Fuzzy Logic Based Detection of SLA Violation in Cloud Computing- A Predictive Approach

Prabhat Kumar Upadhyay
Department of Electrical and Electronics Engineering, Birla Institute of Technology Mesra, Ranchi, India
uprabhatbit@gmail.com

Archana Pandita
Department of Computer Science and Engineering, Birla Institute of Technology, offshore campus, RAK, UAE
archanapandita86@gmail.com

Nisheeth Joshi
Department of Computer Science, Banasthali Vidyapeeth, Rajasthan, India
jnisheeth@banasthali.in

Scheduling of a large number of submitted tasks is a central operation in cloud computing. Efficient scheduling and resource allocation for the submitted tasks ensures that Service-Level-Agreements (SLA) violations are minimized. We present a fuzzy logic-based approach for predicting submitted tasks which are likely to encounter SLA violations. It may help Cloud Service Providers (CSPs) to design corrective interventions in terms of additional resource allocation to prevent SLA violations. The proposed mechanism assists in reducing SLA violations and improves the end-user quality-of-service experience along with enhancement of CSP revenues. The appropriate selection of performance metrics has enabled the proposed model to achieve the highest classification accuracy of 92.6 percent in predicting SLA violation

Keywords: Cloud Computing, Service Level Agreement, Violation, Prediction, Fuzzy Logic.

1. INTRODUCTION

Over the past few years, cloud computing has evolved rapidly from large scale distributed computing, concepts of the grid and parallel computing and utility computing as given by Bohm et al. [2010]. It relies on the sharing of computing resources rather than having the local servers and private devices to be used by the applications. Cloud computing provides services to customers over the network and can scale up or down as per the customer requirements. These services as explained in Dhar [2012] are provided by the third-party provider known as Cloud Service Provider (CSP) who owns the infrastructure to support the services and applications which can be used by the customer. It enables the customers to have access to a shared pool of resources such as network, storage, server, services and applications which are provided with minimal management effort by the service providers. The resources and services are distributed over the web, making it more flexible, easier and low cost. Benefits of cloud computing include maintenance, reliability and cost-saving. Services using more than one redundant sites can aid enterprise continuity and recovery after a disaster. The barrier to access new services are lowered and requires few in-house IT resources to support the system.

The services provided by the service providers are classified into three models explained in Gorelik [2013] focused on how services are provided to the users. Services models are differentiated from one

another based on their levels of abstraction. Platform as a service (PaaS) service model works without any need of installing or downloading software and offers all the tools needed to build the applications and services fully using the internet. It includes application design, development and hosting. Users dont need to worry about buying hardware and software or appoint specialists for maintaining and managing them. Another model, Software as a Services (SaaS) facilitates the users to use various software programs, operating systems and resources without having to install them on the machine. In this model, an application is hosted as a service and users access it through the internet. Infrastructure as a Service (IaaS) is based on virtualization technology used in the sharing of hardware resources to perform the services. The key objective is to make it easier for applications and operating systems to access resources such as servers, networks and storage. Thus, it offers basic infrastructure and uses APIs for interactions with hosts, switches and routers.

In Liu et al. [2011] and Ardagna et al. [2014], QoS refers to the level of performance, consistency and availability that an application and the system or network offers. The minimum QoS is guaranteed between the enterprises and service providers which necessitates the legal agreement known as Service Level Agreement. Availability of CPU, network and data storage during peak hour are the QoS parameters stated in SLA. For example, it may state that the availability of the server machine must be 99.99% during peak hours and 85% during non-peak hours. Another SLA may state that any problem reported needs to be addressed within 10 minutes in peak hour and within 1 hour in non-peak hours. Any deviation to the agreed QoS parameters is subject to the penalty in terms of money or extra service. For example, in Leitner et al. [2013a] the penalty clause in SLA may state that 1000 USD has to be paid for every minute the service level objective is breached .

In the customers perspective, if the agreed parameters are surpassed, it may result in the downgrading of services and which in turn affects the business of enterprises. In the service providers perspective, frequent violation in SLA can cause monetary penalty and damage to their image. Therefore, the mechanism for detection of SLA and monitoring the performance of the cloud service is required. Chana and Singh [2014] have established the impact of SLA violation and presented an architecture to provision resources optimally so that SLA can be achieved and hence penalty can be avoided. Leitner et al. [2013b] have shown different examples depicting how penalties are captured and how it leads to reduced customer satisfaction. There are many other studies which have proved the significance of customer satisfaction in the service industry for example one given by Yeo et al. [2015].

Hence, anything that causes an interruption to the availability of services leads to SLA Violation which needs to be tackled. In this paper, we have proposed a machine learning-based model built using fuzzy inference system (FIS), which can predict if a task generated in the system can result in a violation. In addition to simplicity and flexibility, FIS is efficient in handling the problems related to incomplete and imprecise data. Making use of interpretation ability of FIS, this model will help CSPs to take proactive action to avoid the SLA violation and hence keep the service level objective intact.

