Cluster Comput (2013) 16:961–977
DOI 10.1007/s10586-012-0232-9

# A framework for SLA management in cloud computing for informed decision making

**Adil Hammadi · Omar Khadeer Hussain · Tharam Dillon · Farookh Khadeer Hussain**

Received: 2 July 2012 / Accepted: 3 October 2012 / Published online: 13 December 2012

**Abstract** In cloud computing, service providers offer cost-effective and on-demand IT services to service users on the basis of Service Level Agreements (SLAs). However the effective management of SLAs in cloud computing is essential for the service users to ensure that they achieve the desired outcomes from the formed service. In this paper, we introduce a SLA management framework that will enable service users to select the best available service provider on the basis of its reputation and then monitor the run time performance of the service provider to determine whether or not it will fulfill its promise defined in the SLA. Such analysis will assist the service user to make an informed decision about the continuation of service with the service provider.

**Keywords** Service level agreement · Reputation · Transactional risk · Performance risk · Financial risk

A. Hammadi · O.K. Hussain (✉)
School of Information Systems, Curtin Business School, Perth, Australia
e-mail: O.Hussain@cbs.curtin.edu.au

A. Hammadi
e-mail: adil.hammadi@postgrad.curtin.edu.au

T. Dillon
Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia

F.K. Hussain
Decision Support and e-Service Intelligence Lab, Quantum Computation and Intelligent Systems, School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
e-mail: Farookh.Hussain@uts.edu.au

## 1 Introduction

Big data, due to its huge volume and variety, is considered as the next frontier for innovation, competition and productivity [1]. Such data which is generated from numerous activities has resulted in a huge growth of information in the digital universe, and is a valuable asset for businesses, enterprises and users. However, before such data can be utilized by the various users for their benefit, appropriate data analytic tools are needed that assist them to understand the vast variety and volume of big data in real-time and synthesize meaningful knowledge from it. Big data analytics are the new generation of technologies designed to economically extract value from large volumes and a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis [2]. *Business Intelligence* which involves performing analytics on the underlying data to synthesize actionable knowledge from it is one type of big data analytic techniques. Business Intelligence has various applications in different domains. In this paper, we focus on one such application of Business Intelligence, namely making informed service-based decisions in the domain of cloud computing.

Cloud computing is defined as *a parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements* [3]. By using cloud computing architecture, users can access massive computing resources for short time periods without having to build their own infrastructure. This is achieved in the form of services. Also, by using the cloud computing platform, service providers will be able to provide cost effective and on-demand IT services to the multiple-service users on a multi-tenancy basis for their various needs. Cloud computing architecture works on the vision of the services (required by the service users

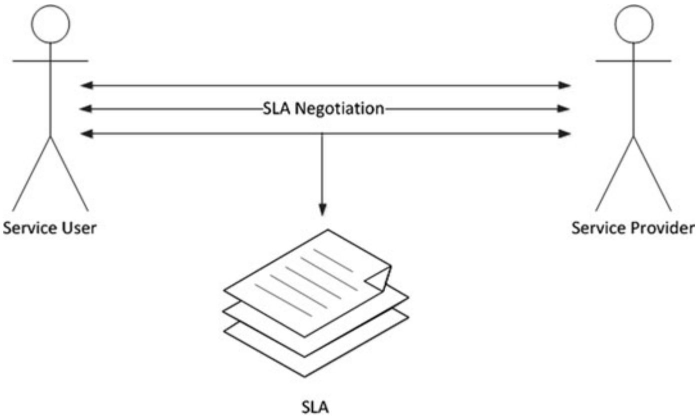

**Fig. 1** SLA negotiation and formation

and provided by the service providers) being delivered as promised at the required capacity, time and need. However, the two main challenges that need to be addressed in order to ensure that this vision is achieved are:

(a) the service user must select a capable service provider (from the possible ones) from whom s/he can achieve the desired outcomes;
(b) the service provider has to ensure that s/he has the capability and ample resources to deliver on the agreed service to the service user. This will ensure in the achievement of financial gains to the service providers as an outcome of the service.

To guarantee Quality of Service (QoS), Service Level Agreements (SLAs) are established between a service provider and a service user. SLA is a formal contract and contains terms and conditions to which both the service provider and service user agree. The establishment of SLAs benefits both providers and users. For a service provider, the SLA provides performance metrics to be committed to, and for a service user, it provides assurance of QoS. SLAs are formed between a service provider and a service user through an iterative process of negotiations [4] as shown in Fig. 1. SLAs can be broken down into various smaller criteria called Service Level Objectives (SLOs). The SLOs describe in detail the QoS properties for the agreed-upon service. In other words, the SLOs also establish a threshold against which a service is expected to be delivered. The performance of the service provider is measured against this threshold.

### 1.1 Problem definition

Even though a SLA defines the agreed QoS properties to which the service provider will commit, there may be scenarios where it may not meet those properties (and hence the SLA) due to various factors such as incapability to meet the service, service degradation, service outages etc. Service degradation is defined as the reduction in the quality of the service, due to any event. This may have the potential to lead to an outage in the service. Service degradation and/or service outage may have an impact on both the service users and service providers in relation to the business outcomes to be achieved and may result in a loss of investment. For example, outages which occurred at Amazon Simple Storage Service (S3), AppEngine and Gmail in 2008 caused significant losses to service users [5]. In order to avoid such undesirable outcomes, *SLA management* is important to ensure that the run-time service properties meets the criteria that are established in the agreement. However, a framework for SLA management should be a two-step process; the first being the establishment of the SLA with a service provider who is capable of providing the required service, and the second being the monitoring or predicting of the performance of the service on delivery to ensure that the quality of service is according to the defined parameters as stated in the SLA. In the literature, SLA monitoring techniques are mentioned, however, none of these techniques considers SLA monitoring as a two-step conjoint process as discussed here. In this paper, we propose an approach for SLA management by which a service user is able to make an informed decision about the selection and continuity of the service with a service provider.

## 2 Literature review

In this section, we explore the work done in the area of SLA formation, negotiation and QoS evaluation, based on SLA for Service-Oriented Computing and Grid Computing. He et al. [6] discussed the SLA formation and negotiation process. The entire SLA process can be accomplished using agreement protocols such as a WS-Agreement where a service requester submits his/her request through a web portal. The client submits its requirement to a Business Rule Consultant which activates the agreement. The service provider's service templates are already available through a service registry. The Business Rule Consultant maps the user requirements to available templates of service providers. After completing this phase, negotiation starts between negotiator agents. If negotiation is successful, a final agreement between agents is reached through an agreement factory. After this, to enforce the agreed SLA, monitoring begins. At this stage, for reasons of credibility, the authors propose the commissioning of a third-party agent to monitor the SLA. Quillinan et al. [7] mention that the Quality of Service (QoS) provided by a service provider can be assessed on the basis of the SLA. When it comes to monitoring the SLA, penalties can be imposed as a result of violation by the service provider. Online monitoring is one way of monitoring service violations. Online monitoring means that the continuous monitoring of service is done based on monitoring intervals. These monitoring intervals depend upon agreement terms and can be in seconds, minutes, hours or days. A second monitoring approach discussed is reactive monitoring, whereas the third monitoring approach is offline monitoring. In terms of penalties for service violations, two types are defined: reputation-based penalty and monetary-based penalty. Khader et al. [8] discuss the reactive monitoring technique for SLA. After discussing online and offline monitoring, they mention that the reactive monitoring approach balances the trade-off between online and offline monitoring. Oliveira et al. [9] provide a benchmark to monitor SLA by using a grid system. They argue that since most of the monitoring techniques are embedded in a particular SLA specification, they rely on specific protocols which impede the wide-spread adoption of grid infrastructures. They introduced a benchmark called Jawari to monitor the SLA. Chang et al. [10] propose a measurement methodology for Quality of Service assessment. According to Chang et al., one way to obtain QoS assessment is by determining the trustworthiness of the service provider. By using a combination of their proposed metrics, the trustworthiness value of a service provider can be determined. In an extension of this work, Schmidt et al. [11] used these metrics to determine the reputation of a service provider. After each business interaction, the credibility of the service provider is adjusted which results in either an increase or decrease of its reputation. If the quality of service provided falls short of the promised service quality, the reputation of the agent is significantly decreased. This work, even though is significant, applies only to QoS assessment after the initiation of the service. However, sometimes the service user needs QoS assurance *before* the start of a business interaction. Hussain et al. [12] address pre-interaction QoS assessment using risk assessment techniques, whereby a service user determines the risk before any interaction with the service provider begins. The level of risk is determined by considering past trustworthiness values obtained through past experience with the service provider or through the recommendations obtained by third-party agents.

