Risk-based framework for SLA violation abatement from the cloud service provider's perspective

Walayat Hussain[1], Farookh Khadeer Hussain[2], Omar Hussain[3], Ravindra Bagia[1] and Elizabeth Chang[3]

[1] School of Systems, Management and Leadership, Faculty of Engineering and IT, University of Technology Sydney, NSW, Australia
[2] Centre for Artificial Intelligence, School of Software, Faculty of Engineering and IT, University of Technology Sydney, NSW, Australia
[3] School of Business, University of New South Wales, Canberra, Australia
[1,2] {Walayat.Hussain, Farookh.Hussain, Ravindra. Bagia}@uts.edu.au, [3] {O.Hussain, E.Chang}@adfa.edu.au

Abstract: *The constant increase in the growth of the cloud market creates new challenges for cloud service providers. One such challenge is the need to avoid possible SLA violations and their consequences through good SLA management. Researchers have proposed various frameworks and have made significant advances in managing SLAs from the perspective of both cloud users and providers. However, none of these approaches guides the service provider on the necessary steps to take for SLA violation abatement; that is, the prediction of possible SLA violations, the process to follow when the system identifies the threat of SLA violation, and the recommended action to take to avoid SLA violation. In this paper, we approach this process of SLA violation detection and abatement from a risk management perspective. We propose a Risk Management-based Framework for SLA violation abatement (RMF-SLA) following the formation of an SLA which comprises SLA monitoring, violation prediction, and decision recommendation. Through experiments, we validate and demonstrate the suitability of the proposed framework for assisting cloud providers to minimize possible service violations and penalties.*

Keywords: SLA management, SLA violation, risk-based decision making

1. Introduction

Cloud computing has captured a huge customer base in enterprise and small business due to its ability to provide users with a wide range of flexible services at reduced cost. Due to its wide adoption, cloud computing is often referred to as a fifth utility for human beings [2]. Enterprises and businesses using the operational paradigm of cloud computing have drastically reduced their business costs by moving from capital expenditure (e.g. buying resources by building datacentres) to operational expenditure, thus enabling them to focus on their core business activities [3]. Features such as the elastic scaling of resources, pay-as-you-go, and metered resource usage have also enabled users of such enterprises and businesses to reduce their operational costs [4]. However, while such features are beneficial from the user's perspective, they create the illusion that businesses have an infinite quantity of resources that can be accessed as required. This may not be true in all cases, especially when the business is a small to medium (SME) cloud service provider. Unlike large scale cloud service providers, such as Amazon, an SME has a finite quantity of computing resources with which to manage their users' requests [5]. As shown in the literature, these issues between a service provider and service user are addressed by defining and managing a Service Level Agreement (SLA). An SLA describes all the Service Level Objectives (SLOs) and agreed Quality of Service (QoS) parameters [6] and shows the commitment and obligations of each party, including the deliverability and penalties to be applied in the case of SLA violation [7].

As is the case in any business activity, the primary aim of a service provider in cloud computing is to fulfill its commitment to the many users with whom it has formed an SLA, to avoid violations. This falls within the broad domain of cloud service management. Recent contributions in this area [8-11] have looked at different methods, such as the automatic extraction of metrics, ontology-based semantic reasoning, and linked USDL (unified service description language) to manage the SLA and avoid service violations. However, these approaches consider the management of a service after a violation has taken place; in other words, they adopt a reactive approach to service management, which may be detrimental to the cloud service provider's reputation and may negatively impact the likelihood of attracting future business from existing or new cloud service users. This can be avoided if service providers proactively manage their services. In this form of service management, service providers constantly monitor the SLOs after the SLA has been formed to ensure that possible violations are averted before they occur. In our previous work [5], we observed that this proactive management after an SLA has been formed but before violation occurs works well for large scale cloud service providers. This is because large providers have abundant resources and can easily obtain additional resources if/when required to avert possible violations as they are detected. However, for an SME cloud service provider that has a finite quantity of computing resources, obtaining such additional resources at the time and in the quantity required after an SLA has been formed may not be possible. For SLA violations to be proactively managed by such cloud service providers, we emphasize that the service management process should start *before* the formation of the SLA [5], during the SLA negotiation/formation phase *(referred to here as the pre-interaction phase)* in which the cloud service provider pre-allocates its available resources to users after conducting a vetting process. In the SLA execution phase *(referred to here as the post-interaction phase*), which includes SLA monitoring, SLA violation prediction and decisions on violation abatement, the SLOs are constantly monitored to ensure that possible violations are averted. From the perspective of SME cloud service providers, therefore, active service management in both the pre-interaction phase and the post-interaction phase will lead to the better administration of the SLA, maximizing the likely commitment of the service provider, reducing the prospect of SLA violation, and achieving maximum financial returns [5, 12, 13].

Our previous work proposed the provider-based Optimized Personalized Viable SLA (OPV-SLA) framework for service management [1, 14]. OPV-SLA is divided into two parts, namely the pre-interaction phase and the post-interaction phase. In the pre-interaction phase, the provider starts the process of SLA management by negotiating and forming a viable SLA, which is then proactively managed in the post-interaction phase. In this paper, we explain the workings of the OPV-SLA post-interaction phase, which we term the Risk Management Framework for SLA violation abatement (RMF-SLA). In this framework, the runtime performance of the SLOs is captured and predicted, and the service provider recommends the appropriate actions to take to proactively mitigate the risk of SLA violation. The rest of the paper is organized as follows. Section 2 describes the related literature on SLA management. Sections 3 and 4 detail the components of the RMF-SLA along with their workings. Section 5 describes the evaluation of RMF-SLA and Section 6 concludes the paper.

2. Literature Review

The activities in SLA management can be broadly categorized into two time periods, namely the *pre-interaction phase* and the *post-interaction phase*, as mentioned in the Introduction. The activities in the pre-interaction phase are the SLA negotiation and formation, while the activities in the post-interaction phase are QoS prediction for future intervals, runtime QoS monitoring, the comparison between actual and promised QoS parameters, and determining the best course of action for SLA management in the event of observed differences [5]. As our focus in this paper is on the post-interaction phase, we present a summary of some of the existing approaches to SLA management and violation abatement in the literature.

Wood et al. [15] proposed the Sandpiper approach for SLA monitoring and resource management to detect hotspots that indicate a possibility of violation. To eliminate a hotspot, Sandpiper resizes and shifts the virtual machine or adjusts resources. It gathers the usage records of virtual and physical servers and flags a hotspot when resource usage exceeds a defined threshold. The proposed approach manages the runtime workload of the servers. Other approaches such as [16-19] map low-level resource metrics to SLA parameters. This is done by mapping the service status to the predefined threshold and identifying the deviation between the agreed and actual behavior to detect SLA violations using case-based reasoning (CBR) approaches. Although the proposed idea of mapping service resource metrics to SLA parameters helps the service provider to identify potential violation on current performance measures, it may not guarantee commitment to the requirements of all customers, as the performance measures are not formed and agreed in the pre-interaction phase. These approaches do not describe what needs to be done when the system identifies a likely violation. Some approaches offer a limited set of rules and use a CBR approach, which has its own limitations such as adaption, processing time, and storage, and usually does not produce optimal results [20]. Another work in this category by Falasi et al. [21] presents the Sky framework, which adaptively implements SLAs to manage changes in a federated cloud environment. The framework is capable of managing multilevel SLAs but does not describe the process for handling SLA violation. Also, SLAs are not formed during the negotiation process of the pre-interaction phase in this framework, which may not guarantee the requirement commitment of customers in the post-interaction phase. There are a number of approaches with self-management features which try to manage SLA violation before end users are affected. Brandic et al. [18] proposed a bottom-up hierarchical layered approach for the propagation of SLA violation when a violation threat is found. Mosallanejad and Atan [22] proposed a hierarchical self-healing approach in which each layer of cloud is responsible for managing the problem by itself. If the problem cannot be, the framework informs the upper layer for possible remedial action. Lu et al. [23] proposed an actor system framework that adopts a parent-child relationship for managing SLA violation. When the actor system detects a possible SLA violation, it first tries to resolve it, and then sends the error information to the upper parent actor if it is unable to do so. A multilayer monitoring approach was proposed by Katsaros et al. [24] that monitors SLAs based on observing time intervals and SLA parameters. The proposed approach has the features of runtime adaptability of resource provision, estimation, and decision taking. Although the self-management approaches [18, 22, 23] attempt to adjust violations when they are detected by the system, they do not suggest what action to take to avoid violation occurring. Moreover, these approaches lack the agreement process in the pre-interaction phase of SLA management.

Other approaches in the literature use a third party broker to manage SLAs. Lee et al. [25] proposed a cloud service broker portal that provides a gateway for cloud service providers and users to interact with each other. The portal has a single entry point for a cloud service broker, a cloud service provider, and a cloud service user. It interacts with a unique interface designed for each stakeholder, and has a brokerage API that integrates various cloud service providers into the cloud service broker portal. The cloud service brokerage model has five components [26]. The framework helps cloud users to select a suitable cloud provider that satisfies the functional and non-functional requirements of the SLA; from a provider's perspective, however, they lack the pre-interaction processing steps and actions to be taken if an SLA violation is detected. Other SLA management approaches focus on trusted relationships between the provider and the user. Noor and Sheng [27] proposed an adaptive credibility model that offers trust as a service to the service provider. The proposed approach helps the service provider to differentiate between biased and unbiased feedback. Although this approach is helpful for the service provider, it is only effective for a system that has existing users. It lacks a process for differentiating between possible users and the recommended action to be taken on SLA formation and violation. Fan and Perros [28] differentiated between biased and unbiased feedback based on the familiarity and consistency of the feedback. They proposed a trust value range and ranked users based on that value. However, without a bootstrapping mechanism, this method cannot be applied to new users who have only recently subscribed to services. Another category of SLA management approaches uses proactive mechanisms to identify and predict likely SLA violations. Quality of Service (QoS) parameters are predicted using a user-based collaborative filtering (CF) mechanism, item-based CF mechanism, and stream processing framework [29-31]. Cardellini et al. [32] proposed heuristic policies to predict QoS parameters and determine the resources needed in future intervals using the recursive least squares method; however the process of managing SLA violations when they are predicted by the system is not defined.

It can be seen from the above discussion that even though many approaches have been proposed in the literature for cloud SLA management, not all of them guide the service provider on the steps required for SLA violation abatement. In Table 1, we compare SLA management approaches on the three criteria required for SLA abatement, namely the ability to predict possible SLA violations, a description of the process to be followed when the system identifies an SLA violation threat, and the SLA violation abatement recommendation. It is important to mention that most of the existing approaches focus on the post-interaction phase of SLA management, that is, after a user and provider have formed their SLA. As mentioned in Section 1, this is not beneficial for SME cloud service providers, since the careful prior negotiation of SLOs is necessary to maximize the likelihood of a consumer commitment, reduce the possibility of SLA violation, and gain maximum financial returns. To address these drawbacks, we proposed the OPV-SLA (Optimized Personalized Viable SLA) management framework, shown in Figure 1, in our previous work [1, 14, 33]. This framework first assists the user and provider to agree on QoS expectations and then monitors the terms of the agreement for violations.