2. BACKGROUND AND RELATED WORK

Technological advancements have given rise to innovative and complex cloud systems changing throughout the decades. For example, the problem caused by the adaptive survival migration of virtual machines in intra-cloud load balancing with significant hosts was addressed by Zhao and Huang [2009]. The author proposes a load balancing model at the migration time of virtual machines based on their processor or IO use, and reports zero-downtime in the cycle of virtual machine migration. The algorithm ensures that VM migration takes place from high-cost physical hosts to low-cost host at all times. It also assumes each physical host to have sufficient memory which may be regarded as weak assumption. Bhadani and Chaudhary [2010] presented a Central Load Balancing Policy for Virtual Machines (CLBVM) balancing the load equally among the virtual machines in a cloud distributed environment. This technique improves the system performance, but system fault tolerance was not taken into account . Bag-of-Tasks proposed in Benoit et al. [2007] schedules multiple applications that are a series of similar and separate tasks operating on a heterogeneous system of master-slave. It minimizes the ratio of the actual time taken by the system

to perform the task and the time taken to perform alone**.** In Varalakshmi et al. [2011], a workflow-based scheduling algorithm is introduced to find a solution that considers the user's task's service quality concerning the user's preference. It provides process planning which reflects the significant improvement in CPU utilization and illustrates the efficient improvement in task allocation and execution. Hu et al. [2013], showed that their modification on a generic Support Vector Regression (SVR) algorithm would lead to a specific CPU Load forecast which can be used to make better use of resources .

In Serrano et al. [2013], authors have proposed a mechanism of integrating quality of service and SLA. They have also introduced a new language to explain QoS oriented SLA associated with cloud services. However, few significant QoS metrics like throughput is neglected. Michlmayr et al. [2009] provided a system for tracking QoS on both client-side as well as the server-side . The framework uses its event detection capability of finding the existing QoS values and possible SLA breaches. In **?**] puts forward a model which resolves the problem of making choices between the alternatives when several points contradict while considering all options that best fits the users requirement. A mechanism for resource allocation is also proposed by Wu et al. [2011], emphasizing SaaS providers reduced SLA violation and infrastructure costs. It guarantees that the SaaS providers are capable of managing the instant change of users demand.

In Xu et al. [2011], Berger model is proposed for the first time to address the equality of the task distribution. Based on distributive justice's social theory, it defines the dual restriction in which the activities are graded according to the criteria of the quality of service and specifies the role of the justice assessment to determine the fairness of the resource allocation. Researchers in Li and Wang [2014], have proposed the algorithm nn-dwrr which resulted in reduced average response time as compared to traditional capacity-based algorithms which are used to schedule incoming requests for VMs. A management system based on Business Process Execution Language (BPEL) that tracks cloud-based Web services was put forth by authors in Grati et al. [2012]. The framework gathers information, analyzes it, and takes corrective action if breaches are found in SLA. The tracking during runtime is focused on BPEL-generated workflow trends. A framework has been proposed in Mabrouk et al. [2009] to assess QoS by concentrating on network complexity, background sensitivity and device flexibility. It is a linguistic framework of four ontological layers for a network asset, device resource, client system and end-user. Another study is focused on a technique for managing cloud resources based on QoS to ensure a reliable and cost-effective cloud environment. This smart approach focuses on the prevention of SLA violations whereby the self-healing function takes over to optimize resources when there are sudden failures as proposed by Gill et al. [2018]. Researchers in Xiaoyong et al. [2015] presented a framework for calculating the penalty based on the unavailability of resources. Its a business framework for the providers to get the best penalty for their SLAs. However, the framework lacks empirical data to validate the model.

Importance of SLA prediction for both service providers and customers has been established by authors in Upadhyay et al. [2019]. Scaled Conjugate Gradient based classifier intends to predict if an incoming task results in a violation. Different datasets are tested and the accuracy of 96.7% is claimed by the authors. The two-level solution put forward by Vázquez-Poletti et al. [2017] uses approximate computing and reduces the demand for resources. It also guarantees optimal provisioning by using admission control policy. Several researchers have addressed a plethora of applications of machine learning techniques for various predictions related works in cloud computing are proposed by Sandıkkaya et al. [2019], Samir and Pahl [2019] and Moreira et al. [2020]. Machine Learning-based auto-scaling mechanism for time-series forecast is proposed by authors in Moreno-Vozmediano et al. [2019]. In this work, resource allocation is based on processing load, predicted on distributed servers aiming at the optimization of server response time and fulfilment of parameters mentioned in the SLA. Another study by Hussain et al. [2018], compares various machine learning techniques for predicting QoS and ranks them as per the accuracy achieved. Authors claim that the study can be helpful to find the appropriate algorithm based on the available input parameters which may be used to avoid penalties caused because of SLA Violations. The prediction technique and feature selection are integrated into the Intelligent Regressive Ensemble Approach for Prediction (REAP) that further increases the accuracy rate and reduces the accuracy time by authors in Kaur et al. [2019].