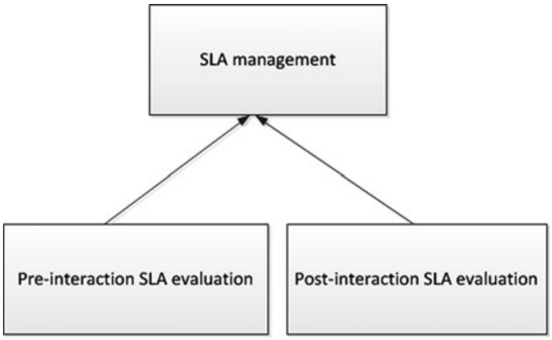
SLA management
Pre-interaction SLA evaluation
Post-interaction SLA evaluation

**Fig. 2** SLA management framework

Even though a great amount of work has been done in the literature to ensure QoS based on the SLA, most of the work places an emphasis on either the pre- or post- service initiation phase. However, as discussed in the earlier section, in order to have a framework for SLA management, a conjoint assessment and consideration of the analysis of both these phases is required to make an informed decision. Such an approach is needed for a service user to make an informed decision of service selection and the continuation of the service with a service provider. In the next section, we propose such an approach for SLA management in cloud computing.

## 3 Proposed framework for SLA management in Cloud Computing

To address the problem raised in Sect. 1.1, in this section, we propose an SLA management framework comprising two parts: (a) pre-interaction SLA evaluation; and (b) post-interaction SLA evaluation, as shown in Fig. 2. Pre-interaction SLA evaluation assists the service user to select the service provider who has the best capability to commit to the established SLAs, whereas post-interaction SLA evaluation assists the service user to monitor the performance of the chosen service provider to ensure that the delivered service meets the service levels defined in the SLOs in different periods of time.

We term the total time over which SLA management has to be carried out as the ‘time space’. In other words, time space is defined as that total period of time which the service user takes into consideration to ascertain the QoS while interacting with the service provider. However, as the capability of the service provider to deliver a service with the required level of quality varies over a period of time, in order for SLA management to be meaningful for both the time parts defined in Fig. 2, we divide the interaction time period into two different non-overlapping parts namely, ‘pre-interaction’ and ‘post-interaction’ time phases [12] as shown in Fig. 3.

The pre-interaction time phase is the portion of time before the start of service between the service user and service

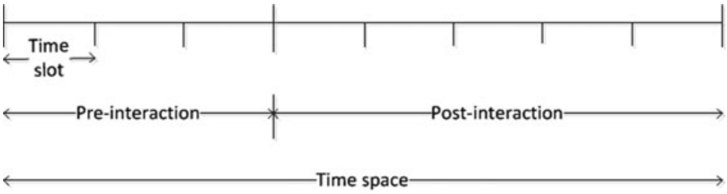

**Fig. 3** Time space and related concepts

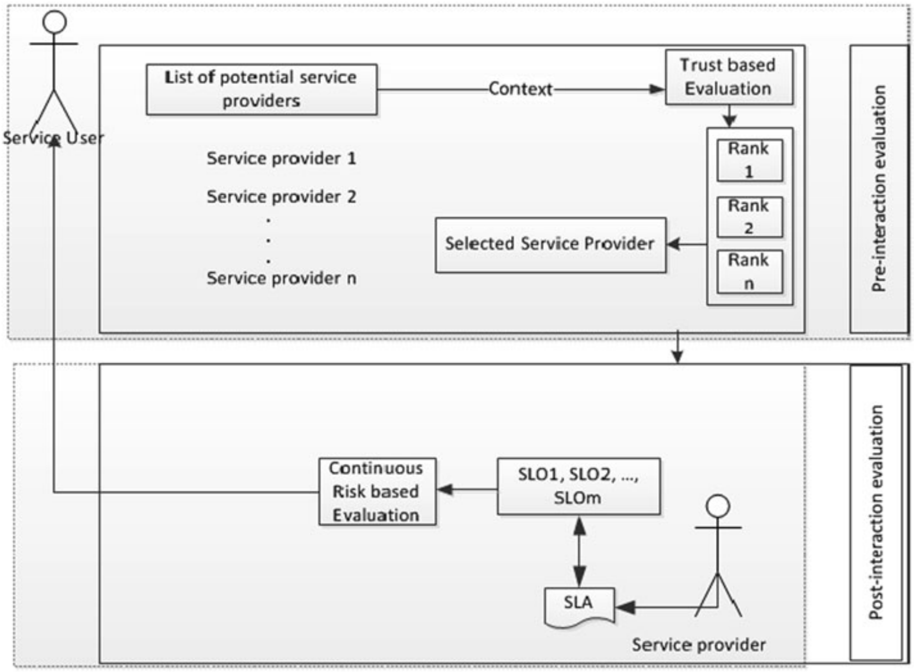

**Fig. 4** Conceptual framework for QoS assessment in cloud computing

provider and the post-interaction time phase is the portion of time after the start of service between the service user and service provider. 'Time slot' is defined as a non-overlapping interval of time within the time space of the interaction. The time slot is obtained by dividing the time space into different equal, non-overlapping parts of time. QoS evaluation is done in each time slot. In the pre-interaction phase it is done by determining the reputation of the service provider in being capable of and committed to complying with the terms of the established SLAs; whereas, in the post-interaction phase, it is done by determining the transactional risk as a result of the service being provided by the service provider not committing to the defined metrics of the SLAs. In order to implement the solution in a pragmatic setting, we lay out the architectural foundation to make the solution scalable and modular. Modularity has been achieved by introducing a multi-layered architecture, as shown in Fig. 4.

In pre-interaction evaluation step of Fig. 4 for pre-interaction evaluation, there is a *trust-based evaluation* service which evaluates service providers who have the capability of fulfilling service user's request. The output of this evaluation is the selection of a service provider who has highest trust value. This output is then passed on to post-interaction evaluation step which then evaluates or predicts the performance of selected service provider against the formed SLAs by using the *continuous risk-based evaluation* service. To monitor SLA evaluation in pre- and post-interaction time phases in automatic fashion, we introduce a third-party service provider called Third-Party Service Level Agreement Monitor (TP SLA Monitor) who is responsible for executing a *trust-based evaluation* service and *continuous risk-based evaluation* service. A trust-based evaluation service consists of a *reputation assessment module* and a continuous risk-based evaluation service consists of a *risk assessment module*, as depicted in Figs. 5a and 5b, respectively.

To start the monitoring process, the TP SLA monitor should first establish the total time space for which QoS has to be assessed. It then divides this time space into pre-interaction and post-interaction time phases. QoS evaluation in the pre-interaction time phase is achieved by using a reputation assessment methodology by soliciting the recommendations from *Recommending Users* (RU). RUs in our framework are a group of users who receive a reputation query from TP SLA monitor and they reply on the basis of their past experiences with the service provider. Each RU stores past experiences in the form of trust values assigned to a service provider in an information repository [10] as depicted

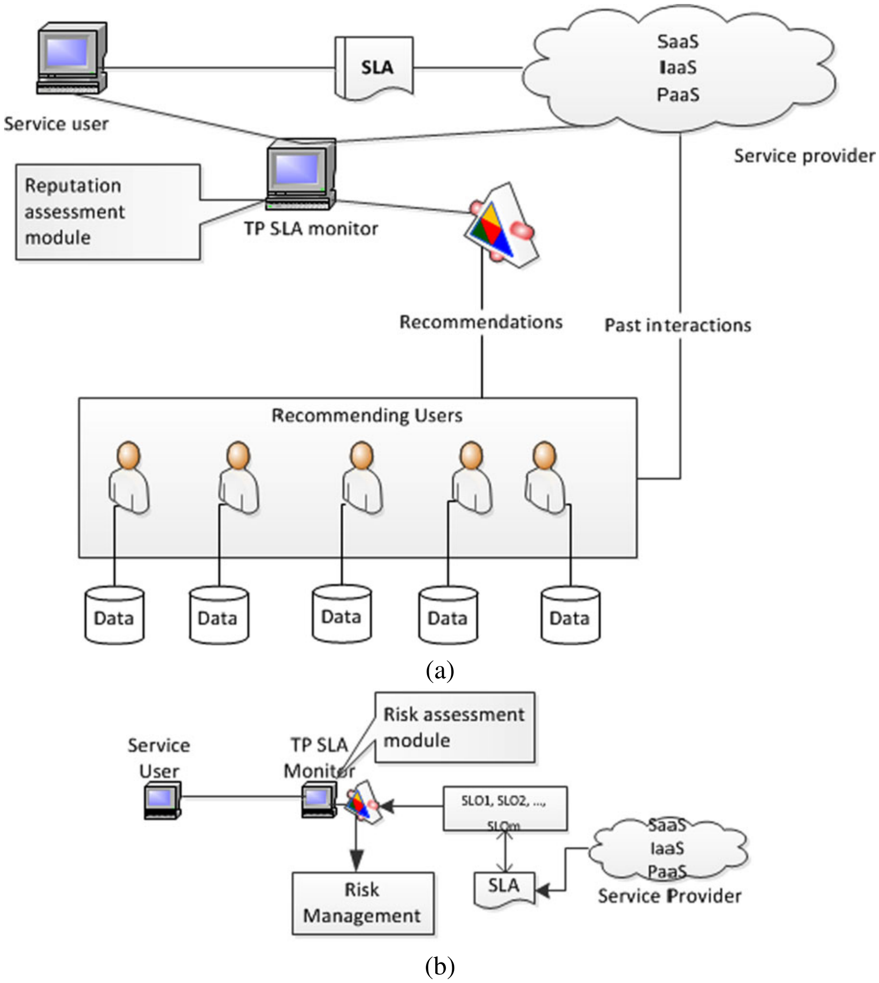

**Fig. 5** (a) Framework for pre-interaction SLA evaluation. (b) Framework for post-interaction SLA evaluation

in Fig. 5a. In addition, TP SLA Monitor stores the credibility of each RU in providing opinions about a service provider in its information repository.