Table 1: Comparison of SLA management approaches in the post-interaction phase

Source	Predicts SLA / SLO /QoS	Defines procedure for when a violation threat is detected	SLA violation recommendation
Emeakaroha et al. [16, 17]	✓	✗	✗
Brandic et al. [18]	✓	✓	✗
Haq et al. [19]	✓	✓	✗
Emeakaroha et al. [34]	✓	✓	✗
Mosallanejad et al. [22]	✗	✓	✗
Katsaros et al. [24]	✗	✗	✗
Al Falasi et al. [21]	✗	✗	✗
Chandrasekar et al. [35]	✓	✗	✗
Alhamad et al. [36]	✗	✗	✗
Wang et al. [37]	✗	✗	✗
Hammadi and Hussain [38]	✗	✗	✗
Muchahari and Sinha [39]	✗	✗	✗
Cicotti et al. [31]	✓	✗	✗
Romano et al. [30]	✓	✗	✗
Sun et al. [40]	✓	✗	✗
Hussain et al. [41]	✓	✓	✓

Source	Predicts SLA / SLO /QoS	Defines procedure for when a violation threat is detected	SLA violation recommendation
Leitner et al. [42]	✓	✗	✗
Ciciani et al. [43]	✓	✗	✗
Cardellini et al. [44]	✓	✗	✗
Son et al. [45]	✗	✗	✗
Silaghi et al. [46]	✗	✗	✗
Badidi et al. [47]	✗	✗	✗
Pacheco-Sanchez et al. [48]	✓	✗	✗
Wood et al. [15]	✗	✓	✗
Schmieders et al. [49]	✓	✓	✗
Noor and Sheng [27]	✗	✗	✗
Fan and Perros [28]	✗	✗	✗

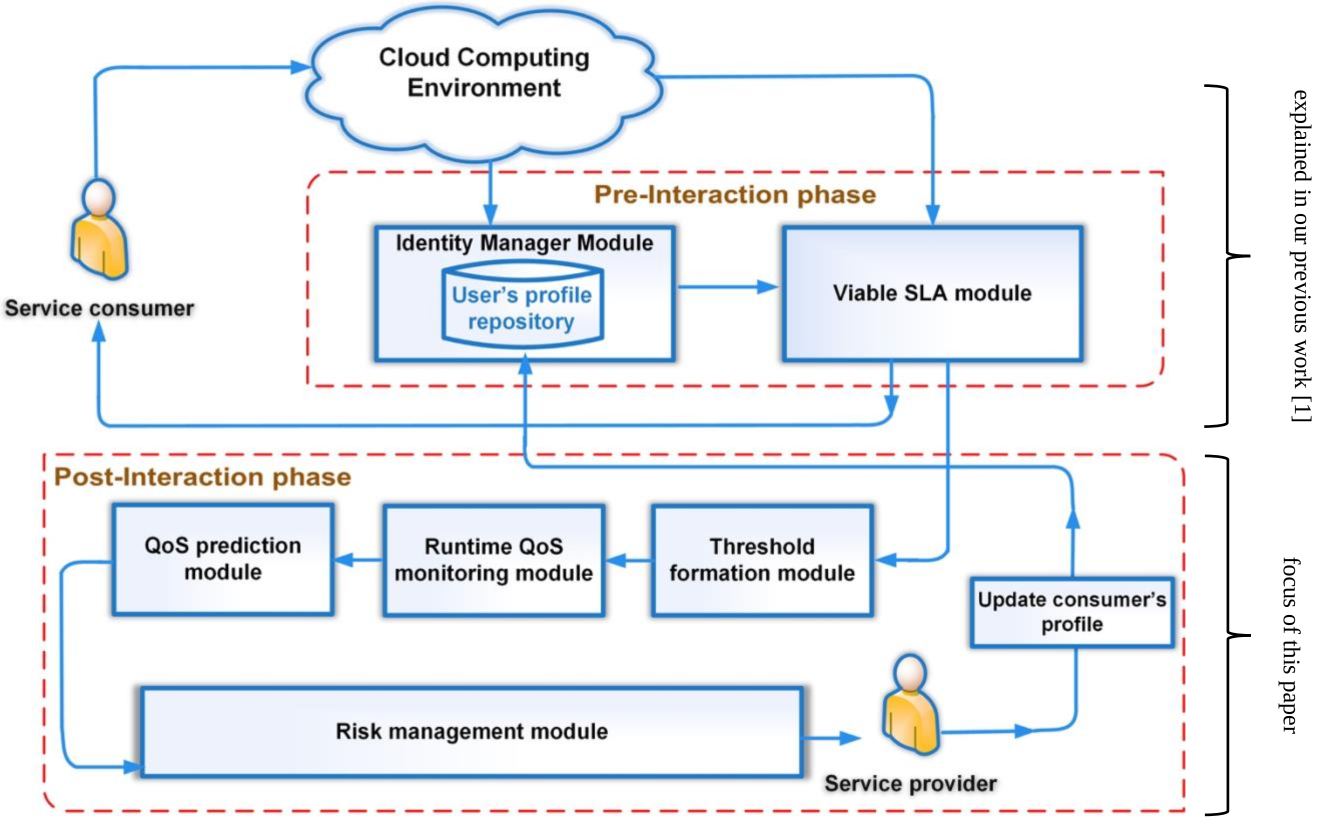

Figure 1: OPV-SLA management framework (reproduced from [1])

The process of forming customized SLAs between providers and users in the pre-interaction phase is explained in our previous work [1]. In this paper, we explain the process for determining and abating SLA violation, which forms part of the post-interaction phase. It is important to mention that various approaches in the literature have used techniques such as QoS prediction [29], workflow detection control model [40], machine learning regression technique [42], and workload analyzer [50] to ascertain the possibility of SLA violation. In our method, we analyze the notion of risk as the criterion for ascertaining possible SLA violation and the subsequent actions to take for its abatement. A related work in this category that uses risk as the criterion for SLA management is Kiran et al. [51], who proposed a risk assessment framework for cloud service provisioning. Their proposed framework assists both SaaS and IaaS providers to identify, evaluate and mitigate risk in service provisioning. The risk assessment between a service provider and an infrastructure provider consists of six steps: the infrastructure provider's business dealings, the service provider's business dealings, the potential for service failure under the SLA, the reliability of the services offered under the SLA, the service provider for runtime operation and monitoring of QoS parameters, and lastly, the infrastructure provider for potential infrastructure failure. Zhang et al. [52] proposed a risk management framework that analyzes, assesses and mitigates risk to help the service provider to achieve better management of SLAs. Risk assessment in this approach is comprised of four steps: to define the likelihood of vulnerabilities and associated threats, to determine the magnitude of risk, to find the level of risk, and to take all necessary actions to mitigate risk. Cicotti et al. [53] proposed a model that predicts future QoS based on runtime monitoring data and data from a probabilistic model-checking method. The system generates an alert when it detects probable QoS violation and helps service providers to stop or minimize possible service violation. Albakri et al. [54] proposed a security risk management framework that allows users to evaluate risk and contribute to the risk assessment process. The framework permits users to define the legal requirements, identify the risk factors, and obtain feedback from a service provider. While there are approaches that consider the notion of risk in SLA management, most of them are unable to guide a service provider in relation to the steps to be taken to determine and address possible SLA violation. In the next section, we define our RMF-SLA, which assists cloud service providers to identify and assess the risk of SLA violation occurring in the post-interaction phase and to manage it by considering a set of decision parameters.

3. Risk Management framework for SLA violation abatement (RMF-SLA)

As shown in Figure 1, RMF-SLA is a combination of five modules that address the detection and abatement of SLA violation. They are:

- *Module 1: Threshold Formation Module (TFM)*
- *Module 2: Runtime QoS Monitoring Module (RQoSMM)*
- *Module 3: QoS Prediction Module (QoSPM)*
- *Module 4: Risk Identification Module (RIM)*
- *Module 5: Risk Management Module (RMM).*

The workings of each module of RMF-SLA are explained in the following sub-sections.

3.1 Module 1: Threshold formation module (TFM):

This is the first module of the RMF-SLA framework, as shown in Figure 2. It takes the QoS values of the SLOs determined between the cloud provider and the user in the pre-interaction phase and forms *two* thresholds for determining and managing violations. These two thresholds are the *Agreed threshold (T_a)* and the *Safe threshold (T_s).*

- *Agreed threshold (T_a):* This threshold value is described in the SLA and is mutually agreed by the user and the provider. When both parties have finalized their SLAs, they agree on certain thresholds for each Service Level Objective (SLO) and QoS parameter. A service provider that does not comply with the agreed QoS parameters commits a service violation and is liable for violation penalties.
- *Safe threshold (T_s):* To avoid possible service violation and penalties, we propose that a provider should define a safe threshold *(T_s)* that is stricter than the agreed threshold *(T_a).* This is a customized threshold defined by the provider. It raises an alarm of possible SLA violation when a runtime QoS reaches or exceeds the threshold and invokes *Module 5*, the *Risk Management Module (RMM),* to take necessary action to avert the violation.

To explain the importance of T_s and T_a, let us consider a provider and user forming an SLA in the pre-interaction phase who agree on having 80% availability of a resource (memory). This 80% availability of memory is the T_a value agreed by both parties. For service management and SLA violation abatement, the provider defines a customized threshold for the total memory, say 90%, which is the T_s value. When the availability at runtime falls below this 90% threshold, the framework alerts the service provider and activates the RMM to manage the risk of the QoS value of the SLO falling below the T_a value.

