived by A's reference trust in B and B's criterion trust in $S.C$.

It is common to collect reference trusts from several different service users to make better decisions. This can be called consensus trust. Assume a service consumer A needs to obtain the value of the trust in a criterion C of a service S, but he has no information about the trust of $S.C$. However, he has information about trust in other consumer X and Y, and both of they have a trust in $S.C$. Then the trust relationship between A and $S.C$ can be defined by :

$$A \rightarrow S.C = ((A \rightarrow X) \cap (X \rightarrow S.C)) \cup ((A \rightarrow Y) \cap (Y \rightarrow S.C)) \tag{8}$$

In this equation, the symbol $\cup$ means that A's trust in $S.C$ can be derived by combining X and Y's criterion trust in $S.C$.

Definition 3. ***Transitive Trust:*** *Suppose there are two service consumers A and B, where A has a reference trust in B. Additionally, B has a function trust in criterion C of service S. A's trust in P.S.C can be derived by using both B's function trust in S.C and A's trust in B:*

$$T(A \rightarrow P.S.C) = T(A \rightarrow B) \cdot T(B \rightarrow P.S.C) \tag{9}$$

Definition 4. ***Consensus Trust:*** *The consensus trust of two consumers' trust in P.S.C is a trust that reflects both trust in a fair and equal way. Then derived consensus trust in P.S.C, $T(A \rightarrow P.S.C)$, is calculated by:*

$$\frac{|T(A \rightarrow X) \cdot T(X \rightarrow P.S.C)| + |T(A \rightarrow Y) \cdot T(Y \rightarrow P.S.C)|}{|T(A \rightarrow X)| + |T(A \rightarrow Y)|} \tag{10}$$

IV. Criterion Value Estimation and Malicious Value Detection

Malicious service consumer can be classified as either adulating service consumer, which tries to falsely improve the trust in certain service providers, or defaming service consumer, trying to degrade the trust in certain service providers.

A. Criterion Value Estimation

It is reasonable to model the distribution of the value of $P.S.C$ as Normal distribution with (μ, σ), because the values of $P.S.C$ obtained by a consumer A are independent. For each criterion, its value follows 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.

At one time, each service consumer i has an estimated value of $P.S.C$'s $\{\mu^r, \sigma^r\}$, denoted as $\{\mu_i^e, \sigma_i^e\}$, μ_i^e and σ_i^e represent the estimated real value of $P.S.C$ and estimated variance, respectively. A service consumer A is going to use estimated values of all other service consumers to predict $P.S.C$'s $\{\mu^r, \sigma^r\}$. After using service consumer i's estimated value, A's estimated values are denoted as $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$. Because of incomplete knowledge of the criterion of the service, i's estimated value usually has a deviation from A's estimated value $\{\mu^e_{A,i}, \sigma^e_{A,i}\}$. Because the estimated value from many independent service consumers, the relation between i's estimate and A's estimate is modeled as

$$\begin{aligned} \mu_i^e = & \quad \mu^e_{A,i} + \lambda_\mu \text{ and } p(\lambda_\mu) \sim \text{Normal}(0, \Lambda_\mu) \\ \sigma_i^e = & \quad \sigma^e_{A,i} + \lambda_\sigma \text{ and } p(\lambda_\sigma) \sim \text{Normal}(0, \Lambda_\sigma) \end{aligned} \tag{11}$$

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

Based on Kalman Filter [16], the estimates of $\{\mu^r, \sigma^r\}$ are governed by the following linear stochastic difference equations:

$$\begin{aligned} \mu^e_{A,i} &= F_\mu \mu^e_{A,i-1} + Bu_{i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma^e_{A,i} &= F_\sigma \sigma^e_{A,i-1} + Bu_{i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned} \tag{12}$$

where, F is the factor for relationship between the previous estimate based on the estimate from consumer $i-1$ and the current estimate based on i's estimate, and u is the optional control input to the estimate $\{\mu^e_A, \sigma^e_A\}$. Because in our model there is no control input, u is 0. Hence, our estimate is governed by the following linear difference equation:

$$\begin{aligned} \mu^e_{A,i} = F_\mu \mu^e_{A,i-1} + w_{\mu,i-1};\ p(w_\mu) \sim \text{Normal}(0, W_\mu) \\ \sigma^e_{A,i} = F_\sigma \sigma^e_{A,i-1} + w_{\sigma,i-1};\ p(w_\sigma) \sim \text{Normal}(0, W_\sigma) \end{aligned} \tag{13}$$

