# Comparative Analysis of Consumer Profile-based Methods to Predict SLA Violation

Walayat Hussain, Farookh Khadeer Hussain
School of Software
Decision Support and e-Service Intelligence Lab
Centre for Quantum Computation & Intelligent Systems
University of Technology Sydney
Sydney, Australia
walayat.hussain@student.uts.edu.au;
Farookh.Hussain@uts.edu.au

Omar Khadeer Hussain
School of Business
University of New South Wales Canberra (UNSW Canberra)
Australian Defence Force Academy
Canberra, ACT, 2601
O.Hussain@adfa.edu.au

***Abstract* – A Service Level Agreement (SLA) is a contract between a service provider and a consumer which specifies in detail the level of service expected from the service provider, obligations, commitment and objectives. In the cloud computing environment, both the cloud provider and the cloud consumer want to know of a likely service violation before the actual violation occurs and to adjust the scaling of the cloud resources appropriately. A consumer's previous resource usage profile is a key element in determining the possibility of service violation in the cloud computing environment, which has not been an area of research focus so far. In this paper, we analyze and compare QoS prediction by considering the consumer's previous resource usage profile in various conditions. From comparative analysis, we observe that by combining a consumer's previous resource usage profile history along with the previous resource usage profile history of its nearest neighbors, we obtain an optimal result.**

***Keywords—SLA violation prediction; cloud computing; consumer's profile.***

## 1. Introduction

In today's world, cloud computing is emerging very rapidly because of its flexibility in managing information architecture, data and computation resources. It provides a platform for providers and consumers to exchange their computation needs and resources over the Internet. In 2013 survey Gartner Research anticipated that, $921 billion will be expected to spend on public cloud computing market from 2013 to 2017 which is 17% increased from 2011.Cloud computing frees consumers from managing, maintaining and updating various computing resources and the cloud providers charge consumers for the amount of resources they use.

Due to the dynamic nature of the cloud, it is esential for all cloud stakeholders to manage, monitor and maximise the utilization of cloud resources by ensuring security, efficiency, high availability, trustworthiness, reliablility and to fullfill all service level objectives (SLOs) as defined in the service level agreement (SLA). An SLA is the key agreement on which all business relations are formed. For successful business relations, it is vital to predict any service violation before it affects the consumer and provider. The service provider needs to know the consumer's likely usage of different cloud resources before finalizing the service level agreement to arrange appropriate cloud resources or to warn or alert a provider about a possible service violation before it occurs.

In any business, a consumer's previous resource usage profile is a key element in determining anticipated future transactions and the possibility of service violation, which has not been an area of research focus to date. The majority of existing approaches predict and monitor the SLA from a consumer's perspective using various techniques, such as mapping low-level resource metrics to SLA parameters [1], exponential smoothing and autoregressive [2].

A consumer's previous resource usage profile is a record of their various business transactions and violation history. We assume that a consumer who has a previous service violation history is more likely to violate a service level agreement in the future.

We proposed a profile-based SLA violation prediction model to help a service provider to predict a likely service violation by a consumer and compare profile-based Quality of Service (QoS) prediction techniques by considering:

- Consumer's previous resource usage profile only
- Previous resource usage profile of the nearest neighbors using the top-K nearest neighbors.
- Resource usage profile of both the consumer and its nearest neighbors.

To evaluate the prediction accuracy of each method we used Root Mean Square Error (RMSE) and Mean Absolute Deviation (MAD) to measure the variation with the Actual result.

The rest of the paper is arranged as follows. Section 2 reviews the related literature on SLA violation prediction. Section 3 discusses system architecture. Section 4 describes our proposed approach. Section 5 describes the implementation and evalutaion of SLA violation prediction approaches. Section 5 concludes the paper and proposes future work.

## 2. Related Studies

A service level agreement is the key agreement that describes the deliverables, obligations and commitment of the provider to the consumer. The reputation and trust of the provider and consumer depend on the successful completion of all objectives defined in SLA. To maintain a trusted relationship between the provider and consumer, it is important to be able to predict service violation before actual violation occurs. As discussed in our previous work [3], there are a number of existing studies which assist cloud consumers and cloud providers to detect violation before it occurs. Some of the related literature is described as follows:

[1] monitored different low-level resource metrics relating to different SLOs and mapped these resource metrics to SLOs to define the threat threshold. At runtime, each resource metrics is compared with the threshold to check whether the service satisfies the threshold. If it is below the threshold, an alarm is generated for early remedial action.

An early warning service management architecture is proposed by [2] that helps the cloud consumer select a suitable provider. By using exponential smoothing and an autoregresive integrated moving average, they predict the QoS for a future time interval. Runtime QoS values are compared with the predicted ones to identify variations and service violation. If there is a variation in the SLO parameters, the system generates an early warning to alert the consumer to a possible future violation.

Recursive least square and the autoregressive process method is used to predict future resource usage and virtual machine allocation [4]. The authors proposed a method that helps the cloud provider manage resources by considering QoS objectives and resource utilization cost. The approach can predict SLA violation but is unable to prevent it.

[5] proposed an approach that prevents SLA violation by cross-layer adaptation. They assumed service-based architecture includes the provider, consumer as well as a third-party service. The monitoring component monitors the QoS and compares it with the expected result. When it detects a violation, the service composition is adapted to avoid the delay difference of the service-based application instance, SLA renegotiation or service infrastructure adaptation.

