Evidence Based Trust Estimation Model for Cloud Computing Services

Shilpa Deshpande and Rajesh Ingle
(Corresponding author: Shilpa Deshpande)

Department of Computer Engineering, College of Engineering Pune, Savitribai Phule Pune University
Wellesely Road, Shivajinagar, Pune 411005, Maharashtra, India
(Email: shilpshree@yahoo.com)

(Received Nov. 8, 2016; revised and accepted Feb. 1, 2017)

Abstract

With growing acceptance of cloud environment, demand for cloud services based applications is rapidly increasing. Cloud environment is inherently distributed and highly dynamic. Usually, many cloud services having similar functionalities but varying performances are offered. This makes difficult for user to identify the appropriate cloud service. Consequently, trust assessment is becoming essential to locate and to continue using the suitable cloud service. Quality of Service (QoS) attributes are significant for trust assessment in cloud environment. These attributes are dynamically changing and are not considered by the traditional trust assessment approaches. Hence, these approaches are inadequate to withstand in cloud environment. This paper proposes an evidence based trust estimation model (EBTEM) for trust assessment of cloud services. EBTEM uses various attributes of cloud service as evidence factors. It performs adaptive trust computation, which is sensitive to changes in the service behavior. EBTEM also presents a method for dynamic trust prediction. Experimental results demonstrate that the proposed model outperforms other models in respect of accuracy and efficiency.

Keywords: Adaptive Trust Computation; Cloud Computing; Dynamic Trust Assessment; Evidence Based Trust; Quality of Service (QoS)

1 Introduction

Real world software systems are increasingly becoming large, complex and business critical. Such enterprise applications demand flexibility in terms of compute capability, location of data, resources and users. Cloud computing paradigm fulfills these needs on demand by dynamically provisioning services and resources over the network. Cloud computing infrastructure and services offer several benefits such as simplicity to the end- users, reduced costs, dynamic resource sharing, pay-per-use and dynamic resource availability. On one side while cloud environment offers these benefits to the user community, on the other side it also poses challenges of increased system complexity, dynamicity, non-transparency of cloud services and geographically distributed data centers [2, 24].

On this background, ensuring availability of services and predicting performances of applications hosted on cloud infrastructure become more and more challenging. Security of applications and data deployed on cloud and maintaining privacy of users, add up further to these challenges. Thus, from the consumer perspective, service not being available when needed, longer time to get response than expected and security and privacy risks, result in lack of trust toward the provider [7].

Across a broad spectrum, enterprises such as banking, hospitals and the like, adopting the cloud computing for its cost-benefits, need to maintain the confidentiality and integrity [9] of the huge amount of data placed in the cloud environment. But as the cloud services are designed to be offered in non-transparent fashion, enterprises may believe that they can no longer manage their data. Hence the users may be hesitant about the probable service quality [6, 20]. This motivates the need of establishing efficient mechanism for trust estimation in the cloud environment. However, trust assessment in cloud environment poses the important issues, which are revealed as part of the following discussion.

A service level agreement (SLA) formed between a cloud user and a provider contains the technical and functional details of the offered service [8]. Contents of SLA are not consistent among the cloud service providers offering similar services. Hence users cannot assess the trust of cloud service provider based on its SLA only [8, 29]. The conventional reputation based trust mechanism reveals only the general thinking of cloud service consumers towards the cloud service and does not reflect the judgement about the performance of the cloud service [10]. In the context of cloud computing, the trust mainly depends on the performance of the cloud service depicted in terms of various cloud service attributes [6, 10]. Hence the cloud Quality of Service (QoS) attributes like availability, performance, security are critical and their evidences are needed to be considered by the trust estimation mechanism.

Trust in cloud computing can be viewed as an indicator of service behavior. Hence trust value of a cloud service may change dynamically in response to the experience of the cloud service by the end user [10, 25]. Consequently, the trust estimation needs to be a continuous dynamic process and not a one-time assessment. Cloud auditor as a third party may perform the trust assessment of cloud services. However, as the audit is conducted only after a certain period, corresponding trust assessment does not represent a dynamic trust evaluation of a cloud service [6].

Cloud environment being highly dynamic, method of trust assessment needs to be responsive to the changes in the behavior of cloud service. Thus in turn, it requires an adaptive trust assessment [24]. Consequently, trust assessment of a cloud service needs to consider the relative importance of each individual QoS attribute in trust calculation. Assigning weights manually to the various attributes of a cloud service requires a judgement by an expert and is time-intensive [27]. Moreover, trust assessment based on manually weighing of attributes does not indicate an adaptability to the cloud service in operation.

In this paper, we present an evidence based trust estimation model (EBTEM), addressing the above mentioned issues. More specifically, the contributions are towards development of:

1) A methodology for computation of trust at an instant of time using evidences of multiple attributes of cloud service.

2) Adaptive trust assessment mechanism containing mathematical formulation of weights which are computed adaptively in response to the changes in the behavior of cloud service.

3) Formulation of dynamic trust prediction over a period of time.

4) An algorithm for adaptive and dynamic trust assessments of a cloud service based on service evidence factors.

5) Comparison of the proposed trust model with other models with regard to accuracy and efficiency.

The paper is organized as follows. Section 2 presents a review of related work. In Section 3, the architecture of the system meant for the proposed trust model and the functional overview of trust estimation are described. Section 4 defines the EBTEM and presents the details of adaptive and dynamic trust assessment. Section 5 depicts the algorithm for trust estimation of a cloud service based on service evidence factors. Section 6 covers the performance evaluation of the proposed trust model including the results and analysis. Section 7 concludes the paper.

2 Related Work

The initial approaches of trust assessment in cloud environment are exclusively based on the traditional technique of reputation. This technique employs feedbacks about a particular cloud service from many service consumers to obtain the trust of that service. In reputation-based technique, source of feedback is not known to the cloud service users. Hence credibility of feedbacks is a major issue in the trust assessment [24]. Reputation based mechanism is helpful only in initial judgement about the cloud service. The mechanism may be inadequate as trust placed on the service evolves with experience. Trust assessment approaches proposed by [1, 22, 23] are based on reputation. These approaches lack in the capability to perform dynamic assessment of trust. A framework proposed by Noor and Sheng [22, 23] offers Trust as a Service (TaaS) for evaluation of cloud services. The framework provides a credibility model that differentiates the reliable feedbacks from the deceptive ones. Trust mechanism suggested by Abawajy [1] uses fading factor to keep track of the drop in satisfaction ratings over a period of time.

Along with the user feedbacks as part of reputation mechanism, few of the approaches employ other factors such as declarations by provider, user's own ratings and certificates for the trust assessment. However, authenticity of the factors used for trust assessment is a major concern in these approaches. Habib *et al.* [8] proposed an architecture to evaluate trust of cloud service providers using the combination of multiple factors such as provider statements, user feedbacks, certificates and expert assessment. Trust model proposed by Pawar *et al.* [26] takes into account the fulfillment of service level agreement (SLA) parameters. The approaches [8, 26] perform the trust evaluation in the form of opinions. These approaches do not consider dynamic trust assessment over a period of time. Ghosh *et al.* [5] proposed a framework to evaluate the risk of interaction with cloud service provider. The approach in turn involves evaluating the trust of cloud provider. The trust is estimated using the combination of direct and indirect interactions between user and service provider. The approach calculates a time window based trust using customer's ratings about previous interactions. However, the approach does not reflect the periodic trust update during the interaction. Ratings submitted by customers may be subjective in nature.