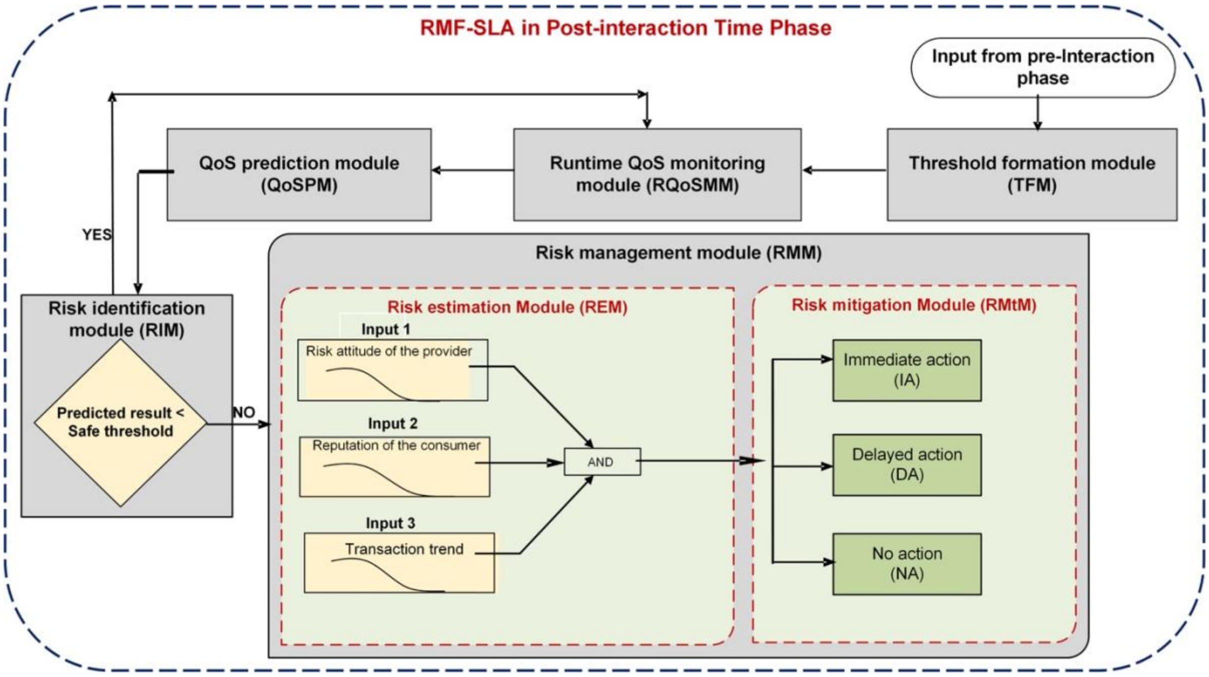
RMF-SLA in Post-interaction Time Phase
Input from pre-Interaction phase
QoS prediction module (QoSPM)
Runtime QoS monitoring module (RQoSMM)
Threshold formation module (TFM)
YES
Risk management module (RMM)
Risk identification module (RIM)
Predicted result < Safe threshold
NO
Risk estimation Module (REM)
Input 1
Risk attitude of the provider
Input 2
Reputation of the consumer
Input 3
Transaction trend
AND
Risk mitigation Module (RMtM)
Immediate action (IA)
Delayed action (DA)
No action (NA)

Figure 2: Provider-based Risk Management Framework for SLA violation abatement (RMF-SLA)

3.2 Module 2: Runtime QoS monitoring module (RQoSMM):

This is the second module of RMF-SLA, which is responsible for monitoring the runtime QoS parameters of each SLO in the SLA. The captured runtime QoS values are sent to *Module 3 - QoSPM* where the QoS values of the SLOs in the near future are determined.

3.3 Module 3: QoS prediction module (QoSPM):

The *QoSPM* is the third module of RMF-SLA, which predicts users' resource usage in each SLO. The module uses an optimal prediction algorithm for each SLO to predict the user's likely resource usage based on his or her usage history. The choice of an optimal prediction algorithm plays a key role in decision making, since the accuracy of of the prediction method depends on the choice of dataset. In our previous work [55], we considered the stochastic, neural network, and different time series prediction methods and analyzed their prediction accuracy on a dataset from the Amazon EC2 cloud. We observed from the evaluation results that an optimal prediction result is obtained by considering small intervals for prediction and using the Autoregressive Integrated Moving Average (ARIMA) method. ARIMA is one of the most efficient versions of the Autoregressive Moving Average (ARMA) method formulated by mathematical statisticians George Box and Gwilym Jenkins in the 1970s [56] for use with business and economic data. It has been widely used as an optimal prediction method in the cloud service domain. For example, Calheiros et al. [57] developed a cloud virtual machine workload prediction model and observed 91% prediction accuracy when using the ARIMA method. Other researchers, such as Rehman et al. [58], have used the ARIMA method to forecast the QoS values from the user perspective with good accuracy. Hence, for the prediction of QoS we use the ARIMA method in *QoSPM*. To enhance the accuracy of the prediction result, *RQoSMM* constantly inputs the value of the SLOs in previous time intervals to *QoSPM*. For example, *RQoSMM* captures the QoS values from time interval 1 to time interval 10 (t_1 - t_{10}) to predict them over time interval 11 to 14 (t_{11} - t_{14}). When the QoS values over the interval

t_{15} to t_{18} are predicted, *RQoSMM* gives *QoSPM* the captured QoS values up to $t_{14,}$ to ensure that an optimal prediction result closely related to the observed data with minimum deviation [55] is achieved. The pseudocode of the workings in *QoSPM* is as follows:

```
for (i=start limit; i <= endlimit ; i++ )
 if (RQoSMM is empty)
    input[i]= prev_observation[i];
 else
    input[i]= RQoSMM[i]+ prev_observation[i];
Pred_output= Prediction_algo(input);
```

The algorithm starts by ensuring that the runtime data is available. If a transaction has just started and runtime QoS data is not available, *QoSPM* considers the user's data from the *Identity Management Module (IMM)* in the pre-interaction phase. As explained in our previous work [1], this module stores the interaction history of a user. If a user is new and has no previous transaction records, the commitment of the user to the QoS values is ascertained by *IMM* using top-K nearest neighbors and their transactions. Once these values have been obtained, the *QoSPM* prediction is not made by taking the relative values of the SLOs, but by taking the percentage value of the level of the SLO commitment to the level of the SLO requested. This is because the level of resources requested by a user in the current SLO interaction may be different from what was requested in the past, so a standardized scale on which to represent these values is needed for fair analysis. To obtain the standardized scale, we take the relative values of the previous transaction, as presented in Equation 1.

$$Rpred = \sum_{i=1}^{m} \left(\frac{Rused}{Rrequest}\right)^{i} * Rc^{i} \qquad \text{Equation 1}$$

where

Rused is the amount of resources used for the previous SLA

Rrequest is the amount of resources requested for the previous SLA

i is the time interval from 1 to m

m is the total time interval

Rc^{i} is the predicted resource for the new SLA at ith interval

Rpred is the total predicted amount of resources for the current requested SLA

The output of the *QoSPM* is forwarded to *Module 4 - Risk Identification Module* (RIM) which invokes *Module 5 - Risk Management Module (RMM)* when the possibility of SLA violation is detected.

3.4 Module 4: Risk identification module (RIM):

RIM is responsible for comparing the predicted values from *QoSPM* with the formed T_s value in *Module 1*. If the value of *QoSPM* reaches or exceeds the T_s value, *Module 5 - Risk Management Module (RMM)* is activated to abate possible SLA violation.

3.5 Module 5: Risk management module (RMM):

As mentioned earlier, *RMM* is invoked when *RIM* determines the possibility of an SLA violation occurring. Once invoked, *RMM* estimates the severity of the risk of SLA violation and determines how to manage it. *RMM* as shown in Figure 4 is comprised of two sub-modules, the *risk estimation module (REM)* and the *risk mitigation module (RMtM).*

a) *Risk estimation module (REM)*: This sub-module is responsible for estimating the risk of an SLA violation occurring. The notion of risk is subjective, as is the process for managing it, so to determine the severity of the risk from the subjective viewpoint of the provider, the following three inputs are considered in *REM*:

- *Risk attitude of the provider:* The risk attitude of the provider represents its capacity to deal with risk. A provider's risk attitude is risk averse, risk neutral, or risk taking. A provider with a risk averse attitude is more reluctant to take a risk (in this case, to allow an SLA violation to occur) than a provider with a risk neutral or risk taking attitude [1].
- *Reputation of the user for whom the possibility of SLA violation is being determined*: Reputation is the reliability or trust value a provider places on a user to uphold the terms of the SLA. The reputation of a user shows the user's commitment to previously formed SLAs with the provider and is represented as being bronze, silver or gold. The process for determining the class of reputation is described in our previous work [1]. The reputation of a user is an input of *REM*, because we consider that if a provider values a user highly (represented as being silver or gold class), the provider will prefer to take immediate action to minimize the possible risk of SLA violation, in contrast to a similar situation with a user whose reputation is bronze class.
- *Transaction trend curve over future time intervals*: The third input to *REM* is the transaction trend curve that shows a user's use of an SLO over future time intervals. This shows the prevailing use of resources by the user over a period of time (from *QoSPM)* and how this usage maps against the formed T_a and T_s values. When the transaction trend curve exceeds T_s, it may either move towards T_a, as shown in Figure 3a, or away from T_a, as shown in Figure 3b. *REM* captures the direction of the transaction trend curve to ascertain the risk of SLA violation and estimate the steps required to mitigate the risk.

b) *Risk mitigation module (RMtM)*: As discussed above, *REM* estimates the risk of possible SLA violation occurring by considering the relevant inputs. Subsequently, *RMtM* recommends an appropriate action to manage and mitigate the risk. A fuzzy inference system is used to perform the computation with the recommendation to take *immediate action, delayed action,* or *no action*. When a risk of violation is assessed as high, *RMtM* recommends that the service provider should take *immediate* action. In taking this action, the service provider stops accepting new requests and arranges for sufficient resources to be provided in the fastest possible time to avoid service violation. When the risk of violation is estimated as medium or low, *RMtM* decides and recommends whether to take *delayed action* or *no action*. Here, it is implied that the provider accepts the risk but keeps the situation under observation, with the intention of taking any necessary action within a certain timeframe.

To summarize, the working of RMF-SLA is as follows and as shown in Figure 5:

a) Step 1: After forming the T_s and T_a thresholds, QoSPM collects data from IMM and RQoSMM.
b) Step 2: QoSPM predicts the QoS usage values in future time intervals.
c) Step 3: RIM compares the predicted values from the QoSPM with the T_s value.
d) Step 4: If the value from QoSPM is below T_s, then no action is taken and the runtime QoS parameters of the SLO are monitored. However, if the value of the QoSPM reaches or exceeds T_s, RMM is activated to manage the risk of SLA violation.
e) Step 5: REM of RMM estimates the risk of SLA violation by capturing the risk attitude of the provider, the reputation of the user and the transaction trend curve. A fuzzy inference system (FIS) is used to estimate the risk of SLA violation occurring.

f) Step 6: Depending on the estimated risk in REM, RMtM suggests the appropriate action that the cloud provider should take to mitigate the risk of violation occurring. The type of action is immediate action, delayed action, or no action.

The fuzzy inference system for estimating the possible risk of SLA violation and determining the appropriate mitigation action is explained in the next section.