The TP SLA Monitor determines the reputation of the service provider before the start of a transaction considering the context and criteria of the SLA parameters to measure performance. For example, if the context of a transaction is bandwidth, initially, it is useful to determine the reputation of the service provider to deliver a promised bandwidth. Reputation is a general perception of a service provider's ability to provide a committed service which is useful for a cloud service user before it starts a transaction with a service provider. After the service user decides to enter into a service with the service provider, the TP SLA Monitor can assess or predict the performance of the service provider at run-time by evaluating its commitment to the defined SLOs, as shown in Fig. 5b.

For assessment of a service's performance at run-time, as proposed by [13], in our framework we consider that the TP SLA Monitor obtains the run-time SLA parameter of the service through the service provider's measurement interface when it is in the current time slot. For example, to evaluate the performance of the service provider offering a bandwidth service, the TP SLA Monitor needs to obtain the run-time value of the bandwidth and compare this with the bandwidth threshold given in the SLA. As a result of a comparison between the SLA threshold of a parameter and its run-time value, the TP SLA monitor determines the transactional risk. Transactional risk ascertains the subcategories of *performance risk* and *financial risk*. Performance risk represents the level of non-commitment of the service provider in the SLOs to the threshold values defined in the SLAs, whereas financial risk represents the financial impact to the service user as a result of this. In this paper, we utilize these two subcategories to ascertain the transactional risk. Such analysis can then be utilized by the service user to make a decision on the continuation of the service. In the next sections, we propose our approach for pre- and post-interaction SLA evaluation.

## 4 Pre-interaction SLA evaluation

The steps in this phase are as follows:

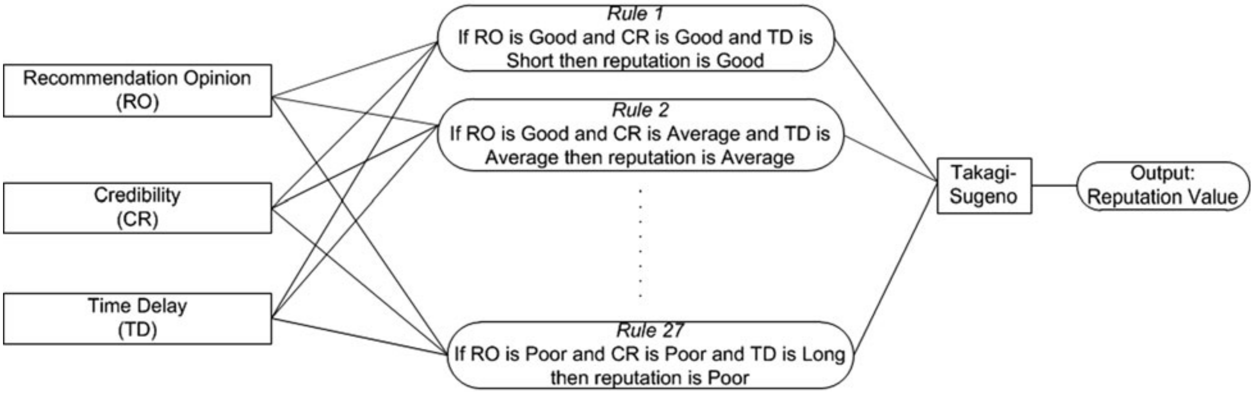
Recommendation Opinion (RO)
Credibility (CR)
Time Delay (TD)
Rule 1
If RO is Good and CR is Good and TD is Short then reputation is Good
Rule 2
If RO is Good and CR is Average and TD is Average then reputation is Average
Rule 27
If RO is Poor and CR is Poor and TD is Long then reputation is Poor
Takagi-Sugeno
Output: Reputation Value

**Fig. 6** T-S based Fuzzy Inference System

1. The TP SLA Monitor sends a reputation query soliciting about a service provider's reputation to RUs.
2. For those users with whom the TP SLA monitor has solicited recommendations in the past, it stores their credibility values in its information repository. Credibility here is defined as the trustworthiness of the opinion of an RU. In our proposed solution, credibility is represented on the scale of [0, 5] where 0 represents the lowest (poor) credibility of the RU and 5 represents the highest (good) credibility of the RU.
3. Only those RUs which have interacted with the service provider in the same context and criteria in which the service user wants to interact with the service provider reply to the reputation query.
4. The reply from each RU contains the trust value assigned to the service provider on the basis of its past interaction with that service provider and the time in which this interaction had occurred [11]. When a RU shares this trust value with the TP SLA Monitor, this value is called recommendation opinion. It is represented on a scale of [0, 5] where 0 represents lowest (poor) recommendation opinion given by RU and 5 represents the highest (good) recommendation opinion of RU.
5. The time factor is important since a RU can have multiple interactions with the same service provider in the same context but in different time slots. The time factor can be represented by a time delay which indicates the time elapsed since the last interaction of a RU with the service provider and we represent time delay on a scale of [0, 5] where 0 represents the long time delay and 5 represents the short time delay.
6. For each reputation query response received from a RU, the TP SLA Monitor consolidates the credibility value, time delay value and the recommendation opinion. The outcome of this consolidation is the *reputation* of the service provider provided by a RU.
7. After consolidating these factors for each RU, the TP SLA Monitor then aggregates all reputations by all RUs involved in the reputation query and gives the final reputation value for a service provider.

In our approach we use a fuzzy logic-based approach to consolidate the recommendation opinion of the RU, the credibility of their opinion and the time delay after the last interaction to ascertain the reputation value of a service provider in a particular context. We discuss about that further in the next section.

## 4.1 Fuzzy logic-based approach

To build a fuzzy logic-based system to determine reputation, we use the factors affecting reputation, namely, recommendation opinion (RO), credibility (CR) and time delay (TD) as input, and the output is the reputation contribution $R_i$ of RU$i$. We need a fuzzy inference method to map input variables to the output value. Fuzzy linguistic control rules can be used for this mapping. For the fuzzy reasoning, we use the Takagi-Sugeno approach (T-S method) [14] which can be trained using the Adaptive-Neuro Fuzzy Inference System (ANFIS) algorithm. An example of input variables, linguistic control rules, fuzzy reasoning method and output value is given in Fig. 6.

One of the main advantages of the Takagi-Sugeno approach is the use of a tuning algorithm to achieve the best generalization capability of the fuzzy inference system, therefore we use the fuzzy inference system with the Adaptive-Neuro Fuzzy Inference System (ANFIS) algorithm. This approach is called a 'soft computing'-based approach, which is discussed in Sect. 4.2.

The fuzzy sets that we use for input variables RO and CR are [Poor, Average, Good] and TD variables are [Long, Average, Short]. All fuzzy variables are expressed on a scale of 0 to 5, as shown in Figs. 7 and 8 for input variables RO and TD, respectively.

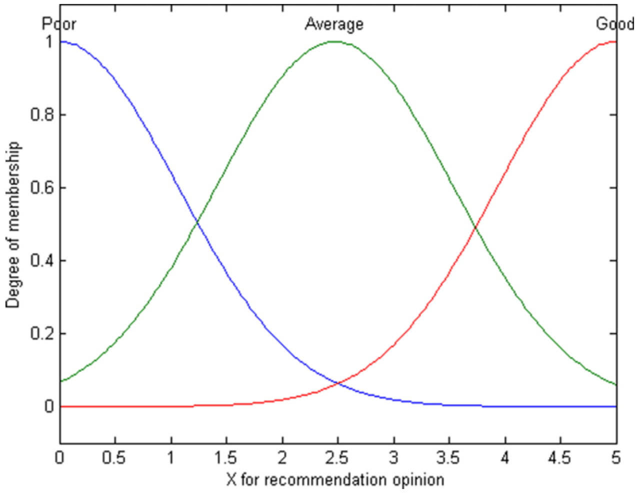

**Fig. 7** Membership functions for Recommendation Opinion

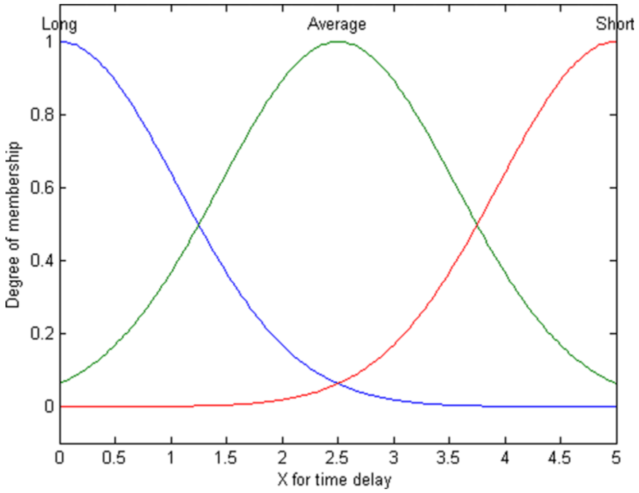

**Fig. 8** Membership functions for Time Delay

### 4.2 Takagi-Sugeno inference approach

The antecedent part of the Takagi-Sugeno rule system is the same as the Mamdani approach. The consequent part of it consists of a linear equation. So, a rule in the Sugeno system takes the form [14]:

$\text{IF}(x_1 \text{ is } X_1 \text{ AND } x_2 \text{ is } X_2, \dots, x_n \text{ is } X_n)$

$\text{THEN}(y_q = a_{q0} + a_{q1}x_1 + \cdots + a_{qn}x_n)$

where: $x_1$, $x_2$ are scalar inputs; $X_1$, $X_2$ are fuzzy sets; $a_{q0}, a_{q1}, \dots, a_{qn}$, are real numbers; and $y_q$ is the consequent of the rule.

