Computers, Materials & Continua
DOI:10.32604/cmc.2022.023480
Article

Tech Science Press

Intelligent Model for Predicting the Quality of Services Violation

Muhammad Adnan Khan[1,2], Asma Kanwal[3], Sagheer Abbas[3], Faheem Khan[4] and T. Whangbo[4,*]

[1]Pattern Recognition and Machine Learning Lab., Department of Software, Gachon University, Seongnam, Gyeonggido, 13120, Korea
[2]Riphah School of Computing & Innovation, Faculty of Computing, Riphah International University Lahore Campus, Lahore, 54000, Pakistan
[3]School of Computer Science, National College of Business Administration and Economics, Lahore, 54000, Pakistan
[4]Department of Computer Engineering, Gachon University, Seongnam, 13557, Korea
*Corresponding Author: T. Whangbo. Email: tkwhangbo@gachon.ac.kr
Received: 09 September 2021; Accepted: 15 October 2021

Abstract: Cloud computing is providing IT services to its customer based on Service level agreements (SLAs). It is important for cloud service providers to provide reliable Quality of service (QoS) and to maintain SLAs accountability. Cloud service providers need to predict possible service violations before the emergence of an issue to perform remedial actions for it. Cloud users' major concerns; the factors for service reliability are based on response time, accessibility, availability, and speed. In this paper, we, therefore, experiment with the parallel mutant-Particle swarm optimization (PSO) for the detection and predictions of QoS violations in terms of response time, speed, accessibility, and availability. This paper also compares Simple-PSO and Parallel Mutant-PSO. In simulation results, it is observed that the proposed Parallel Mutant-PSO solution for cloud QoS violation prediction achieves 94% accuracy which is many accurate results and is computationally the fastest technique in comparison of conventional PSO technique.

Keywords: Accountability; particle swarm optimization; mutant particle swarm optimization; quality of service; service level agreement

1 Introduction

Cloud computing is a novel domain for the implementation of accessible computing resources by integrating the concept of virtualization, distributed application design, and grid computing. The core purpose of this paradigm is its convenient behavior which is provided through its real-time access to the network and configuration of the computing resources. These resources involve hardware (network, servers, and data storage) and software (as service) resources. Cloud service provider facilitates its users to rapidly provisioned and release these resources [1]. In the current era; the demand for cloud systems rapidly increases which is a result that generates a large number of cloud service providers in the market. The interaction between the service provider and the consumer is a transfer of values between the interacting bodies which is called service provisioning [2]. The cloud hypervisors are

virtual machine monitors to handle resource provisioning, this software is used to map or schedule the instantiated virtual machine that lies on the cloud physical servers [3,4]. Through this interaction, a cloud user can deploy their application on the pool of networked resources with practically minimal operating cost in comparison to actual resource purchasing [5]. For example, commercial cloud service providers like Amazon web services, Microsoft Azure, Google App Engine are running their services on millions of physical hosts. Each host is hosting many virtual machines which can be invoked and removed dynamically. These technical and economic aids like the on-demand service of cloud computing increasing the trends of migration from traditional enterprise computing to cloud computing [6].

Service provisioning like all subscription-based services relies on Service level agreements (SLAs) which is a signed contract between the cloud service provider and customer [7]. These SLAs are designed by considering all functional and non-functional requirements to assure the Quality of service (QoS). SLAs contain all the roles and responsibilities of all the interacting parties involved, scope and quality of service, the performance of services, rates of charges of services, time, etc [8]. So, these QoS provide an acceptance ratio of the customer for the cloud services. In case of agreement violation obligations, service charges or penalties will be paid by the service provider. This violation decreases the trust of the customer for migration towards cloud computing services by decreasing the cloud service quality [9]. Cloud service quality consists of many related attributes like availability, speed, accessibility, and response time of the service [10]. These quality attributes are the key factors for the selection of the best cloud service providers among similar functionality service providers. These quality attributes are the basis for the contract between customer and service provider to assure the customer that their service expectations are fulfilled. These contracts will manage by service management systems to assess and monitor the quality of service that is being executed. Some adequate actions will perform to enforce these contracts, for example, upscaling the relevant resources, substitute the erroneous service. This will also define the overall cost that will be paid by the relevant body on the termination of the contract.

QoS includes very significant attributes response time and speed that are explicitly mentioned in the SLA [10]. The nature of service quality is dynamic that changes according to the functionality of the server, so the values of the service quality attribute can be changed without affecting the basic functionalities of the main service [11]. For the assurance of QoS; it is necessary to monitor cloud resources, existing provided data, and SLAs for the effective working of the cloud environment [12]. The dynamic monitoring of provided quality attributes could provide consistent and real-time information about the running services and resources for all types of cloud services: Platform as a service (PaaS), Infrastructure as a service (IaaS), and Software as a service (SaaS) [13]. Any deviation from the agreed quality attribute ranges which are explicitly defined is SLA may become the cause of system failure and dissatisfaction of customers. So, for effective monitoring of cloud services, it is important to detect the SLA violations that occur in the system [13].