A model is suggested by Moyano *et al.* [21] for the trust assessment of cloud provider based on the factors such as SLA, transparency, accounting and auditing. The method uses a trust interval formed by the combination of value of the factor and its associated confidence value. Although the model offers simplicity in trust assessment, trust values are assigned using only the self-assessment based information available on the web sites of cloud providers. Moreover, subjective quantification of the factors may take place during the trust evaluation. Li *et al.* [16] proposed a model to judge the credibility of a cloud service by assessment of its trust. It uses multiple factors which include user ratings, record of service call, service certification and service quality monitoring for evaluation of trust. However, the details of service attributes are not specified explicitly. The weights assigned to the various factors are decided subjectively by the users themselves.

Few of the mechanisms consider QoS attributes for trust assessment. The approach proposed by Manuel *et al.* [19] computes the trust of a cloud resource as a simple summation of values assigned to user feedbacks, security level and reputation. A model is suggested by Manuel *et al.* [18] to compute the reputation based trust of a resource. Trust value of a resource is obtained as a combination of its identity, capability and behavior values. The approaches [18, 19] do not consider the dynamic trust evaluation over a period of time. Fan *et al.* [4] suggested a mechanism for evaluating trust of a cloud service using multiple attributes. The mechanism obtains the trust value by the user's direct interaction with the service. This trust value is combined with the reputation value of a cloud service to obtain the final evaluation. Both the assessment values rely on the feedbacks given by the users. However, authenticity of feedbacks is not addressed by the authors. A fuzzy trust evaluation approach for cloud services is suggested by Huo *et al.* [11]. The approach uses a set of cloud service attributes to evaluate a reputation based trust value. Weights to the various factors in the approaches [4, 11, 18] are assigned manually. Hence these weights may be static and subjective.

The existing QoS based approaches make use of performance, security, availability and reliability as the general attributes of cloud service for trust evaluation. Response time, throughput, capability and network bandwidth are the commonly used performance related factors for trust assessment. System proposed by Qu and Buyya [27] estimates trust of a cloud service by taking into consideration the performance variations of the service due to the dynamic attributes. The authors focus on evaluating the trust of a service prior to the user interaction by retrieving the past data of service attributes. However, updating the trust value of a service during the period of user interaction is not considered by the approach. A framework is proposed by Sidhu and Singh [28] for trust evaluation of cloud service providers based on QoS attributes. The approach monitors the QoS attributes and evaluates the compliance with regard to the SLA. However, the approach does not consider the dynamic trust evaluation and updating trust over a period of time.

Manuel [17] proposed a model which computes the trust of a resource as a combination of its past credentials and present capabilities. Past credentials of the resource are represented in the form of QoS attributes. However, the various attributes for trust assessment are merged by assigning static weights to them. Li *et al.* [15] proposed a dynamic trust management method for the resources in cloud environment. The model evaluates the trust degree of a resource based on the data obtained by monitoring of multiple attributes. The method assigns information entropy based weights to various factors and combines them to generate the trust value. However, the operations involved in the computation of trust are considerably complex in nature. The static factors such as capacity of a resource are treated similarly as dynamic factors during the trust computation.

In summary, the above review of the related work indicates that only few of the approaches [15, 17, 27, 28] employ monitoring based cloud QoS attributes for trust assessment. However, cloud QoS attributes are the vital factors for trust estimation. Values of QoS attributes obtained through monitoring are objective in nature and are more reliable factors for trust assessment. Dynamic cloud environment entails the trust to be evaluated and updated continuously with time. However, the approaches [27, 28] do not consider dynamic trust update of a cloud service according to the periodically changing values of the service attributes. Moreover, the approach [17] combines the various attributes for trust assessment by assigning static weights to them. Static weights may be subjective in nature and the corresponding trust computation does not reflect the adaptability to the changing behavior of a cloud service. Our trust model EBTEM, intends to address these shortcomings in the earlier work. EBTEM performs adaptive and dynamic trust assessments of a cloud service by taking into account multiple quality attributes of the service. Evidence based trust computation used in our model, enables dynamic update of trust values by collecting evidences at different times. EBTEM facilitates adaptive computation of weights for the various service attributes by considering the correlation among the attributes.

3 Architecture of Trust Estimation System

Figure 1 shows the overall layout of the system meant for the proposed trust model. It depicts the main trust estimator module which is connected with the other supplementary modules. Here, the cloud user can be the end-user who intends to use the trustworthy cloud service or the cloud user can be the service provider willing to deploy the application onto the cloud.

The service specification collector compiles the functional requirements of the cloud service, submitted by cloud user. Based on the kind of application to be executed, the user decides the functional specifications of the service. Service extraction module then finds the services from service repository whose functional specifications match with the required one.

Trust estimator module is the core component performing an adaptive and dynamic trust assessment of the cloud service. Direct interaction between a cloud user and the service, is the main source of evidence for trust estimation. Consequently, for the cloud service in execution, at each instant of time, trust estimator obtains the evidence factors compiled by an evidence collector over the designated period of time. The module makes use of these evidence

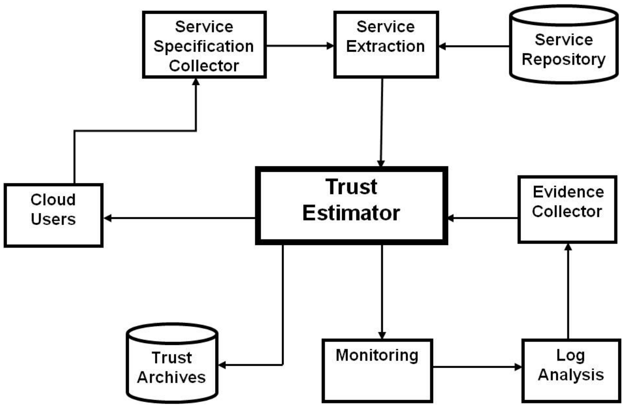

Figure 1: Architecture of trust estimation system

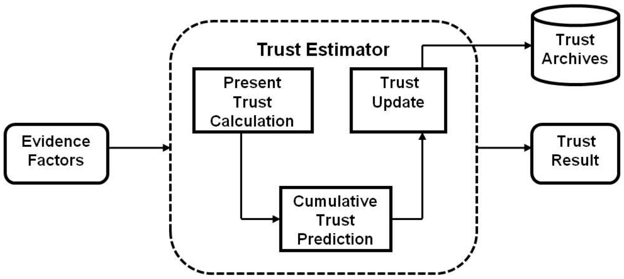

Figure 2: Functional overview of trust estimation

factors of the service attributes, for the trust assessment of the service. The cloud user can decide whether to continue using the cloud service based on the state of trust indicated by the trust estimator. The result of trust assessment is recorded in the trust archives.

Monitoring is a real-time dynamic process tracking the performance of the cloud service in operation. It observes the variation in the performance and creates a log containing values of cloud service attributes such as response time, throughput, availability and security. Log analysis module retrieves the evidence factors recorded as part of continuous monitoring process after every fixed time interval. The evidence collector collects these evidence factors which are then used for trust assessment of a cloud service.