We consider a system with $m$ fuzzy rules of the T-S form. The form of crisp output is

$$y(\underline{x}) = \frac{\sum_{q=1}^{m} \alpha_q (a_{q0} + \sum_{s=1}^{n} a_{qs}x_s)}{\sum_{q=1}^{m} \alpha_q} \tag{1}$$

In (1), $\alpha_q$ is the firing strength of rule $q$. The actual approach to fuzzy reasoning in this case has the following steps:

(a) fuzzify inputs
(b) obtain the firing strength $\alpha_q$ associated with each rule $q$
(c) obtain the output function of $y_q$ associated with each rule $q$ using the firing strength $\alpha_q$
(d) obtain the overall output $y(\underline{x})$ using expression (1) given above

We use the first-order Takagi-Sugeno (T-S) approach for fuzzy inference with a linear function for the right-hand side. The inputs on the left-hand side of the fuzzy rule will consist of the factors that affect reputation.

### 4.3 Working of the FIS for reputation

Using MATLAB's fuzzy logic toolbox, we performed the following four major steps for ANFIS training:

1. loaded training and monitoring datasets
2. created initial FIS
3. trained initial FIS
4. validated trained FIS

Before we detail the experiment process, we explain the source of the training data, what is input-output data, and provide an explanation of our dataset.

### 4.4 Obtaining training data

ANFIS uses a training data set to tune input and output fuzzy variables. Obtaining training data is a critical part of the whole process [15]. The first step is input selection. Input vectors should cover important aspects of the system without which optimum results cannot be achieved. The omission of any important input points may result in some output values that are either very large or negative. The second part is the values of output vectors in the training data set. The output vectors selected for the training data set must confirm to the rule base. We generated an artificial dataset containing values that span all possible scenarios for the considered inputs in a real-life situation. These values were processed on linguistic rules to generate reputation values (output).

Each tuple of our training dataset consists of an input vector of the form [RECOMMENDATION OPINION, CREDIBILITY, TIME DELAY] and an output value that represents Reputation. As an example, one tuple of our dataset is [3.6 3.7 3.2] with output 3.94. This tuple corresponds to the linguistic rule:

IF RO is Good AND CR is Good AND TD is Average

THEN REPUTATION is Good

We generated 588 such tuples encompassing every aspect of the reputation system, 97 of which we use for monitoring

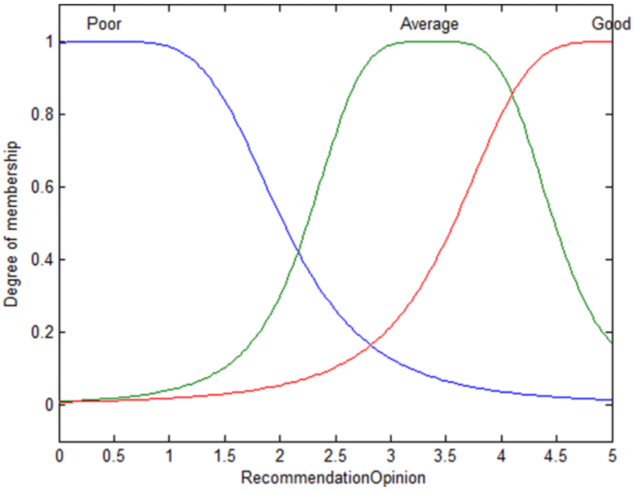

**Fig. 9** Membership Function for Recommendation Opinion

purposes, leaving 491 data points for training purposes. The monitoring data set was also used for cross-validation of the FIS structure alongside the training data set during the training.

### 4.5 Training process

The training and monitoring data sets are used with tuning algorithms. The actual tuning algorithm used was the adaptive-network-based fuzzy inference system (ANFIS) algorithm [16], since it tunes the parameters of both the input membership function as well as the output coefficients. For the studies carried out in this paper, the version of ANFIS implemented in the Fuzzy Logic Toolbox of MATLAB was used. Initially, we split the data set into three parts: (1) training set (2) monitoring set and (3) a generalization test set. In the training stage, we used the training set error to derive the adjustments in the parameters. This training set error is the difference between outputs provided by the model and the actual value for each instance. We used the monitoring set which is unseen by the training model, defining the output training to monitor the generalization capability of the model. Thus, the monitoring set error = (predicted value − actual value)$^2$ goes through to minimum before increasing, even though the training set error is still decreasing. This minimum corresponds to the best generalization capability of the model and training ceases when we achieve this. The final training error achieved was 0.2076 and the final monitoring error achieved was 0.2054.

### 4.6 Results obtained and discussion

Using experimental investigation of the fuzzy logic-based reputation measure, we investigate three fuzzy sets corresponding to the linguistic term set [Good, Average, Poor] to span the input space of RO and CR input factors and three

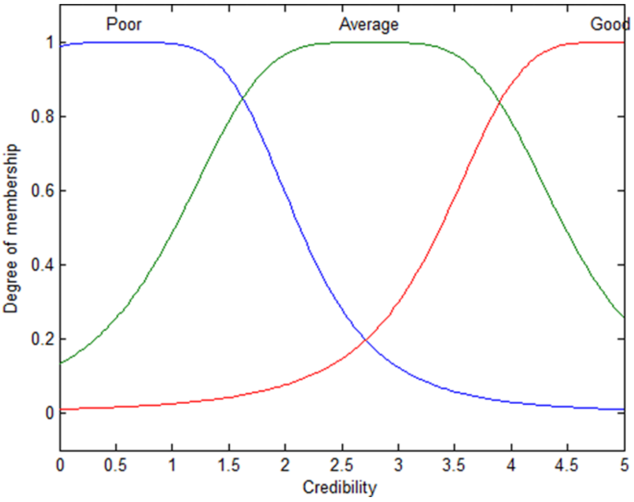

**Fig. 10** Membership Function for Credibility

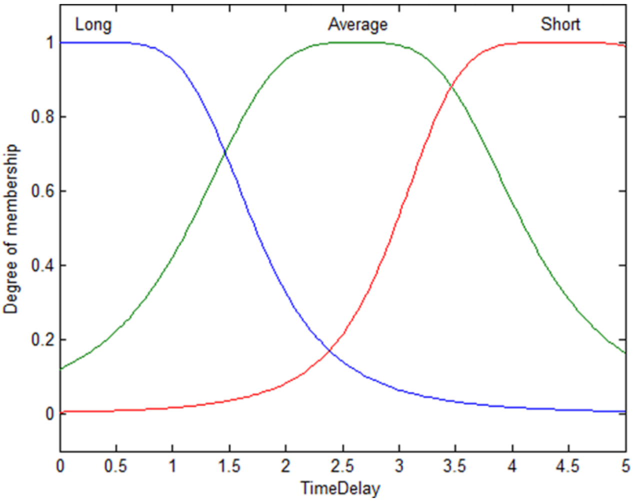

**Fig. 11** Membership Function for Time Delay

fuzzy sets corresponding to the linguistic term set [Long, Average, Short] to span the input space of TD input factor. We obtain the following curves for the input membership functions shown in Figs. 9, 10, 11:

Note that the ‘Poor’ membership function for the Recommendation Opinion and Credibility in Figs. 9 and 10 spans to the right which implies that the ‘Poor’ recommendation and credibility has a great influence on the reputation of the system. We also note that the ‘Average’ membership function in the Recommendation Opinion in Fig. 9 is skewed to the right which further emphasizes the influence of the ‘Poor’ data set in the Recommendation Opinion, whereas the ‘Average’ membership function in Credibility in Fig. 10 spans wide which implies that it has great influence on ascertaining the reputation. The ‘Short’ membership function in Fig. 11 spans left which implies that the time delay function is slow, indicating that Time Delay has less influence on

**Table 1** Input test cases derived from varying different inputs about two nominal points for 'poor' Reputation

| Test case number | Input vector | Reputation |
|---|---|---|
| 1 | [0.8 0.6 1] | 0.95 nominal pt 1 |
| 2 | [1.5 0.6 1] | 1.91 |
| 3 | [0.1 0.6 1] | 0.67 |
| 4 | [0.8 1.1 1] | 1.21 |
| 5 | [0.8 0.1 1] | 0.72 |
| 6 | [0.8 0.6 1.5] | 0.93 |
| 7 | [0.8 0.6 0.5] | 0.94 |
| 8 | [0.5 1 0.8] | 1.04 nominal pt 2 |
| 9 | [1 1 0.8] | 1.22 |
| 10 | [0.1 1 0.8] | 0.88 |
| 11 | [0.5 1.4 0.8] | 1.19 |
| 12 | [0.5 0.6 0.8] | 0.82 |
| 13 | [0.5 1 1.4] | 1.07 |
| 14 | [0.5 1 0.2] | 0.94 |

reputation when compared to other input variables namely, Recommendation Opinion and Credibility.