The proposed research is focused on cloud QoS violation detection and prediction using the Parallel Mutant-PSO machine learning technique for SaaS's SLA violation using response time, speed, availability, and accessibility.

2 Related Work

The following section provides the literature review and its related work on cloud quality of service for SaaS and also the detection of QoS violations.

2.1 Service Level Agreement for Cloud Base Services

SLA obliges a contract between service providers and service users to define cloud services. SLA lists down the quality attributes, commitments, and guarantees between both parties. These QoS attributes are Service-level objectives (SLOs) that define the criteria to measure QoS metrics [14]. Set of SLOs specify Service level (SL). Cloud service providers define various SLs in a single SLA. These SLA represents a different type of services and also execute at different time durations. If these SLs are violated then downgrading to service may consider. The resultant service provider will execute remedial actions to cover up the existing flaws of the service. So, there is a need to define an effective mechanism to continuously measure cloud service performance and detection of SLA violations. The study shows that several types of research have been conducted in this field. Like, Business process execution language (BPEL) is a framework that manages cloud-based web services [13]. The framework was designed to gather cloud information, detect the SLAs violations and also provide corrective measures to remove these violations. This framework performs monitoring at run-time based on the workflow pattern generated by BPEL.

Michlmayr et al. [15] in another research define a framework that monitors client and server QoS. The framework performs event-based processing to detect the ongoing QoS metrics and possible SLAs violations. On the occurrence of SLAs violation, an adaptive approach will be adopted like hosting a new service instance for the avoidance of undesired QoS that trigger at run-time.

2.2 Cloud Resource Monitoring and Service Violation Detection

An effective monitoring mechanism is required to ensure that software or hardware that is remotely deployed by the cloud service provider is meeting all the quality levels defined in SLAs. This approach will gather QoS parameters values and detect QoS violations [13]. Typically the cloud resource monitoring system on the abnormal working of any cloud service performs the remedial actions to recover the service.

A variety of IT infrastructure tools are available before the evolution of cloud computing. These tools are specially designed to monitor cluster and grid computing. Many of these existing tools are now used in cloud computing to monitor the cloud resources at an abstract level. These monitoring tools collect metrics from all the components and servers playing their role in providing current cloud services and analyses the collected data. These tools generate SLA reports based on noted data. Some cloud service provider provides their service monitoring facility to their consumers for the better management of applications. For example, Azure fabric controller (AFC) is a monitoring service that monitors Azure-based cloud resources, web services, Windows Azure storage facility [16], hosted websites, etc. Amazon cloud watch (ACW) is a service to monitor Amazon web services (AWS) [17]. Some other cloud monitoring services are used to monitors diverse cloud platforms, [18] like Amazon EC2, GoogleApp Engine, Salesforce CRM, S3 Web Services, Rackspace Cloud Microsoft Azure, CloudKick, etc.

2.3 Software as a Service (SaaS's) QoS

International Standard Organization (ISO) 9126 model defines the distinct software quality attributes relevant to cloud services. These cloud services expose their external attributes to their users during the advertising and utilization phase. External attributes like response time, security, accessibility, availability, speed, performance, reliability, etc. Many internal attributes are hidden for a user like changeability. So, ISO 9126 [19] does not define the SaaS's service quality sufficiently. Resource monitoring to achieve quality is a significant procedure in SaaS cloud computing. This monitoring defines how much user expectations have been met. The study shows that response time throughput, and availability is quality attributes that estimate the effectiveness of cloud cost and cloud QoS ranking prediction [11–14]. Consequently, by monitoring these SLAs' metrics degree of QoS no-violation or violation can be predicted.

Cappiello et al. [20] proposed a QoS-based technique to manage cloud resources effectively and reliably. This approach prevents SLA violation on the sudden failure of service using intelligent optimization for the self-healing process. Availability and performance QoS attributes are considered in this research for the violation evaluation. Penalties are calculated by managing this violation data.

Kritikos et al. [21] define a model for QoS monitoring and adaptation. Through this model service quality categorization is proposed in which service performance is a foremost category that defines how well any service performs. One more category that this research considers is response time. Response time is a composite attribute that can be calculated through network latency and delay. The fundamental parameters for this research were performance, throughput, and response time to evaluate QoS parameters, using these parameters performance of cloud service can be predicted.

Alotaibi et al. [22] highlight the functional and non-functional aspects of QoS. This research defines that semantic web and web services description models help any user to select the best quality services. The authors define that response time and throughput are domain-independent parameters that depict QoS. Whereas, service availability is a dependent parameter that is based on security and reliability of service. This paper proposed a classification mechanism for the management of service-based applications.