[6] proposed a middleware architecture for service providers that can dynamically configure and manage resources to respond to a consumer's request for an application. The architecute is able to change the amount of resources dynamically with changing QoS requirements. The proposed architecture compares the QoS that it delivers against the predefined values in SLA. In the case of violation, the architecture reconfigures itself dynamically to add additional resources to meet the requirements.

The Workload Analyzer in the Cloud-TM project was proposed by [7]. The proposed architecture monitors resource data and is able to anticipate future workload fluctuations for SLA violation prediction. The Workload Analyzer monitors and classifies the consumption of data, both at the infrastructure and platform layers. It combines all the data from different nodes of the Cloud-TM platform, then filters and correlates them. Once data is gathered, it makes a complete workload outline of all applications, describing the current and future requirements for hardware and software resources. Using different statistical functionalities, it is able to predict future trends in workload variations. Based on the predicted data, the Workload Analyzer generates an alert to alarm users to possible violations of SLA.

## 3. SLA Violation Prediction Architecture

In this section, we propose our SLA violation prediction architecture that predicts SLA violation from a provider's perspective presented in Fig 1.

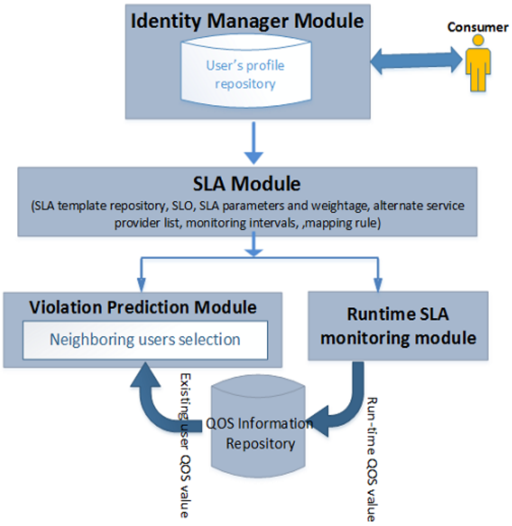
Identity Manager Module
User's profile repository
Consumer
SLA Module
(SLA template repository, SLO, SLA parameters and weightage, alternate service provider list, monitoring intervals, ,mapping rule)
Violation Prediction Module
Neighboring users selection
Runtime SLA monitoring module
Existing user QOS value
QOS Information Repository
Run-time QOS value

Figure 1: Profile-based SLA violation prediction architecture

The system comprises different components that consider a consumer's or the top-K nearest neighbours' previous resource usage profile to predict SLA violation. The functionality of each component is described below:

- Identity Manager Module: this module is responsible for the authentication and identification of the requesting consumer.
- SLA Module: this is responsible for finalizing the SLA and its objectives.
- Violation prediction module: this is a key module which is responsible for predicting SLA violation based on the consumer's or its nearest neighbor previous resource usage profile.
- Runtime SLA Module: this is responsible for monitoring the runtime behavior of consumers and providers, finalizing the SLA and its objectives.
- Resource usage information repository: this module contains the previous resource usage profile of all consumers.

## 4. Violation Prediction Approach

Most of the existing approaches predict SLA violation from a consumer's perspective, however it is as important to predict violation from a provider's side too. Providers want to know the expected resource usage of consumers so that they can manage their resources in order to avoid violation and penalties.

A consumer's previous resource usage profile can be taken into consideration for predicting SLA violation. Therefore, it is important that a provider has the previous resource usage profile history of a given user or its nearest neighbours. It is assumed that a consumer who has a previous record of service violation can be expected to violate in the future too. Prediction for given requesting consumer is represented in Fig 2. By proceeding with this assumption, there may be one of two conditions:

- The requesting consumer has a previous resource usage profile history. The provider considers the previous resource usage profile to determine the likelihood of service violation.
- The requesting consumer is new and does not have any previous transaction record. In this case, the provider selects the nearest neighbours and calculates their previous resource usage profile. This can be done by using the weighted average of the top-K nearest neighbours.

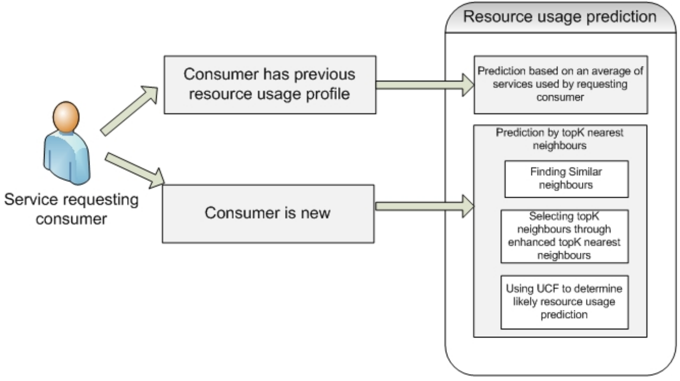
Service requesting consumer
Consumer has previous resource usage profile
Consumer is new
Resource usage prediction
Prediction based on an average of services used by requesting consumer
Prediction by topK nearest neighbours
Finding Similar neighbours
Selecting topK neighbours through enhanced topK nearest neighbours
Using UCF to determine likely resource usage prediction

Figure 2: Method for resource usage prediction

### *4.1 Similarity computation*

Before predicting cloud resource usage, we need to identify all the nearest neighbours of a given consumer, because the degree to which the consumer and its neighbours are similar can significantly affect prediction. We use the Pearson Correlation Coefficient (PCC) [8] to determine the nearest neighbours. PCC is widely used in different recommender systems due to its accuracy and precision. PCC calculates the similarity between requesting consumer x and its neighbour y by the following equation 1:

$$\boldsymbol{Sim(x,y)} = \frac{n(\sum xy)-(\sum x)(\sum y)}{\sqrt{[n\sum x^2-(\sum x)^2][n\sum y^2-(\sum y)^2]}} \quad (1)$$

We assume that similarity is calculated between a consumer and its neighbours based on the basic parameters that the consumer has entered during the registration process.