## 4.7 Model validity and sensitivity studies

In order to study the validity of the model developed, we used three different methods [17] as follows:

(1) input values corresponding to the training data were loaded and the output obtained from FIS was checked against the output value of the actual training data set;
(2) input values that were not in the original training set were run with this fuzzy model;
(3) sensitivity studies were conducted, varying each of the input factors in turn about a nominal point from the initial measured data.

The first two types of validations can be achieved using the ANFIS interface in MATLAB. In both cases, we obtained correct results within an acceptable tolerance. These validations indicate that our input data set includes all of the representative features of our reputation model. In the case of the second validation method, our model predicts output values as expected in response to input values that were not in the original training set. Discussion on the third validation technique, sensitivity analysis, is given in the next subsection.

### *4.7.1 Sensitivity study for fuzzy trust model*

Sensitivity analysis is a technique used to determine how different values of an independent variable will impact on a particular dependent variable [18]. The results of our sensitivity studies are given in Tables 1–3; the input vector consists of the following factors: [Recommendation Opinion, Credibility, Time Delay].

These results are obtained by perturbing each good, poor and average recommendation opinion and credibility in addition to each long, average and short time delay. The fuzzy trust model is then used to find the overall measure of reputation for each of these perturbed values. Let us initially consider all the results and then the results with respect to some of the input factors in turn.

Nearly all of the values generated for Reputation indicate movements in the correct direction, that is, an increase in input values leads to an increase in reputation and vice versa for a perturbation in a single input factor about the nominal point. The exceptions are indicated by an asterisk (*). However, even for these, while the movement with respect to nominal points might be slightly in the wrong direction (test case number 4 in Table 2), if two adjacent points corresponding to a change in the same input factor are considered, we notice one of these points indicates a movement in the correct direction. Let us consider the value of Reputation for poor Recommendation Opinion, poor Credibility and long Time Delay is approximately between 0 and 1.5, for average Recommendation Opinion, Credibility and Time Delay is between 1.6 and 3.5 and for good Recommendation Opinion, Credibility and short Time Delay is between 3.6 to 5. Almost all points obtained by the change in the input variable, except for test cases 4, 13, 14 in Table 2 and test cases 6, 11, 12 in Table 3, result in a change in Reputation in the expected direction. Hence, the Fuzzy Trust Model seems to adequately model the relationship between input factors and the overall Reputation. The influences of some of these input factors are discussed in the next section.

*Recommendation opinion* Let us consider the input factor, Recommendation Opinion. We notice the following:

(1) for poor Recommendation Opinion test cases 3, 9 and 10 (in Table 1) and 2 (in Table 2);

**Table 2** Input test cases derived from varying different inputs about two nominal points for 'average' Reputation

| Test case number | Input vector | Reputation |
|---|---|---|
| 1 | [2.3 2 2.2] | 2.13 nominal pt 1 |
| 2 | [1.9 2 2.2] | 1.42 |
| 3 | [2.7 2 2.2] | 2.84 |
| 4 | [2.3 1.6 2.2] | 2.15 * |
| 5 | [2.3 2.4 2.2] | 2.14 |
| 6 | [2.3 2.4 1.8] | 1.98 |
| 7 | [2.3 2.4 2.6] | 2.25 |
| 8 | [3 2.8 3] | 2.93 nominal pt 2 |
| 9 | [2.5 2.8 3] | 2.59 |
| 10 | [3.5 2.8 3] | 3.34 |
| 11 | [3 2.3 3] | 2.77 |
| 12 | [3 3.2 3] | 3.34 |
| 13 | [3 2.8 3.5] | 2.61 * |
| 14 | [3 2.8 2.5] | 3.37 * |

**Table 3** Input test cases derived from varying different inputs about two nominal points for 'good' Reputation

| Test case number | Input vector | Reputation |
|---|---|---|
| 1 | [3.6 3.7 3.2] | 3.94 nominal pt 1 |
| 2 | [4.1 3.7 3.2] | 4.12 |
| 3 | [3.1 3.7 3.2] | 3.56 |
| 4 | [3.6 3 3.2] | 3.42 |
| 5 | [3.6 4.2 3.2] | 4.05 |
| 6 | [3.6 3.7 2.8] | 4.22 * |
| 7 | [3.6 3.7 3.8] | 3.93 |
| 8 | [4.1 4.5 4.3] | 4.18 nominal pt 2 |
| 9 | [3.8 4.5 4.3] | 4.11 |
| 10 | [4.6 4.5 4.3] | 4.62 |
| 11 | [4.1 3.9 4.3] | 4.23 * |
| 12 | [4.1 4.9 4.3] | 3.92 * |
| 13 | [4.1 4.5 4.8] | 4.33 |
| 14 | [4.1 4.5 3.7] | 4.11 |

(2) for average Recommendation Opinion test cases 2 (in Table 1) and 3, 9 and 10 (in Table 2);
(3) for good Recommendation Opinion test cases 2, 3, 9 and 10 (in Table 3).

All indicate that an improvement in the Recommendation Opinion leads to an improvement in Reputation and a deterioration in the Recommendation Opinion leads to a deterioration in Reputation, which is the expected result. We notice that the first perturbation in the value of Recommendation Opinion in Table 1 takes the value of Reputation from poor to average. In test case 2 in Table 2, the perturbation in the value of Recommendation Opinion takes the value of Reputation from average to poor. This indicates that Recommendation Opinion has a significant influence on Reputation.

*Credibility* Let us next consider the input factor Credibility. We notice the following:

(1) for poor Credibility test cases 4, 5, 11 and 12 (in Table 1);
(2) for average Credibility test cases 4, 5, 11 and 12 (in Table 2) and 4 (in Table 3);
(3) for good Credibility test cases 5, 11 and 12 (in Table 3).

Almost all indicate that improvement in Credibility leads to improvement in Reputation and vice versa with the exception of a few test cases. These test cases are numbers 11 and 12 in Table 3 and number 4 in Table 2. However, the other test cases in the same tables indicate movement in the right direction. In nearly all cases, we notice that Credibility has a strong influence on Reputation for all subsets. In some cases, the influence is more pronounced (e.g. test case 4 in

Table 3). Hence Credibility, with Recommendation Opinion, is a significant indicator of Reputation.

*Time delay* Now we consider the input factor, Time Delay. We notice the following:

(1) for long Time Delay test cases 6, 7, 13 and 14 (in Table 1);
(2) for average Time Delay test cases 6, 7, 13 and 14 (in Table 2);
(3) for short Time Delay test cases 6, 7, 13 and 14 (in Table 3).

Almost all indicate movement in the right direction; that is, reducing time delay (higher value) increases Reputation and increasing time delay (lower value) decreases Reputation. The exceptions are test cases 13, 14 in Table 2 and 6 in Table 3. Although these test cases indicate movement in the wrong direction, all other points lead Reputation in the right direction.

Hence, we conclude that Credibility and Recommendation Opinion have a significant impact on Reputation followed by Time Delay. In other words, the effect of Time Delay is slowly changing which is consistent with our model. From the trained system, the TP SLA monitor can obtain $R_i$, reputation value given by one recommending user $i$. Reputation $R_k$ of the service provider $K$ is the aggregation of $n$ recommendations (reputation values) of all recommending users. It is given by:

$$R_K = \sum_{l=1}^{n} \frac{R_l}{n} \tag{2}$$

In the next section, we will demonstrate the working of the proposed FIS to determine the reputation of the service provider.

### 4.8 Example

To illustrate the working details of our proposed framework, let us consider the following hypothetical scenario. Consider a service user ‘A’ who wants to select a video conferencing service and Voice over IP (VoIP) service for his business needs. There are two service providers, namely service provider ‘B’ and service provider ‘C’ that closely match the business and operational requirements of service user ‘A’. In order to do the pre-screening of potential service provider candidates, service user ‘A’ requests reputation assessments from the TP SLA Monitor. On receiving the request, the TP SLA solicits recommendations from RUs for service providers ‘B’ and ‘C’. Suppose RU1 and RU2 send their recommendation about service provider ‘B’ and RU1 and RU3 send their recommendations about service provider ‘C’. Using the soft-computing-based approach introduced earlier, the TP SLA monitor computes the final reputation values of service providers ‘B’ and ‘C’. The process of selection is depicted in Fig. 12.