Brandic et al. [23] determine QoS through perspective awareness and user mobility. This is a semantic model consisting of four ontological layers represented as network, applications, hardware resources, and the end-user layer. Quality attributes response time, speed and throughput define the performance of service. This research presents the QoS knowledge representation as a general framework in the form of the ontology layer. So, this research gives another direction in QoS service engineering.

Truong et al. [24] work on grid computing and present an approach to focus on the comprehensive set of QoS to give the high-level workflow of the system. The author proposed the QoS aware grid workflow language (QoWL) that experimentally specifies the performance of SaaS. Müller et al. [25] present the significance of QoS monitoring for the selection of services. QoS Classification Tree is used to evaluate QoS using response time, availability, and throughput. Noor et al. [26] provide a comprehensive conceptual reference model named SALMonADA that is responsible to get the SLA agreed between both parties and define the explanation of SLA violations. SALMonADA continuously monitors the configurations and through constraint satisfaction problem identifies the SLA violations.

Deep & Machine learning arose over the last two decades from the increasing capacity of computers to process large amounts of data empowered with cloud computing [27,28]. Computational Intelligence approaches like Swarm Intelligence [29], Evolutionary Computing [30] like Genetic Algorithm [31], Neural Network [32], Deep Extreme Machine learning [33] and Fuzzy system [34–38] are strong candidate solutions in the field of the smart city [39–41], smart health empowered with cloud computing [42,43], and wireless communication [44,45,46], etc.

3 Proposed Methodology

This Section presents an overview of the methodology applied for SLA violation detection using the machine learning mutant-Particle swarm optimization (PSO) technique.

Fig. 1 shows the proposed conceptual model for the Detection and Prediction of SLA Violations.

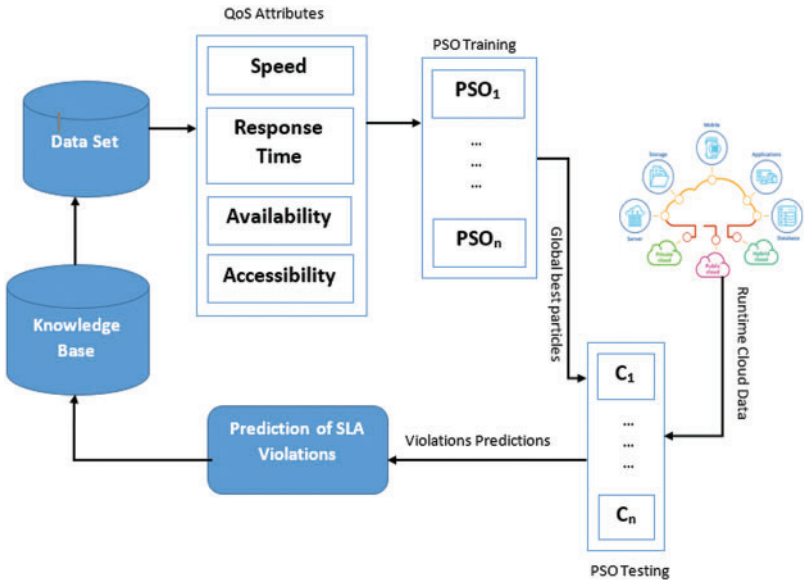

Figure 1: Proposed conceptual model for detection and prediction of SLA violations

The proposed model consists of many modules for the learning state of PSO based on available data in its knowledge base. A part of the dataset is used for the training of PSO. The knowledge base consists of the QoS metrics data. These metrics were transferred to the PSO for training. In the proposed work five Mutant-PSO run parallel for the training of category-based SLA violation. After training the trained PSO will test for the life prediction of SLA violation. The newly predicted rules will send to the knowledge base for the extension of existing knowledge. The response time, speed, availability, and accessibility are the QoS attributes of cloud service that are quantitative value, which does not represent any meaningful terms. The knowledge base for the proposed system is taken from Cloud Armour [27] due to its accommodation of multiple QoS attributes.

3.1 Data Preprocessing

It is a fact that if data is insufficient, the machine learning algorithm may produce inaccurate results or there is the possibility that the machine learning algorithm simply fails to determine anything useful due to this deficiency of data. It is observed that the knowledgebase that is provided having some

missing values in its defined attributes so, these missing values estimation is a valuable part to properly process and learn the class of SLA violation from the given knowledgebase. The proposed approach use is the simple method used to find out the missing value that is represented as α, the system searches for a record having similar values of any attribute β in the observed data and extracts that missing value of α.

A sample consists of a part of the dataset for the training of the Mutant-PSO to learn the algorithm for the five classes defined in Tab. 1.