The top-K nearest neighbours is used to select the most similar neighbours, with the similarity ranging from a positive to a negative value. Prediction is affected significantly by a negative value, therefore we enhanced the top-K nearest neighbour selection to select only positive values. The top-K nearest neighbours for consumer c can be determined by the following equation 2:

$$TNN(c) = \{c_a | c_a \in T_k(c), siml'(c_a, c) > 0, c_a \neq c\} \quad (2)$$

where $T_k(c)$ is the top-K nearest neighbour of consumer c. $TNN(c)$is the enhanced top-K nearest neighbours for consumer c.

### *4.2 Resource usage prediction*

After obtaining the top-K nearest neighbours, we can determine the consumer's expected resource usage. We use the user-based collaborative filtering method (UCF) [8] which is widely used in user-based recommender systems. For a given consumer, we select all the nearest neighbours who have used a similar service for a period of time. Resource prediction by using the nearest neighbours can be determined by the following equation 3:

$$\bar{\mathrm{U}}\mathrm{bi} = \frac{\sum\{c\in \mathrm{N_b} | i\in \mathrm{R_u}\}\ \mathrm{sim(b,c)} * \mathrm{U_{c,i}}}{\sum\{c\in \mathrm{N_b} | i\in \mathrm{R_u}\} |\mathrm{sim(b,c)}|} \quad (3)$$

where $R_u$ is the set of resource usage metrics used by consumer c and $U_{c,i}$ is the usage of resource i by consumer c. Using the weighted sum of resource usage of all nearest neighbours $c\in N_b$ that have previously used resource i can determine the predicted resource usage $\bar{U}bi$ of resource i for consumer b.

## 5. Implementation and Evaluation Of The SLA Vioaltion Prediction Approaches

In this section, we implement and evaluate our SLA violation prediction approach by considering a consumer's previous resource usage profile history in three scenerios:

- The consumer has a previous resource usage profile history. Resources are predicted by considering the previous resource usage of a given consumer for different services for a certain time interval.
- The consumer is new and doesn't have a previous resource usage profile. Resource usage is calcuated by considering the previous resource usage of the nearest neighbours for the same service for a certain time interval.
- By considering both the consumer's and its nearest neighbours' previous resource usage profile for the same service for a certain time period

We use an exisiting dataset from [9]. The dataset comprises 142 users, using 4532 web services for 64 time intervals for the QoS parameters i.e. throughput and response time. We consider throughput for this experiement and evalute our approach by considering the above three scenerios. We use RMSE and MAD metrics to compare the prediction accuracy of the three approaches. RMSE is defined by the equation 4:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}} \quad (4)$$

where $x_i$is the observed throughput value for time interval and $\hat{x}_i$is the predicted throughput.

MAD is defined by the equation 5:

$$MAD = \frac{\sum_{i=1}^{n}|x_i - \hat{x}_i|}{n} \quad (5)$$

The lower the value of RMSE and MAD, the better the prediction accuracy.

### 5.1 Influence of the value of top-K in prediction

To determine the impact of top-K in prediction accuracy, we vary the value of top-K from 2 to 35. We conduct the experiment and obtain the prediction value of throughput for consumer 5, using service number 1942 for 64 time intervals. The prediction results of RMSE and MAD are compared with the actual throughput values. Table 1 shows the values of RMSE, MAD and top-K nearest neighbours. Fig 3 represent RMSE and Fig 4 represent MAD for different top-K nearest neighbours.

Table 1: RMSE and MAD for different top-K nearest neighbors

| TOP-K | RMSE | MAD |
|---|---|---|
| 2 | 0.820856 | 0.60703 |
| 3 | 0.802337 | 0.635128 |
| 4 | 0.793873 | 0.641566 |
| 5 | 0.860021 | 0.690758 |
| 6 | 0.694338 | 0.579813 |
| 7 | 0.724104 | 0.607998 |
| 8 | 0.686049 | 0.5831 |
| 9 | 0.668933 | 0.550948 |
| 10 | 0.707908 | 0.628897 |
| 11 | 0.735434 | 0.657866 |
| 12 | 0.764868 | 0.681089 |
| 13 | 0.759547 | 0.678688 |
| 14 | 0.73903 | 0.655641 |
| 15 | 0.773842 | 0.68822 |
| 16 | 0.777221 | 0.690981 |
| 17 | 0.765566 | 0.678838 |
| 18 | 0.763594 | 0.676608 |
| 19 | 0.826001 | 0.735019 |
| 35 | 0.815077 | 0.722747 |

From the results, we observe that by keeping the value of 9 for top-K, we obtain an optimal result. Therefore, for the rest of the experiment, we keep the value of the top-K nearest neighbours as 9 to obtain the best result.