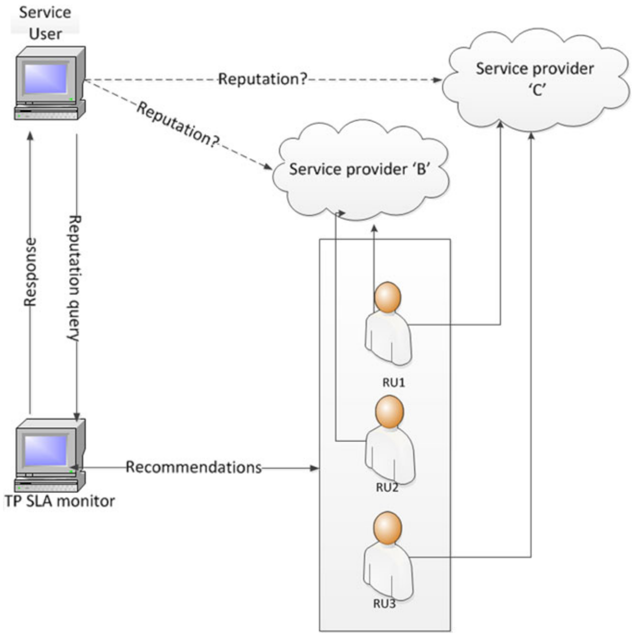

**Fig. 12** Service selection process

Based on RO, CR and TD values, the TP SLA monitor computes the reputation value provided by one RU. For example, using our reputation assessment methodology, the reputation values provided by RU1 and RU2 about service provider ‘B’ are 3.93 and 4.11, respectively. The reputation values provided by RU1 and RU3 for service provider ‘C’ are 4.12 and 3.56, respectively. Using Eq. (2), the TP SLA monitor computes the final reputation value of service provider ‘B’ as 4.02 and the final reputation value of service provider ‘C’ as 3.84. On the basis of the high reputation value, service user ‘A’ may want to enter into an interaction with the service provider ‘B’.

After determining the reputation of a service provider and selecting it for service provision, a service user now needs to ascertain that the service provider will deliver the promised service. This can be assessed by risk assessment in the post-interaction SLA evaluation methodology which is discussed in the next section.

## 5 Post-interaction SLA evaluation

As depicted in Fig. 4, once a service selection process is complete, then starts using the *continuous risk-based evaluation* service which is implemented by the TP SLA monitor as shown in Fig. 5b. As discussed in Sect. 2, there are several methods to evaluate a service provider’s performance after the initiation of the service in order to decide on the continuation of the service, one of these being transactional risk analysis. In this section, we discuss how we use transactional risk analysis for SLA evaluation and management in the post-interaction time-phase.

Risk assessment is that phase of risk analysis where a hazard or threat is identified, quantified. This then leads to the evaluation and management in the risk evaluation and risk management phases. In its simplest form, risk assessment starts with the identification of a hazardous event $E_i$, followed by determining the likelihood or probability $P_i$ of this undesirable event occurring [19]. The second phase of risk assessment is to determine the consequence $C_i$ as a result of this event occurring. This consequence $C_i$, is a measure of the impact of $E_i$. The overall risk $R$ then can be given as:

$$R = \sum_i P_i \cdot C_i \tag{3}$$

It should be noted that determination of $P_i$ is objective and depends upon past experience or historical data, whereas the determination of $C_i$ is subjective. In the literature, various techniques have been proposed for risk assessment. One such approach by which risk can be analyzed is convolution [20]. Convolution is the integral operator which expresses the amount of overlap and impact of one function as it shifts over the other. Convolution has been used in different applications such as power generation systems, to determine the expected demand not supplied by determining the effect of load demand on the generation capacity of the generation system. Mathematically, convolution of $X$ and $Y$, both of which are independent random variables, is given by:

$$Z = X \oplus Y \tag{4}$$

In our approach, we utilize the principles of convolution and use it to ascertain the transactional risk of the service provider not meeting the defined threshold levels in the SLAs.

### 5.1 Example

Continuing with the case study introduced in Sect. 4.8, once service user 'A' enters into a business interaction with service provider 'B', it is important for him to monitor whether service provider 'B' provides the desired level of service to achieve his desired outcomes. From user 'A's perspective, video conferencing and VoIP services are important as business meetings are arranged with his business customers through these services. Not being able to conduct business meetings may result in financial loss and damage the business reputation of service user 'A'.

Let us consider that service user 'A' on a particular day expects to secure a business contract of $15,000 through his video conferencing meetings. Since services such as VoIP and video conferencing depend on bandwidth, let us consider that service provider 'B' has the maximum capacity to provide a bandwidth of 24 Mbps. Further, let us consider that service provider 'B' establishes a threshold of 12 Mbps in SLA against which the bandwidth service is expected to be delivered. If a service provider 'B' delivers a bandwidth below 12 Mbps, these bandwidth levels are considered as service level degradation since they deviate from the threshold defined in the SLAs. It is important to measure the service deviation levels in order to measure the performance of service provider 'B'. Moreover, let us consider that the video conferencing service does not work if the bandwidth falls below 6 Mbps.

### 5.2 Determining the service deviation levels

Suppose we want to calculate the probability that the bandwidth falls below the threshold (12 Mbps) for the video conferencing service. Depending on the time phase in which the current time slot is, the TP SLA monitor can either use the past performance of the service provider or predict the performance of the service provider in the future time slots, based on its past performance history by using the techniques mentioned in Hussain et al. [21]. By using the techniques we determine the level of bandwidth delivery of the service provider in an advance time period of 24-hour duration. If a discrete random variable $Q$ represents the levels of bandwidth for a 24-hour interval, the Probability Mass Function (pmf) for this random variable is given by:

$$f : q \to P(Q = q) \quad \text{or} \quad f(q) = P(Q = q)$$

where variable $q$ indicates the values within the range of the random variable.

Relative frequency for each class interval is then calculated as:

$$P(Q = q) = \frac{Fr}{n} \tag{5}$$

where $Fr$ denotes the frequency of bandwidth on class intervals representing bandwidth and $n$ represents the total bandwidth. By definition of pmf:

$$P(Q) = \sum_i \frac{Fr_i}{n} = 1 \tag{6}$$

where $i$ represents the class intervals.

Using (5) for our case study, we obtain the graph shown in Fig. 13.

In Fig. 13, where $q < 12$ it indicates the chances of the service not meeting the required threshold. To determine the level of service degradability, we convert this portion of the distribution into another distribution that represents the service deviation levels of a bandwidth. If a discrete random

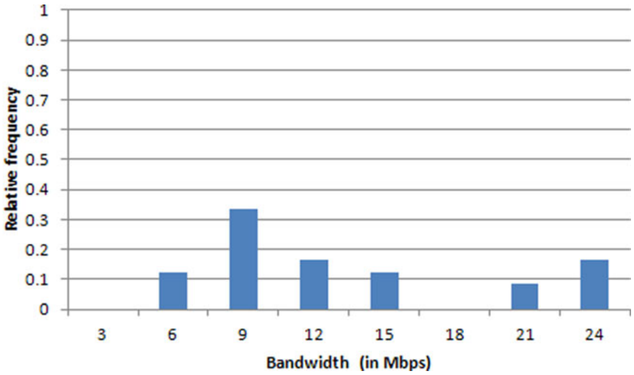

**Fig. 13** Probability Mass Function for bandwidth levels

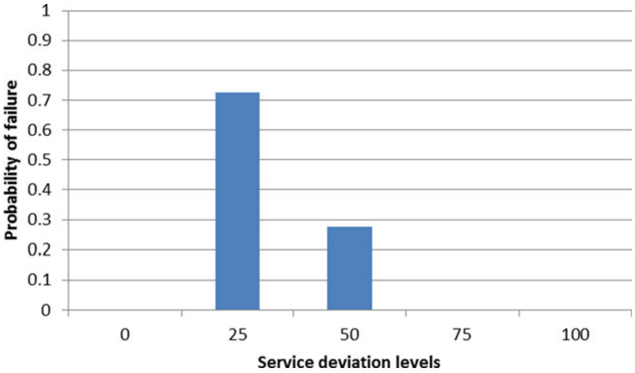

**Fig. 14** Level of service deviation from minimum threshold requirements

variable $R$ represents service deviation levels and a variable $r$ indicates the values within the range of the random variable, the probability of the level of service deviation is calculated as:

$$P(R = r) = \frac{P(q_i)}{\sum_{i=0}^{i<12} P(q_i)} \tag{7}$$

Consequently, we obtain a probability mass function for random variable $R$ given as:

$$P(R) = \sum_j P(r_j) = 1, \quad \text{where } j = 0, 3, \dots, 9 \tag{8}$$

By normalizing service deviation distribution to a percentage scale, as shown in Fig. 14, we obtain the different levels of deviation from the threshold point defined in the SLAs. The next step in determining the transactional risk is to ascertain the impact as a result of the deviation in the service.

### 5.3 Determining the financial loss as a result of service deviation levels

Service user 'A', through his investment of resources in service provider 'B', wants to benefit financially over a specific period of time. However, due to service degradation, the expected outcome may not be achieved which may lead to the service user experiencing a financial loss. In order to determine this, the first step in this process is to determine the expected financial gain which user 'A' wants to achieve over a period of time. The service user expects to achieve financial gain in each time interval if the service is delivered as promised by the service provider. From such expected financial gain in each time slot over a time period, we propose that the TP SLA monitor plots the Expected Financial Gain Curve (EFGC).