Table 1: QoS attributes classes

Class name	Value
Strongly weak	0–0.99
Weak	1–1.99
Medium	2–2.99
Strong	3–3.99
Very strong	4–5

3.2 Learning Mechanism for Particle Swarm Optimization

PSO is an optimization algorithm based on the intelligent behavior of the particles to reach an optimized goal [28]. The group of particles is initially generated and their corresponding velocities and next positions are defined. Each particle is considered as an eligible candidate solution, therefore, the fitness of each candidate will be calculated. This fitness determines the movement vector of every particle in search space. This movement defines its current best-fit location. The next iteration of the algorithm takes place when all the particles move to their best next location.

The first step to perform the SLA violation prediction using the Mutant-PSO algorithm is to encode the particle among multiple QoS attributes. Where each element of a particle represents a single attribute. The value of every dimension of the particle is randomly defined under the defined upper and lower bound. Where the upper bound for every dimension is 5 and the lower bound is 0.01.

3.3 Initialization of the Particle Groups

An initial population matrix is generated which consists of many particles using random methods. A combination of defined QoS attributes is a single composition particle.

In the current algorithm, we have performed some modifications to calculate the next position of a particle instead of using the traditional velocity calculation technique we are using the average velocity of every particle. The basic formula of PSO is as in Eq. (1). The velocity of the ith particle is denoted as $v_i = v_{i1}, v_{i2}, \ldots\ldots\ldots, v_{in}$ and position of the ith particle is represented as $x_i = x_{i1}, x_{i2}, \ldots\ldots\ldots, x_{in}$ and local best particle of any iteration is presented as $L_i = L_{i1}, L_{i2}, \ldots\ldots\ldots, L_{in}$ and the global best particle is presented as $G_i = G_{i1}, G_{i2}, \ldots\ldots\ldots, G_{in}$. For every particle, its nth dimensional velocity at time $t+1$ is calculated using the given formula in Eq. (1) using by calculating local and global intelligence.

$$v_{in}^{t+1} = v_{in}^{t} + \alpha_1 r_1 (L_{in}^{t} - x_{in}^{t}) + \alpha_2 r_2 (G_{in}^{t} - x_{in}^{t}) \tag{1}$$

where α_1, α_2 the learning rate for is local and global best particles respectively and r_1, r_2 is the random number that generates under the range (0–1).

The average velocity calculated through Eq. (2)

$$v^t_{average} = \sum_{t=1}^{m} \sum_{i=1}^{n} v^t_i / n \tag{2}$$

For the learning of the particle, the existing particle is updated through Eq. (3)

$$x^{t+1}_{in} = x^t_{in} + v^t_{average} \tag{3}$$

Then fitness of every particle is calculated through Root mean square error (RMSE) and if the particle reaches to desired fitness, then the algorithm will stop otherwise the whole procedure will repeat until each particle attains the required fitness or the Number of cycles (NoC) will achieve.

3.4 Fitness Function

The PSO algorithm selects the next generation to evolve based on the fitness function and then it will search for the best solution in the search space. In the proposed methodology, the fitness function is mainly used to calculate the SLA violation that occurs under the given dataset. The fitness of the particle is evaluated by multiplying every particle with the given dataset and find the overall fitness of the particle for the whole dataset. This composite particle has four parameters of response time, availability, accessibility, speed. SLA violation occurs if the values of given attributes are very low or below the average value. The values of every attribute are classified in Tab. 1. For fitness RMSE of the learning particle with the given class is calculated and if the error ratio is minimum then the particle is considered as the best-fitted particle.

4 Results and Discussion

The following diagrams Figs. 2 and 3 represent the results of simulation in the learning process of parallel mutant-PSO for different classes of QoS attributes in the context of SLA violations.