Many researchers have put forward different mechanisms to tackle the problem of SLA violation and each one of them has their pros and cons. Literature reveals that problems pertaining to resource allocation of the submitted task are yet to be explored more in terms of prediction of SLA violation. Few models in the literature have also used the attributes such as users information and data generated through experimentation. However, in this work we have considered only those features from the real-world dataset that are related to resource allocation of different tasks.

3. SYSTEM MODEL

System model in the proposed system is shown in figure 1. The SLA provides high-level service/application-related attributes and the service modelling phase provides additional information regarding the system to be implemented in a cloud infrastructure. The cloud service providers analyze their business goals through a process of business modelling to optimize their offerings to the customers. After sufficient analysis, the SLA contract is defined and the costs and penalties are also negotiated along with the quality attributes. Finally, the agreement is enforced through various triggering events. The user starts sending the task for execution which is submitted to the system by the cloud service provider. The incoming tasks are evaluated for the resources they require to satisfy the quality attribute as per agreed SLA. The cloud service provider will use the monitoring mechanism here using fuzzy logic to check the possibility of violation of SLA. Continuous monitoring of resources and SLAs needs to be carried out for smooth functioning and effective operations. The resource provisioning is further handled by cloud hypervisors which are responsible for mapping and scheduling of virtual machines. Allocating the incoming task to a VM is the responsibility of scheduler. It has all the information about the machines and it matches the incoming task to a corresponding VM. Load balancing is also performed by the scheduler so that load can be transferred from overloaded machines to under loaded machine. Every VM maintains a queue of incoming tasks and the length of the queue represents the load on a particular VM.

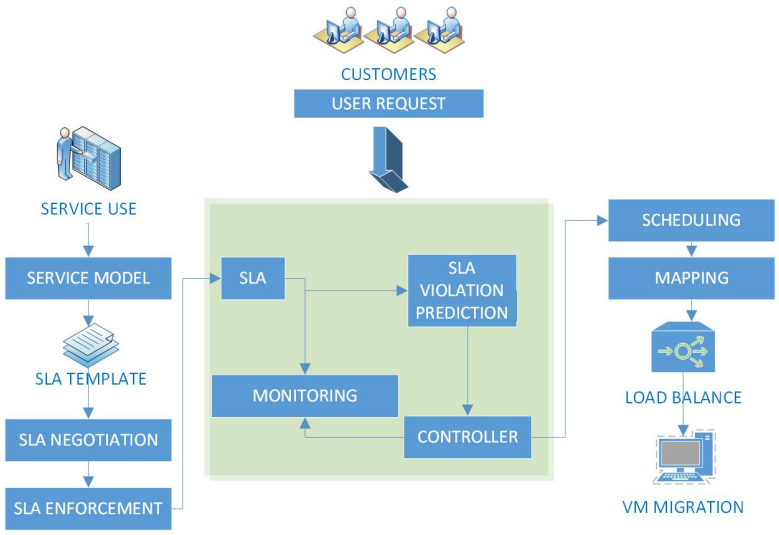


Figure 1: System Model

In this work, we have focused on dynamic monitoring which is important to figure out any possible deviation from the agreed values and detect Violation. This mechanism shall increase the trust between the service provider and the customers. We have used the Fuzzy logic-based model for SLA Violation prediction so that proactive remedial action can be taken. In this proposed model violation prediction is based on the attributes related to the resources requested and assigned. Overview of dataset and methodology along with the proposed fuzzy model is elaborated in sections below.

3.1 Dataset Used

The real-world dataset used in this paper is Google Cluster Trace proposed by Reiss et al. [2011] dataset which contains 29-days trace of Google clusters. A Google cluster is a set of machines which are packed into racks and linked by a high-bandwidth network in the form of clusters. Jobs comprising of one or more tasks are allocated to the machines using the cluster management system. Each task has associated with it the resources requirement which is used for scheduling of tasks onto machines. Every single usage trace is a set of datasets which comprise of few days of workload on each of these machines. The trace data is made public for researchers to explore different types of jobs, scheduling constraints and also the wrong estimation of resource consumption for Googles workload. Linear scaling has been carried out on resource sizes carefully to keep the data usable for research studies. The dataset is derived from a monitoring data which is periodically collected using remote procedure calls. In this work the following tables and attributes have been considered.