### 5.1 Prediction by a consumer's previous resource usage profile only

First, we randomly select any user from 1 to 142. In our approach, we select consumer 5 and the last ten services from the expected one to observe the usage behavior of the consumer for 64 time intervals. We obtain the commulative throughput value by averaging the throughput value for each time interval. We then compare the predicted result with the eleventh service to determine the accuracy of the prediction. For comparision, we use RMSE and MAD. Table 3 and Fig 5 shows the predicted and actual throughput values for 64 time intervals.

### 5.2 Prediction by top-K NN excluding the consumer's resource usage profile

In this experiment, we consider the top-K nearest neighbours for consumer number 5 using service 145. We remove the data for consumer 5 and then use PCC to identify the nearest neighbours. The top-K nearest neighbours with the PCC values are shown in Table 2 and Fig 6. Using the weighted average of the throughput values from all the selected nearest neighbours, we can predicte the likely throughput values for 64 time intervals. We then compare the predicted throughput values with the actual throughput values for consumer 5 using service 145. Table 3 and Fig 7 shows the predictd and actual throughput values for the actual vs predicted throughput for 64 time intervals.

### 5.3 Prediction by a consumer's resource usage profile and top-K nearest neighbours

In this experiment, we predict the throughput values for consumer 5 using service 145. We consider both the throughput values of consumer 5 and its nearest neighbours. Using the weighted average, we obtain the predicted throughput values for 64 time intervals. We then compare the results of the predicted throughput values with the actual throughput values of user 5 using service 145. Table 3 and Fig 8 shows the predicted and actual throughput values for 64 time intervals.

Table 2: Top-K nearest neighbors with PCC values

| # | Top-K user ID | PCC |
|---|---|---|
| 1 | 106 | 0.9999779 |
| 2 | 94 | 0.9999685 |
| 3 | 34 | 0.9999406 |
| 4 | 82 | 0.9999311 |
| 5 | 63 | 0.9997260 |
| 6 | 23 | 0.9992655 |
| 7 | 92 | 0.9992068 |
| 8 | 81 | 0.9992068 |
| 9 | 15 | 0.9992068 |