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

$Predict$ step:

$$\bar{\mu}^e_{A,i} = F_{\mu,i} \mu^e_{A,i-1}, \quad \bar{\sigma}^e_{A,i} = F_{\sigma,i} \sigma^e_{A,i-1} \tag{14}$$

$$\bar{P}_{\mu,i} = F^2_{\mu,i} P_{\mu,i-1} + W_{\mu,i}, \quad \bar{P}_{\sigma,i} = F^2_{\sigma,i} P_{\sigma,i-1} + W_{\sigma,i} \tag{15}$$

$Update$ step:

$$K_{\mu,i} = \bar{P}_{\mu,i} / (\bar{P}_{\mu,i} + \Lambda_{\mu,i}), \quad K_{\sigma,i} = \bar{P}_{\sigma,i} / (\bar{P}_{\sigma,i} + \Lambda_{\sigma,i}) \tag{16}$$

$$\begin{aligned} \mu^e_{A,i} = & \quad \bar{P}_{\mu,i} + K_{\mu,i}(\mu_i^e - \bar{\mu}^e_{A,i}), \\ \sigma^e_{A,i} = & \quad \bar{P}_{\sigma,i} + K_{\sigma,i}(\sigma_i^e - \bar{\sigma}^e_{A,i}) \end{aligned} \tag{17}$$

$$P_{\mu,i} = (1 - K_{\mu,i}) \bar{P}_{\mu,i}, \quad P_{\sigma,i} = (1 - K_{\sigma,i}) \bar{P}_{\sigma,i} \tag{18}$$

In order 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:

$$F_{\mu,i} = \frac{\sum_{j=1}^{i-1} \mu^e_{A,j} \mu^e_{A,j-1}}{\sum_{j=1}^{i-1} (\mu^e_{A,j})^2}, \quad F_{\sigma,i} = \frac{\sum_{j=1}^{i-1} \sigma^e_{A,j} \sigma^e_{A,j-1}}{\sum_{j=1}^{i-1} (\sigma^e_{A,j})^2} \tag{19}$$

$$\Lambda_{\mu,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\mu_j^e - \mu^e_{A,j})^2, \quad \Lambda_{\sigma,i} = \frac{1}{i} \sum_{j=1}^{i-1} (\sigma_j^e - \sigma^e_{A,j})^2 \tag{20}$$

$$\begin{aligned} W_{\mu,i} &= \frac{1}{i} \sum_{j=1}^{i-1} (\mu^e_{A,j} - F_i \mu^e_{A,j-1})^2, \\ W_{\sigma,i} &= \frac{1}{i} \sum_{j=1}^{i-1} (\sigma^e_{A,j} - F_i \sigma^e_{A,j-1})^2 \end{aligned} \tag{21}$$

B. Malicious Value Detection

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

$$P(|\mu_i^e - \mu^e_{A,i}| \leq \Delta_{\mu,i}) = \delta_\mu, P(|\sigma_i^e - \sigma^e_{A,i}| \leq \Delta_{\sigma,i}) = \delta_\sigma \tag{22}$$