Task events table contains attributes like priority, resource request for CPU, resource request for RAM, resource request for local disk space. Each task in the dataset has a priority assigned to it with 0 as the lowest priority. Higher priority tasks get preference for resources over the lower priority tasks. The resource request depicts the limit of the permissible amount of CPU, memory and disk. The tasks that use resources more than the limit can be killed or throttled. Over-commit of resources by the scheduler can be the reason resulting scarcity of resources to meet runtime requests. In such cases, one or more tasks with the lowest priority become the victim and maybe killed. Also, the tasks are permitted to use more resources than what they requested by the runtime environment. Task resources table contains fields like CPU usage, memory usage and disk usage. CPU usage is defined as CPU core seconds per second, memory usage is measured as number pages accessible by the user and blkio system is used to measure the disk usage which represents runtime local disk usage.

3.2 Feature Selection

Any Job requested by the user is divided into different tasks and each task is assigned an ID for its unique identification. The required resources for each job is assigned by the system and hence the list of required memory (MEMR), required CPU (CPUR), required disk (DISKR) is generated. Depending upon the availability of resources, the server assigns the resources to the task in such a way that resources are sufficient for a task to execute gracefully and wastage is also avoided. Since the number of variables is less the dimension reduction is not required and the features are extracted based on their relevance. Priority feature is strongly relevant as priority is directly proportional to the penalty. Higher priority jobs having associated with them a lesser number of resources than required are surely going to be violated. The timestamp is not a relevant feature because the finish time is not known. To find the optimal feature is usually very difficult in real-world and hence we have chosen an approximation of optimal subset.

Scaling of data values of selected features priority, memory, disk and CPU, has been done by dividing each value by their corresponding largest number. We have performed data analysis of features and tabulated as shown in Table 1. The analysis was used further to divide the data points three bands: Low, Medium and High as shown in Table 2.

Table I: Statistical Analysis

Data Set	Parameters	Priority	CPU	Memory	Disk
S1	Min	0	0.0006	0.0009	0.000001
	Max	10	0.18	0.50	0.001
	Mean	8.24	0.065	0.05	0.0003
	Standard Deviation	2.54	0.045	0.037	0.0002
S2	Min	0	0.0006	0.0006	0.000001
	Max	11	0.22	0.50	0.003
	Mean	8.98	0.048	0.02	0.0006
	Standard Deviation	2.88	0.044	0.029	0.0009

Table II: Fuzzy Input Variable

Ranges				Linguistic Terms
Priority	CPU	Memory	Disk	
0 -4	0- 0.07	0- 0.16	0 0.00033	Low
3 - 7	0.06- 0.15	0.15- 0.33	0. 00022 0.00066	Medium
6- 11	0.13- 0.22	0.3 0.50	0.00066 0.001	High

CPU, disk and memory are the main resources assigned to a task when it comes for execution and any deviation between the assigned resources and requested resources are the primary reason for the violation. Comparing both tables of events and usage in the Google Cluster Trace dataset we can detect a possible violation by comparing the requested resources and its mean utilization during the tasks execution. The data is divided into two random sets S1 and S2. S1 has a total of 77 Violations out of total 870 task executions and hence the violation percentage is 9% . S2 has 16 % Violation with 316 Violations in 2000 executions. The description of datasets is given in Table 3.

Table III: Dataset Description

Samples	Total	Violation	Success	Violation percent
S1	870	77	798	9%
S2	2000	316	1684	16%

3.3 Fuzzy Logic

Fuzzy set theory linguistically helps to deal with information and provides systematic calculus which is set by membership functions as shown by Castillo et al. [2007] and Nagpal and Upadhyay [2018]. Human expertise is modelled using if-then rules which is an essential component of the fuzzy inference system. Fuzzy systems can be used to approximate the system's behaviors where numerical functions and analytical functions do not exist. The fuzzy logic comprises four main components as shown in figure 2.

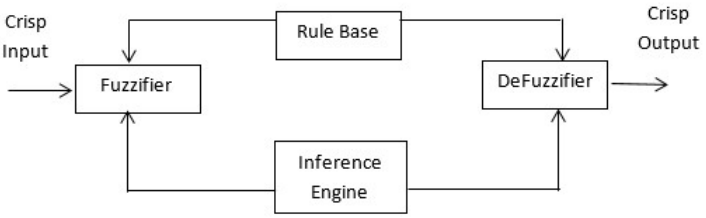


Figure 2: Fuzzy Logic

Rule-Based contains a set of rules and IF-THEN conditions which facilitate decision making. A simple fuzzy rule has the following form

If a is $\widetilde{X}$ then b is $\widetilde{Y}$

Where a is the input, b is output, $\widetilde{X}$ and $\widetilde{Y}$ are the respective fuzzy sets. Fuzzifier converts crisp value input to fuzzy values. A matching degree is determined by the inference engine concerning each rule. The values obtained by the inference engine are converted to crisp value by defuzzifier. The defuzzification techniques viz. mean of maximum, smallest of maximum, the bisector of area are applied at the fuzzy output.