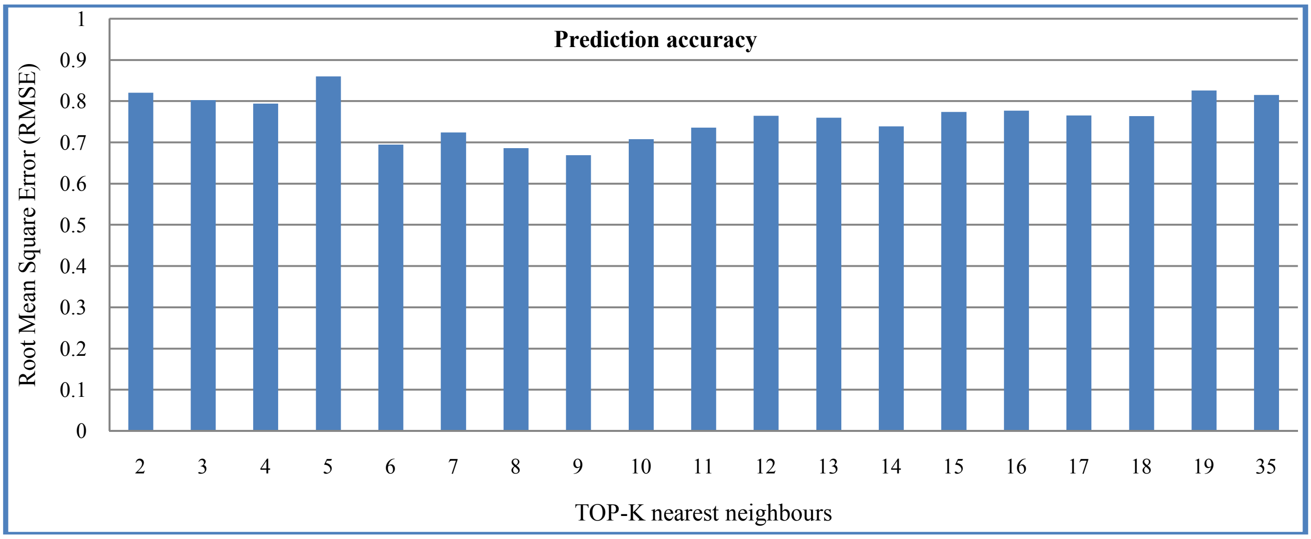

Figure 3: RMSE with different top-K values

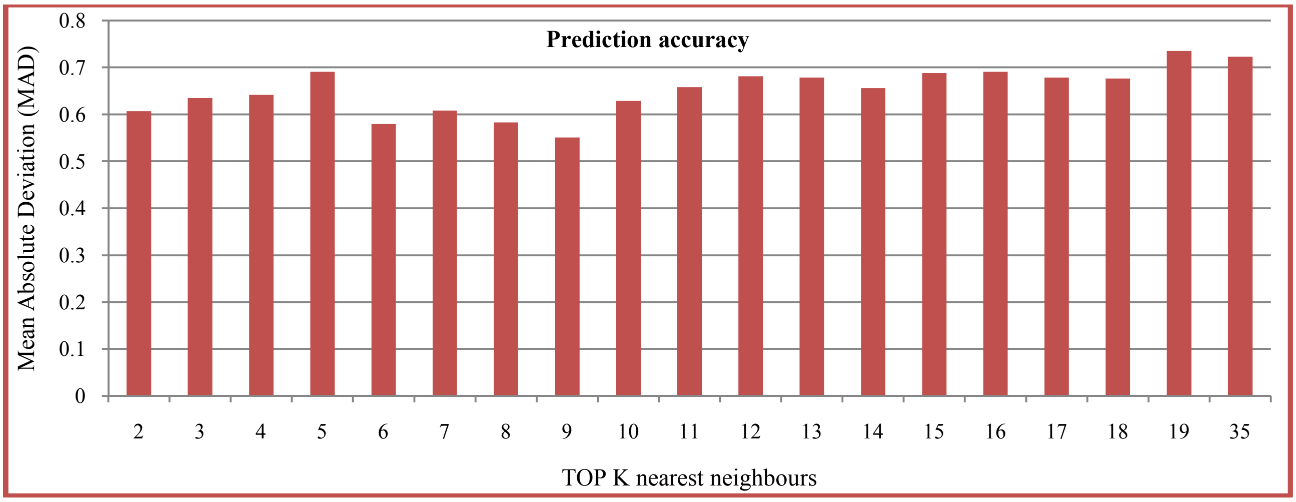

Figure 4: MAD with different top-K values

Table 3: Throughput value from different prediction methods

| S.No | Actual throughput | Prediction only by consumer profile for different services | Prediction by nearest neighbours only | Prediction by nearest neighbours and consumer profile | S.No | Actual throughput | Prediction only by consumer profile for different services | Prediction by nearest neighbours only | Prediction by nearest neighbours and consumer profile |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2.9750 | 1.0214 | 0.5010 | 0.7489 | 33 | 2.6842 | 1.5707 | 3.3675 | 3.3295 |
| 2 | 3.0000 | 1.2617 | 2.9074 | 2.9167 | 34 | 1.0689 | 1.4334 | 2.0218 | 2.0954 |
| 3 | 3.1043 | 1.7508 | 3.0665 | 3.0703 | 35 | 2.8560 | 1.3099 | 1.8874 | 1.7964 |
| 4 | 3.0254 | 1.7592 | 2.5640 | 2.6102 | 36 | 1.9944 | 1.5843 | 1.8147 | 1.9304 |
| 5 | 1.2982 | 1.5915 | 1.9278 | 1.8648 | 37 | 2.6642 | 1.5210 | 1.7939 | 1.8162 |
| 6 | 3.0254 | 1.6690 | 1.6101 | 1.7517 | 38 | 3.1316 | 1.3827 | 1.8740 | 1.9618 |
| 7 | 3.1316 | 1.4992 | 2.4291 | 2.4994 | 39 | 1.0884 | 1.5271 | 2.0913 | 2.2069 |
| 8 | 3.0513 | 1.7568 | 2.8751 | 2.8927 | 40 | 3.1593 | 1.3945 | 2.4104 | 2.2635 |
| 9 | 3.0513 | 1.4711 | 2.2936 | 2.3694 | 41 | 2.3333 | 1.2588 | 1.5972 | 1.7709 |
| 10 | 3.0000 | 1.6408 | 2.2687 | 2.3419 | 42 | 2.2037 | 1.7534 | 1.9613 | 2.0027 |
| 11 | 2.9024 | 1.4568 | 1.9604 | 2.0546 | 43 | 1.5063 | 1.4182 | 2.7335 | 2.6746 |
| 12 | 2.9750 | 1.8379 | 1.9599 | 2.0615 | 44 | 3.1043 | 1.3649 | 1.5862 | 1.5773 |
| 13 | 1.2353 | 1.4275 | 2.1450 | 2.0540 | 45 | 2.9504 | 1.7047 | 1.9773 | 2.1026 |
| 14 | 2.5870 | 1.1348 | 1.7139 | 1.8013 | 46 | 3.1043 | 1.4486 | 1.6076 | 1.7569 |
| 15 | 3.0000 | 1.5833 | 1.9832 | 2.0849 | 47 | 1.8030 | 1.5994 | 2.1865 | 2.2885 |
| 16 | 3.1043 | 1.8665 | 2.9983 | 3.0089 | 48 | 3.1316 | 1.4182 | 2.9677 | 2.8382 |
| 17 | 2.4286 | 1.8653 | 2.2695 | 2.2854 | 49 | 2.9262 | 1.5220 | 2.1530 | 2.2754 |
| 18 | 1.3029 | 1.4212 | 1.9122 | 1.8512 | 50 | 3.2162 | 1.1705 | 2.0907 | 2.1952 |
| 19 | 2.1506 | 1.4456 | 2.0616 | 2.0705 | 51 | 2.9504 | 1.5888 | 2.3765 | 2.4815 |
| 20 | 2.9504 | 1.3571 | 2.1178 | 2.2011 | 52 | 2.8110 | 1.3253 | 2.5565 | 2.6058 |
| 21 | 2.9750 | 1.9532 | 2.4055 | 2.4625 | 53 | 2.9750 | 1.5728 | 2.8483 | 2.8437 |
| 22 | 1.1743 | 1.4807 | 2.4953 | 2.3632 | 54 | 3.0776 | 1.5566 | 2.2418 | 2.3335 |
| 23 | 2.9750 | 1.3870 | 2.2910 | 2.3594 | 55 | 1.2615 | 1.3764 | 2.6800 | 2.7297 |
| 24 | 2.6058 | 1.4840 | 3.1437 | 3.0899 | 56 | 2.9750 | 1.7808 | 2.0311 | 1.9211 |
| 25 | 2.4452 | 1.7113 | 2.6940 | 2.6691 | 57 | 2.9750 | 1.4646 | 1.6986 | 1.8810 |
| 26 | 2.6842 | 1.5564 | 1.6976 | 1.7963 | 58 | 2.9750 | 1.2768 | 2.6264 | 2.6762 |
| 27 | 3.0000 | 1.6004 | 1.8451 | 1.9606 | 59 | 2.4621 | 1.1639 | 4.3172 | 4.1254 |
| 28 | 2.7674 | 1.8919 | 1.6445 | 1.7569 | 60 | 1.2842 | 1.3873 | 2.6941 | 2.6609 |
| 29 | 2.9750 | 1.8880 | 2.6344 | 2.6685 | 61 | 2.4621 | 1.7417 | 2.2842 | 2.1412 |
| 30 | 2.9504 | 1.6943 | 1.4941 | 1.6398 | 62 | 2.9750 | 1.3651 | 1.2192 | 1.3968 |
| 31 | 1.0851 | 1.7375 | 2.7154 | 2.5523 | 63 | 3.0000 | 1.7325 | 0.9337 | 1.2254 |
| 32 | 2.9504 | 1.6476 | 1.6703 | 1.8126 | 64 | 2.6842 | 1.6860 | 2.2549 | 2.3614 |