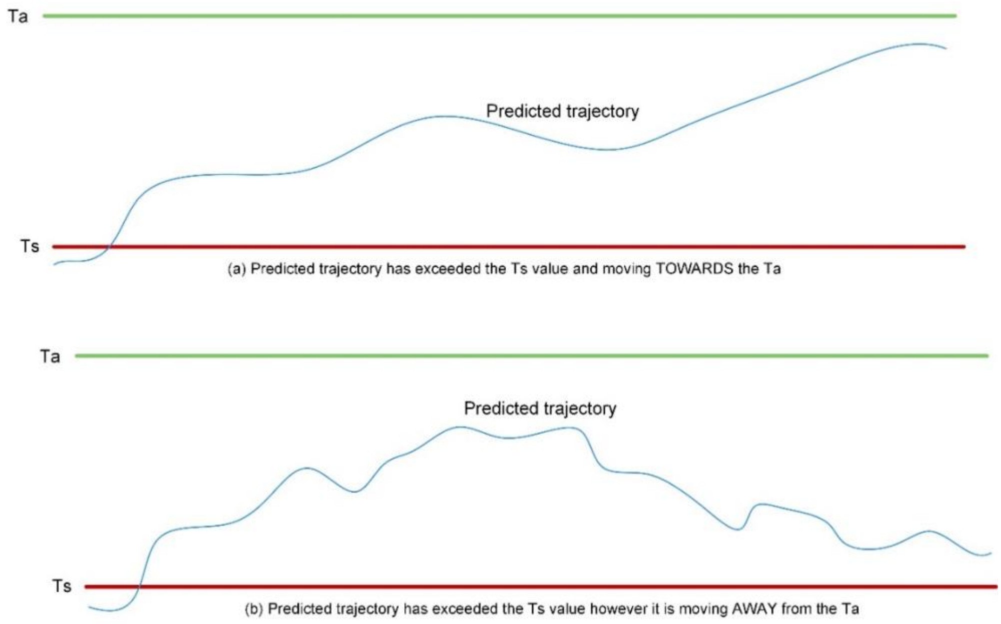
Ta
Predicted trajectory
Ts
(a) Predicted trajectory has exceeded the Ts value and moving TOWARDS the Ta
Ta
Predicted trajectory
Ts
(b) Predicted trajectory has exceeded the Ts value however it is moving AWAY from the Ta

Figure 3: Transaction trend curve moving towards and away from T_a

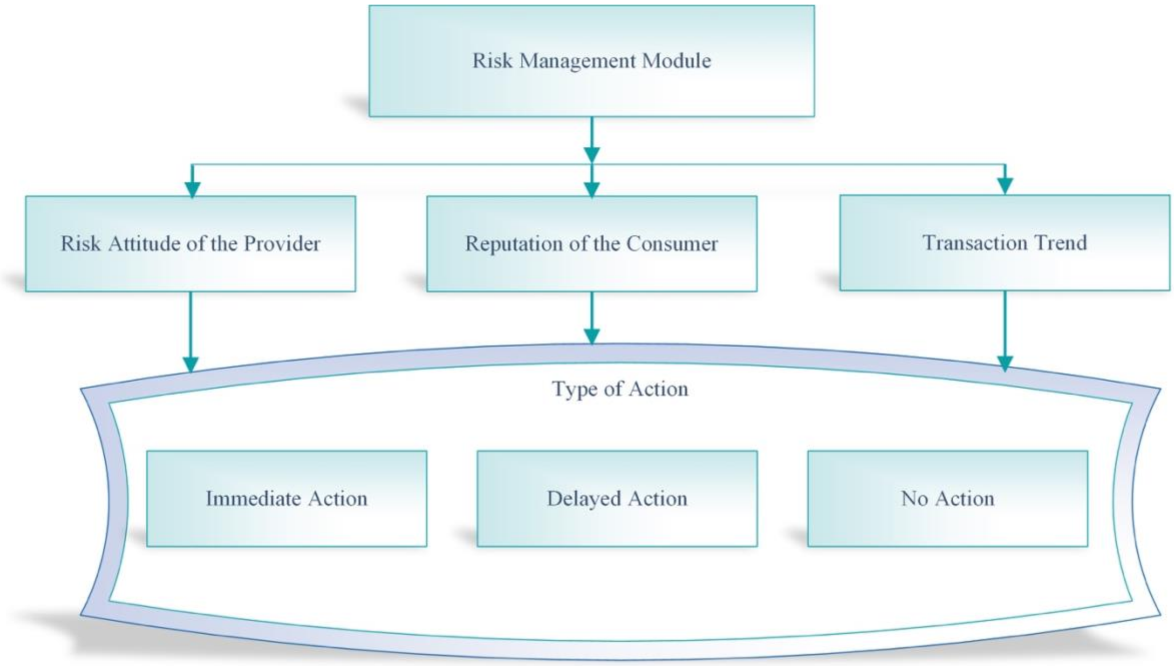

Figure 4: Working of RMM in RMF-SLA

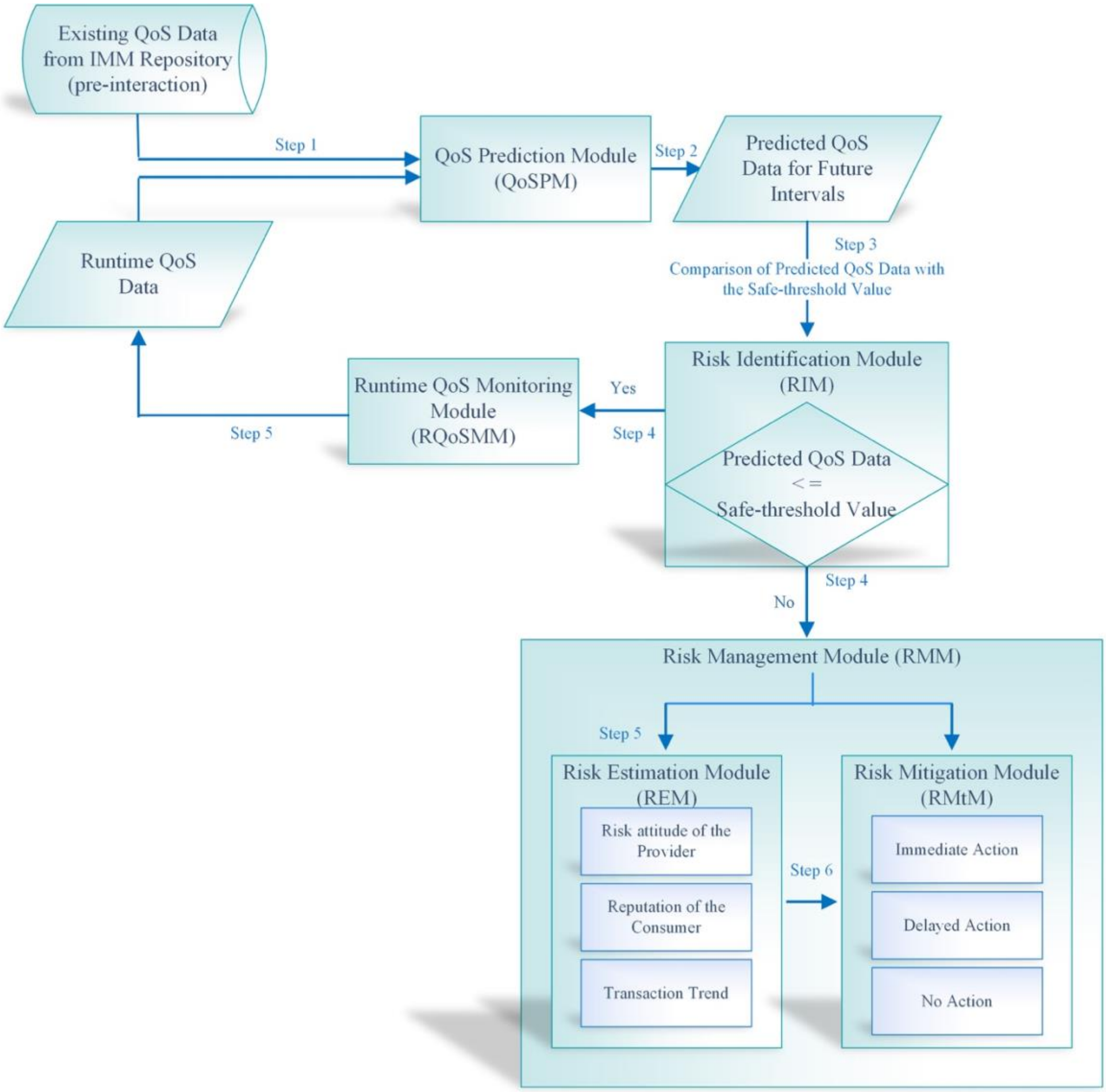
Existing QoS Data from IMM Repository (pre-interaction)
Step 1
QoS Prediction Module (QoSPM)
Step 2
Predicted QoS Data for Future Intervals
Runtime QoS Data
Step 3
Comparison of Predicted QoS Data with the Safe-threshold Value
Risk Identification Module (RIM)
Predicted QoS Data <= Safe-threshold Value
Runtime QoS Monitoring Module (RQoSMM)
Yes
Step 4
Step 5
Step 4
No
Risk Management Module (RMM)
Step 5
Risk Estimation Module (REM)
Risk attitude of the Provider
Reputation of the Consumer
Transaction Trend
Step 6
Risk Mitigation Module (RMtM)
Immediate Action
Delayed Action
No Action

Figure 5: The working of RMF-SLA in the post-interaction phase of OPV-SLA

4. Fuzzy Inference System (FIS) for determining possible SLA violations and their abatement in RMF-SLA

To assess the possible risk of SLA violation and manage its abatement, we use a Mamdani type FIS [59] to combine the various inputs. Figure 6 represents the input and the output used to manage possible SLA violations. The FIS and the membership functions of each of its inputs and outputs are explained in the following sub-sections.