Fuzzy logic deals with uncertainties easily and can work with a different type of input including noisy, distorted and imprecise data. Its based on set theory and the fuzzy set is having a membership degree between 0 and 1. If U is the universe of discourse, a set of ordered pairs $\widetilde{A}$ defines the fuzzy set as given in equation 1. When the universe of discourse is discrete and finite, a fuzzy set is as represented in equation 3.

$$\widetilde{A} = \left\{\left(x,\ \mu_{\{\widetilde{A}\}}(x)\right) | x \in X\right\} \quad (1)$$

$$\widetilde{A} = \sum_{i=1}^{n} \frac{\mu_{\widetilde{A}}(x)}{x_i} \quad (2)$$

$$\widetilde{A} = \frac{\mu_{\widetilde{A}}(x_i)}{x_i} + \frac{\mu_{\widetilde{A}}(x_i)}{x_1} + \ldots + \frac{\mu_{\widetilde{A}}(x_{ni})}{x_n} \quad (3)$$

$$\widetilde{A} = \frac{\mu_{\widetilde{A}}(x)}{x} \quad (4)$$

3.4 Fuzzy Rules for Classification

Fuzzy rules are integrated within the fuzzy inference engine to predict an effective output, based on the inputs given to the system. In this model, the jobs are submitted to the cluster along with its required resources. In FIS, we have designed the model with four inputs, priority, MEMR, DISKR, CPUR. The server assigns the resources in terms of memory (MEMu), CPU (CPUu), disk (DISKu). While performing the analysis of data, it was noticed that as priority increases, the required amount of resources also increases and hence tends to have more possibility of violation. Mentioned features were chosen because any difference between requested and assigned resources can cause a fault in the task and which in turn affects the SLA. To handle the fuzziness of the system, rules have been incorporated as shown in table 4. The fuzzy rules, based on the fuzzy input variables classify the task into three classes: Definite No (0), Probable Yes (0.5) or Definite Yes (1).

Table IV: Fuzzy rules

Rule	IF	Priority is	AND	CPU or Memory or Disk is	THEN	Violation
1		Low		Low		Definite No
2		Low		Medium		Definite No
3		Low		High		Probable Yes
4		Medium		Low		Definite No
5		Medium		Medium		Probable No
6		Medium		High		Definite Yes
7		High		Low		Probable Yes
8		High		Medium		Definite Yes
9		High		High		Definite Yes

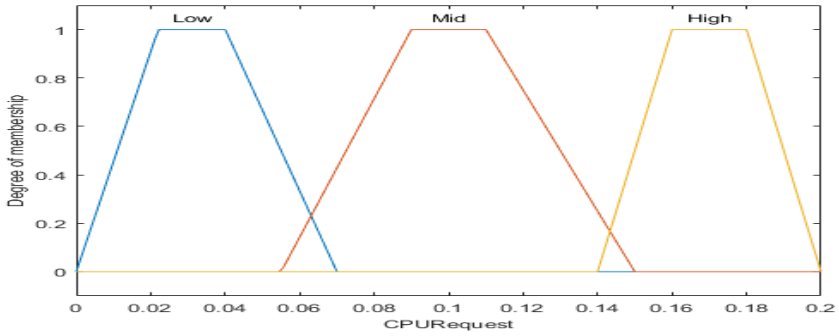
3.5 Membership Functions for Fuzzy Input Variables

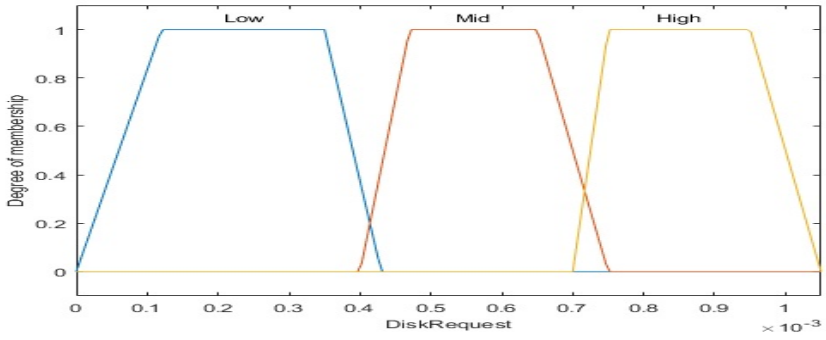
Membership function scales each fuzzy input to a membership grade. Membership function characterizes a fuzzy set and can be expressed as a mathematical formula. In this study, we have used a computationally efficient trapezoid membership function for all four input variables. The trapezoid membership function is used extensively in real-life applications. Fuzzy membership values are computed by using equation 5, where, a, b, c and d are membership function parameters defining the shape of the membership function for the given input x.