Table 4: RMSE and MAD of different prediction methods

|  | Prediction by consumer profile for different services | Prediction by nearest neighbors only | Prediction by nearest neighbors and consumer profile |
|---|---|---|---|
| **RMSE** | 1.254443804 | 0.94899142 | 0.841908455 |
| **MAD** | 1.155383125 | 0.81080156 | 0.719467188 |

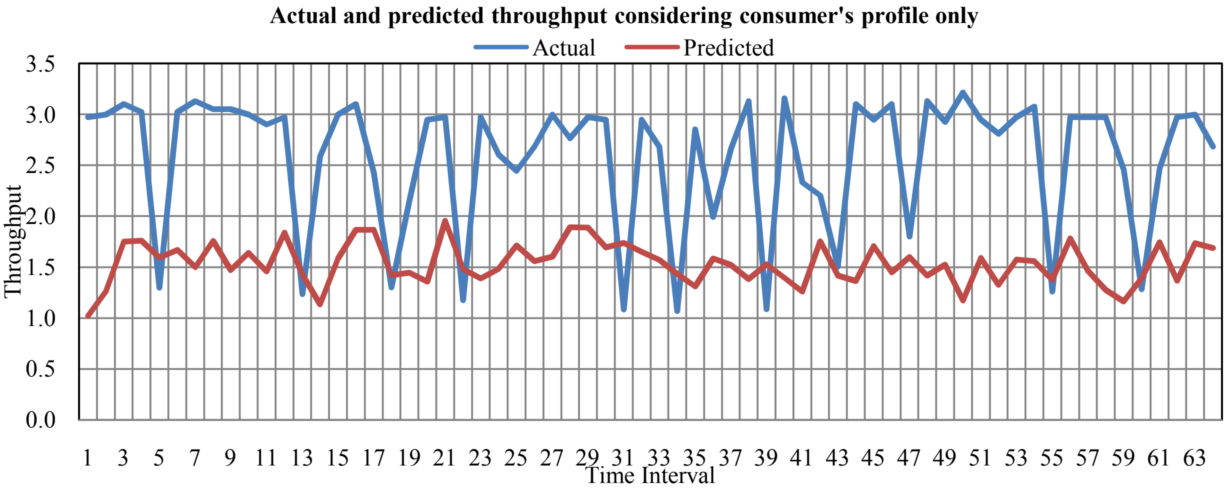

Figure 5: Actual and predicted throughput by considering a consumer’s previous resource usage profile only