In addition, $\mu_i^e - \mu^e_{A,i}$ and $\sigma_i^e - \sigma^e_{A,i}$ 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:

$$\begin{aligned} P(|\mu_i^e - \mu^e_{A,i}| \leq \Delta_{\mu,i}) = 1 - 2\Phi(\frac{-\Delta_{\mu,i}}{\sqrt{P_{\mu,i} + \Lambda_{\mu,i}}}), \\ P(|\sigma_i^e - \sigma^e_{A,i}| \leq \Delta_{\sigma,i}) = 1 - 2\Phi(\frac{-\Delta_{\sigma,i}}{\sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}}}) \end{aligned} \tag{23}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Hence, after solving Equations. (22) and (23), $\Delta_{\mu,i}$ and $\Delta_{\sigma,i}$ can be obtained:

$$\begin{aligned} \Delta_{\mu,i} = -\Phi^{-1}((1 - \delta_\mu)/2) \sqrt{P_{\mu,i} + \Lambda_{\mu,i}}, \\ \Delta_{\sigma,i} = -\Phi^{-1}((1 - \delta_\sigma)/2) \sqrt{P_{\sigma,i} + \Lambda_{\sigma,i}} \end{aligned} \tag{24}$$

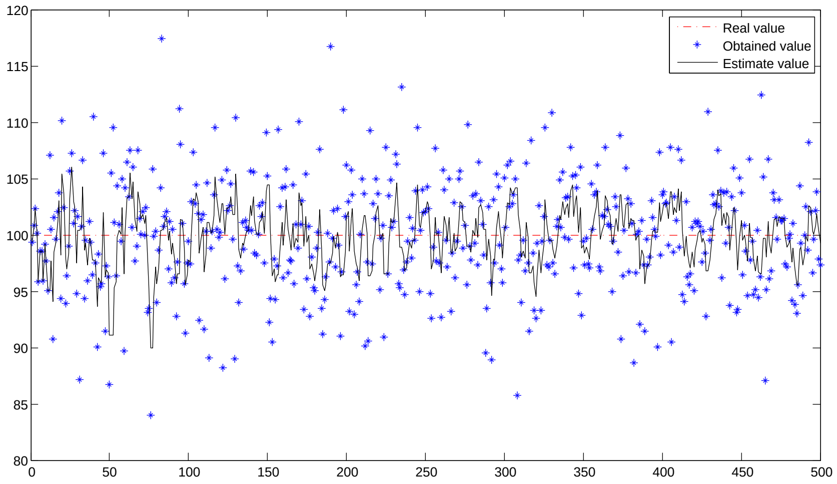

Fig. 1. The real, estimate and obtained values

C. Calculation Algorithm

Definition 5. ***Time Domain****: Each time a service consumer invokes a service, it can obtain values of all criteria. Then it can use the values to estimate real criterion value and the values are called values of quality criterion at Time Domain.*

Because at time domain all values are collected by a service consumer A itself, it is unnecessary to detect malicious values. For ratio criteria, it is easy to calculate the values and it is accurate. For instance, the success rate c_{sr} of a service S can be calculated by the following equations:

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

$$\mathrm{num}_{\mathrm{success},i} = \mathrm{num}_{\mathrm{success},i-1} + 1 \tag{25}$$

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

$$\mathrm{num}_{\mathrm{success},i} = \mathrm{num}_{\mathrm{success},i-1} \tag{26}$$

Finally:

$$\mathrm{num}_{\mathrm{total},i} = \mathrm{num}_{\mathrm{total},i-1} + 1 \tag{27}$$

$$c_{sr} = \frac{\mathrm{num}_{\mathrm{success},i}}{\mathrm{num}_{\mathrm{total},i}} \tag{28}$$

Each time a service S invoked by a service consumer i, then i can get values of non-ratio criteria. If exact values of non-ratio criteria C can be obtained, such as price, then the value $\{\mu_i^e, \sigma_i^e\}$ can also be calculated by the equations as follows:

$$\mu_i^e = E[C_i] \tag{29}$$

$$\sigma_i^e = E[(C_i - \mu_i^e)^2] \tag{30}$$

However, exact values of certain non-ratio criteria cannot be obtained, such as response time. 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, at this point, the method of criterion value estimation in Section IV is used to obtain the estimate value $\{\mu_i^e, \sigma_i^e\}$.

Definition 6. ***Aggregation Domain****: Each service consumer can collect criterion values of various services from numerous service consumers. Further these values can be aggregated by this service consumer to estimate real criterion value too, although some of these values may be malicious. These values are called values of quality criterion at Aggregation Domain.*

At aggregation domain, values of all criteria including ratio criteria and non-ratio criteria are estimated by using the method of criterion value estimation in Section IV, not only because malicious values need to be filtered out, but also because all these values may not be accurate due to incomplete knowledge on service providers.