Trust estimation is the primary focus of this paper. Therefore, the details of supplementary modules which cover service extraction, monitoring and related functionalities, are not discussed further, in this paper. We assume these as the already existing valid services and are available in the form of external interfaces to the trust estimator.

Figure 2 shows the high-level functional overview for the trust estimation of the cloud service in operation. Evidence factors over the period of time, representing the QoS attributes of the cloud service, are taken as input by the trust estimator. The trust estimator calculates present trust of a service by aggregating all the evidence factors at an instant of time. Subsequently, the module performs computation of cumulative trust over a period of time. The trust result generated in the form of cumulative trust indicates the predicted trust level of a cloud service. The trust estimator updates the trust value stored in the trust archives by the latest cumulative trust value. The details of present trust and cumulative trust assessments of a cloud service are described in Section 4. The steps depicting the control flow for trust estimation are presented in Section 5 in the form of algorithm.

4 Evidence Based Trust Estimation Model

Trust value of a cloud service is a function of cloud service attributes. Value of a cloud service attribute is termed as an evidence factor.

Definition 1. *Evidence Based Trust Estimation Model (EBTEM) is defined by a 9-tuple* $(L, AC, TI, C, M, PT, CT, E, D)$ *where*

L: Set of v cloud services: $\{s_1, s_2, ..., s_v\}$

AC: Set of m cloud service attributes: $\{R_1, R_2, ..., R_m\}$

TI: Ordered discrete set of n time instances: $\{1, 2, ..., n\}$

C: An evidence matrix which depicts m evidence factors at each of the n time instances.

M: Normalized evidence matrix.

PT: Present Trust of a cloud service at a particular time instant.

CT: Cumulative Trust of a cloud service over a period of time.

E: A set of core trust estimation functions: $\{f_{PT}, f_{CT}\}$*; where* f_{PT} *indicates a function to compute Present Trust (PT) and* f_{CT} *is a function to assess Cumulative Trust (CT).*

D: A set of allied functions: $\{f_{NE}, f_{CW}\}$*; where* f_{NE} *is a function to normalize evidence factors and* f_{CW} *indicates a function to compute weights of cloud service attributes.*

While a cloud service is running, evidence factors are retrieved after every fixed time interval. Representation of the evidence factors is devised in the form of an evidence matrix as shown below.

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{bmatrix} \tag{1}$$

In Equation (1), at a time instant i such that $1 \leq i \leq n$, a row in the matrix indicates a sample of evidence factors as $\{c_{i1}, c_{i2}, ..., c_{im}\}$ and each value c_{ij} in the sample, denotes a value of an attribute R_j. Thus, there are n samples of evidence factors. Column position in the matrix specifies a particular attribute within the sample.

In order to transform values of all the attributes to uniform range and to make them independent of units, values of the attributes in the evidence matrix need to be normalized. Normalization involves scaling of the values. Thus, for further processing of trust assessment, each value in the evidence matrix is normalized in the range denoted by $[R^{new\_min}, R^{new\_max}]$. From the perspective of desired performance of a cloud service, attributes can be categorized in two types: one where higher value of an evidence factor c_{ij} is desired and the other where lower value of c_{ij} is desired. The category where higher value of c_{ij} is desired, the corresponding normalized value h_{ij} is formulated as shown below.

$$h_{ij} = \frac{(c_{ij} - R_j^{min})(R^{new\_max} - R^{new\_min})}{(R_j^{max} - R_j^{min})} + R^{new\_min} \tag{2}$$

The other category where lower value of c_{ij} is desired, the corresponding normalized value h_{ij} is devised as:

$$h_{ij} = \frac{(R_j^{max} - c_{ij})(R^{new\_max} - R^{new\_min})}{(R_j^{max} - R_j^{min})} + R^{new\_min} \tag{3}$$

In Equations (2) and (3), R_j^{min} is the minimum value of the attribute R_j in a time window of n samples and R_j^{max} is the maximum value of R_j in the same time window of n samples. Higher the resultant value h_{ij}, better is its contribution to the high quality of a cloud service. The normalized evidence matrix is:

$$M = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nm} \end{bmatrix} \tag{4}$$

4.1 Attributes-aggregation Based Calculation of Present Trust

For effective trust assessment of a cloud service, corresponding evidence factors need to be evaluated on the basis of their utility and inter-relationship. Accordingly, evidence factors can be combined to find a trust value. A method is developed for trust estimation of a cloud service using the values of attributes, at a particular instant of time. Within a time window of size n, after every fixed time interval, evidence factors are retrieved and a trust value is calculated at a particular time instant.

Definition 2. *Trust value of a cloud service (s_l), at a time instant i, termed as Present Trust (PT) is devised as an aggregation of corresponding all m evidence factors. It is given as:*

$$PT^i(s_l) = \sum\nolimits_{j=1}^{m} w_j h_{ij} \tag{5}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i, w_j is a weight assigned to it such that $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$.

Subjectively or manually assigned weights are static in nature. Trust assessment techniques based on such weights are not suitable for effective assessment of trust in cloud environment. Thus, weights need to adapt to changes in cloud service behavior [24] and hence computation of weights is crucial for adaptive trust estimation.

4.2 Computation of Weights

Weight designated to an attribute highlights the significance of the attribute in trust calculation. Weight of an attribute is computed based on values of that attribute at changing time instances. This results in adaptive weight assignment which is sensitive to the changes in the values of the attribute over a time period while the cloud service is running. This leads to an adaptive trust assessment. Degree of variation of each attribute within a time window is estimated for deciding weight of that attribute.

Definition 3. *Variation factor of an attribute R_j is formulated as given below.*

$$V(R_j) = \sum_{i=1}^{n} (h_{ij} - A(R_j))^2 \tag{6}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i.

$A(R_j)$ is the average of n evidence factors of attribute R_j in a time window containing n samples, shown as:

$$A(R_j) = (\sum_{i=1}^{n} h_{ij})/n \tag{7}$$

Effect of $V(R_j)$ on weight calculation is defined in terms of impact of variation factor (F_j) as shown below.

Definition 4. *Impact of variation factor (F_j) for attribute R_j is devised as:*

$$F_j = 1/(V(R_j) + (1/n)) \tag{8}$$

where $(1/n)$ is a negligible positive term which ensures a finite value for (F_j), especially in the situation when $V(R_j)$ is zero. However, such situation is rare in practice.

From Equation (8), less is the variation factor $V(R_j)$ of the attribute, higher is its impact (F_j) on the weight of the attribute R_j.
Weight w_j of an attribute R_j is computed as given below.

$$w_j = F_j / \sum_{k=1}^{m} F_k \tag{9}$$

where $0 < w_j < 1$ and $\sum_{j=1}^{m} w_j = 1$. Thus, weights computed using Equation (9) when substituted in Equation (5), result in adaptive trust estimation. Less is the variation factor of the attribute, higher is its resultant weight. This implies an effective trust assessment of a cloud service over a long duration from the perspective of a user. Weight computation of the attribute is also dependent on the variation factors of all other attributes. Weight of an attribute is higher if combined effect of variation factors of all other attributes is higher.