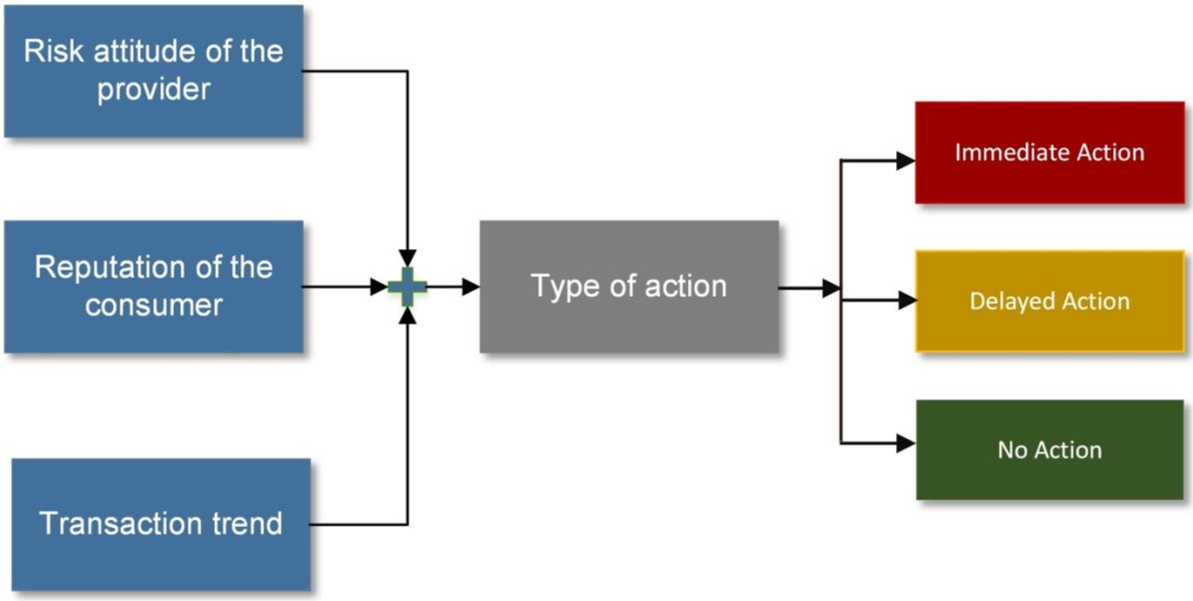

Figure 6: FIS for assessing and managing the risk of SLA violation in RMF-SLA

4.1 Defining the fuzzy sets and membership function for input - risk attitude of the provider: The risk attitude (RA) of the provider defines the provider's propensity to take risk. Depending on its RA, a service provider may be *risk averse, risk neutral*, or *risk taking*. These are the fuzzy sets over which the RA will be represented. A risk averse provider attempts to avoid any risk, whether it is small or large. A risk neutral provider takes the middle ground; depending on the nature of the risk, it may decide to take action or to ignore the risk. A risk taking provider has a bold attitude, ignoring small risks and taking action only for risks that will have a significant effect. We consider 1 to 5 as the Universe of Discourse over which the fuzzy sets of this input will be represented. The membership function for this input is as shown in Figure 7, and the corresponding membership function for each fuzzy set is as follows:

$$\mu_{\text{Risk Averse}}(\text{RA}) = \frac{3-x}{2} \text{ if } 1 < x \leq 3;\ 0 \text{ if } 3 < x \leq 5$$

$$\mu_{\text{Risk Neutral}}(\text{RA}) = \frac{x-1}{2} \text{ if } 1 < x \leq 3;\ \frac{5-x}{2} \text{ if } 3 < x \leq 5$$

$$\mu_{\text{Risk Taking}}(\text{RA}) = 0 \text{ if } 0 < x \leq 3;\ \frac{x-3}{2} \text{ if } 3 < x \leq 5$$

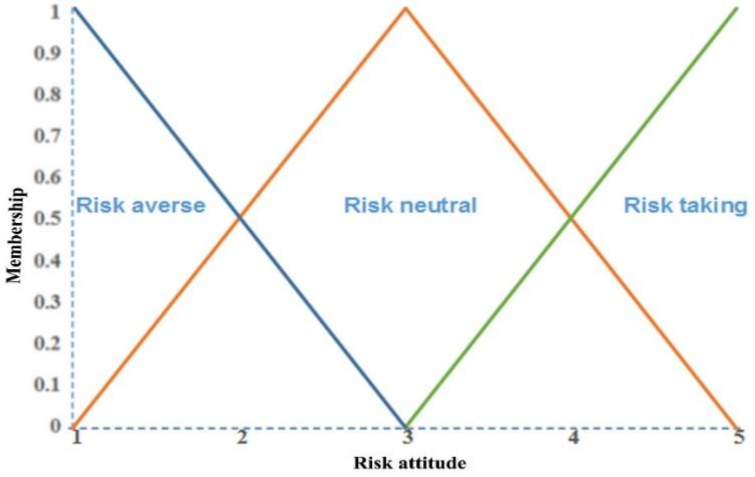

Figure 7: Risk attitude of a provider in assessing the possibility of SLA violation occurring

4.2 Defining the fuzzy sets and membership function for input - reputation of the user:

User reputation (R) is the trustworthiness of the user's commitment in previous transactions to the defined SLAs with the service provider. The fuzzy set over which the reputation value of a user is represented is bronze, silver, or gold, and the universe of discourse is from 0 to 100. The membership function for this input is as shown in Figure 8, and the corresponding membership function for each fuzzy set is as follows:

$\mu_{\text{Bronze}}(R) = 1 \text{ if } 0 < x \leq 40; \ \frac{45-x}{5} \text{ if } 41 < x \leq 45; \ 0 \text{ if } 46 < x \leq 100$

$\mu_{\text{Silver}}(R) = 0 \text{ if } 0 < x \leq 40, \ \frac{x-40}{5} \text{ if } 41 < x \leq 45, \ 1 \text{ if } 45 < x \leq 70, \ \frac{75-x}{5} \text{ if } 71 < x \leq 75, \ 0 \text{ if } 76 < x \leq 100$

$\mu_{\text{Gold}}(R) = 0 \text{ if } 0 < x \leq 70, \ \frac{x-70}{5} \text{ if } 71 < x \leq 75, \ 1 \text{ if } 76 < x \leq 100$

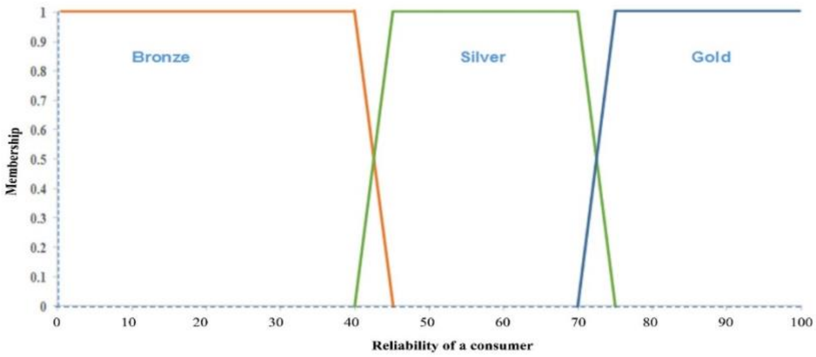

Figure 8: Membership function for the reliability of a user

4.3 Defining the fuzzy sets and membership function for input – transaction trend:

Transaction trend (TT) shows the trajectory of the predicted resource usage in future intervals. The values of the predicted trajectory are obtained from the *QoSPM*. The fuzzy sets used to represent input TT are Towards the T_a or Away from the T_a. The universe of discourse over which the input TT is represented is from 0 to 1. The membership function for this input is as shown in Figure 9, and the corresponding membership function for each fuzzy set is as follows:

$\mu_{\text{Away}}(TT) = \frac{1-x}{1} \quad \text{if } 0 < x \leq 1$

$\mu_{\text{Towards}}(TT) = \frac{x-1}{1} \quad \text{if } 0 < x \leq 1$

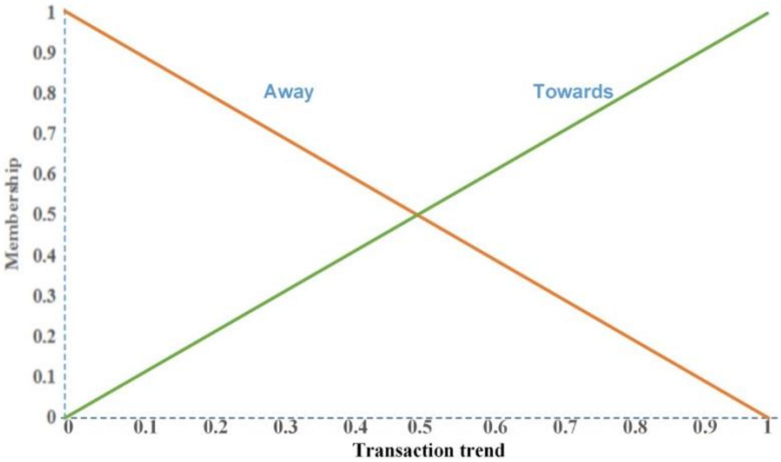

Figure 9: Membership function for the transaction trend

4.4 Defining the fuzzy sets and membership function for output – recommended action:

The output Recommended Action (RAc) is the appropriate action to be taken to manage the possible risk of violation occurring, and the recommended output of Immediate Action, Delayed Action, or No Action. These are the fuzzy sets used to represent the output, and the universe of discourse over which these fuzzy sets are represented is 0 to 1. The membership function for this input is as shown in Figure 10 and the corresponding membership function for each fuzzy set is as follows:

$\mu_{\text{No Action}}(\text{RAc}) = 1 \text{ if } 0 < x \leq 0.01, 0 \text{ if } 0.01 < x \leq 1$

$\mu_{\text{Delayed Action}}(\text{RAc}) = 0 \text{ if } 0 < x \leq 0.01; 1 \text{ if } x = 0.01; \dfrac{x - 0.01}{0.99} \text{ if } 0.01 < x \leq 1;$

$\mu_{\text{Delayed Action}}(\text{RAc}) = 0 \text{ if } x=0.01; \dfrac{1 - x}{0.99} \text{ if } 0.01 < x \leq 1$

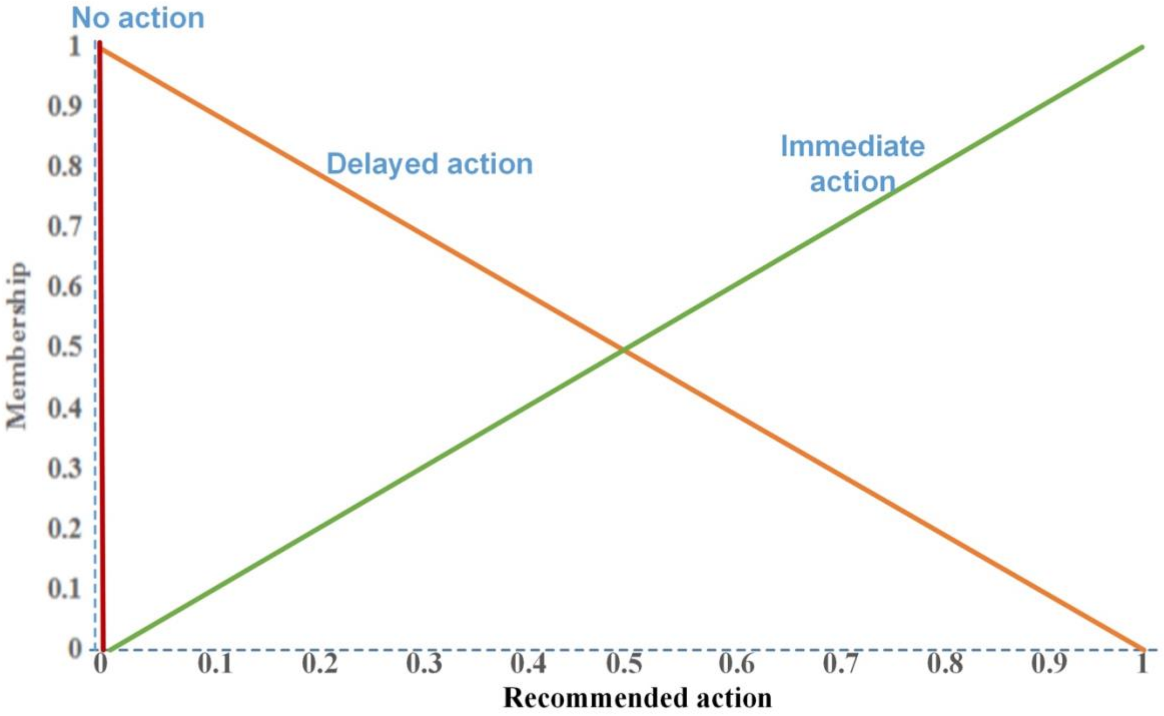

Figure 10: Membership function for recommended risk mitigation action

4.5 Fuzzy rules for possible risk of SLA violation occurring and mitigation action to be taken:

The combination of linguistic variables for the inputs, resulting in a total of eighteen rules, is presented in Table 2. The variables are: the risk attitude of the provider [risk averse (Ra), risk neutral (Rn), or risk taking (Rt)], the reputation of the user [bronze (B), silver (S), or gold (G)], and the transaction trend [towards (T) or away (A)]