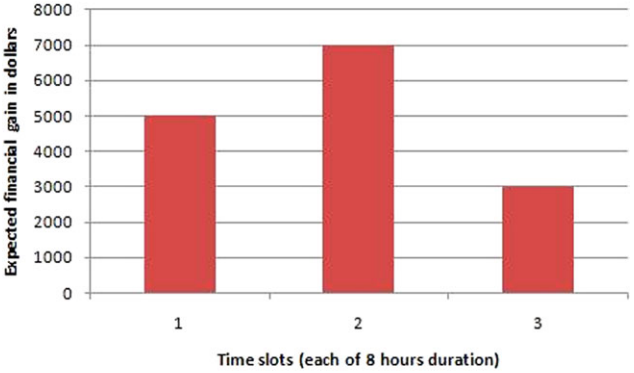

**Fig. 15** Expected financial gain in each time slot

#### 5.3.1 *Expected financial gain curve*

We use our case study to demonstrate the process of plotting the Expected Financial Gain Curve (EFGC). As discussed in our case study, service user 'A', over a period of 24 hours, expects to gain a benefit of $15,000 as a result of a service provided by service provider 'B'. In order to capture the dynamic nature of risk over this period, we consider the methodology proposed by Chang et al. [10], where the time space of the business interaction is established and then divided into different time slots. In this case study, the time space is 24 hours and is divided into 3 time slots of 8 hours each. In 24 hours, as a result of a video conference service, let us suppose that the service user expects to achieve a financial gain, as shown in Fig. 15.

In order to determine the impact of service deviation levels on the expected financial gain, we first need to determine the collective expected financial gain that was anticipated throughout the time period. This is achieved by plotting the cumulative probability density function of the amount of resources invested over the time period [22, 23]. The curve, called an Expected Financial Gain Curve (EFGC), is shown in Fig. 16.

To determine the financial loss, the impact of service deviation levels on the expected financial gain has to be determined. We achieve this by convolution. However, for convolution both the random variables need to be on a uniform scale. So the expected financial gain first has to be converted

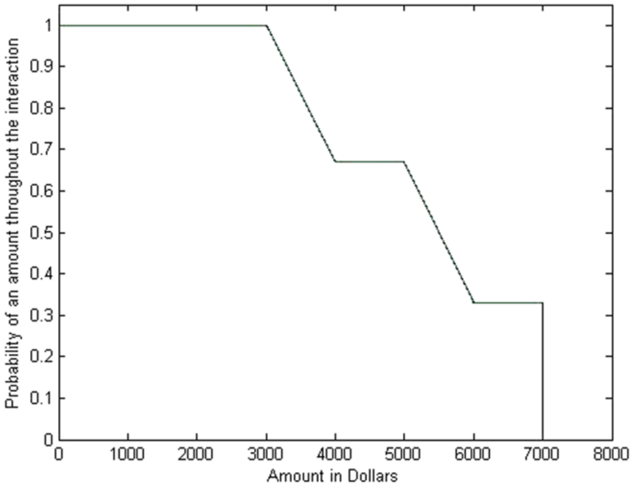

**Fig. 16** Expected Financial Gain Curve

to the percentage scale. Convolution results in a curve which we call the Actual Resource Investment Curve (ARIC). As opposed to the EFGC, the ARIC indicates the probability of an amount required to be kept at stake in order for service user 'A' to achieve the same outcome with service provider 'B'.

### 5.3.2 *Convolution process*

Using (4), convolution for our random variables can be written as:

$$ARIC = FL \oplus EFGC$$

where: $ARIC$ = Actual Resource Invested Curve, $FL$ = random variable indicating service deviation levels, $EFGC$ = Expected Financial Gain Curve.

In other words, convolution is the sliding of the $EFGC$ over the projections of service deviation levels. We use the following formulae to perform this operation:

$$ARIC(x) = \sum_{i=1}^{n} p_i * EFGC(x - FL_i) \quad \text{for } (x - FL_i) \geq 0$$

or

$$ARIC\,(x) = \sum_{i=1}^{n} p_i \quad \text{for } (x - FL_i) < 0 \tag{9}$$

where: $n$ = the number of service deviation levels, $x$ = the point at which $ARIC$ has to be determined, $FL_i$ = magnitude of service deviation level $i$, $p_i$ = magnitude of occurrence of service deviation level $i$, $EFGC(x - FLi)$ = Expected Financial Gain Curve at point $(x - FLi)$.

The effect of the service deviation levels $FL_i$ over each point of curve $(x)$ is determined and after the convolution process, the $ARIC$ curve is produced, as shown in Fig. 17.

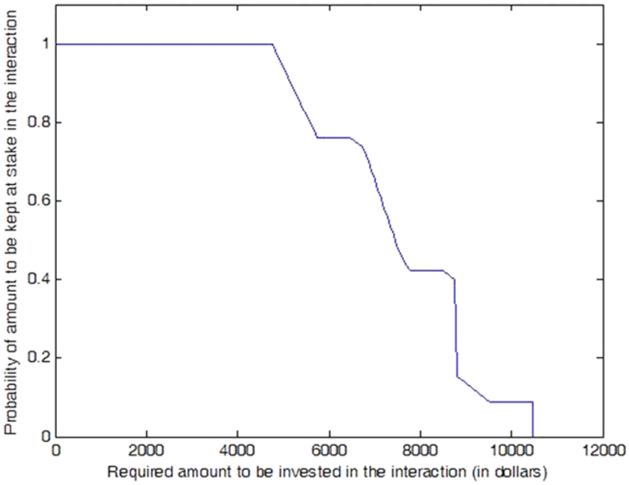

**Fig. 17** Curve representing the financial amount to be kept at stake due to service degradation

Comparing Figs. 16 and 17, it is evident that the curve in Fig. 17 is inflated compared to the curve in Fig. 16 because of the additional resources that need to be kept at stake to obtain the financial gain due to the different levels of service deviation or service degradation. These additional levels of resources can be viewed as the financial loss that arises as a result of service degradation. However an important point to note is the curve obtained in Fig. 17 is based on dependable criteria where a service provider fails to provide a required level of service. Apart from this, there may be other associated costs outside the dependence of the service provider. These costs are called as non-dependable criteria and in cloud computing they may be costs such as the migration cost that occurs due to the migration of a service from one cloud service provider to another. To achieve business goals, service user 'A' also needs to determine the effect of non-dependable criteria in addition to dependable criteria.

## 5.4 Determining the impact of additional factors when ascertaining financial risk as a result of service degradability

Continuing our case study, if service user 'A' chooses to migrate VoIP or video conference service from service provider 'B' to another service provider due to degradation of bandwidth service, the loss from other non-dependable factors such as migration cost adds to the financial loss that could be experienced as a result of service degradation. Let us consider that the migration cost for service user 'A' to transfer its service to another provider is $3000. This may be either a one-off payment or it may be spread over a period of time. Let us consider that service user 'A' opts to make a payment over a period of time with the probability mass function (pmf) as shown in Fig. 18.

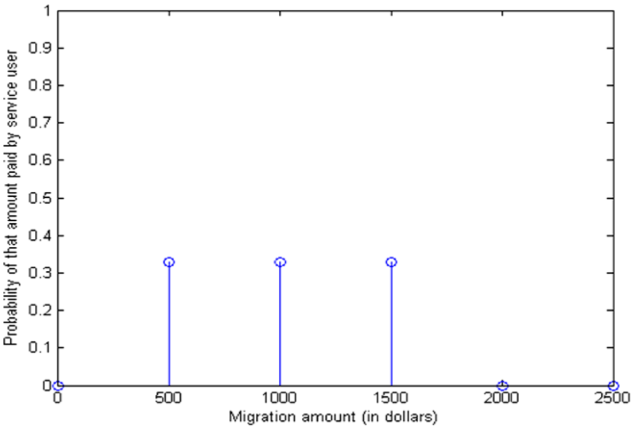

**Fig. 18** The probability of extra level of resources for migration cost

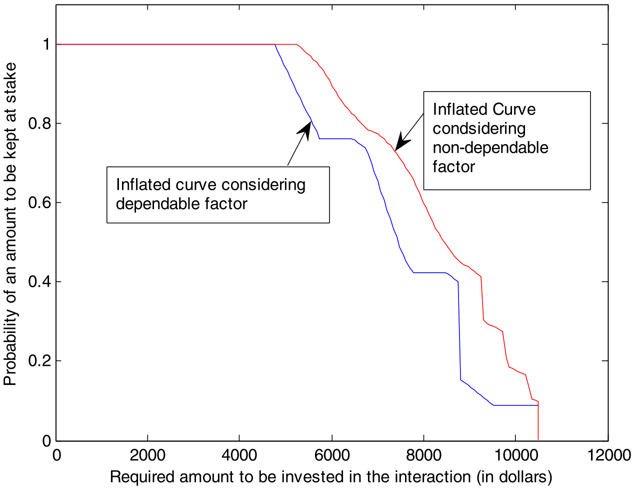

**Fig. 19** Comparison of Actual Resource Investment Curve and Total Resource Investment Curve

We call the curve depicted in Fig. 18 the Extra Investment Curve (EIC). To determine the total financial resources that service user 'A' has to keep at stake, we need to combine the EIC with the ARIC. We achieve this through convolution and we obtain the Total Resource Invested Curve (TRIC) as a result. If we compare the two curves, namely TRIC and ARIC, we obtain the graph shown in Fig. 19.

It can be seen from Fig. 19 that the TRIC is inflated compared to the ARIC. This indicates the extra level of resources that service user 'A' has to keep at stake for the migration of bandwidth service. The higher the levels and probabilities of the resources in the EIC, the greater will be the inflation in the TRIC. Further, the comparison between the total resource investment curve (TRIC) and the expected financial gain curve given in Fig. 20 indicates the extra amount that service user 'A' has to keep at stake to achieve business objectives.