Theorem 1. *Weight assigned to any attribute R_j of a cloud service, is directly proportional to the combined effect of variation factors of all $(m-1)$ attributes other than R_j of the cloud service.*

Proof. Substituting F_j and F_k from Equation (8) in Equation (9), weight of any attribute R_j is:

$$w_j = \frac{[1/(V(R_j)+(1/n))]}{\sum_{k=1}^{m}[1/(V(R_k)+(1/n))]} \tag{10}$$

where $1 \le j \le m$. Substituting $V(R_j)$ and $V(R_k)$ from Equation (6) in Equation (10),

$$w_j = \frac{[1/(\sum_{i=1}^{n}(h_{ij}-A(R_j))^2+(1/n))]}{\sum_{k=1}^{m}[1/(\sum_{i=1}^{n}(h_{ik}-A(R_k))^2+(1/n))]} \tag{11}$$

Upon simplification, w_j becomes:

$$w_j = \frac{\prod_p [V(R_p)+(1/n)]}{[term1]+[term2]+...+[termm]} \tag{12}$$

where $1 \le p \le m$, $p \ne j$ and

$term1 = (V(R_2)+\frac{1}{n})(V(R_3)+\frac{1}{n})...(V(R_m)+\frac{1}{n})$

$term2 = (V(R_1)+\frac{1}{n})(V(R_3)+\frac{1}{n})...(V(R_m)+\frac{1}{n})$

$termm = (V(R_1)+\frac{1}{n})(V(R_2)+\frac{1}{n})...(V(R_{m-1})+\frac{1}{n})$.

In Equation (12), for any weight w_j, where $1 \le j \le m$, the denominator on the right-hand side, is the same. Hence, for any attribute R_j weight w_j is directly proportional to the product of $(m-1)$ terms in the numerator where each term indicates variation factor of corresponding attribute other than R_j. This proves the theorem. □

Thus, as indicated by Theorem 1, computation of weight for a cloud service attribute takes into consideration correlation of the attribute with all other attributes of the service. This achieves balancing effect while weighing the service attributes and subsequently aggregating them to compute a present trust of the cloud service.

4.3 Prediction of Cumulative Trust

A set of present trust (PT) values of a cloud service computed at different time instances forms a time series. From Equation (5), at time instant n, time series (TS) is:

$$TS = \{PT^1(s_l), PT^2(s_l), ..., PT^n(s_l)\} \tag{13}$$

This set serves as a basis to predict the future value of trust termed as cumulative trust. Prediction of cumulative trust is essential to assess the future quality of a cloud service for the duration of its execution. Hence, a method is developed for dynamic prediction of cumulative trust over a period of time.

Definition 5. *Cumulative Trust (CT) of a cloud service (s_l) over a period of time, predicted at a time instant n is defined as:*

$$CT^n(s_l) = \alpha PT^n(s_l) + (1-\alpha)CT^{n-1}(s_l) \tag{14}$$

where $PT^n(s_l)$ is PT of a cloud service (s_l) at n^{th} time instant as defined by Equation (5) and $CT^{n-1}(s_l)$ is a cumulative trust of a cloud service (s_l) at time instant $(n-1)$. $CT^{n-1}(s_l)$ is subsequently substituted repeatedly in Equation (14), with initial value $CT^1(s_l) = PT^1(s_l)$. α is a smoothing factor such that $0 < \alpha < 1$.

In consequent expansion of Equation (14), weights assigned to PT values, decrease exponentially from the most recent PT value to the PT values at earlier time instances. At the same time, it achieves uniform effect in the prediction of cumulative trust, by tuning of smoothing factor α. We recommend that the smoothing factor α should be tuned to a value in the range from 0.1 to 0.4. This enables predicted cumulative trust to match closely with the actual trust value.

5 Algorithm for Trust Estimation

Algorithm 1 shows the steps for adaptive and dynamic trust estimation of a cloud service. The trust assessments are performed during the interaction between a user and the service, based on service evidence factors. The algorithm takes a set of cloud service attributes and a number of time instances as input for trust computation of a service. Threshold trust (TH) taken as another input, is the minimum expected trust by the user. The algorithm takes historical trust value and user ratings as input for trust initialization of a cloud service. The algorithm gives the output as sets of present trust and cumulative trust values for service s_l over the period of interaction. Historical trust value (HS) indicates the cumulative trust value of the cloud service saved as a result of the last interaction of the user with the service. The steps of Algorithm 1 are explained as below.

Step 1. (Line 4) At the start of the user interaction with the service, trust is initialized as follows:

i) If the user has interacted with the service before, the trust is initialized to a value as indicated by *HS*.

ii) Otherwise initial trust is set based on the interactions of other users with the service. Here, the ratings for the service, given by other users are used for trust initialization. Computation of reputation score using beta probability density function is described by Josang *et al.* [12]. Here, in Algorithm 1, the same notion of beta

Algorithm 1 Trust Estimation for cloud service s_l

```
 1: Input:
      a. Set of $m$ cloud service attributes,
         $(AC) = \{R_1, R_2, ..., R_m\}$
      b. Number of time instances $(n)$
      c. Threshold trust $(TH)$
         // minimum expected trust value
      d. Historical trust value $(HS)$
      e. User ratings $(ER_l) = \{p_l, n_l\}$
         // number of positive and negative ratings
         // for service $s_l$
 2: Output:
      a. Set of Present Trust values for service $s_l$,
         $LP = \{PT[1], PT[2], ..., PT[n]\}$
      b. Set of Cumulative Trust values for service $s_l$,
         $LC = \{CT[1], CT[2], ..., CT[n]\}$
 3: Begin
 4: $IT$ = Trust_initiate($s_l$,HS,$ER_l$);
         // Trust initialization for service $s_l$
 5: if $IT \geq TH$ then
 6:   Matrix C = Get_evidences($s_l$,AC,n);
 7:   Matrix M = Normalize_evidences(C,AC);
          // Function $f_{NE}$ in Definition 1
 8:   Set W = Compute_weights(M,AC,n);
          // W is a set of weights of $m$ attributes,
          // computed by Algorithm 2,
          // $W = \{w_1, w_2, ..., w_m\}$
 9:   $i = 1$;
10:   while $i \neq n$ do
11:     Compute Present Trust of service $s_l$ at
        time instant $i$ as: $PT[i] = \sum_{j=1}^{m} w_j h_{ij}$;
            // Function $f_{PT}$ in Definition 1
            // From Algorithm 2, $w_j$ is a weight
            // and $h_{ij}$ is an element of matrix $M$
12:     Add $PT[i]$ in set $LP$;
13:     if $i = 1$ then
14:       $CT[i] = PT[i]$;
15:     else
16:       Compute Cumulative Trust of service $s_l$
          as: $CT[i] = \alpha PT[i] + (1 - \alpha) CT[i-1]$;
          // Function $f_{CT}$ in Definition 1
          // $\alpha$ is a smoothing factor, $0 < \alpha < 1$
17:     end if
18:     Add $CT[i]$ in set $LC$;
19:     if $CT[i] < TH$ then
20:       Notify incident;
21:     end if
22:     Update_trust($s_l$,$CT[i]$);
23:     $i = i + 1$;
24:   end while
25: end if
26: End
```

probability density function is used to compute the Initial Trust (IT) and is given by:

$$IT = \frac{p_l + 1}{p_l + n_l + 2} \tag{15}$$

where p_l and n_l are number of positive and negative ratings about the service s_l as shown by the set ER_l. The individual rating submitted by the user is in the range $[0, 1]$. A rating greater than or equal to 0.5 is considered as positive rating and a rating less than 0.5 is treated as a negative rating.

iii) When ratings about the service are not available, then IT becomes equal to 0.5 as deduced by Equation (15).