Table 2: FIS rules for the assessment and abatement of SLA violation risk

Rule #		Risk attitude		Reputation		Transaction trend	then	Recommended risk mitigation action
1	If	Ra	and	B	and	T	then	IA
2	If	Ra	and	B	and	A	then	IA

3	If	Ra	and	S	and	T	then	IA
4	If	Ra	and	S	and	A	then	DA
5	If	Ra	and	G	and	T	then	DA
6	If	Ra	and	G	and	A	then	DA
7	If	Rn	and	B	and	T	then	IA
8	If	Rn	and	B	and	A	then	DA
9	If	Rn	and	S	and	T	then	IA
10	If	Rn	and	S	and	A	then	NA
11	If	Rn	and	G	and	T	then	DA
12	If	Rn	and	G	and	A	then	NA
13	If	Rt	and	B	and	T	then	DA
14	If	Rt	and	B	and	A	then	NA
15	If	Rt	and	S	and	T	then	DA
16	If	Rt	and	S	and	A	then	NA
17	If	Rt	and	G	and	T	then	NA
18	If	Rt	and	G	and	A	then	NA

In the next section, we demonstrate how the service provider can assess and manage the risk of possible SLA violation with the consumer using the RMF-SLA framework of OPV-SLA.

5. Validation of RMF-SLA framework for managing possible SLA violation abatement

To demonstrate the applicability of the RMF-SLA framework for service providers in the abatement of possible SLA violations, we utilize the dataset from Amazon EC2 IaaS cloud services – EC2 US West collected from CloudClimate [60] through the PRTG monitoring service [61]. This dataset is used for QoS prediction and for managing SLAs and the abatement of possible violation. The prototype was built using Microsoft Visual Studio 2010 to develop the interface, Microsoft SQL Server Management Studio 2008 for the databases, and MATLAB to design the FIS application. To implement RMF-SLA, we first need to form an SLA between the user and provider in the OPV-SLA pre-interaction phase. Readers should refer to our previous work in [1], in which we explain these computations in detail. The outcome of this phase is a well-formed SLA between the service provider and service user which maximizes the likelihood of the service provider's commitment to the formed SLOs, reduces the potential for SLA violation, and achieves the maximum financial return for the available resources. To ensure the successful fulfillment of the SLA, the service provider needs to undertake the following management steps in the post-interaction phase: prediction, monitoring, and decision-making, which are assisted by RMF-SLA, as explained next.

Using the EC2 US West dataset from Amazon EC2 IaaS cloud services, we adopt CPU usage as the SLO we want to monitor to proactively pre-determine possible SLA violations. As discussed in Section 4, the first module of RMF-SLA is *TFM*, which defines the T_s value for the SLO being monitored. This is different from the T_a value, which is decided during the formation of the SLA. The next two stages in the RMF-SLA are *RQoSMM* and *QoSPM*. *QoSPM* predicts the QoS values over a future period. A number of prediction methods are available, each of which generates a different output depending on the nature of the dataset being used. RMF-SLA uses ten prediction methods, namely Cascade Forward Backpropagation (CFBP), Elman Backpropagation (EBP), Generalized Regression (GR), Nonlinear autoregressive neural network with external input (NARX), Simple Exponential

Smoothing (SES), Simple Moving Average (SMA), Weighted Moving Average (WMA), Extrapolation (EXP), Holt-Winters Double Exponential Smoothing (HDES), and Autoregressive Integrated Moving Average Method (ARIMA). Root Mean Square Error (RMSE) and Mean Absolute Deviation (MAD) are used as the benchmark to measure prediction accuracy, and the method which gives the least error is used for prediction.

We use an example to explain the process. Figure 11 shows the observed QoS of the SLO CPU usage for the period of one hour on 6th September 2016 from 06:35AM to 7:30AM. To test the accuracy of the prediction methods, we use the QoS values for that SLO from a previous time period and use them to predict the QoS values for 06:35AM to 7:30AM on 6th September 2016. The neural network-based methods were trained by considering 1002 data sets from the previous six days. The results of the observed and predicted QoS values are shown in Table 3 and Figure 12. The prediction results at five-minute intervals are given, and all units are measured in millisecond (ms). The accuracy of each method is measured using Root Mean Square Error (RMSE) and Mean Absolute Deviation (MAD). The prediction accuracy of all methods is presented in Table 4 and Figure 13.

We evaluate CPU usage every five minutes for one hour, starting on 6th September 2016 at 06:35AM and ending on 6th September 2016 at 7:30AM. Figure 12 presents the CPU usage for the period.

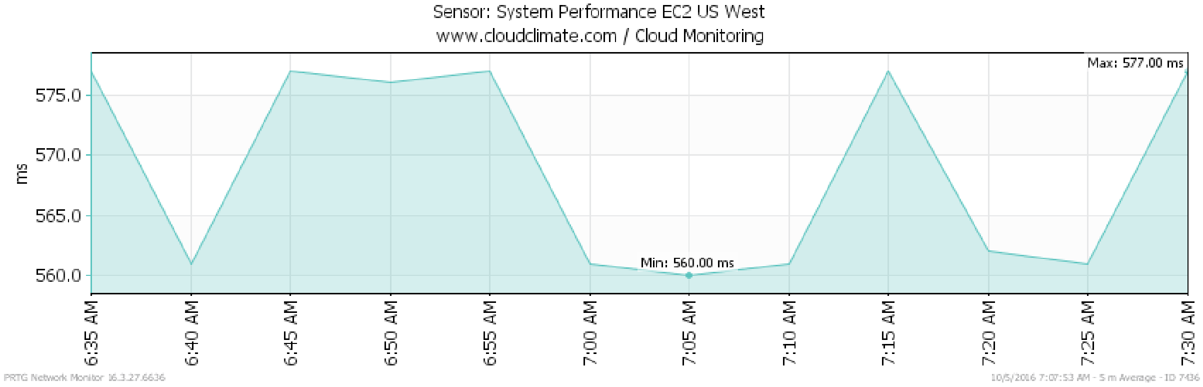

Figure 11: CPU usage data for one hour **[60]**

Table 3: Prediction results of ten methods at five-minute intervals

Time	6:35:00	6:40:00	6:45:00	6:50:00	6:55:00	7:00:00	7:05:00	7:10:00	7:15:00	7:20:00	7:25:00	7:30:00
Observed data	577	561	577	576	577	561	560	561	577	562	561	577
CFBP	565.7835	569.1289	565.7835	565.9902	565.7835	569.1289	569.3407	569.1289	565.7835	568.9174	569.1289	565.7835
EBP	561.1551	560.3656	561.1551	560.921	561.1551	560.3656	560.4925	560.3656	561.1551	560.2562	560.3656	561.1551
GR	563.7587	562.58	568.6284	566.7982	563.7587	560.2106	560.2518	562.58	563.9038	559.747	562.58	563.7587
NARX	579.6209	578.983	561.8996539	575.5140582	564.1123463	563.7748057	561.8996539	576.9943986	577.0424372	575.5140582	562.7078841	576.8213273
SES	577.0000	577.0000	569.0000	573.0000	574.5000	575.7500	568.3750	564.1875	562.5938	569.7969	565.8984	563.4492
SMA	577.0000	569.0000	571.6667	571.3333	576.6667	571.3333	566.0000	560.6667	566.0000	566.6667	566.6667	566.6667
WMA	577.0000	566.3333	571.6667	576.3333	576.6667	566.3333	560.3333	560.6667	571.6667	567.0000	561.3333	571.6667
EXP	577.0000	561.0000	545.0000	593.0000	575.0000	578.0000	545.0000	559.0000	562.0000	593.0000	547.0000	560.0000
HWDES	577.0000	572.2000	572.9680	573.4475	574.2363	570.1543	566.4477	563.7661	566.5231	564.5816	562.7676	566.1918
ARIMA	576.5432	561.0000	577.0000	576.9987	577.0009	561.0887	560.0087	561.0000	577.0000	562.0000	561.0000	577.0594