The difference between the expected financial gain curve and the inflated curve due to dependable and non-dependable criteria is shown in Fig. 21. This curve is termed a 'loss of investment' curve.

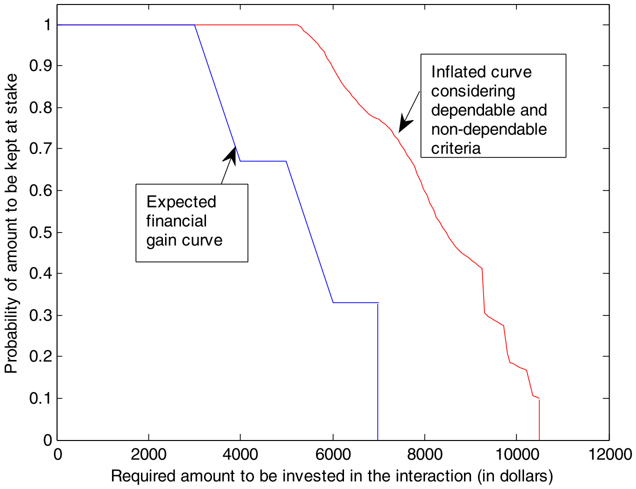

**Fig. 20** Comparison of Expected Financial Gain Curve and Total Resource Investment Curve

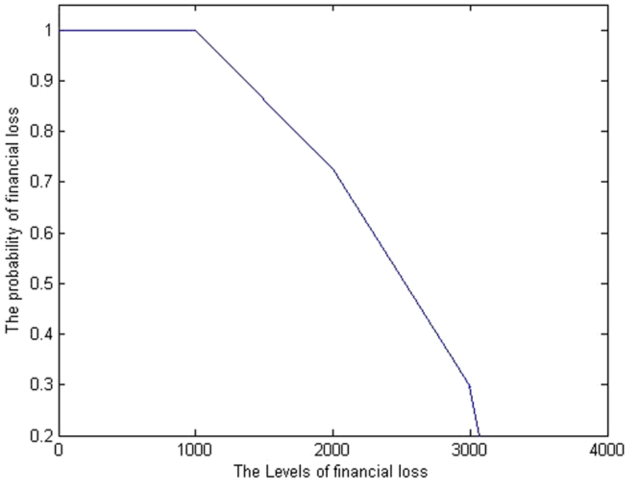

**Fig. 21** The loss curve in the interaction

Initially, the service user was expecting to achieve the financial gain as shown in Fig. 16 but due to possible level/s of service degradation as shown in Fig. 14, the service user has to keep extra amounts of resources at stake to achieve the desired outcomes. The TP SLA monitor informs about the determined analysis to the service user who may then decide whether to take the risk management step and continue the service with the service provider or migrate to a new service provider. Having such continuous evaluation of service provider performance on regular intervals allows the service user to take timely action to ensure that s/he achieves the desired outcomes.

## 6 Conclusion

In this paper, we proposed a framework for SLA management which consists of pre-interaction and post-interaction SLA evaluation processes. In the pre-interaction SLA evaluation process, we used a fuzzy-logic-based approach to determine the reputation of a service provider on the basis of recommendations provided by a group of users. This reputation value is used to select the best possible service provider for the requested service. In the post-interaction time phase, we used a transactional-risk-based approach to determine the level of service degradability in the performance of a service provider and the impact to the service user as a result of that. By using analysis, the service user can make an informed decision regarding the continuity of a service. In our future work we aim to propose a decision making technique that considers the output of these two approaches conjointly and recommend an informed decision on the continuity of the service to the service user.

## References

1. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute, May (2011)
2. Gantz, D.R.J.: (2011). Extracting value from chaos. Available at http://idcdocserv.com/1142. Accessed on June 5, 2011
3. Buyya, R., Pandey, S., Vecchiola, C.: Market-oriented cloud computing and the cloudbus toolkit. In: The First International Conference on Cloud Computing (CloudCom), Beijing, pp. 24–44 (2009)
4. Sun, Y.L., Perrott, R., Harmer, T.J., Cunningham, C., Wright, P., Kennedy, J., Edmonds, A., Bayon, V., Maza, J., Berginc, G., Hadalin, P.: SLA-aware resource management. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 35–44. Springer, New York (2010)
5. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**, 50–58 (2010)
6. He, Q., Yan, J., Kowalczyk, R., Jin, H., Yang, Y.: Lifetime service level agreement management with autonomous agents for services provision. Inf. Sci. **179**, 2591–2605 (2009)
7. Quillinan, T.B., Clark, K.P., Warnier, M., Brazier, F.M.T., Rana, O.: Negotiation and monitoring of service level agreements. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 167–176. Springer, New York (2010)
8. Khader, D., Padget, J., Warnier, M.: Reactive monitoring of service level agreements. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 13–22. Springer, New York (2010)
9. de Oliveira, E., Pfreundt, F.-J.: Monitoring service level agreements in grids with support of a grid benchmarking service. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 1–11. Springer, New York (2010)
10. Chang, E., Hussain, F., Dillon, T.: Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence. Wiley, New York (2005)
11. Schmidt, S., Steele, R., Dillon, T., Chang, E.: Building a fuzzy trust network in unsupervised multi-agent environments. In: On the Move to Meaningful Internet Systems Workshops, Agia Napa, Cyprus, pp. 816–825 (2005)
12. Hussain, O.K., Chang, E., Hussain, F.K., Dillon, T.S.: A methodology to quantify failure for risk-based decision support system in digital business ecosystems. Data Knowl. Eng. **63**, 597–621 (2007)
13. Kaliski, B.S. Jr., Pauley, W.: Toward risk assessment as a service in cloud environments. In: 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, MA, pp. 1–7 (2010)
14. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)
15. Jang, J.-S.R.: Input selection for ANFIS learning. In: 5th International Conference on Fuzzy Systems, New Orleans, LA, pp. 1493–1499 (1996)
16. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**, 665–685 (1993)
17. Chang, E., Dillon, T.: A usability-evaluation metric based on a soft-computing approach. IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. **36**, 356–372 (2006)
18. http://www.investopedia.com/terms/s/sensitivityanalysis.asp#axzz1z0Oi7az5
19. Modarres, M.: What Every Engineer Should Know About Reliability and Risk Analysis. Dekker, New York (1993)
20. Li, W.: Risk Assessment of Power Systems Models, Methods, and Applications. Wiley, New York (2005)
21. Hussain, O., Chang, E., Hussain, F., Dillon, T.: Determining the Failure Level for Risk Analysis in an e-Commerce Interaction. Advances in Web Semantics, vol. I, pp. 290–323. Springer, Berlin (2009)
22. Hussain, O., Dillon, T., Hussain, F.K., Chang, E.: Probabilistic assessment of financial risk in e-business associations. Simul. Model. Pract. Theory **19**, 704–717 (2011)
23. Walpole, R.E., Myers, R.H.: Probability and Statistics for Engineers and Scientists, 5th edn. Macmillan Co., London (1993)

**Adil Hammadi** is a Ph.D. student at School of Information Systems, Curtin University, Perth, WA. In 2006, he received his Master of Science (by research) degree from Deakin University, Australia. In 1998, he received Master of Information Technology degree from Charles Sturt University, Australia. His areas of interests are Software Engineering, Data Mining, Business Intelligence, Trust and Risk based Informed Decision Support Systems. He is a member of IEEE, ACS and AUSOUG.

**Omar Khadeer Hussain** received his Ph.D. degree in Computer Science in 2008 from Curtin University. Since 2008, he is currently a Research Fellow with the School of Information Systems at Curtin University. His research interests are in the area of Risk Assessment and Management, Trusted Computing, Informed Decision Support Systems. He is a member of the IEEE and the IEEE Computer Society.

**Tharam Dillon** received the Ph.D. degree in electrical and computer systems engineering from Monash University, Australia, in 1974. He is the Editor-in-Chief of the International Journal of Computer Systems Science and Engineering and Engineering Intelligent Systems. He has published more than 750 papers published in international conferences and journals. He is the author of five books and the editor of five books. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hybrid neurosymbolic systems, neural nets, software engineering, database systems, and data mining. Prof. Dillon is the head of the International Federation for Information Processing (IFIP) International Task Force WG2.12/24 on Semantic Web and Web Semantics, the Chairman of the IFIP WG12.9 on computational intelligence, the IEEE Industrial Electronics Society Technical Committee on Industrial Informatics, and the IFIP Technical Committee 12 on Artificial Intelligence.

**Farookh Khadeer Hussain** received the Bachelor of Technology degree in computer science and computer engineering; the M.S. degree in Information Yechnology from the La Trobe University, Melbourne, Australia; and the Ph.D. degree in Information Systems from Curtin University of Technology, Perth, Australia, in 2006. He is currently a Faculty member at School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW. His areas of active research are trust, reputation, trust ontologies, data modeling of public and private trust data, semantic web technologies and industrial informatics. He works actively in the domain of making informed business decisions (business intelligence) through the use of trust and reputation technology. He is interested in the application of trust and reputation as a technology, as a business analysis and intelligence tool, and the applications of trust and reputation to various domains.