If the initial trust satisfies the threshold trust requirement, then the algorithm proceeds with the next steps.

Step 2. (Line 6) Trust estimator gets the evidence factors of the cloud service and gives the resultant evidence matrix C as shown in Equation (1).

Step 3. (Line 7) Normalization function takes the evidence matrix as input and transforms values of all the attributes in the matrix to uniform range as specified by Equations (2) and (3). It results into the normalized evidence matrix M as depicted by Equation (4).

Step 4. (Line 8) Here, the algorithm invokes the function to perform adaptive computation of weights for attributes of the cloud service. The details of the function to compute weights are given by Algorithm 2 in Section 5.1.

Step 5. (Lines 9 - 24) At each instant of time, adaptive computation of present trust is performed using the normalized evidence factors and the weights of service attributes. Subsequently, cumulative trust, representing the dynamic trust prediction of a service, is computed. The output sets of present trust and cumulative trust values are populated with the corresponding computed trust values. At any time, if cumulative trust falls below TH, then the user is notified about the incident. At each time instant, the HS of the service is revised by the latest computed cumulative trust value. Computations of present trust and cumulative trust values are described in Section 4 with elaboration.

5.1 Algorithm for Computation of Weights

Algorithm 2 describes the steps for adaptive computation of weights for attributes of the cloud service. The algorithm takes a normalized evidence matrix, a set of cloud service attributes and a number of time instances as input. The algorithm in turn, gives the set of weights for the attributes of a cloud service as the output. As shown in the algorithm, average of the evidence factors, variation factor and the impact of variation factor are computed for each attribute. From the values of impact of variation factor, weight of each attribute is computed. The computed weight of each attribute is added to the output set of weights. The details of computation of adaptive weights are presented in Section 4.2.

Algorithm 2 Computation of weights for the attributes of a cloud service: *Compute_weights(M,AC,n)*

```
Input:
    a. Matrix $M = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} \\ h_{21} & h_{22} & \dots & h_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{nm} \end{bmatrix}$
        // Normalized evidence matrix
    b. Set of $m$ cloud service attributes,
       $(AC) = \{R_1, R_2, ..., R_m\}$
    c. Number of time instances $(n)$
Output: Set of weights of m attributes,
        $W = \{w_1, w_2, ..., w_m\}$
Begin
$sum = 0$;
foreach $R_j \in AC$ do
    Compute average of evidence factors as:
        $A(R_j) = (\sum_{i=1}^{n} h_{ij})/n$;
    Calculate variation factor as:
        $V(R_j) = \sum_{i=1}^{n} (h_{ij} - A(R_j))^2$;
    Calculate impact of variation factor as:
        $F[j] = 1/(V(R_j) + (1/n))$;
    $sum = sum + F[j]$;
end
$j = 1$;
while $j \neq m$ do
    Compute weight of an attribute $R_j$ as:
        $w_j = F[j]/sum$;
        // Function $f_{CW}$ in Definition 1
    Add $w_j$ in set $W$;
    $j = j + 1$;
end while
End
```

5.2 Computational Complexity of Trust Estimation

The details of computational complexity of Algorithm 1 are described as follows. The computational complexity of a function to get the evidence factors is $O(mn)$. The computational complexity of normalization operation is $O(mn)$. Computation of present trust has the complexity of $O(nm)$. The computational complexity of cumulative trust prediction is $O(n)$. Computational complexity of a function to update trust is $O(n)$. As shown by the steps in Algorithm 2, computation of weights has the complexity of $O(mn^2)$. Therefore, the overall computational complexity (CC) of trust estimation (including Algorithm 1 and Algorithm 2) is given as below:

$$CC = O(mn) + O(n) + O(mn^2) = O(mn^2) \qquad (16)$$

Thus, the overall computational complexity of trust estimation depends on the number of cloud service attributes (m) and the number of time instances (n).

6 Performance Evaluation

For the evaluation of trust estimation model EBTEM described in Section 4, a prototype is developed in Java. The general and the most relevant attributes of a cloud service, as discussed in Section 2, which include availability, throughput, response time and security, are used during the experimentation. These attributes are important as they reflect the ability of a cloud service to perform the various operations effectively. For the values of throughput (kbps) and response time (seconds), real world QoS data set [30] is referred. The availability implies the percentage of time the cloud service is accessible. Security attribute is considered as the percentage of the number of violation incidents related to authentication or authorization. Weibull distribution is the suitable theoretical distribution for modeling failure time and can also be employed for modeling inputs in the absence of real data [14]. Hence, values of availability (%) and security violation incidents (%) are generated using the Weibull distribution. Various values of the attributes are normalized in the range [0.01, 0.99]. For a cloud service, higher values of availability and throughput are desired. Hence, values of these attributes are normalized using Equation (2). Whereas, lower values of response time and security violation incidents are expected. Hence, values of these attributes are normalized using Equation (3).

6.1 Trust Models for Comparison

In addition to our trust model EBTEM, two other trust models have also been implemented for comparative assessment. They are, averaging based simple trust model (ASTM) and weighted summation based trust model (WSTM). Both, ASTM and WSTM make use of multiple cloud QoS attributes for trust evaluation. These trust models are selected to compare the performance of EBTEM from two perspectives: i) To compare with the model where weights assigned to the various factors for trust assessment are static and subjective in nature. Thus, ASTM represents this trust model where equal weights are assigned to all the factors. ASTM is analogous to the model proposed by Manuel [17]. ii) To compare with the other model where dynamic weights are assigned to all the factors for trust assessment. WSTM represents this trust model and it corresponds to the model proposed by Li *et al.* [15]. The trust models [15, 17] are discussed in Section 2.

In ASTM, present trust (PT) of a cloud service (s_l) is computed as an average of all m evidence factors at a time instant i. It is given by:

$$AT^i(s_l) = (\sum_{j=1}^{m} h_{ij})/m \tag{17}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i. Cumulative Trust (CT), at time instant n, is calculated as an average of PT values, given as:

$$T^n(s_l) = (\sum_{i=1}^{n} AT^i(s_l))/n \tag{18}$$

where $AT^i(s_l)$ is a PT at time instant i.

In WSTM, PT of a cloud service (s_l) is computed as a weighted summation of all m evidence factors at time instant i. It is given by:

$$DT^i(s_l) = \sum_{j=1}^{m} w_j^{'} h_{ij} \tag{19}$$

where h_{ij} is a normalized evidence factor of attribute R_j at time instant i and $w_j^{'}$ is a weight assigned to it such that $0 < w_j^{'} < 1$ and $\sum_{j=1}^{m} w_j^{'} = 1$. Here, weights are computed using an approach of information entropy based weights, proposed by Li *et al.* [15]. CT at time instant n, is given by:

$$IT^n(s_l) = \sum_{i=1}^{n} w_i^{''} DT^i(s_l) \tag{20}$$

where $DT^i(s_l)$ is a PT of a cloud service (s_l) at time instant i and $w_i^{''}$ is a weight assigned to it such that $0 < w_i^{''} < 1$ and $\sum_{i=1}^{n} w_i^{''} = 1$. Here, weights are computed using the approach proposed by Li *et al.* [15] for assigning weight to each Real-Time Trust Degree (RTD). Here decreasing weights are assigned from latest RTD to RTDs at previous time instances.