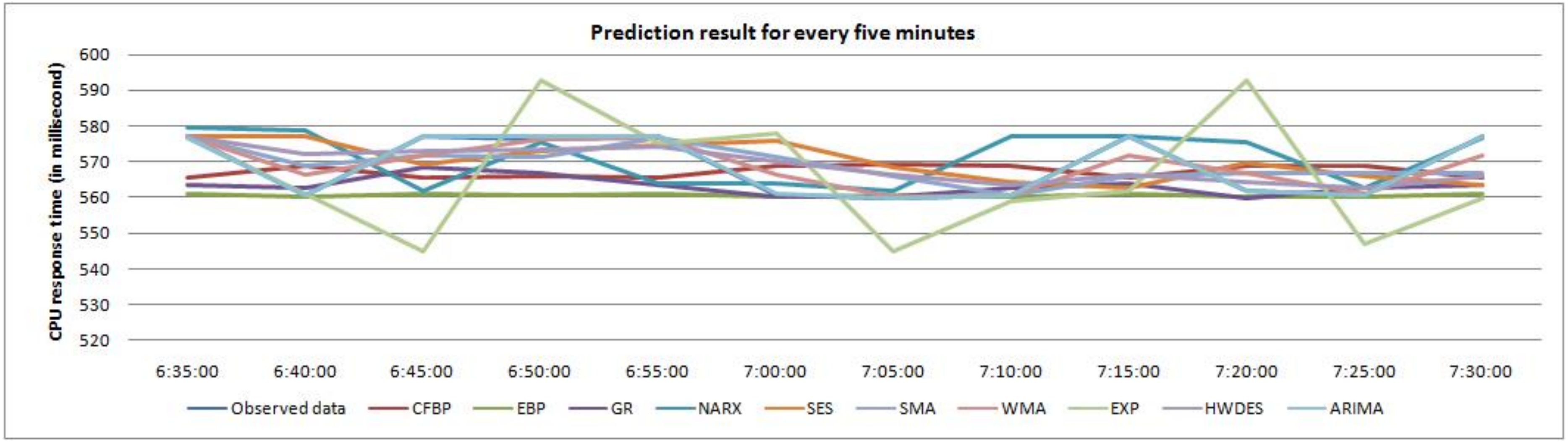

Figure 12: Prediction output of each approach at five-minute intervals

Table 4: Prediction accuracy of all methods

Prediction Method	RMSE	MAD
CFBP	9.489859312	9.365921429
EBP	10.31080812	7.167585714
GR	7.873491133	5.757521429
NARX	9.959856064	7.185961724
SES	9.317532303	7.878696987
SMA	6.52346193	5.523809524
WMA	3.743777907	2.761904762
EXP	18.48937919	15.00000000
HWDES	6.216755602	4.999325847
ARIMA	0.461865174	0.249580313

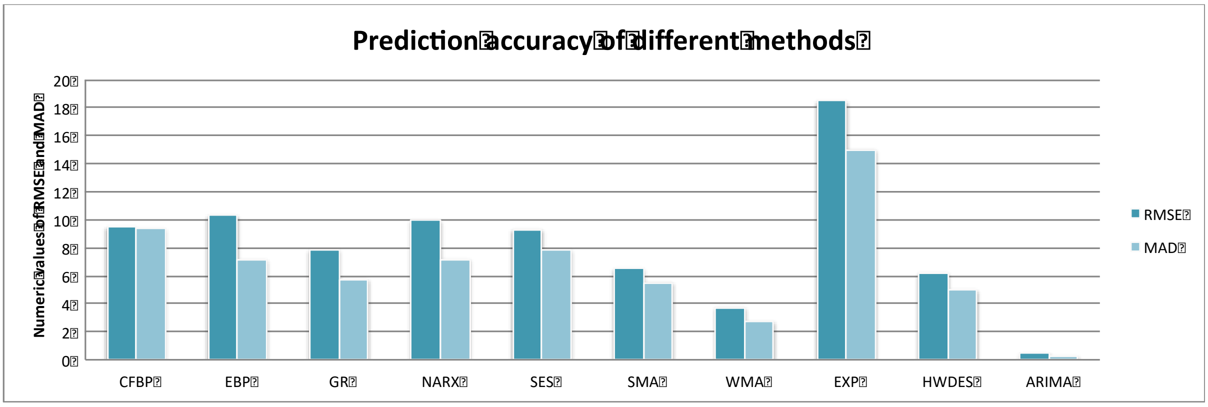

Figure 13: Prediction accuracy of all methods using RMSE and MAD as a benchmark

From Table 4, we can see that of all the prediction methods, ARIMA gives the optimal prediction result with an RMSE value of 0.461865174 and a MAD value of 0.249580313. Extending our example, *QoSPM* uses the ARIMA method to predict the QoS of the CPU usage for the next hour from 7:40 AM to 8:35 AM, as shown in Table 5. To determine the possibility of SLA violation and its abatement, we consider that the values for T_s and T_a are 575ms and 599ms respectively, as shown in Figure 14. T_a is the value of the SLO determined on the formation of the SLA and T_s is the safe threshold defined by the provider. *RIM* compares the predicted QoS value with these values, and if the T_s value is exceeded, the *RMM* is activated to ascertain and manage the risk of SLA violation.

Table 5: Prediction of the SLO over a period of one hour using the ARIMA method

Time	7:40:00	7:45:00	7:50:00	7:55:00	8:00:00	8:05:00	8:10:00	8:15:00	8:20:00	8:25:00	8:30:00	8:35:00
ARIMA	567	569	577	579	580	582	585	589	571	591	593	594

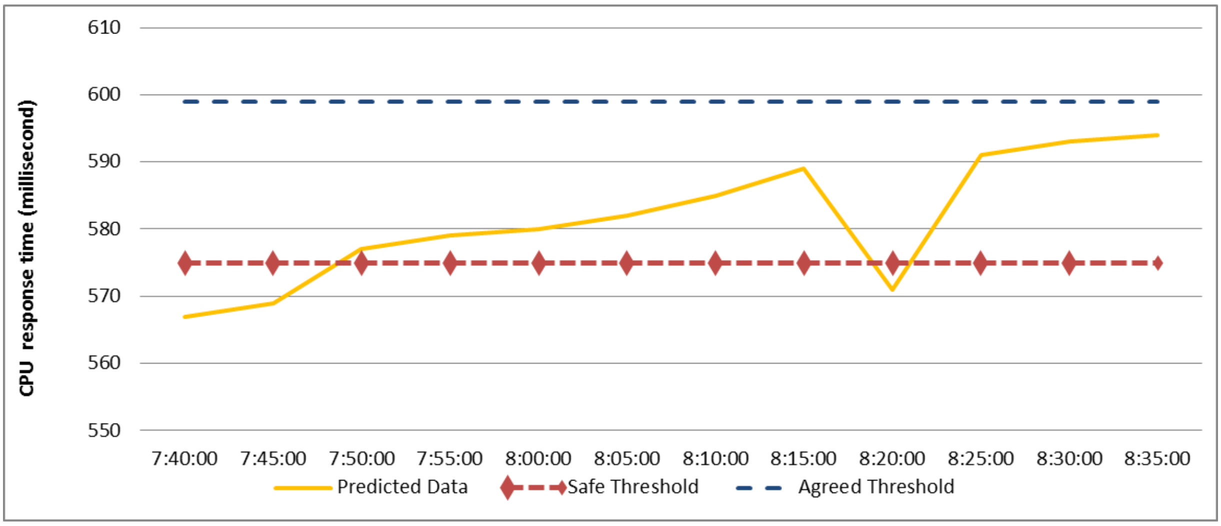

Figure 14: Showing the T_s and T_a values of the predicted QoS over a future period

From Figure 14, we see that at the third time interval (7:50AM) the predicted result exceeds the T_s threshold. *RMM* at this stage considers the risk attitude of the provider, the reputation of the user, and the projected transaction trend to suggest an appropriate action. In this scenario, we consider that the reputation of the user at the pre-interaction phase is 45 (silver), the risk attitude of the provider is risk neutral, and the transaction trend is moving towards the agreed threshold value. These inputs are processed by the FIS rules and the recommended output is *immediate action*. This is because the provider is risk neutral and the transaction trend has exceeded the T_s and is moving towards the T_a value, so if the provider does not take action, there is a high risk of SLA violation. The provider needs to take immediate action by arranging supply of the deficient resources, either itself or from external resources, to avoid possible violation. Similarly, at 8:15 AM we see that the predicted QoS value moves towards T_s and drops below it. In this scenario, the output from the FIS recommends no action to be taken, as no likelihood of SLA violation is determined. From the above example we see that RMF-SLA suggests the appropriate action to be taken to manage potential SLA violation according to the risk attitude of the provider, the user's reputation, and the transaction trend. The combination of RMF-SLA with the pre-interaction phase module of OPV-SLA assists an SME service provider to first form viable SLA and then manage the risk associated with possible SLA violations.

6. Conclusion

The service level agreement (SLA) is the key agreement made between a service provider and a service user in a cloud computing environment. To increase and maintain their reputation, service providers need a viable SLA management framework that helps them to first form viable SLAs and then intelligently predict the occurrence of possible SLA violations before recommending an appropriate action to be taken. Our proposed OPV-SLA management framework helps service providers, particularly SME providers with limited resources, to achieve this. In this paper, we have briefly explained the OPV-SLA framework and focussed on its post-interaction phase module, namely the RMF-SLA, which is responsible for QoS prediction, detecting the possible occurrence of SLA violations, and recommending the best possible decision to avert violation. We have demonstrated the application of RMF-SLA with an example and have shown how the proposed method assists cloud service providers in SLA management. In our future work, we will find the hidden patterns between SLOs and low-level metrics to predict likely violation for SLA management.

7. References

[1] W. Hussain, F. Hussain, O. Hussain, and E. Chang, "Provider-based Optimized Personalized Viable SLA (OPV-SLA) Framework to Prevent SLA Violation," *The Computer Journal* vol. 59, pp. 1760-1783, 2016.

[2] C. Weinhardt, D.-I.-W. A. Anandasivam, B. Blau, D.-I. N. Borissov, D.-M. T. Meinl, D.-I.-W. W. Michalk*, et al.*, "Cloud computing–a classification, business models, and research directions," *Business & Information Systems Engineering,* vol. 1, pp. 391-399, 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski*, et al.*, "Above the clouds: A berkeley view of cloud computing," Electrical Engineering and Computer Sciences University of California at Berkeley, Available at http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, Technical Report No. UCB/EECS-2009-282009.

[4] J. Rhoton, *Cloud Computing Explained: Implementation Handbook for Enterprises*. London, United Kingdom: Recursive Press, 2013.

[5] W. Hussain, F. K. Hussain, O. Hussain, E. Damiani, and E. Chang, "Formulating and managing viable SLAs in Cloud Computing from a small to medium service provider's viewpoint: A state-of-the-art review," *Information Systems,* In press 2017.

[6] H. Ludwig, A. Keller, A. Dan, R. King, and R. Franck, "A service level agreement language for dynamic electronic services," *Electronic Commerce Research,* vol. 3, pp. 43-59, 2003.

[7] W. Hussain, F. K. Hussain, and O. K. Hussain, "Maintaining Trust in Cloud Computing through SLA Monitoring," in *International Conference on Neural Information Processing*, Kuching, Malaysia, 2014, pp. 690-697.

[8] S. Mittal, K. P. Joshi, C. Pearce, and A. Joshi, "Automatic Extraction of Metrics from SLAs for Cloud Service Management," in *2016 IEEE International Conference on Cloud Engineering (IC2E)*, Berlin, Germany, 2016, pp. 139-142.

[9] B. Karim, T. Qing, J. R. Villar, and E. d. l. Cal, "Resource brokerage ontology for vendor-independent Cloud Service management," in *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, China, 2017, pp. 466-472.

[10] J. M. García, P. Fernández, C. Pedrinaci, M. Resinas, J. Cardoso, and A. Ruiz-Cortés, "Modeling Service Level Agreements with Linked USDL Agreement," *IEEE Transactions on Services Computing,* vol. 10, pp. 52-65, 2017.