Assume a consumer A is going to aggregate all values from others to calculate the trust in a quality criterion C of a service S provided by service provider P. Then each step after estimating the value of C by the use of the value provided by other service consumer X, the trust in C is calculated using the method presented in Section III-A. When A use this value to estimate the trust in $P.S.C$, the trust in other consumer X calculated by the use of the method introduced in Section III-B is used to weight the trust in a service provider from that consumer X . Furthermore, second hand values are also aggregated using the approach presented in Section III-C.

V. Performance Evaluation

In this section, this trust model is evaluated in a simulated environment. The model is validated first, and then another experiment is carried out to evaluate the robustness of this model, compared to some other approaches.

The first experiment is carried out in a clean environment without malicious values in order to validate the model. The value of a QoS criterion C is estimated and the accuracy of the value estimation is evaluated each step. One result calculated by our model is shown in Fig.1. The dashed line represents C's real values, while the obtained value with noise is denoted by the stars. The real line represents the values estimated by our model. It is seen that in this experiment the obtained values

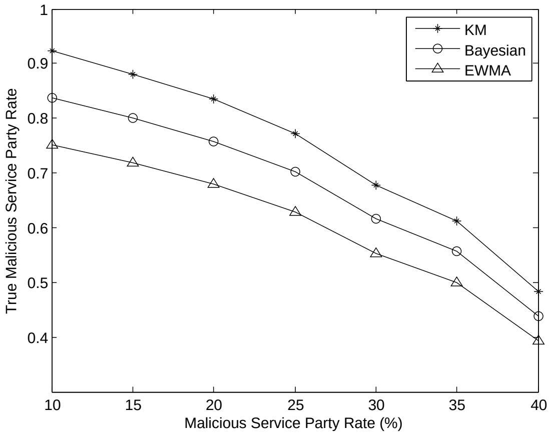

Fig. 2. Average true malicious rate

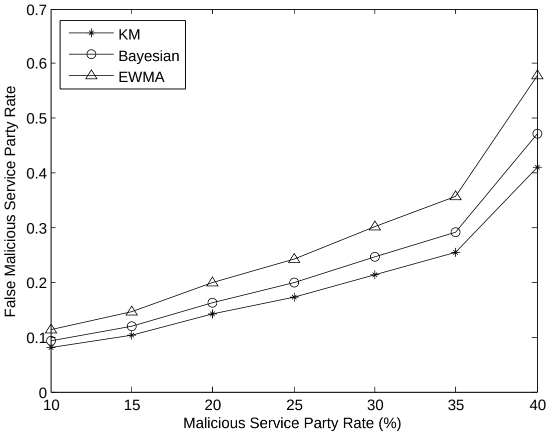

Fig. 3. Average false malicious rate

are not accurate, but C's values are still being estimated well, which is closer to the real values than the obtained values.

In order to evaluate the robustness of this model, another experiment is carried out in an environment with malicious values. Our approach is compared with the methods in [9], [13], respectively represented by EWMA and Bayesian, to evaluate the robustness of malicious value detection. Because it is almost impossible that more than a half of all service consumers within an environment behaviour maliciously and it is impossible to perform well in an environment with more than 50% malicious service consumers, the probability of malicious service consumers is set up to 40%.

Definition 7. ***True Malicious Rate:*** *The percentage of correctly detected malicious service consumers.*

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

Definition 8. ***False Malicious Rate:*** *The percentage of wrongly detected non-malicious service consumers.*

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

As shown in Figs. 2 and 3, as the increase of the malicious service consumer probability, our model performs better than those two approaches. Hence, the accuracy of those approaches is lower than ours.