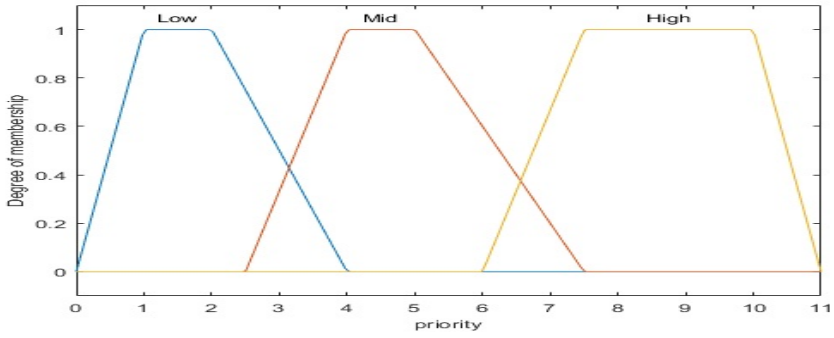
$$f(x; a, b, c, d) = max(min(\tfrac{x-a}{b-a}, 1, \tfrac{d-x}{d-c}), o\ ) \quad (5)$$

Figure 3 shows membership function for each linguistic variable: Low, Medium and High. The curve shown in the figure defines the mapping of each point in the input space to a membership value.

No: 3 starts here


Low
Mid
High
Degree of membership
CPURequest


Low
Mid
High
Degree of membership
DiskRequest
× 10^-3


Low
Mid
High
Degree of membership
priority

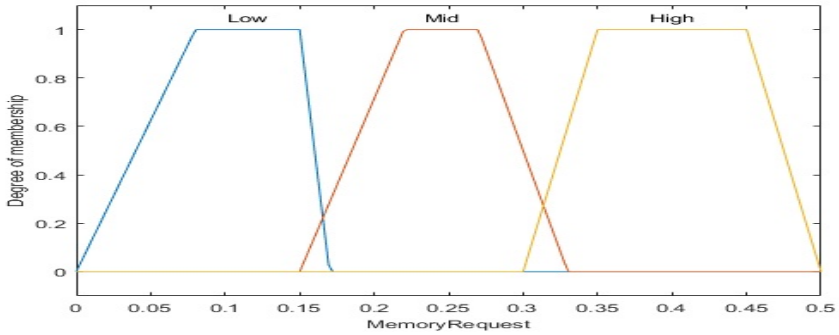


Figure 3: Membership functions- Fuzzy input variables-CPU, Disk Priority and memory

3.6 FIS Implementation

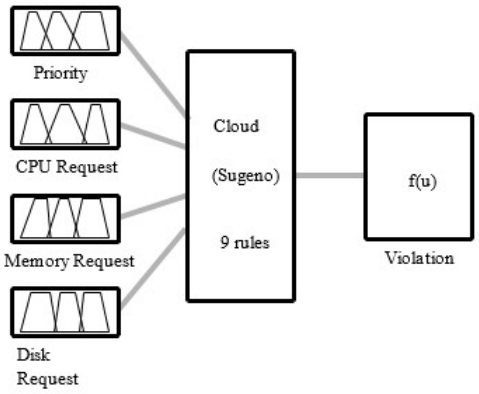


Figure 4: Fuzzy Inference System

Based on fuzzy reasoning, fuzzy if-then rules and fuzzy set theory, a computing framework known as Fuzzy Inference System can be built. It is an aggregation of three components viz. rule base, knowledge-base and reasoning mechanism. Rule base comprises fuzzy rules, a knowledge-based engine facilitates the fuzzy rules with the help of membership functions and the reasoning mechanism derives inferences from the rules.

The FIS used in this work as shown in figure 4 has 4 inputs and 1 output where 9 rules have been embedded in the rule base engine. The implementation of the fuzzy system has been carried out with the codes written and executed in MATLAB 2016.a. The proposed system was modelled using the input variables as shown in table 2 and the fuzzy rules as shown in table 4. To implement the fuzzy inference system, input variables were presented to the system followed by firing the corresponding rules. Finally, the output is estimated in the form of a fuzzy score which is used to classify the task.

4. PERFORMANCE EVALUATION

The analysis performed to evaluate the model is presented in this section. The results obtained are analyzed in terms of fuzzy score and percentage accuracy.

4.1 Fuzzy Score

Membership functions, inputs and constraints are considered by FIS to calculate the fuzzy score. The fuzzy score calculated for a few input values using fuzzy rules is shown in table 5. The values recorded depict that the model can categorize the output appropriately as Definite Yes, Probable Yes or Definite No, which can further be used for decision making in terms of allocation of resources to the incoming task.