[11] G. E. Jaramillo, C. A. Ardagna, and M. Anisetti, "A hybrid representation model for service contracts," in *2015 International Conference on Information and Communication Technology Research (ICTRC)*, Abu Dhabi, United Arab Emirates, 2015, pp. 246-249.

[12] F. Messina, G. Pappalardo, C. Santoro, D. Rosaci, and G. M. Sarné, "A multi-agent protocol for service level agreement negotiation in cloud federations," *International Journal of Grid and Utility Computing,* vol. 7, pp. 101-112, 2016.

[13] G. Feng and R. Buyya, "Maximum revenue-oriented resource allocation in cloud," *International Journal of Grid and Utility Computing,* vol. 7, pp. 12-21, 2016.

[14] W. Hussain, F. K. Hussain, and O. K. Hussain, "SLA Management Framework to Avoid Violation in Cloud," in *International Conference on Neural Information Processing*, Kyoto, Japan, 2016, pp. 309-316.

[15] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Computer Networks,* vol. 53, pp. 2923-2938, 2009.

[16] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*, 2010, pp. 48-54.

[17] V. C. Emeakaroha, M. A. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. De Rose, "Towards autonomic detection of SLA violations in Cloud infrastructures," *Future Generation Computer Systems,* vol. 28, pp. 1017-1029, 2012.

[18] I. Brandic, V. C. Emeakaroha, M. Maurer, S. Dustdar, S. Acs, A. Kertesz*, et al.*, "Laysi: A layered approach for sla-violation propagation in self-manageable cloud infrastructures," in *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, Seoul, South Korea, 2010, pp. 365-370.

[19] I. U. Haq, I. Brandic, and E. Schikuta, "Sla validation in layered cloud infrastructures," in *International Conference on the Economics of Grids, Clouds, Systems, and Services*, Ischia, Italy, 2010, pp. 153-164.

[20] W. Cheetham, A. Varma, and K. Goebel, "Case-Based Reasoning at General Electric," in *PROCEEDINGS OF THE FOURTEENTH INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY CONFERENCE*, Florida, USA, 2001, pp. 93-97.

[21] A. Al Falasi, M. A. Serhani, and R. Dssouli, "A Model for Multi-levels SLA Monitoring in Federated Cloud Environment," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 2013 IEEE 10th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, Vietri sul Mere, Italy, 2013, pp. 363-370.

[22] A. Mosallanejad and R. Atan, "HA-SLA: A Hierarchical Autonomic SLA Model for SLA Monitoring in Cloud Computing," *Journal of Software Engineering and Applications,* vol. 6, p. 114, 2013.

[23] K. Lu, R. Yahyapour, P. Wieder, E. Yaqub, M. Abdullah, B. Schloer*, et al.*, "Fault-tolerant Service Level Agreement lifecycle management in clouds using actor system," *Future Generation Computer Systems,* vol. 54, pp. 247-259, 2015.

[24] G. Katsaros, G. Kousiouris, S. V. Gogouvitis, D. Kyriazis, A. Menychtas, and T. Varvarigou, "A Self-adaptive hierarchical monitoring mechanism for Clouds," *Journal of Systems and Software,* vol. 85, pp. 1029-1041, 2012.

[25] J. Lee, J. Kim, D.-J. Kang, N. Kim, and S. Jung, "Cloud Service Broker Portal: Main entry point for multi-cloud service providers and consumers," in *2014 16th International Conference on Advanced Communication Technology (ICACT)*, Pyeongchang, South Korea, 2014, pp. 1108-1112.

[26] F. Jrad, J. Tao, and A. Streit, "SLA based Service Brokering in Intercloud Environments," in *CLOSER*, 2012, pp. 76-81.

[27] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *Web Information System Engineering–WISE 2011*, Sydney, Australia, 2011, pp. 314-321.

[28] W. Fan and H. Perros, "A reliability-based trust management mechanism for cloud services," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Melbourne, VIC, Australia, 2013, pp. 1581-1586.

[29] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *2011 30th IEEE Symposium on Reliable Distributed Systems (SRDS)*, Madrid, Spain, 2011, pp. 1-10.

[30] L. Romano, D. De Mari, Z. Jerzak, and C. Fetzer, "A novel approach to QoS monitoring in the cloud," in *2011 First International Conference on Data Compression, Communications and Processing (CCP)*, Palinuro, Italy, 2011, pp. 45-51.

[31] G. Cicotti, L. Coppolino, S. D'Antonio, and L. Romano, "How to monitor QoS in cloud infrastructures: the QoSMONaaS approach," *International Journal of Computational Science and Engineering,* vol. 11, pp. 29-45, 2015.

[32] V. Cardellini, E. Casalicchio, F. Lo Presti, and L. Silvestri, "Sla-aware resource management for application service providers in the cloud," in *2011 First International Symposium on Network Cloud Computing and Applications (NCCA)*, Toulouse, France, 2011, pp. 20-27.

[33] W. Hussain, F. K. Hussain, and O. Hussain, "Allocating Optimized Resources in the Cloud by a Viable SLA Model," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada 2016, pp. 1282-1287.

[34] V. C. Emeakaroha, T. C. Ferreto, M. A. Netto, I. Brandic, and C. A. De Rose, "Casvid: Application level monitoring for sla violation detection in clouds," in *2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC)*, Izmir, Turkey, 2012, pp. 499-508.

[35] A. Chandrasekar, K. Chandrasekar, M. Mahadevan, and P. Varalakshmi, "QoS monitoring and dynamic trust establishment in the cloud," in *Advances in Grid and Pervasive Computing*, ed: Springer, 2012, pp. 289-301.

[36] M. Alhamad, T. Dillon, and E. Chang, "Sla-based trust model for cloud computing," in *2010 13th International Conference on Network-Based Information Systems (NBiS)*, Takayama, Japan, 2010, pp. 321-324.

[37] M. Wang, X. Wu, W. Zhang, F. Ding, J. Zhou, and G. Pei, "A conceptual platform of SLA in cloud computing," in *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Sydney, Australia, 2011, pp. 1131-1135.

[38] A. M. Hammadi and O. Hussain, "A framework for SLA assurance in cloud computing," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Fukuoka, Japan, 2012, pp. 393-398.

[39] M. K. Muchahari and S. K. Sinha, "A new trust management architecture for cloud computing environment," in *2012 International Symposium on Cloud and Services Computing (ISCOS)*, Mangalore, India, 2012, pp. 136-140.

[40] Y. Sun, W. Tan, L. Li, G. Lu, and A. Tang, "SLA detective control model for workflow composition of cloud services," in *2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Whistler, BC, Canada, 2013, pp. 165-171.

[41] O. K. Hussain, F. K. Hussain, J. Singh, N. K. Janjua, and E. Chang, "A User-Based Early Warning Service Management Framework in Cloud Computing," *The Computer Journal,* vol. 58, pp. 472–496, 2014.

[42] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime prediction of service level agreement violations for composite services," in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, Stockholm, Sweden, 2010, pp. 176-186.

[43] B. Ciciani, D. Didona, P. Di Sanzo, R. Palmieri, S. Peluso, F. Quaglia*, et al.*, "Automated workload characterization in cloud-based transactional data grids," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, 2012, pp. 1525-1533.

[44] V. Cardellini, E. Casalicchio, F. Lo Presti, and L. Silvestri, "Sla-aware resource management for application service providers in the cloud," in *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*, 2011, pp. 20-27.

[45] S. Son, D.-J. Kang, S. P. Huh, W.-Y. Kim, and W. Choi, "Adaptive trade-off strategy for bargaining-based multi-objective SLA establishment under varying cloud workload," *The Journal of Supercomputing,* vol. 72, pp. 1597-1622, 2016.

[46] G. C. Silaghi, L. D. ŞErban, and C. M. Litan, "A time-constrained SLA negotiation strategy in competitive computational grids," *Future Generation Computer Systems,* vol. 28, pp. 1303-1315, 2012.

[47] E. Badidi, "A cloud service broker for SLA-based SaaS provisioning," in *2013 International Conference on Information Society (i-Society)*, Toronto, Canada, 2013, pp. 61-66.

[48] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson, "Markovian workload characterization for qos prediction in the cloud," in *2011 IEEE International Conference on Cloud Computing (CLOUD)*, Washington, DC, USA, 2011, pp. 147-154.

[49] E. Schmieders, A. Micsik, M. Oriol, K. Mahbub, and R. Kazhamiakin, "Combining SLA prediction and cross layer adaptation for preventing SLA violations," *Available at: http://eprints.sztaki.hu/6563/1/2ndwoss_Micsik.pdf*, 2011.

[50] B. Ciciani, D. Didona, P. Di Sanzo, R. Palmieri, S. Peluso, F. Quaglia*, et al.*, "Automated workload characterization in cloud-based transactional data grids," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, Shanghai, China, 2012, pp. 1525-1533.

[51] M. Kiran, M. Jiang, D. J. Armstrong, and K. Djemame, "Towards a service lifecycle based methodology for risk assessment in cloud computing," in *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Sydney, NSW, Australia, 2011, pp. 449-456.

[52] X. Zhang, N. Wuwong, H. Li, and X. Zhang, "Information security risk management framework for the cloud computing environments," in *2010 IEEE 10th International Conference on Computer and Information Technology (CIT)*, Bradford, UK, 2010, pp. 1328-1334.

[53] G. Cicotti, L. Coppolino, S. D'Antonio, and L. Romano, "Runtime Model Checking for SLA Compliance Monitoring and QoS Prediction," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA),* vol. 6, pp. 4-20, 2015.

[54] S. H. Albakri, B. Shanmugam, G. N. Samy, N. B. Idris, and A. Ahmed, "Security risk assessment framework for cloud computing environments," *Security and Communication Networks,* vol. 7, pp. 2114-2124, 2014.

[55] W. Hussain, F. K. Hussain, and O. Hussain, "QoS Prediction Methods to Avoid SLA Violation in Post-Interaction Time Phase," in *11th IEEE Conference on Industrial Electronics and Applications (ICIEA 2016)* Hefei, China, 2016, pp. 32-37.

[56] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control* vol. 734: John Wiley & Sons, 2011.

[57] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Transactions on Cloud Computing,* vol. 3, pp. 449-458, 2015.

[58] Z. ur Rehman, O. K. Hussain, F. K. Hussain, E. Chang, and T. Dillon, "User-side QoS forecasting and management of cloud services," *World Wide Web,* vol. 18, pp. 1677–1716, 2015.

[59] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International journal of man-machine studies,* vol. 7, pp. 1-13, 1975.

[60] CloudClimate. *Watching the Cloud*. Available: http://www.cloudclimate.com

[61] P. N. Monitor. Available: https://prtg.paessler.com