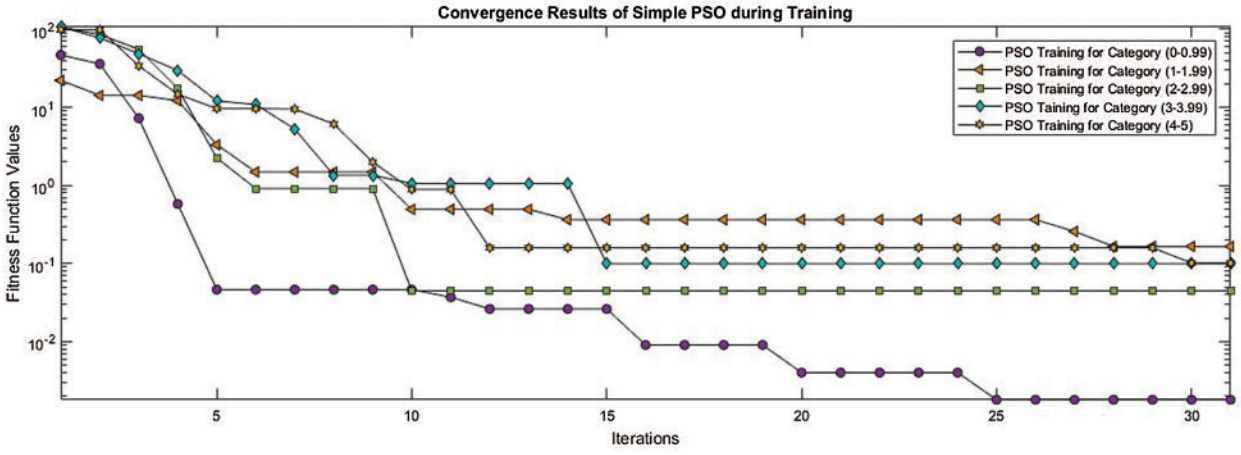

Figure 2: Simple PSO convergence results for training of all classes

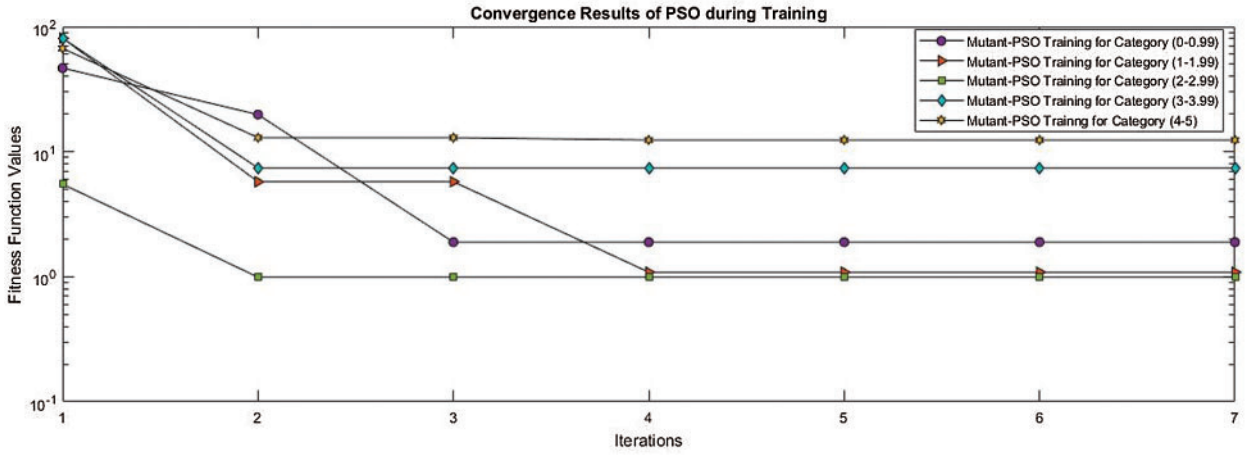

Figure 3: Parallel mutant-PSO convergence results for training of all classes

4.1 PSOs Performance in Training Phase

The proposed solution performance compares with the simple PSO in the training phase. The same dataset provides to both methodologies (Simple PSO and Mutant-PSO) for the training of algorithms. For the proposed solution MATLAB is used for the simulation. The parameters are defined in Section 3.4.3 for the training of both PSOs. The values of each parameter are pre-defined in the given dataset according to Tab. 1 defined values.

4.2 Convergence of Simple PSO in Training Phase

Fig. 2 shows the simple PSO convergence for all categories. The algorithm converges to its global minima in a minimum of 25–30 iterations. For the Strongly Weak class, the algorithm starts converging to fitness value at the 25th iteration. For the Weak class, it converges at the 27th iteration. The algorithm converges very speedily for Medium class, Strong class at 10th and 15th iteration respectively. The Very Strong class algorithm converges very slowly at the 29th iteration.

4.3 Convergence of Parallel Mutant-PSO in Training Phase

Fig. 3 represents the convergence of fitness values for training is calculated by Parallel Mutant-PSO for the iterations of different categories of SLA violation proposed in Fig. 1. Initially, the particles are generated randomly, therefore, the initial fitness error for the system is very high. Initial fitness error relates to the 0th iteration. As the algorithm advances, the convergence of particles is extreme and this particle finds its global minima very fast. As for Medium, Strong, and Very Strong classes [3(2–2.99), 4(3–3.99), and 5(4–5)], the algorithm converges at its second iteration. For the Strongly Weak class (0–0.99) Mutant-PSO converges at the third iteration and for Weak class (1–1.99) it converges at the fourth iteration. It is also observed that the number of iterations required for the convergence Parallel Mutant-PSO is seen to be 4–7 for our simulation. It is observed that the proposed Parallel Mutant-PSO converges more speedily for all categories comparative to Simple PSO for the same dataset. Simple PSO takes approximately 25–30 iterations and in its comparison, Mutant-PSO takes only 5–7 iterations for the overall convergence of the algorithm for all categories.