6.2 Evaluation Metrics

The effectiveness of trust assessment method depends on the accuracy of trust estimation. Mean Absolute Error (MAE) [3] is a metric to assess an error in the prediction process. Here it is devised to compute an error in the prediction of cumulative trust and thus, to analyze the accuracy of trust estimation. Consequently, MAE is formulated as below.

$$MAE = \frac{1}{n}(\sum_{i=1}^{n} |P^{i+1}(s_l) - C^i(s_l)|) \tag{21}$$

where $P^{i+1}(s_l)$ is present trust of a cloud service (s_l) at time instant $(i+1)$, $C^i(s_l)$ is a predicted cumulative trust of a cloud service (s_l) at time instant i and n is the total number of time instances for assessment of MAE. Smaller value of MAE indicates higher accuracy of trust estimation and hence better performance of the trust model.

MAE as an absolute error based measure is complemented by a measure based on relative percentage error to analyze accuracy of trust estimation. Symmetric Mean Absolute Percentage Error (SMAPE) [13] is such a measure of error. SMAPE is formulated as below.

$$SMAPE = \frac{(\sum_{i=1}^{n} |C^i(s_l) - P^{i+1}(s_l)|) \times 100}{\sum_{i=1}^{n}(P^{i+1}(s_l) + C^i(s_l))} \tag{22}$$

where $P^{i+1}(s_l)$ is present trust of a cloud service (s_l) at time instant $(i+1)$, $C^i(s_l)$ is a predicted cumulative trust of a cloud service (s_l) at time instant i and n is the total number of time instances for assessment of SMAPE. From Equation (22), it enables to evaluate unbiased assessment of error in the prediction process. Similar to MAE, a smaller value of SMAPE signifies that predicted trust values closely match with actual trust values, leading to better accuracy of trust estimation.

Along with the accuracy in the calculation, trust model should be able to accomplish the task of rapid trust assessment. This is essential to service the upcoming requests for trust assessment efficiently. Hence, Mean Execution Time (MET) is used as a metric to evaluate the computational efficiency of trust assessment.

6.3 Results and Analysis

The performance of our trust model EBTEM is evaluated in terms of the accuracy and computational efficiency of trust estimation.

6.3.1 Assessment of Accuracy

The comparative evaluation of accuracy of trust estimation is performed during three sets of experiments, by observing MAE and SMAPE values of three trust models. The number of evidence samples is varied from 50 to 500, in all the sets of experiments. The comparisons of error and accuracy are made in all the sets of experiments, at a mid-point of sample counts, which is 250. The results show less error and better accuracy for EBTEM as against the ASTM and WSTM. The details are explained below. For other sample counts, the results may vary within marginal limits. However, the consistency of better accuracy of EBTEM remains more or less the same, in comparison with ASTM and WSTM, for any number of attributes of a cloud service.

In the first set of experiments, evidence samples for the two attributes which include throughput and response time, are considered. The results in Figure 3 illustrate that, the MAE of EBTEM is 0.035 whereas for ASTM it is 0.144 and for WSTM, it is 0.166. Thus, MAE of ASTM is 4.11 times the one in EBTEM and that of WSTM is 4.74 times the one in EBTEM. This implies that the accuracy of EBTEM is higher than ASTM by 11.3% and is higher than WSTM by 13.6%. As seen in Figure 4, the SMAPE of EBTEM is 1.85 whereas for ASTM it is 9.06 and for WSTM it is 10.85. Thus, SMAPE of ASTM is 4.9 times the one in EBTEM and that of WSTM is 5.86

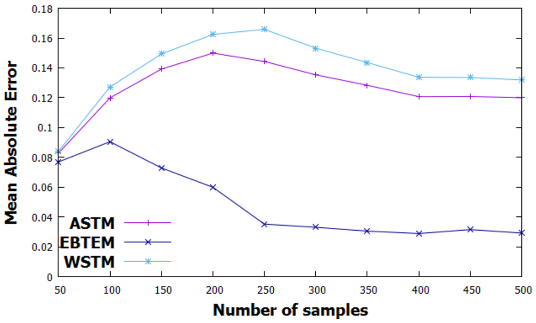

Figure 3: MAE with two cloud service attributes

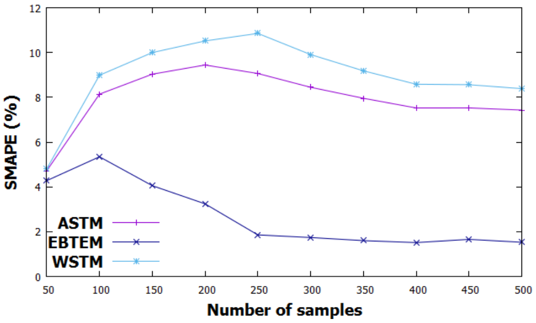

Figure 4: SMAPE with two cloud service attributes

times the one in EBTEM. This signifies that the accuracy of EBTEM is higher than ASTM by 7.35% and is higher than WSTM by 9.17%. Thus, the results in Figures 3 and 4 signify that, the performance of EBTEM is better than that of the other models for various number of samples of two attributes.

For the second set of experiments, evidence samples for the three attributes which cover availability, throughput and response time, are taken. The results in Figure 5 show that, the MAE of EBTEM is 0.041 whereas for ASTM it is 0.163 and for WSTM, it is 0.165. Thus, MAE of ASTM is 3.98 times the one in EBTEM and that of WSTM is 4.02 times the one in EBTEM. This implies that accuracy of EBTEM is higher than ASTM by 12.72% and is higher than WSTM by 12.93%. From Figure 6, the SMAPE of EBTEM is 2.24 whereas for ASTM it is 10.39 and for WSTM it is 10.51. Thus, SMAPE of ASTM is 4.64 times the one in EBTEM and that of WSTM is 4.69 times the one in EBTEM. This signifies that accuracy of EBTEM is higher than ASTM by 8.34% and is higher than WSTM by 8.46%. Thus, the results in Figures 5 and 6 depict that, with the evidences of three cloud service attributes as well, the performance of EBTEM is better than that of the other models for various number of samples.

The third set of experiments makes use of evidence samples for the four attributes which incorporate availability, throughput, response time and security violation incidents. The results in Figure 7 show that, the MAE of EBTEM is 0.045 whereas for ASTM it is 0.126 and for