Table V: Fuzzy Score Recorded

Input				Output	
Priority	Memory	CPU	Disk	Fuzzy Score	Violation
11	0.424	0.162	0.000961	1	Definite Yes
7	0.33	0.184	0.001	1	Definite Yes
5	0.253	0.109	0.0008	0.5	Probable Yes
3	0.35	0.19	0.001	0.5	Probable Yes
1	0.035	0.024	0.00016	0	Definite No

4.2 Accuracy

Performance of the fuzzy system is evaluated in terms of percentage accuracy. The accuracy of S1 is noted as 91% and S2 is recorded as 88.4% . Since the two samples chosen randomly may be biased in terms of feature space, it may lead to overfitting problem. To verify this we have generated another dataset S3 using SMOTE-Tomek link method of sampling. S3 has 50% Violation with 948 Violations in 1900 data points. The accuracy recorded for S3 is 92.6% which is quite close to the accuracy of S1 and S2. The comparison of the accuracy of three datasets is shown in figure 5 and the values confirm that the proposed model can classify the task with an average accuracy of 90.6% .

Figure 5: Percentage accuracy comparison

5. CONCLUSION

Inference system deduces new information to the knowledge base by applying logical rules. In this work, Fuzzy Inference System has been employed to address an issue of SLA violation by using the real-world dataset to make appropriate predictions based on historical data. The proposed model will help service providers to avoid violations and enhance Quality of Service. Highest accuracy recorded is 92.6% which is very encouraging and signifies that fuzzy logic and fuzzy inference system can be applied for the classification of tasks. Moreover, the testing has been done on real-world datasets hence the model is more suitable to tackle the real-world challenges.

Our future work aims to focus on using other machine learning techniques to further enhance the accuracy and efficiency of the proposed model. We also aim to utilize the obtained decisions to design a mechanism for remedial action so that the possible violation can be tackled.

References

Ardagna, D., Casale, G., Ciavotta, M., Pérez, J. F., and Wang, W. 2014. Quality-of-service in cloud computing: modeling techniques and their applications. *Journal of Internet Services and Applications 5,* 1, 11.

Benoit, A., Marchal, L., Pineau, J.-F., Robert, Y., and Vivien, F. 2007. Offline and online scheduling of concurrent bags-of-tasks on heterogeneous platforms.

Bhadani, A. and Chaudhary, S. 2010. Performance evaluation of web servers using central load balancing policy over virtual machines on cloud. In *Proceedings of the Third Annual ACM Bangalore Conference*. 1–4.

Bohm, M., Leimeister, S., Riedl, C., and Krcmar, H. 2010. Cloud computing and computing evolution. technische universität münchen (tum).

Castillo, O., Melin, P., Kacprzyk, J., and Pedrycz, W. 2007. Type-2 fuzzy logic: theory and applications. In *2007 IEEE International Conference on Granular Computing (GRC 2007)*. IEEE, 145–145.

Chana, I. and Singh, S. 2014. Quality of service and service level agreements for cloud environments: Issues and challenges. In *Cloud Computing*. Springer, 51–72.

Dhar, S. 2012. From outsourcing to cloud computing: evolution of it services. *Management Research Review*.

Gill, S. S., Chana, I., Singh, M., and Buyya, R. 2018. Chopper: an intelligent qos-aware autonomic resource management approach for cloud computing. *Cluster Computing 21,* 2, 1203–1241.

Gorelik, E. 2013. Cloud computing models.

Grati, R., Boukadi, K., and Ben-Abdallah, H. 2012. A qos monitoring framework for composite web services in the cloud. In *In The Sixth International Conference on Advanced Engineering Computing and Applications in Sciences (Advcomp12)*.

Hu, R., Jiang, J., Liu, G., and Wang, L. 2013. Kswsvr: A new load forecasting method for efficient resources provisioning in cloud. In *2013 IEEE International Conference on Services Computing*. IEEE, 120–127.

Hussain, W., Hussain, F. K., Saberi, M., Hussain, O. K., and Chang, E. 2018. Comparing time series with machine learning-based prediction approaches for violation management in cloud slas. *Future Generation Computer Systems 89*, 464–477.

Kaur, G., Bala, A., and Chana, I. 2019. An intelligent regressive ensemble approach for predicting resource usage in cloud computing. *Journal of Parallel and Distributed Computing 123*, 1–12.

Leitner, P., Ferner, J., Hummer, W., and Dustdar, S. 2013a. Data-driven and automated prediction of service level agreement violations in service compositions. *Distributed and Parallel Databases 31,* 3, 447–470.

Leitner, P., Ferner, J., Hummer, W., and Dustdar, S. 2013b. Data-driven and automated prediction of service level agreement violations in service compositions. *Distributed and Parallel Databases 31,* 3, 447–470.

Li, C.-C. and Wang, K. 2014. An sla-aware load balancing scheme for cloud datacenters. In *The International Conference on Information Networking 2014 (ICOIN2014)*. IEEE, 58–63.

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., and Leaf, D. 2011. Nist cloud computing reference architecture. *NIST special publication 500,* 2011, 1–28.

Mabrouk, N. B., Georgantas, N., and Issarny, V. 2009. A semantic end-to-end qos model for dynamic service oriented environments. In *2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*. IEEE, 34–41.