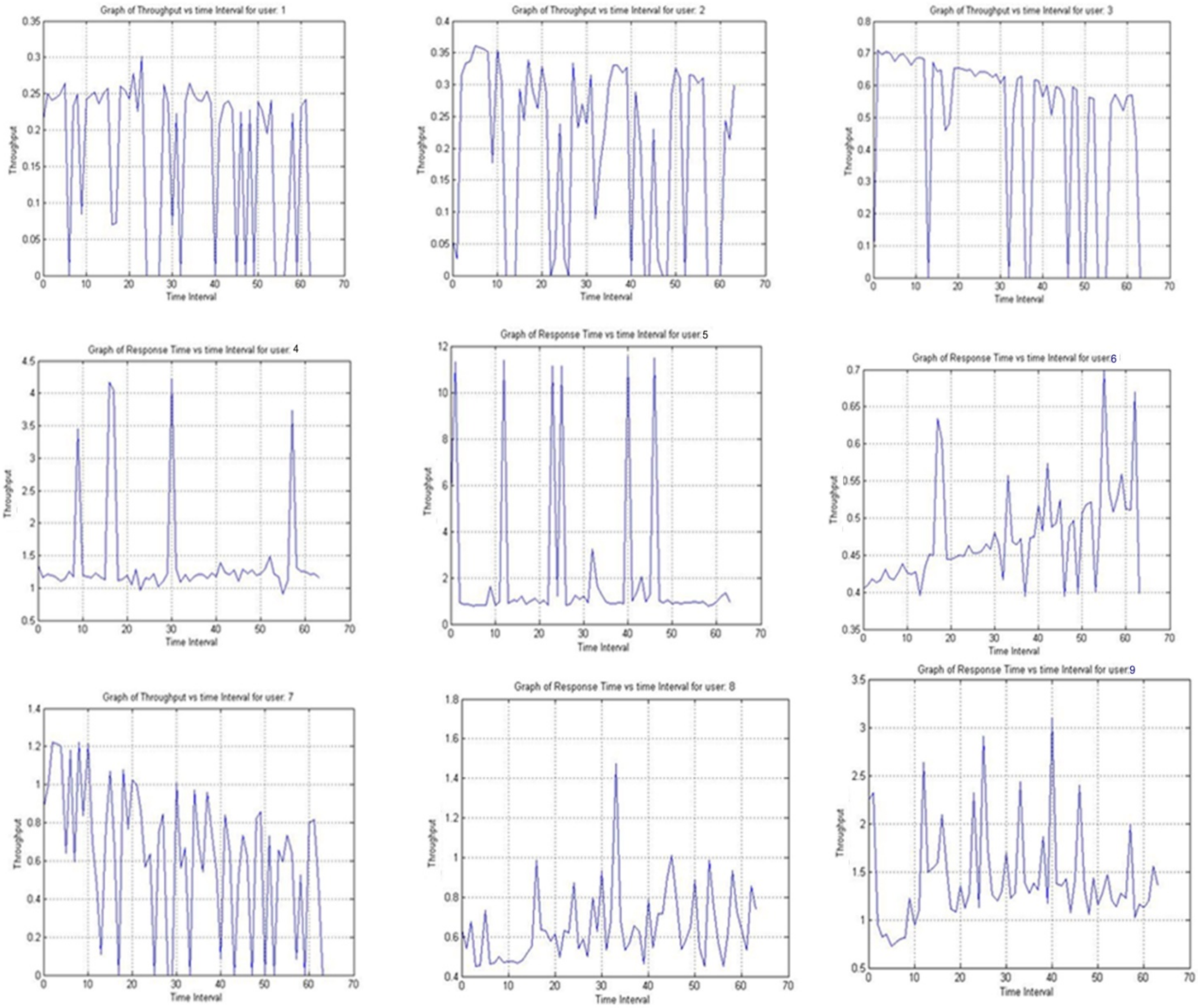

Figure 6: Top-K nearest neighbors with their throughput values

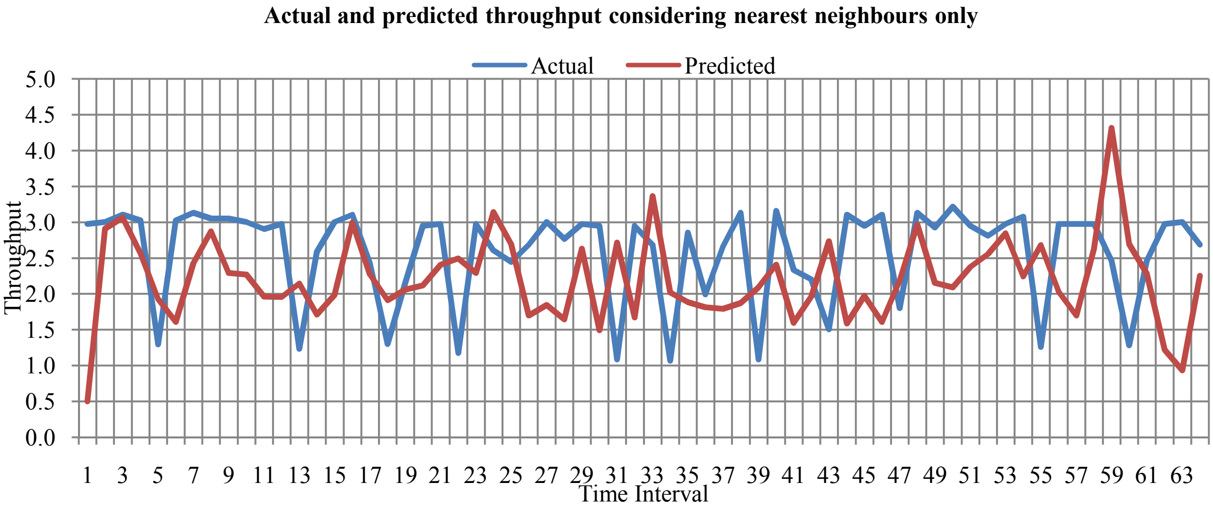

Figure 7: Original and predicted throughput by considering a consumer’s top-K nearest neighbours only