4.4 PSOs Performance in Validation Phase

4.4.1 Simple PSO in Validation Phase

In the validation phase, the learning of Simple PSO is tested on the unknown dataset which is not used in the training of PSO to assess the SLA Violation prediction done by PSO. Sample Dataset of 50 particles is chosen for this purpose which consists of 10 particles for each category. In the Strongly Weak, Medium, and Strong category Simple PSO prediction remain good as it predicts accurately 8, 7, and 9 particles respectively out of 10 particles. Whereas the Simple PSO performance remains very poor for the Weak and Very Strong category, it performs maximum wrong predictions for them as shown in Tab. 2.

Table 2: Simple PSO SLA violation prediction results

Dataset	Strongly weak	Weak	Medium	Strong	Very strong
	10 particles	10 particles	10 particles	10 particles	10 particles
SLA violation prediction results					
Total true predictions	8	3	7	9	0
Total false predictions	2	7	3	1	10
Accuracy level in % age	80%	30%	70%	90%	0%
Average accuracy of system in % age	54%				

4.4.2 Parallel Mutant-PSO in Validation Phase

In a comparison of simple PSO, Parallel Mutant-PSO predictions remain far better. The same dataset of 50 particles is provided to Mutant-PSO for the validation of its learning. Mutant-PSO achieves maximum accurate predictions for all categories only for Weak category its performance remains a little low as out of 10 it predicts 2 particles wrongly. Parallel Mutant-PSO SLA violation prediction results are shown in Tab. 3.

Table 3: Parallel mutant-PSO SLA violation prediction results

Dataset	Strongly weak	Weak	Medium	Strong	Very strong
	10 particles	10 particles	10 particles	10 particles	10 particles
SLA violation prediction results					
Total true predictions	10	8	10	10	9

(Continued)

Table 3: Continued

Dataset	Strongly weak	Weak	Medium	Strong	Very strong
	10 particles	10 particles	10 particles	10 particles	10 particles
Total false predictions	0	2	0	0	1
Accuracy level in % age	100%	80%	100%	100%	90%
Average accuracy of system in % age	94%				

So, in overall comparison proposed Parallel Mutant-PSO is considered as the best solution so far for the prediction of QoS-based SLA violation as it converges in minimum NoC and shows maximum accuracy in its learning.

5 Conclusion

Cloud computing is an emerging paradigm in the computing field due to its flexible use and tractability in providing hardware or software resources and which in result reduce the cost factors. Monitoring is a way in cloud computing that assures the QoS agreed in SLA will provide the cloud resources. The core concern of cloud users is the performance of service which includes service availability, response time, and desired data size and speed for the transfer of data. This study proposed a Parallel Mutant-PSO for SLA violation detection and prediction by considering the QoS attributes response time, speed, availability, accessibility. Our experimental results based on the proposed model show that the Mutant-PSO reaches its convergence state in 5–7 iterations which is better than the performance of Simple PSO that reaches its convergence state 25–30 iterations. Tabs. 2 and 3 define that predictions done by Mutant-PSO are more accurate than the Simple PSO. The Proposed Parallel Mutant-PSO meets a 94% accuracy level. The Mutant PSO is stuck in local minima, which may affect our system performance, to overcome this issue we can use opposite learning-based algorithms like Opposite Mutant-PSO, Opposite PSO, Total Opposite Mutant-PSO, etc.

Acknowledgement: We thank our families and colleagues who provided us with moral support.

Funding Statement: No funding was received for this research.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Mell and T. Grance, "The nist definition of cloud computing," *National Institute of Standards and Technology Special Publication*, vol. 13, no. 5, pp. 800–1457, 2011.

[2] J. O'sullivan, D. Edmond and A. T. Hofstede, "What's in a service?" *Distributed and Parallel Databases*, vol. 12, no. 2, pp. 117–133, 2002.