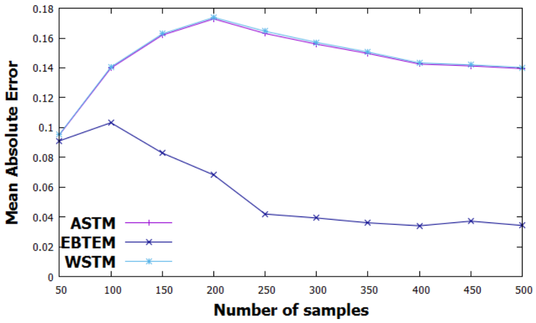

Figure 5: MAE with three cloud service attributes

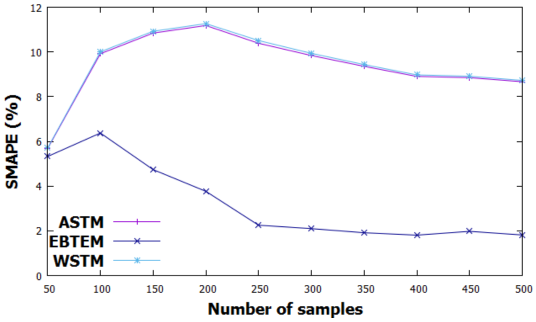

Figure 6: SMAPE with three cloud service attributes

WSTM, it is 0.127. Thus, MAE of ASTM is 2.8 times the one in EBTEM and that of WSTM is 2.82 times the one in EBTEM. This implies that the accuracy of EBTEM is higher than ASTM by 8.48% and is higher than WSTM by 8.59%. From Figure 8, the SMAPE of EBTEM is 2.53 whereas for ASTM it is 8.43 and for WSTM it is 8.46. Thus, SMAPE of ASTM is 3.33 times the one in EBTEM and that of WSTM is 3.34 times the one in EBTEM. This signifies that accuracy of EBTEM is higher than ASTM by 6.05% and is higher than WSTM by 6.08%. Thus, the results in Figures 7 and 8 demonstrate that, with the evidences of four cloud service attributes as well, the performance of EBTEM is better than that of the other models for various number of samples.

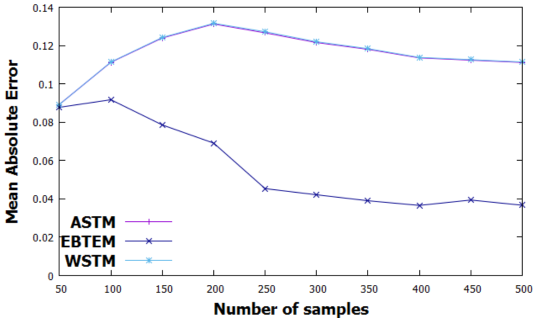

Figure 7: MAE with four cloud service attributes

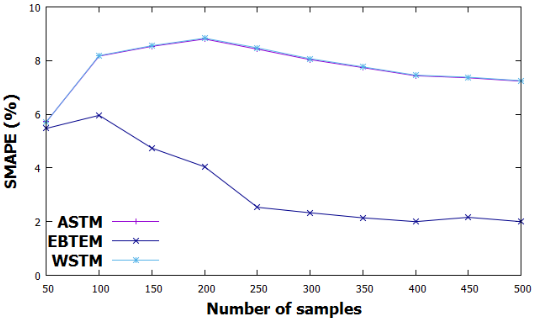

Figure 8: SMAPE with four cloud service attributes

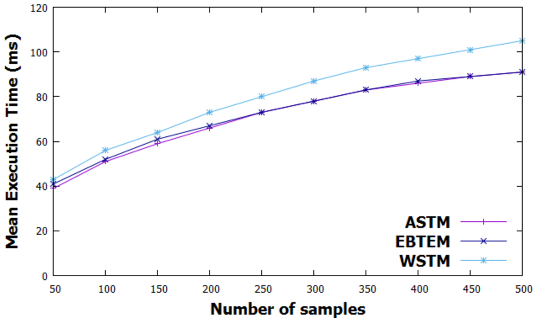

Figure 10: MET with three cloud service attributes

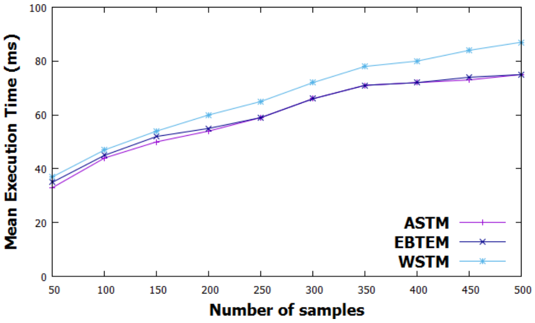

Figure 9: MET with two cloud service attributes

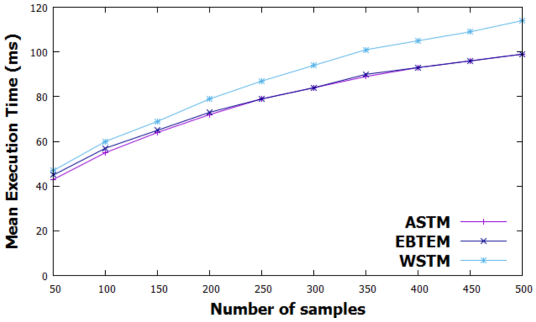

Figure 11: MET with four cloud service attributes

The results in Figure 3 to Figure 8 show that, with the increase in the number of samples, MAE and SMAPE values of EBTEM are decreased. Moreover, even with a smaller number of samples, EBTEM depicts higher accuracy and hence signifies the better performance. This makes EBTEM more suitable for cloud-based deployments in practice where, with limited number of samples, trust estimation can be performed efficiently.

6.3.2 Assessment of Computational Efficiency

The comparative assessment of efficiency of trust computation is performed with a set of three experiments, by observing MET values of three trust models. Experimentation is carried out using a computer with Intel i7-3537, 2.00 GHz CPU and 8 GB RAM. The number of evidence samples is varied from 50 to 500, in all the experiments. Figure 9 shows the results of the first experiment, where evidence samples for the two attributes which include throughput and response time, are considered. Figure 10 depicts the MET values of the second experiment, where evidence samples for the three attributes which cover availability, throughput and response time, are taken. Figure 11 demonstrates the results of the third experiment, where evidence samples for the four attributes which incorporate availability, throughput, response time and security violation incidents, are used.

Results in Figures 9, 10 and 11 show that, initially when the number of samples is reasonably small, observed MET values of the three models are nearby to each other. For instance, from Figure 9, at a sample count of 150, MET of EBTEM is 52, for ASTM it is 50 and for WSTM it is 54. When the number of samples becomes large, MET values of EBTEM and that of ASTM increase along smooth curves and follow the same trend in which the values of MET closely match with each other. On the other hand, MET values for WSTM increase and deviate as compared to the other two models. For example, from Figure 10, at a sample count of 450, MET value reaches to 89 for both, EBTEM as well as ASTM and for WSTM it is 101. Similarly, for instance, from Figure 11, at a sample count of 500, MET value is 99, for both, EBTEM as well as ASTM and for WSTM it is 114. Thus, it can be noted from Figures 9, 10 and 11, that ASTM has the lowest MET values and MET values for EBTEM almost match with those of ASTM. WSTM has comparatively greater MET values. In a nutshell, EBTEM and ASTM demonstrate better computational efficiency.

Although, ASTM coincides with EBTEM along the dimension of efficiency, accuracy of EBTEM is the highest and that of ASTM is much low. This is indicated by the results in Figure 3 to Figure 8. Hence, our trust model EBTEM exhibits much better performance. For the purpose of experimentation and performance analysis, we used the sets of two, three and four cloud service attributes in trust estimation. However, as elaborated in Section 4, the methodology of trust model enables to take into account multiple cloud service attributes for trust assessment.

7 Conclusions

In this paper, we presented the evidence based trust estimation model (EBTEM) and the algorithm for trust estimation of a cloud service. Adaptive and dynamic trust assessments are the main facets offered by the model. The evidence factors of a cloud service are aggregated to compute its present trust by assigning adaptive weights to the attributes of the service. The model enables continuous trust assessment of a cloud service according to the real-time changing values of the service attributes. Thus, the cloud user can decide whether to continue using the cloud service based on cumulative trust indicated by the model.