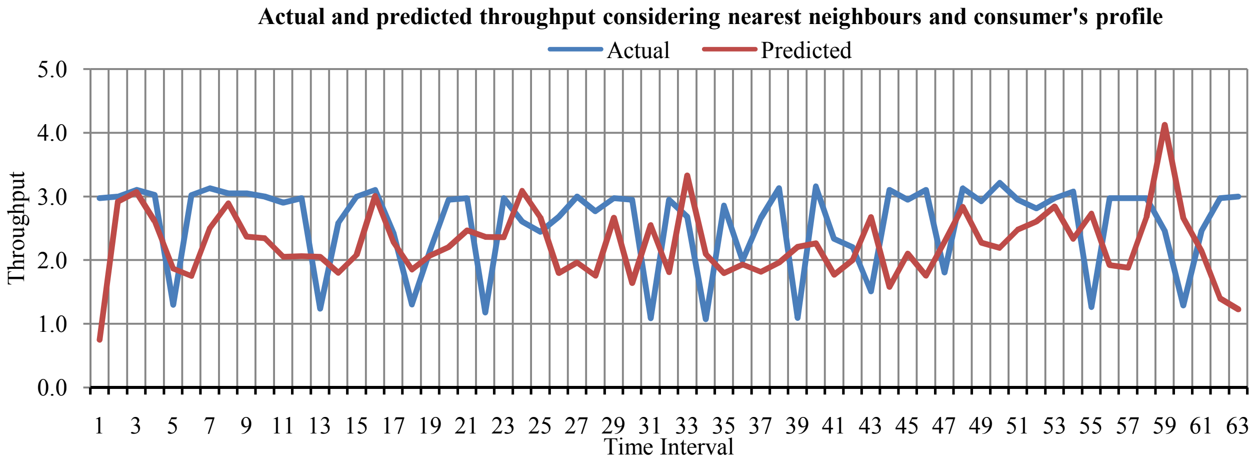

Figure 8: Actual and predicted throughput by considering a consumer’s profile and its nearest neighbor’s profile

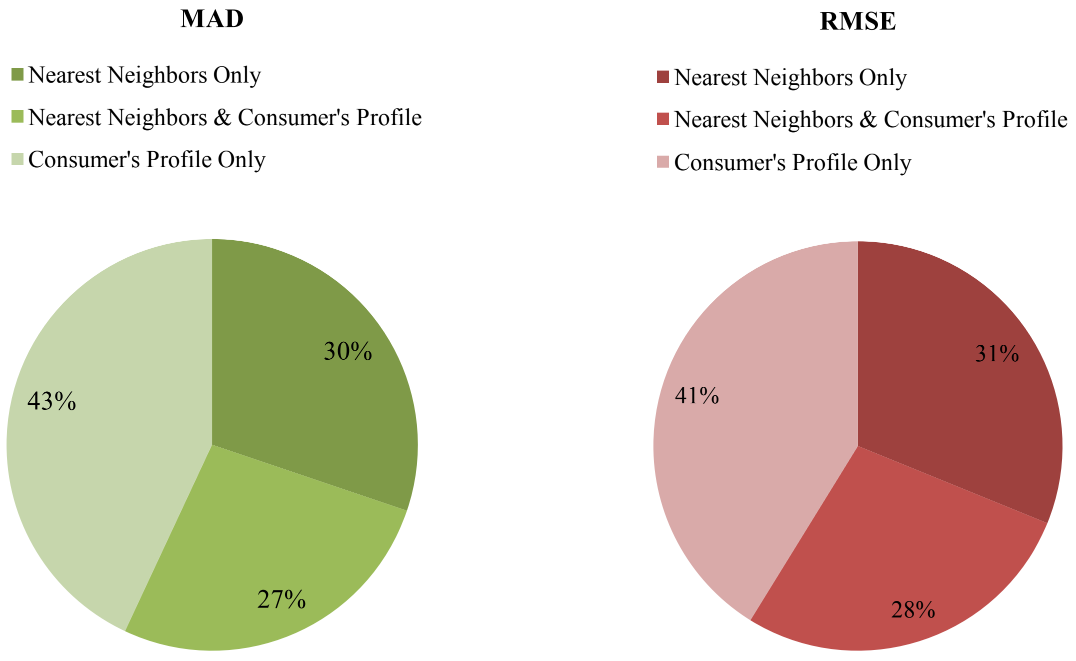

Figure 9: MAD and RMSE with different prediction approaches

To determine the prediction accuracy of each approach we evaluate the results of each method by using the benchmark of MAD and RMSE as shown in table 4 and Fig 9. We get most optimal result by considering consumer's and its nearest neighbours previous resource usage profile i.e. 27% with MAD and 28% with RMSE which is less than the results obtained from the nearest neighbours' resource usage profile i.e. 30% with MAD and 31% with RMSE, and the consumer's resource usage profile, which is 43% with MAD and 41% with RMSE.

Based on the above results, we can say that when we consider both the requesting consumer's previous resource usage profile history along with the previous resource usage history of its nearest neighbours, we obtain the most optimal output.

## 6. Conclusion

In the cloud computing environement, it is very important for a provider to know the likely usage pattern of the consumer before the commencment of any service so it can manage its resources accordingly to avoid any violation and penalities.

A consumer's previous resource usage profile is an important aspect for predicting likely service violation. A consumer who has a previous service violation history is assumed to violate again in the future. Proceeding with this assumption, we predict throughput in three scenerios where we first consider a consumer's previous resource usage profile only; second, we consider all the resource usage profiles of the top-K nearest neighbours; and, third, we consider both the consumer's resource usage profile and its nearest neighbours' profile. From the results, we can see that we obtain optimal prediction results by considering both the resource usage profile of the consumer and its nearest neighbours.

In the future, we will focus on validating the approach by considering different SLO in the SLA.

## 7. References

[1] 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.

[2] 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,* p. bxu064, 2014.

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

[4] 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.

[5] E. Schmieders, A. Micsik, M. Oriol, K. Mahbub, and R. Kazhamiakin, "Combining SLA prediction and cross layer adaptation for preventing SLA violations," 2011.

[6] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, and E. Turrini, "Qos–aware clouds," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 321-328.

[7] 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.

[8] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, pp. 43-52.

[9] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for Web services," in *Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on*, 2011, pp. 210-219.