VI. Conclusion

Trust in Quality of Service of service providers is really important for service consumers to select services. In this paper, a model using Kalman Filter to filter out malicious values was introduced. First, QoS criteria were classified into several groups on the basis of their characteristics. Then a model to estimate the trust in quality criterion was presented, not only based on the trust in service providers but also on the basis of the trust in service consumers, which significantly helped reduce the influence of dishonest service consumers. The trust calculation processes were classified into two groups, including Time and Aggregation Domain. At time domain, a service consumer uses the values obtained by itself while at aggregation domain, a service consumer to calculate the value of trust in a service provider by the use of values from other service consumers, which may be malicious. Hence, at aggregation domain, a method based on Kalman Filter was presented to filter out malicious values and the trust in other consumer X was used to weight the data from X. Finally, our model was evaluated by two experiments and the results shown that a more accurate value estimation can be made, with better detection accuracy, compared with two other approaches.

Although our model works well, a large amount of historic estimation needs to be stored and it needs lots of calculation. Hence, further research will be carried out to reduce the need of storing historic estimation and calculation.

Acknowledgment

This work is partly supported by British Telecommunications Research. The authors would like to thank Keith Briggs, Michael Fitch, Santosh Kawade, etc.

References

[1] S. R. Yan, X. L. Zheng, D. R. Chen, and W. Y. Zhang, “User-centric trust and reputation model for personal and trusted service selection,” International Journal of Intelligent Systems, vol. 26, no. 8, 2011, pp. 687–717.

[2] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in Proceedings of the 12th International Conference on World Wide Web, 2003, pp. 640–651.

[3] Z. Yan and H. Silke, “Trust modeling and management: From social trust to digital trust,” in Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions, 2008, pp. 290–323.

[4] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A survey of attack and defense techniques for reputation systems,” ACM Compute Survey, vol. 42, no. 1, 2009, pp. 1:1–1:31.

[5] Y. Wang and K.-J. Lin, “Reputation-oriented trustworthy computing in e-commerce environments,” Internet Computing, IEEE, vol. 12, no. 4, 2008, pp. 55–59.

[6] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” Decision Support System, vol. 43, no. 2, 2007, pp. 618–644.

[7] D. Ardagna and B. Pernici, “Adaptive service composition in flexible processes,” Software Engineering, IEEE Transactions on, vol. 33, no. 6, 2007, pp. 369–384.

[8] Y. H. Wang and M. P. Singh, “Trust Representation and Aggregation in a Distributed Agent System,” in Proceeding of National Conference on Artificial Intelligence, 2006, pp. 1425–1430.

[9] Y. C. Zhang and Y. G. Fang, “A fine-grained reputation system for reliable service selection in peer-to-peer networks,” Parallel and Distributed Systems, IEEE Transactions on, vol. 18, no. 8, 2007, pp. 1134–1145.

[10] A. Whitby, A. Josang, and J. Indulska, “Filtering out unfair ratings in bayesian reputation systems,” in Proceeding of the International Joint Conference on Autonomous Agenst Systems, 2004, pp. 106–117.

[11] S. Buchegger and J.-Y. Le Boudec, “A robust reputation system for peer-to-peer and mobile ad-hoc networks,” in Proceedings of the Second Workshop Economics of P2P Systems, 2004, pp. 1–6.

[12] X. F. Wang, L. Liu, and J. S. Su, “Rlm: A general model for trust representation and aggregation,” Services Computing, IEEE Transactions on, vol. 5, no. 1, 2012, pp. 131–143.

[13] W. Conner, I. Rouvellou, A. Iyengar, K. Nahrstedt, and T. Mikalsen, “A trust management framework for service-oriented environments,” in International World Wide Web Conference, 2009, pp. 891–900.

[14] L. Xiong and L. Liu, “Peertrust: supporting reputation-based trust for peer-to-peer electronic communities,” Knowledge and Data Engineering, IEEE Transactions on, vol. 16, no. 7, 2004, pp. 843 – 857.

[15] M. Srivatsa, L. Xiong, and L. Liu, “Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks,” in Proceedings of the 14th international conference on World Wide Web, 2005, pp. 422–431.

[16] R. E. Kalman “A New Approach to Linear Filtering and Prediction Problems,” Transaction of the ASME-Journal of Basic Engineering, vol. 82, 1960, pp. 35-45.