Michlmayr, A., Rosenberg, F., Leitner, P., and Dustdar, S. 2009. Comprehensive qos monitoring of web services and event-based sla violation detection. In *Proceedings of the 4th international workshop on middleware for service oriented computing*. 1–6.

Moreira, R., Silva, F. D. O., Rosa, P. F., and Aguiar, R. L. 2020. A smart network and compute-aware orchestrator to enhance qos on cloud-based multimedia services. *International Journal of Grid and Utility Computing 11,* 1, 49–61.

Moreno-Vozmediano, R., Montero, R. S., Huedo, E., and Llorente, I. M. 2019. Efficient resource provisioning for elastic cloud services based on machine learning techniques. *Journal of Cloud Computing 8,* 1, 5.

Nagpal, C. and Upadhyay, P. K. 2018. Wavelet based sleep eeg detection using fuzzy logic. In *International Conference on Advanced Informatics for Computing Research*. Springer, 794–805.

Reiss, C., Wilkes, J., and Hellerstein, J. 2011. Google clusterusage traces: format+ schema, google inc. *Mountain View, CA, USA, White Paper*.

Samir, A. and Pahl, C. 2019. Anomaly detection and analysis for clustered cloud computing reliability. *CLOUD COMPUTING 2019*, 120.

Sandikkaya, M. T., Yaslan, Y., and Özdemir, C. D. 2019. Demeter in clouds: detection of malicious external thread execution in runtime with machine learning in paas clouds. *Cluster Computing*, 1–14.

Serrano, D., Bouchenak, S., Kouki, Y., Ledoux, T., Lejeune, J., Sopena, J., Arantes, L., and Sens, P. 2013. Towards qos-oriented sla guarantees for online cloud services. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, 50–57.

Upadhyay, P. K., Pandita, A., and Joshi, N. 2019. Scaled conjugate gradient backpropagation based sla violation prediction in cloud computing. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 203–208.

Varalakshmi, P., Ramaswamy, A., Balasubramanian, A., and Vijaykumar, P. 2011. An optimal workflow based scheduling and resource allocation in cloud. In *International Conference on Advances in Computing and Communications*. Springer, 411–420.

Vázquez-Poletti, J. L., Moreno-Vozmediano, R., Han, R., Wang, W., and Llorente, I. M. 2017. Saas enabled admission control for mcmc simulation in cloud computing infrastructures. *Computer Physics Communications 211*, 88–97.

Wu, L., Garg, S. K., and Buyya, R. 2011. Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 195–204.

Xiaoyong, Y., Ying, L., Tong, J., Tiancheng, L., and Zhonghai, W. 2015. An analysis on availability commitment and penalty in cloud sla. In *2015 IEEE 39th Annual Computer Software and Applications Conference*. Vol. 2. IEEE, 914–919.

Xu, B., Zhao, C., Hu, E., and Hu, B. 2011. Job scheduling algorithm based on berger model in cloud environment. *Advances in Engineering Software 42,* 7, 419–425.

Yeo, G. T., Thai, V. V., and Roh, S. Y. 2015. An analysis of port service quality and customer satisfaction: The case of korean container ports. *The Asian Journal of Shipping and Logistics 31,* 4, 437–447.

Zhao, Y. and Huang, W. 2009. Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud. In *2009 Fifth International Joint Conference on INC, IMS and IDC*. IEEE, 170–175.

Dr. Prabhat Kumar Upadhyay obtained his PhD (Technology) degree from Birla Institute of Technology, Meesra, India. He has been working as a faculty in the department of Electrical and Electronics Engineering of BIT Mesra campus and its offshore campuses in the UAE and OMAN for last 16 years. He has published more than 20 research papers in various journals and conferences. His current research interests include brain signals, signal processing and soft computing.

Ms Archana Pandita is working as Assistant professor in Birla Institute of Technology, Offshore campus RAK, UAE. She has received an M.Tech. degree in Computer Science from Kurukshetra University, India in 2011 and BE degree in Information Technology from Jammu University, India in 2007. She is pursuing her Ph.D degree from Banasthali University. She has 9 years of teaching experience and 3 years of industry experience. Her research area includes Cloud Computing. She has published seven research papers in international journals and conferences. She is a IBM Certified Big Data Engineer and Security analyst. Her research areas include DataSscience, Bigdata, Machine Learning and Cloud Computing

Dr. Nisheeth Joshii is an Associate Professor at Banasthali Vidyapith, India. He has done his Ph.D. in, the area of Natural Language Processing. Being involved in teaching and research for over 13 years,, he has developed the art of explaining even the most complicated topics in a straight forward and, easily understandable fashion. He also has vast experience in handling large scale research projects., This has helped him in developing practical insights into complex AI systems., He is the recipient of the prestigious ISTE-U.P. Government National Award for Outstanding Work, Done in Specified Areas of Engineering and Technology. He has authored several papers in, international journals and conferences of repute.