[3] S. Shin and G. Gu, "Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in *20th IEEE Int. Conf. on Network Protocols*, Austin, Texas, pp. 1–6, 2012.
[4] K. Alhamazani, R. Ranjan, K. Mitra, F. Rabhi, P. P. Jayaraman *et al.*, "An overview of the commercial cloud monitoring tools: Research dimensions, design issues, and state-of-the-art," *Computing*, vol. 97, no. 4, pp. 357–377, 2015.
[5] M. Hedges, T. Blanke and A. Hasan, "Rule-based curation and preservation of data: A data grid approach using iRODS," *Future Generation Computer Systems*, vol. 25, no. 4, pp. 446–452, 2009.
[6] N. Tabassum, A. Ditta, T. Alyas, S. Abbas, H. Alquhayz *et al.*, "Prediction of cloud ranking in a hyperconverged cloud ecosystem using machine learning," *Computers Materials & Continua*, vol. 67, no. 3, pp. 3129–3141, 2021.
[7] F. Matloob, T. M. Ghazal, N. Taleb, S. Aftab, M. Ahmad *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, pp. 98754–98771, 2021.
[8] A. D'Ambrogio and P. Bocciarelli, "A model-driven approach to describe and predict the performance of composite services," in *In 6th Int. Workshop on Software and Performance*, New York, United States, pp. 78–89, 2007.
[9] F. Qazi, A. Jhumka and P. Ezhilchelvan, "Towards automated enforcement of cloud sla," in *10th Int. Conf. on Utility and Cloud Computing*, Austin, TX, USA, pp. 151–156, 2017.
[10] C. Qu, R. N. Calheiros and R. Buyya, "A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances," *Journal of Network and Computer Applications*, vol. 65, pp. 167–180, 2016.
[11] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu and J. Wang, "Qos ranking prediction for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1213–1222, 2012.
[12] F. Matloob, S. Aftab, M. Ahmad, M. A. Khan, A. Fatima *et al.*, "Software defect prediction using supervised machine learning techniques: A systematic literature review," *Intelligent Automation & Soft Computing*, vol. 29, no. 2, pp. 404–421, 2021.
[13] R. Grati, K. Boukadi and H. Abdallah, "A qos monitoring framework for composite web services in the cloud," in *Sixth Int. Conf. on Advanced Engineering Computing and Applications in Sciences*, Barcelona, Spain, pp. 65–70, 2012.
[14] K. Sultan, I. Naseer, R. Majeed, D. Musleh, M. A. S. Gollapalli *et al.*, "Supervised machine learning-based prediction of COVID-19, *Computers, Materials and Continua*, vol. 69, no. 1, pp. 101–128, 2021.
[15] A. Michlmayr, F. Rosenberg, P. Leitner and S. Dustdar, "Comprehensive qos monitoring of web services and event-based sla violation detection," in *4th Int. Workshop on Middleware for Service Oriented Computing*, USA, pp. 1–6, 2009.
[16] H. M. Khan, G. Y. Chan and F. F. Chua, "A fuzzy model for detecting and predicting cloud quality of service violation," *Journal of Engineering Science and Technology*, vol. 13, pp. 58–77, 2018.
[17] B. Hoßbach, B. Freisleben and B. Seeger, "Reaktives cloud monitoring mit complex event processing," *Datenbank Spektrum*, vol. 12, no. 1, pp. 33–42, 2012.
[18] Q. T. A. Khan, S. Abbas, M. A. Khan, A. Fatima, S. Alanazi *et al.*, "Modelling intelligent driving behavior using machine learning," *Computers Materials & Continua*, vol. 68, no. 3, pp. 3061–3077, 2021.
[19] S. S. Gill, I. Chana, M. Singh and R. Buyya, "Chopper: An intelligent qos-aware autonomic resource management approach for cloud computing," *Cluster Computing*, vol. 21, no. 2, pp. 1203–1241, 2018.
[20] C. Cappiello, K. Kritikos, A. Metzger, M. Parkin, B. Pernici *et al.*, "A quality model for service monitoring and adaptation," in *Workshop on Service Monitoring, Adaptation and Beyond*, Germany, vol. 2, no. 26. pp. 29–40, 2009.
[21] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi *et al.*, "A survey on service quality description," *ACM Computing Surveys*, vol. 4, no. 1, pp. 1–58, 2013.
[22] S. M. Alotaibi, M. I. Basheer, A. Rehman and M. A. Khan, "Ensemble machine learning based identification of pediatric epilepsy," *Computers Materials & Continua*, vol. 68, no. 1, pp. 149–165, 2021.