Experimental results have shown that average Mean Absolute Error (MAE) of averaging based simple trust model (ASTM) and weighted summation based trust model (WSTM) are respectively 3.63 and 3.86 times that of EBTEM. Hence, based on MAE, the average accuracy of EBTEM is higher than ASTM by 10.83% and by 11.71% than WSTM. Also, average Symmetric Mean Absolute Percentage Error (SMAPE) of ASTM and WSTM are respectively 4.29 and 4.63 times that of EBTEM. Hence, with regard to SMAPE, the average accuracy of EBTEM is higher than ASTM by 7.25% and by 7.9% than WSTM. Results have also demonstrated that, even with the limited number of samples, EBTEM achieves higher accuracy. Thus, results have shown that performance of EBTEM is much better than that of other models for the dimensions of accuracy and computational efficiency of trust assessment. In conclusion, our trust model EBTEM depicts effective trust estimation of a cloud service. As future work, we aim to extend our trust model by taking into account Quality of Service (QoS) related requirements of the user for assessment of trust.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments. Sincere thanks are also due to College of Engineering Pune and Cummins College of Engineering for Women, Pune, India.

References

[1] J. Abawajy, "Establishing trust in hybrid cloud computing environments," in *10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pp. 118–125, 2011.

[2] I. M. Abbadi and A. Martin, "Trust in the cloud," *Information Security Technical Report*, vol. 16, no. 3, pp. 108–114, 2011.

[3] C. Chatfield, *Time-series Forecasting*, CRC Press, 2000.

[4] W. J. Fan, S. L. Yang, H. Perros, and J. Pei, "A multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach," *International Journal of Automation and Computing*, vol. 12, no. 2, pp. 208–219, 2015.

[5] N. Ghosh, S. K. Ghosh, and S. K. Das, "Selcsp: A framework to facilitate selection of cloud service providers," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 66–79, 2015.

[6] S. M. Habib, S. Hauke, S. Ries, and M. Mühlhäuser, "Trust as a facilitator in cloud computing: A survey," *Journal of Cloud Computing*, vol. 1, no. 1, pp. 1–18, 2012.

[7] S. M. Habib, S. Ries, and M. Mühlhäuser, "Cloud computing landscape and research challenges regarding trust and reputation," in *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC'10)*, pp. 410–415, 2010.

[8] S. M. Habib, S. Ries, and M. Mühlhäuser, "Towards a trust management system for cloud computing," in *10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*, pp. 933–939, 2011.

[9] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing.," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[10] J. Huang and D. M. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing*, vol. 2, no. 1, pp. 1–14, 2013.

[11] Y. Huo, Y. Zhuang, and S. Ni, "Fuzzy trust evaluation based on consistency intensity for cloud services," *Kybernetes*, vol. 44, no. 1, pp. 7–24, 2015.

[12] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[13] V. Kreinovich, H. T. Nguyen, and R. Ouncharoen, "How to estimate forecasting quality: A system-motivated derivation of symmetric mean absolute percentage error (smape) and other similar characteristics," 2014.

[14] A. M. Law, *Simulation modeling and analysis*, McGraw-Hill, 2007.

[15] X. Li, H. Ma, F. Zhou, and X. Gui, "Service operator-aware trust scheme for resource matchmaking across multiple clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1419–1429, 2015.

[16] Z. Li, L. Liao, H. Leung, B. Li, and C. Li, "Evaluating the credibility of cloud services," *Computers & Electrical Engineering*, 2016.

[17] P. Manuel, "A trust model of cloud computing based on quality of service," *Annals of Operations Research*, pp. 1–12, 2013.

[18] P. D. Manuel, M. I. Abd-El Barr, and S. T. Selvi, "A novel trust management system for cloud computing iaas providers," *Journal of Combinatorial Mathematicsand Combinatorial Computing*, vol. 79, p. 3, 2011.

[19] P. D. Manuel, S. T. Selvi, and M. I. Abd-El Barr, "Trust management system for grid and cloud resources," in *First International Conference on Advanced Computing (ICAC'09)*, pp. 176–181, 2009.

[20] A. Mosa, H. M. El-Bakry, S. M. Abd El-Razek, S. Q. Hasan, "A proposed E-government framework based on cloud service architecture," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 93–104, 2016.

[21] F. Moyano, K. Beckers, and C. Fernandez-Gago, "Trust-aware decision-making methodology for cloud sourcing," in *26th International Conference on Advanced Information Systems Engineering (CAiSE'14)*, pp. 136–149, 2014.

[22] T. H. Noor and Q. Z. Sheng, "Credibility-based trust management for services in cloud environments," in *9th International Conference on Service-Oriented Computing (ICSOC'11)*, pp. 328–343, 2011.

[23] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *12th International Conference on Web Information System Engineering (WISE'11)*, pp. 314–321, 2011.

[24] T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust management of services in cloud environments: Obstacles and solutions," *ACM Computing Surveys*, vol. 46, no. 1, pp. 12, 2013.

[25] D. Patel, "Accountability in cloud computing by means of chain of trust dipen contractor," *International Journal of Network Security*, vol. 19, no. 2, pp. 251-259, 2017.

[26] P. S. Pawar, M. Rajarajan, S. K. Nair, and A. Zisman, "Trust model for optimized cloud services," in *Trust Management VI*, pp. 97–112, 2012.

[27] C. Qu and R. Buyya, "A cloud trust evaluation system using hierarchical fuzzy inference system for service selection," in *28th International Conference on Advanced Information Networking and Applications (AINA'14)*, pp. 850–857, 2014.

[28] J. Sidhu and S. Singh, "Improved topsis method based trust evaluation framework for determining trustworthiness of cloud service providers," *Journal of Grid Computing*, pp. 1–25, 2016.

[29] W. L. Tai, Y. F. Chang, "Comments on a secure authentication scheme for IoT and cloud servers," *International Journal of Network Security*, vol. 19, no. 4, pp. 648-651, 2017.

[30] Y. Zhang, Z. Zheng, and M. R. Lyu, "Wspred: A time-aware personalized qos prediction framework for web services," in *22nd International Symposium on Software Reliability Engineering (ISSRE'11)*, pp. 210–219, 2011.

Biography

Shilpa Deshpande is a research scholar at College of Engineering Pune, Savitribai Phule Pune University, India. Her research interests are in the area of cloud computing and distributed systems. She is currently an Assistant Professor with the Computer Engineering Department, Cummins College of Engineering for Women, Pune. She has received the Bachelor's degree and the Master's degree in Computer Engineering from Savitribai Phule Pune University, in 1996 and in 2002 respectively.

Rajesh Ingle is an Adjunct Professor at Department of Computer Engineering, College of Engineering Pune, India. He is a Professor at Department of Computer Engineering, Pune Institute of Computer Technology, Pune. He has received Ph.D. Computer Science and Engineering from Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai, Mumbai. His research interests include Distributed system security and Cloud security. He has received the B.E. Computer Engineering from Pune Institute of Computer Technology and M.E. Computer Engineering from Government College of Engineering, Savitribai Phule Pune University. He has also received M.S. Software Systems from BITS, Pilani, India. He is a senior member of the IEEE, IEEE Communications Society and IEEE Computer Society.