[23] I. Brandic, S. Pllana and S. Benkner, "An approach for the high-level specification of qos-aware grid workflows considering location affinity," *Scientific Programming*, vol. 14, no. 3–4, pp. 231–250, 2006.
[24] H. L. Truong, R. Samborski and T. Fahringer, "Towards a framework for monitoring and analyzing qos metrics of grid services," in *Second IEEE Int. Conf. on e-Science and Grid Computing*, Amsterdam, Netherlands, pp. 65–65, 2006.
[25] C. Müller, M. Oriol, M. Rodríguez, X. Franch, J. Marco *et al.,* "Salmonada: A platform for monitoring and explaining violations of ws-agreement-compliant documents," in *4th Int. Workshop on Principles of Engineering Service-Oriented Systems*, Zurich, Switzerland, pp. 43–49, 2012.
[26] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar and A. H. Ngu, "Cloudarmor: Supporting reputation-based trust management for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 367–380, 2015.
[27] A. Naseri and N. J. Navimipour, "A new agent-based method for qos-aware cloud service composition using particle swarm optimization algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1851–1864, 2019.
[28] G. Ahmad, S. Alanazi, M. Alruwaili, F. Ahmad, M. A. Khan *et al.*, "Intelligent ammunition detection and classification system using convolutional neural network," *Computers Materials & Continua*, vol. 67, no. 2, pp. 2585–2600, 2021.
[29] B. Shoaib, Y. Javed, M. A. Khan, F. Ahmad, R. Majeed *et al.*, "Prediction of time series empowered with a novel srekrls algorithm," *Computers Materials & Continua*, vol. 67, no. 2, pp. 1413–1427, 2021.
[30] M. A. Khan, S. Abbas, K. M. Khan, M. A. Ghamdi and A. Rehman, "Intelligent forecasting model of covid-19 novel coronavirus outbreak empowered with deep extreme learning machine," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1329–1342, 2020.
[31] A. H. Khan, M. A. Khan, S. Abbas, S. Y. Siddiqui, M. A. Saeed *et al.*, "Simulation, modeling, and optimization of intelligent kidney disease predication empowered with computational intelligence approaches," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1399–1412, 2021.
[32] S. Aftab, S. Alanazi, M. Ahmad, M. A. Khan, A. Fatima *et al.*, "Cloud-based diabetes decision support system using machine learning fusion," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1341–1357, 2021.
[33] A. Haider, M. A. Khan, A. Rehman, M. U. Rahman and H. S. Kim, "A real-time sequential deep extreme learning machine cybersecurity intrusion detection system," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1785–1798, 2020.
[34] S. Hussain, R. A. Naqvi, S. Abbas, M. A. Khan, T. Sohail *et al.*, "Trait based trustworthiness assessment in human-agent collaboration using multi-layer fuzzy inference approach," *IEEE Access*, vol. 9, no. 4, pp. 73561–73574, 2021.
[35] N. Tabassum, A. Ditta, T. Alyas, S. Abbas, H. Alquhayz *et al.*, "Prediction of cloud ranking in a hyperconverged cloud ecosystem using machine learning," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 3129–3141, 2021.
[36] M. W. Nadeem, H. G. Goh, M. A. Khan, M. Hussain, M. F. Mushtaq *et al.*, "Fusion-based machine learning architecture for heart disease prediction," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2481–2496, 2021.
[37] A. I. Khan, S. A. R. Kazmi, A. Atta, M. F. Mushtaq, M. Idrees *et al.*, "Intelligent cloud-based load balancing system empowered with fuzzy logic," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 519–528, 2021.
[38] S. Y. Siddiqui, I. Naseer, M. A. Khan, M. F. Mushtaq, R. A. Naqvi *et al.*, "Intelligent breast cancer prediction empowered with fusion and deep learning," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 1033–1049, 2021.
[39] R. A. Naqvi, M. F. Mushtaq, N. A. Mian, M. A. Khan, M. A. Yousaf *et al.*, "Coronavirus: A mild virus turned deadly infection," *Computers, Materials and Continua*, vol. 67, no. 2, pp. 2631–2646, 2021.

[40] A. Fatima, M. A. Khan, S. Abbas, M. Waqas, L. Anum *et al.,* "Evaluation of planet factors of smart city through multi-layer fuzzy logic," *The Isc International Journal of Information Security*, vol. 11, no. 3, pp. 51–58, 2019.
[41] F. Alhaidari, S. H. Almotiri, M. A. A. Ghamdi, M. A. Khan, A. Rehman *et al.*, "Intelligent software-defined network for cognitive routing optimization using deep extreme learning machine approach," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 1269–1285, 2021.
[42] M. W. Nadeem, M. A. A. Ghamdi, M. Hussain, M. A. Khan, K. M. Khan *et al.*, "Brain tumor analysis empowered with deep learning: A review, taxonomy, and future challenges," *Brain Sciences*, vol. 10, no. 2, pp. 118–139, 2020.
[43] A. Atta, S. Abbas, M. A. Khan, G. Ahmed and U. Farooq, "An adaptive approach: Smart traffic congestion control system," *Journal of King Saud University—Computer and Information Sciences*, vol. 32, no. 9, pp. 1012–1019, 2020.
[44] M. A. Khan, S. Abbas, A. Rehman, Y. Saeed, A. Zeb *et al.*, "A machine learning approach for blockchain-based smart home networks security," *IEEE Network*, vol. 35, no. 3, pp. 223–229, 2020.
[45] M. A. Khan, M. Umair, M. A. Saleem, M. N. Ali and S. Abbas, "Cde using improved opposite-based swarm optimization for mimo systems," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 1, pp. 687–692, 2019.
[46] M. A. Khan, M. Umair and M. A. Saleem, "Ga based adaptive receiver for mc-cdma system," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 23, no. 1, pp. 2267–2277, 2015.